МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МАТЕРИАЛЫ

Международной научной конференции

«МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ, ТЕХНИЧЕСКИХ И ЭКОНОМИЧЕСКИХ СИСТЕМ»

Томск, 28–30 мая 2020 г.

Под общей редакцией
кандидата технических наук И.С. Шмырина

Томск
Издательство Томского государственного университета
2020

ББК 22.17–22.19
УДК 519.2, 519.7, 519.8
Т78

Т78 Труды Томского государственного университета. – Т. 305. Серия физико-математическая: Математическое и программное обеспечение информационных, технических и экономических систем : материалы Международной научной конференции. Томск, 28–30 мая 2020 г. / под общ. ред. И.С. Шмырина. – Томск : Издательство Томского государственного университета, 2020. – 322 с.

ISBN 978-5-94621-970-9

Сборник содержит материалы Международной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем», проводившейся 28–30 мая 2020 г. на базе Института прикладной математики и компьютерных наук Томского государственного университета. Материалы сгруппированы в соответствии с работавшими на конференции секциями.

Для научных работников, преподавателей, аспирантов, магистрантов и студентов.

УДК 539.3.004
ББК 22,25.22.251.22.62

ISBN 978-5-94621-970-9

I. ТЕОРЕТИЧЕСКИЕ И ТЕХНОЛОГИЧЕСКИЕ ОСНОВЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА, ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА И ВИЗУАЛИЗАЦИИ БОЛЬШИХ ДАННЫХ, МАШИННОГО ОБУЧЕНИЯ И СИСТЕМ СЕМАНТИЧЕСКОГО МОДЕЛИРОВАНИЯ

ПОСТРОЕНИЕ ПАРАМЕТРИЗОВАННОГО ЧЕЛОВЕЧЕСКОГО МАНЕКЕНА НА ОСНОВЕ ШВЕЙНЫХ МЕРОК

Головчинер М.Н., Рогожин А.С.
Томский государственный университет
golovchiner@mail.ru, globys7520@mail.ru

Аннотация

Рассматриваются способ построения параметризованного человеческого манекена на основе швейных мерок. Предлагается трех этапный подход, основанный на антропометрии, пластической анатомии и алгоритмах подразделения поверхности. Изложен общий подход для получения параметризованного человеческого манекена на основе размерных признаков.

Введение

Манекен – это изделие, имитирующее форму тела человека: мужчины, женщины или ребёнка. Данное изделие нашло свое применение во многих сферах деятельности, таких, как медицина, пошив одежды, военное дело и в других. Особенное место они заняли в портновском деле и при демонстрации одежды.

В портновской деятельности, манекены используются при проектировании и пошиве одежды. Стоит отметить, что изготовление одежды является сложным технологическим процессом, состоящим из множества этапов, таких как конструирование лекал, создание технологической карты пошива, подготовительная стадия, раскрой деталей из материалов и т.д. Практически на всех этих этапах требуется работа с манекеном, однако при изготовлении одежды по индивидуальным параметрам невозможно использовать готовые манекены, т.к. параметры заказчика редко соответствуют стандартным размерам готовых манекенов.

В связи с развитием сетевых технологий и появлением всемирной сети, появилась возможность распространения, продажи различных товаров через неё. Однако следует иметь в виду, что одежда является товаром, который требуется примерять клиенту на себя, для того чтобы определиться с размером и другими факторами выбора. Можно заметить, что одежда имеет свои таблицы размеров, при этом нет никакой гарантии, в том, что они подойдут для конкретного клиента. Данная тенденция, продажи товаров через всемирную сеть делает актуальной разработку методов построения манекенов на основе определенных параметров, для виртуальной примерки одежды, а также для изготовления одежды под заказ. К одним из подобных работ можно отнести [1,2].

Представленная работа посвящена методу построения параметризованного человеческого манекена на основе швейных мерок. В рамках данной работы разработан метод построение манекена по определённым параметрам с заданным уровнем гладкости. Предлагаемый в данной работе подход основан на использовании методов подразделения поверхности, антропометрических данных человека, пластической анатомии и швейных измерений. Комбинируя данные методы, можно получить высокополигональную (реалистичную) модель по заданным меркам, а также низкополигональную (референтную) модель для её хранения или иных манипуляций.

В разделе 1 дается постановка задачи. В разделе 2 представляется метод построение модели параметризованного человеческого манекена.

1. Постановка задачи

Задана информация о индивидуальных швейных мерках пользователя, снятых по технологии единой методики конструирования одежды СЭВ (ЕМКО СЭВ). В частности, данная методика использует 26 основных размерных признаков и 14 дополнительных (для контроля), таких как рост, высота точки основания шеи, высота талии и т.д. Требуется построить параметризованную модель манекена, удовлетворяющую измерениям, снятым по ЕМКО СЭВ. При этом желательно, что бы построение модели манекена производилось в реальном времени. Отметим, что использование данной модели предполагается для виртуальной примерки одежды, что требует от неё наличия референтного каркаса примерки модели одежды или выкроек.

2. Описание метода построения модели параметризованного человеческого манекена

В нашем методе, пользователю необходимо ввести ряд параметров, снятых по ЕМКО СЭВ, затем на основе этих параметров система будет вычислять дополнительные параметры необходимы для генерации модели. Затем на основе дополнительных параметров и швейных измерений генерируется базовая (референтная) модель манекена. Конечная (реалистичная) модель получается путём применения алгоритмов интерполяционного подразделения поверхности на базовую модель. Таким образом данный подход можно разделить на три этапа: предварительный этап, этап генерации базовой модели и этап детализации базовой модели. На рис. 1 представлена блок-схема данного метода.

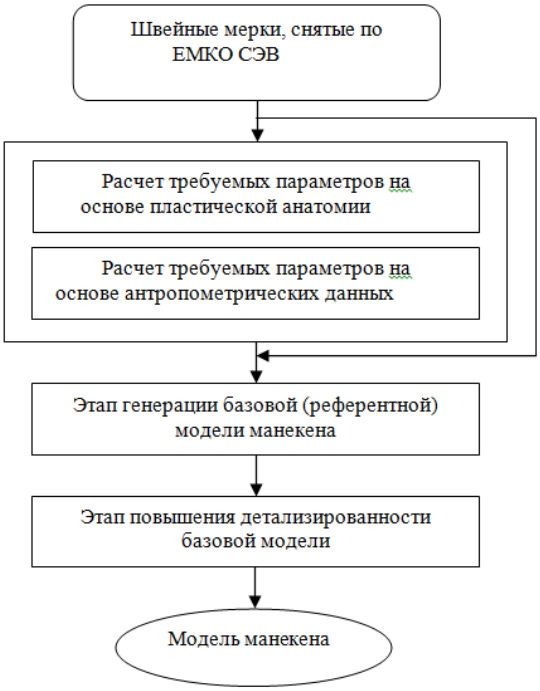


Рис. 1. Блок-схема метода построения модели параметризованного манекена

2.1. Предварительный этап

Размерные признаки (швейные измерения) представляют исчерпывающую информацию для описания индивидуальной формы человека при создании одежды, однако при этом не является исчерпывающими для построения модели манекена. Задача данного этапа заключается в нахождении недостающих (вспомогательных) параметров,

таких как толщина шеи или ширина бедер. Для этого используются знания об антропометрии и пластической анатомии [3,4,5].

2.2. Этап генерация базовой модели

В соответствии с анатомической особенностью человека, модель манекена можно разделить на шесть сегментов: голова, туловище, верхняя конечность, нижняя конечность, рука, ступня. Из этих сегментов туловище является наиболее важным и сложным сегментом, который представляет основные черты человеческого организма. Для построения, к примеру, туловища используется двенадцать параметров (см. табл. 1).

Таблица 1

Швейные параметры для построения модели туловища

Швейное имя	Расшифровка швейного имени
T1	Рост
T4	Высота точки основания шеи
T7	Высота линии талии
T8	Высота остисто-подвздошной передней точки
T12	Высота подъягодичной складки
T16	Обхват груди третий
T17	Обхват груди четвертый
T39	Расстояние от швейной точки до линии обхвата груди первого с учётом выступающих лопаток
T40	Длина спины до талии с учётом выступания лопаток
T46	Расстояние между сосковыми точками
T53	Плечевой диаметр
T57	Передне-задний диаметр руки

Базовая модель – это низкополигональный каркас. Он состоит из ярко выделенных поперечных сечений, каждое из которых имеет свою геометрическую форму, зависящую от мерок и дополнительных параметров, полученных на предварительном этапе. Отметим, что базовая модель неизменна, следовательно, она может использоваться как референтная модель, к примеру, для примерки моделей одежды.

2.3. Этап повышения детализированности базовой модели

На данном этапе происходит повышения полигональности у базовой модели манекена, т.е. превращение низкополигональной модели в высокополигональную модель, для этого используются алгоритмы подразделения поверхности. Заметим, что алгоритмы подразделения поверхности делятся на два вида: аппроксимирующие и интерполирующие. Предполагается, что базовая модель неизменна, следовательно, использование аппроксимирующих методов недопустимо, т.к. они изменяютизначальное положения исходных точек базовой модели. Поэтому на данном этапе требуется использовать интерполирующие методы подразделения поверхности такие как схема подразделения Butterfly [6], схема подразделения Kobbelt [7] и др.

2.4. Архитектура информационной системы

На рис. 2. представлена общая структура информационной системы, состоящая из следующих блоков: пользовательское приложение (App), менеджер графов (GraphManager), конфигурационный файл (Config), блок генерации графа (GraphBuilder), граф (Graph), репозиторий компонентов (ComponentsRepository).

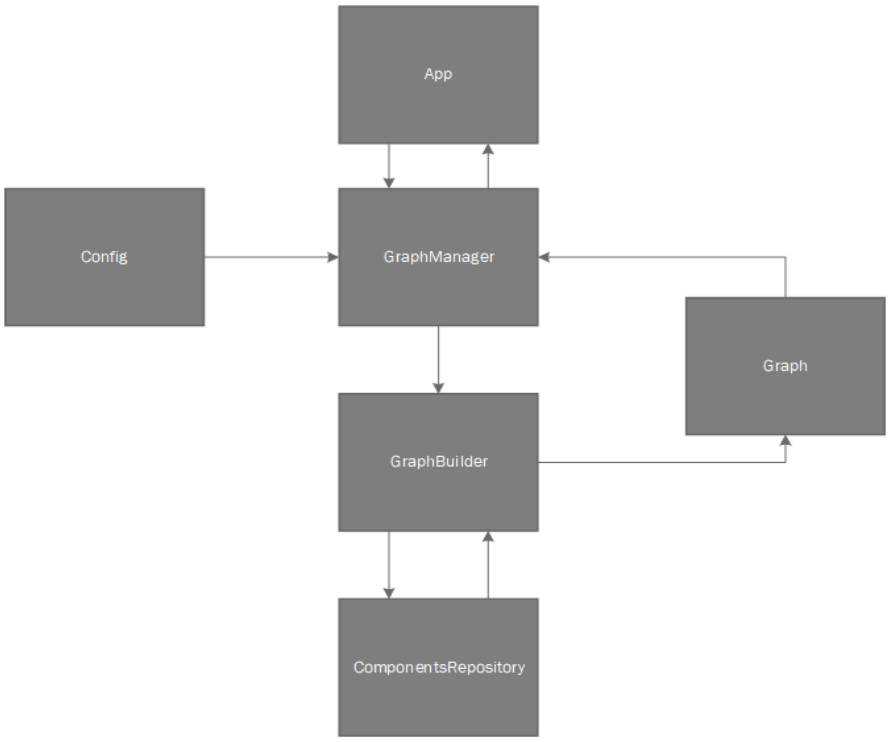


Рис. 2. Общая структурная схема информационной системы

Пользовательское приложение (App) – интерфейс взаимодействия пользователя с системой. В зависимости от типа пользователя может иметь различный функционал. Граф (Graph) – цепочка компонентов (параметров) конкретного пользователя. Конфигурационный файл (Config) – файл, описывающий структуру входящих данных. Репозиторий компонентов (ComponentRepository) – хранилище всех компонентов. Блок генерации графа (GraphBuilder) – блок, предназначенный для создания графа из компонентов, существующих в репозитории компонентов. Менеджер графов (GraphManager) – администратор всех построенных графов.

2.4.1. Пользовательское приложение

Пользовательское приложение – это приложение для взаимодействия с системой. Пользовательское приложение может быть двух типов: для конечных пользователей и для разработчиков. Для конечных пользователей оно может быть представлено в виде десктопного приложения, веб сайта или мобильного приложения с уже предустановленным функционалом. Для разработчиков оно может быть представлено в виде некоторых api или некого графического интейфейса для взаимодействия с данными api.

На рис. 3 представлен пример пользовательского десктопного приложения.

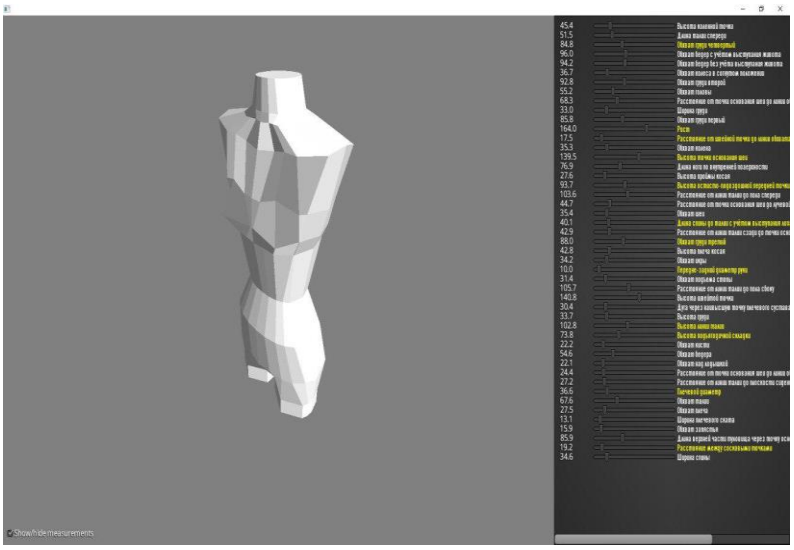


Рис. 3. Пример пользовательского десктопного приложения

Основной задачей пользовательского приложения является передача данных в определенном формате для построения модели и отображение результата (модели).

2.4.2. Менеджер графов и граф

Менеджер графов – это блок управления всеми графами в системе. Основной задачей данного блока является создание графов по заданной конфигурации, управление потоками данных между графами и пользовательскими приложениями, а также управление жизнью графов. Структурная схема менеджера графов имеет следующий вид (рис. 4):

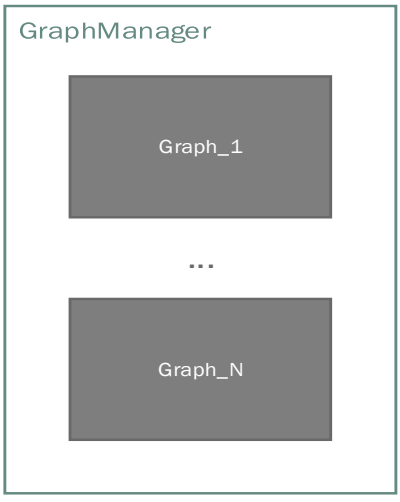


Рис. 4. Структурная схема менеджера графов

Граф – это определенная цепочка элементов, выполнение которой приводит к результату для конкретного пользователя. Архитектура графа основана на паттерне «PipeandFilter», где роль вершин отведена фильтрам, а роль ребер – каналам.

PipeandFilter– это архитектурный шаблон, в котором есть независимые объекты, называемые фильтрами (компонентами), выполняющие некоторые преобразования над данными, и каналы, служащие соединителями для потока преобразовываемых данных, каждый из которых подключен к следующему компоненту в трубопроводе(графе).

Данный паттерн характеризуется последовательным преобразованием потоков данных в одном направлении (рис. 5).

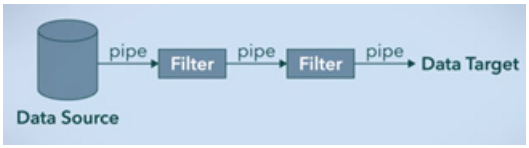


Рис. 5. Преобразование данных в архитектуре канала и фильтра

Заметим, что любой из фильтров способен как получать данные из одного или нескольких каналов, так и отправлять данные. При этом фильтры могут работать одновременно. Поскольку выход одного фильтра является входом другого, то порядок расположения фильтров очень важен.

Таким образом, граф, основанный на паттерне «PipeandFilter», обладает следующими преимуществами:

1) обеспечивает слабую и гибкую связь компонентов (фильтров);
2) слабая связь позволяет заменять фильтры без изменений других фильтров;
3) фильтры могут обрабатываться параллельно;
4) фильтры можно рассматривать как черные ящики, что позволяет пользователям системы не знать логику работы каждого фильтра;
5) возможность повторного использования фильтров.

Структура графа представлена на рис. 6.

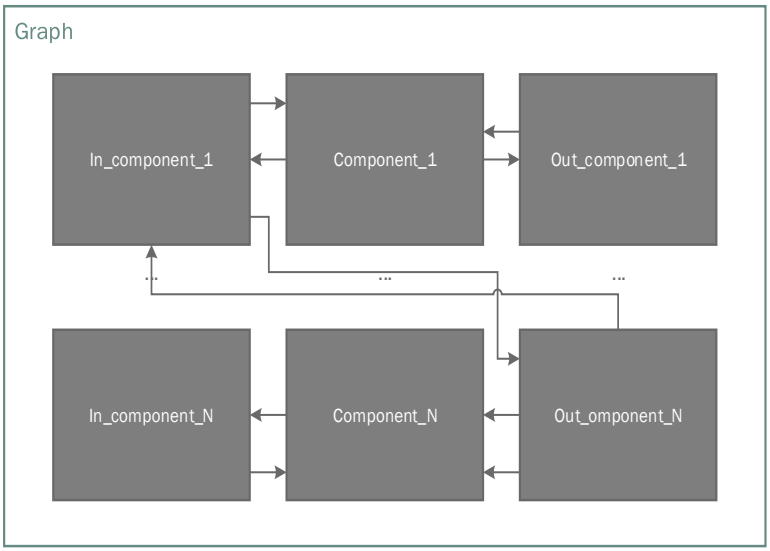


Рис. 6. Структура графа

2.4.3. Компоненты

Компоненты фильтра – это некоторые атомарные задачи, выполняемые над данными или генерирующие новые данные по каким-либо правилам, определенным внутри компонента. Общая структура компонентов представлена на рис. 7.

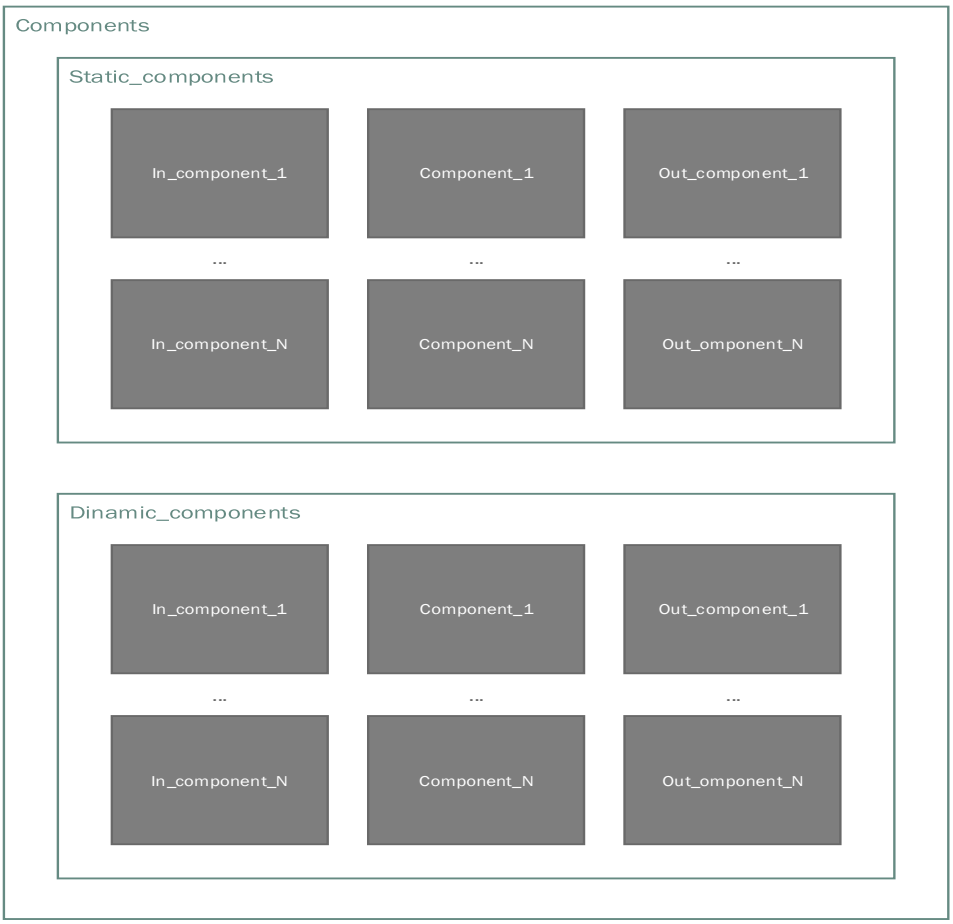


Рис. 7. Общая структура компонентов

Фильтры подразделяются на 3 типа: входные, выходные и стандартные. Кроме того, компоненты делятся на 2 вида:

– статические, которые включаются сразу после выбора;

– динамические, которые можно добавить по ходу выполнения программы.

Входные компоненты – это компоненты, являющиеся началом любого графа. В графе должен присутствовать минимум один компонент данного типа. Особенностью данного типа компонентов является то, что они не имеют входящих каналов, все необходимые данные входные фильтры либо уже содержат внутри, либо получают иными методами.

Структура входного компонента представлена на рис. 8.

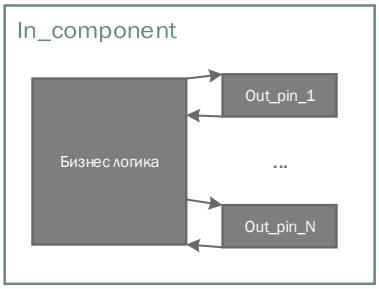


Рис. 8. Структура входного компонента

Выходные компоненты – это компоненты, являющиеся конечным результатом любого графа. Как и во входных компонентах, фильтров данного типа должно быть не менее одного. Главной отличительной чертой данных компонентов является то, что они не имеют выходных каналов, они выдают результат напрямую или каким-либо косвенным путём (например, записывают результат в файл). Структура данных компонентов представлена на рис. 9.

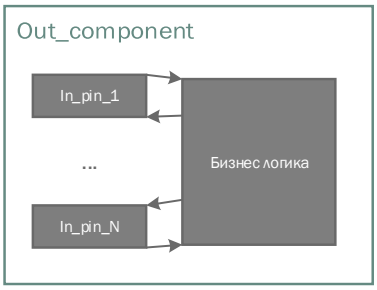


Рис. 9. Структура выходного компонента

Стандартные компоненты – это промежуточные компоненты, выполняющие какую-то определенную задачу в графе. Данные фильтры характеризуются тем, что их вообще может не присутствовать в исходном графе. На рис. 10 представлена структура данных компонентов.

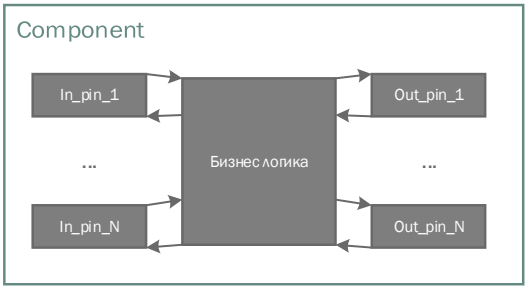


Рис. 10. Структура стандартного компонента

Для того чтобы образовать трубопровод (граф), все компоненты исходного графа соединяются между собой, образуя каналы, роль которых выполняют блоки in_pinи out_pin, соединенные между собой.

Т.к. в каналах не происходят манипуляции с данными, то реализация данных блоков была основана на паттерне «DataTransferObject», т.к. этот шаблон проектирования используется для передачи данных между подсистемами приложения и не обладает каким-либо поведением внутри себя.

Заключение

Предложен подход к построению параметризованного человеческого манекена на основе швейных мерок, основанный на построении референтного (базового) каркаса с последующей постобработкой его для получения более детализированной модели манекена.

Разработана методика построения параметризованного человеческого манекена и архитектура информационной системы построения манекена, функционирование которой основано на использовании искомой методики.

Разработанная методика состоит из трех этапов: предварительного этапа, этапа генерации базовой модели и этапа повышения детализации базовой модели. Данная ме-

тодика позволяет получить на выходе готовый параметризованный манекен с заданным уровнем сглаженности.

Предлагается структура информационной системы, состоящей из шести блоков: пользовательское приложение, менеджер графов, конфигурационный файл, блок генерации графа, граф, репозиторий компонентов.

Такая система позволяет достаточно гибко управлять процессом создания параметризованного манекена.

ЛИТЕРАТУРА

1. *Junfeng Yao* R&D of a Parameterized Method for 3D Virtual Human Body Based on Anthropometry [Электронный ресурс] // PDFS.SEMANTICSCHOLAR.ORG: сервер статей в открытом доступе. URL: https://pdfs.semanticscholar.org/66dd/b7a01bc968ce137b8a3a214f4dc423e701fa.pdf (дата обращения: 30.05.2020).

2. *Azuola F.* Building Anthropometry-Based Virtual Human Models [Электронный ресурс] // REPOSITORY.UPENN.EDU:общедоступный институциональный репозиторий Университета Пенсильвании. URL: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1066&context=hms (дата обращения: 30.05.2020).

3. Размеры тела и их взаимосвязь [Электронный ресурс]: анатомия, физиология, патология. URL: http://anfiz.ru/books/item/f00/s00/z0000029/st051.shtml (дата обращения: 30.05.2020).

4. *Колодовский И.* Пропорции тела человека [Электронный ресурс]: методические рекомендации для художников. URL: https://docplayer.ru/28420253-Proporcii-tela-cheloveka.html (дата обращения: 30.05.2020).

5. Пластическая анатомия [Электронный ресурс] // STUDFILE.NET: интернет-площадку для размещения студентами своих работ и иных полезных для учебы и образования материалов. URL: https://studfile.net/preview/6065594/page:13/ (дата обращения: 30.05.2020).

6. *David Levin* A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control [Электронный ресурс] // CSXSTATIC.IST.PSU.EDU: цифровая библиотека и поисковая система для научной литературы. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.133.8925&rep=rep1&type=pdf (дата обращения: 30.05.2020).

7. *Leif Kobbelt* Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology [Электронный ресурс] // GRAPHICS.RWTH-AACHEN.DE: группа компьютерной графики RWTH Aachen University. URL: https://www.graphics.rwth-aachen.de/media/papers/four.pdf (дата обращения: 30.05.2020).

МЕТОДЫ И АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ ПОИСКА ФОРМАЛЬНЫХ ПОНЯТИЙ ДЛЯ БИНАРНЫХ И НЕЧЁТКИХ КОНТЕКСТОВ

Голышев В.К., Семенова Д.В.
Сибирский федеральный университет
valeriygolyshev@mail.ru, DVSemenova@sfu-kras.ru

Введение

В различных прикладных исследованиях используются матрицы для представления объектов и их признаков из предметной области. Строкам таких матриц соответствуют объекты, а столбцам – признаки. Если объект обладает признаком, то соответствующий элемент матрицы равен 1, и если не обладает, то он равен 0. Имеет место и более общий случай, когда элементы матрицы могут принимать более двух значений (значения от 0 до 1). Часто для таких матриц возникает следующая задача: дано множество объектов A, необходимо найти множество B всех его признаков, которые свойственны всем объектам из A, а также найти множество C всех объектов, обладающих всем признаками из B. Другими, словами нужно найти максимальную единичную подматрицу. Такую подматрицу можно сопоставить понятию, в котором множество объектов является объемом, а множество признаков – содержанием [1], что полностью соответствует традиционному философскому определению понятия. Такое определение понятия представляется парой вида (объем, содержание); при уменьшении объема понятия увеличивается его содержание и наоборот; понятия иерархически упорядочены отношением «быть более общим понятием» [2,3].

Описанной выше задачей занимается анализ формальных понятий (АФП) [1,4,5] – прикладная ветвь алгебраической теории решеток [6]. Также все более популярным становится обобщение анализа формальных понятий на нечеткие контексты [7]. АФП позволяет получать концептуальную модель рассматриваемой предметной области с

помощью решеток понятий, которые можно графически представить в виде диаграммы Хассе. В рамках АФП разработано немалое количество алгоритмов получения множества всех формальных понятий и связывания их в решетку. Однако зачастую возникает проблема вычислительной сложности при большой размерности контекста. Поэтому на сегодняшний день актуальны исследования, позволяющие снизить вычислительную сложность предложенной задачи. Данные исследования направлены на разработку новых, более эффективных алгоритмов, отбирающих наиболее «важные» понятия. Другое направление разработок рассматривает уменьшение размерности входа, что позволяет улучшить работу уже существующих алгоритмов. В данной работе рассматривается второе направление исследований, а именно метод фрагментации и дефрагментации формальных контекстов, предложенный в [8]. Целью работы является исследование методов и алгоритмов решения задачи поиска формальных понятий для бинарных и нечётких контекстов.

1. Классический анализ формальный понятий

Формальным контекстом называется тройка $K=(G,M,I)$, где G – непустое конечное множество объектов, M – непустое конечное множество признаков, а $I: G\times M\to\{0,1\}$ – непустое отношение инцидентности, ставящее каждому объекту $g\in G$ признак $m\in M$, т.е. запись $(g,m)\in I$ означает, что объект *g* обладает признаком *m* (или признак *m* присущ объекту *g*). Формальный контекст может быть представлен 0,1-матрицей.

Формальным понятием контекста $K=(G,M,I)$ называется пара множеств (A,B), $A\subseteq G$, $B\subseteq M$ таких, что $A'=B$ и $B'=A$. Здесь A' – множество признаков, присущих всем объектам из *A*, а B' – множество объектов, имеющих все признаки из *B*. Также полагается, что пустому множеству объектов присущи все признаки из *M* и каждый объект обладает пустым множеством признаков. Отображения $(\cdot)'$ являются соответствиями Галуа между частично упорядоченными множествами 2^G и 2^M [6]. Двойное применение $(\cdot)'$ устанавливает оператор замыкания $(\cdot)''$ на 2^M в алгебраическом смысле. Множество *A* называют объемом формального понятия (A,B), а множество *B* – его содержанием.

Множество всех формальных понятий контекста $K=(G,M,I)$ обозначается *FC*. На данном множестве задается отношение частичного порядка $\sqsubseteq$: $(A_1,B_1)\sqsubseteq(A_2,B_2)\Leftrightarrow A_1\subseteq A_2\ (B_2\subseteq B_1)$, $(A_1,B_1),(A_2,B_2)\in FC$. Говорят, что формальное понятие (A_2,B_2) является более общим, чем понятие (A_1,B_1). Частично упорядоченное множество $(FC,\sqsubseteq)$ образует решетку формальных понятий контекста $K=(G,M,I)$.

Задача нахождения всех формальных понятий в бинарном контексте.

Дано: формальный контекст $K=(G,M,I)$.

Требуется: найти множество всех его формальных понятий *FC*.

Задача поиска всех формальных понятий является комбинаторной перечислительной задачей и считается #*P*-полной. В общем случае мощность множества *FC* экспоненциально зависит от размеров исходного контекста, что говорит о высокой вычислительной сложности данной задачи. Классификация алгоритмов решения задачи нахождения множества *FC* по порядку генерации понятий представлена на рис. 1 [9,10].

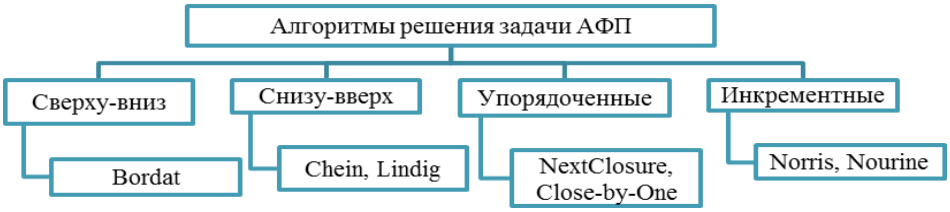


Рис. 1. Алгоритмы решения задачи поиска формальных понятий для бинарных контекстов

2. Нечеткий анализ формальных понятий

Развитие аппарата нечеткой логики повлекло за собой все большее внедрение этого направления в различные области научных исследований. Это явление не обошло стороной и анализ формальных понятий. Классический АФП имеет дело с бинарными контекстами, представляемыми 0,1-матрицами. На практике же могут возникать ситуации неопределенности и неполноты информации об объектах и их признаках, что приводит к размытой оценкой истинности утверждения факта обладания объектом g признаком m.

Пусть задана полная резидуальная решетка $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, *, 0, 1 \rangle$, где L – множество степеней истинности, * – замыкание, $\rightarrow$ и $\otimes$ – «нечеткая импликация» и «нечеткая конъюнкция» соответственно. Нечеткий формальный контекст – это тройка $\tilde{K} = (G, M, \tilde{I})$, где G и M – непустые конечные множества объектов и признаков соответственно, а $\tilde{I} : G \times M \rightarrow L$ – нечеткое отношение между G и M, ставящее каждому объекту $g \in G$ в соответствие признак $m \in M$ с некоторой степенью истинности $\tilde{I}(g,m) \in L$ [11,12,13].

Нечёткое формальное понятие определяется через сопряжённые операции «нечеткой импликации» и «нечеткой конъюнкции» и операцию нечёткого замыкания [11,12]. Тремя наиболее важными парами сопряженных операций являются операции Лукасевича, операции Геделя и операции произведения [11,12]. В работе рассмотрены два граничных случая замыкания: идентичность и глобализация.

Для нечетких множеств $\tilde{A} \in \mathbf{L}^G$ и $\tilde{B} \in \mathbf{L}^M$ рассмотрим нечеткие множества $\tilde{A}^{\uparrow} \in \mathbf{L}^M$ и $\tilde{B}^{\downarrow} \in \mathbf{L}^G$, определяемые как

$$\tilde{A}^{\uparrow}(m) = \bigwedge_{g \in G} \left(\tilde{A}(g)^{*_G} \rightarrow \tilde{I}(g,m) \right), \quad \tilde{B}^{\downarrow}(g) = \bigwedge_{m \in M} \left(\tilde{B}(m)^{*_M} \rightarrow \tilde{I}(g,m) \right).$$

Здесь $\tilde{A}^{\uparrow}(m)$ – нечеткое множество признаков, общих для всех объектов для которых «очень верно», что они принадлежат $\tilde{A}$, а $\tilde{B}^{\downarrow}(g)$ – нечеткое множество объектов, обладающих всеми признаками, для которых «очень верно», что они принадлежат $\tilde{B}$. Операторы $^{\uparrow\downarrow}$ формируют нечеткие соответствия Галуа с замыканиями [11,12].

Нечеткое формальное понятие контекста $\tilde{K} = (G, M, \tilde{I})$ – это пара нечетких множеств $(\tilde{A}, \tilde{B})$, $\tilde{A} \in \mathbf{L}^G$, $\tilde{B} \in \mathbf{L}^M$ таких, что $\tilde{A}^{\uparrow} = \tilde{B}$, $\tilde{B}^{\downarrow} = \tilde{A}$. Здесь нечеткое множество $\tilde{A}$ – (нечеткий) объем понятия $(\tilde{A}, \tilde{B})$, а нечеткое множество $\tilde{B}$ – его (нечеткое) содержание. Множество всех нечетких формальных понятий нечеткого контекста $\tilde{K} = (G, M, \tilde{I})$ обозначается FC. Рассмотрим отношение частичного порядка $\leq$ на FC:

$$\left(\tilde{A}_1, \tilde{B}_1\right) \le \left(\tilde{A}_2, \tilde{B}_2\right) \Leftrightarrow \tilde{A}_1 \subseteq \tilde{A}_2 \text{ (или } \tilde{B}_2 \subseteq \tilde{B}_1\text{)}$$

для любых $\left(\tilde{A}_1, \tilde{B}_1\right), \left(\tilde{A}_2, \tilde{B}_2\right) \in FC$. Структуру $\left(FC, \le\right)$ называют решеткой нечетких формальных понятий.

Задача нахождения всех формальных понятий в бинарном контексте.

Дано: нечеткий формальный контекст $\tilde{K} = \left(G, M, \tilde{I}\right)$.

Требуется: найти множество всех его формальных понятий FC.

Методы построения нечётких контекстов рассмотрены в [7]. На рис. 2 представлены способы нахождения понятий в нечетком контексте.

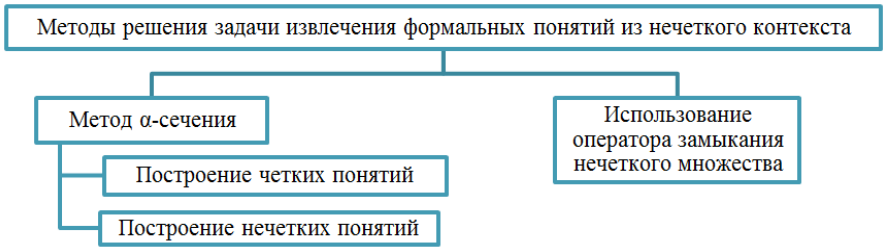


Рис. 2. Подходы к поиску формальных понятий в нечетком контексте

3. Методы решения задачи поиска формальных понятий

В настоящее время было разработано большое количество методов и алгоритмов, позволяющих решать задачу нахождения формальных понятий для бинарных и нечетких контекстов. В данной работе были реализованы одни из наиболее известных алгоритмов анализа формальных понятий: алгоритм NextClosure Гантера [10], Close-by-One Кузнецова [10], а также обобщение алгоритма Гантера на нечеткие контексты [11].

Алгоритмы NextClosure и Close-by-One генерируют понятия в лексикографическом порядке их объемов. Алгоритм Гантера NextClosure вычисляет замыкания только для некоторых подмножеств G и занимает небольшое количество памяти. Алгоритм Кузнецова Close-by-One получает каждое новое замыкание пересечением содержания понятия, сгенерированного на прошлом шаге, и содержания объекта, не принадлежащего объему этого понятия. Обобщенный алгоритм Гантера находит нечеткие формальные понятия нечеткого контекста, используя замыкание нечеткого множества и различные пары сопряженных операций. Вычислительная сложность алгоритмов Close-by-One и NextClosure – $O\left(|FC| \cdot |G|^2 \cdot |M|\right)$. Сложность обобщенного алгоритма Гантера не менее сложности NextClosure и существенно зависит от выбора оператора замыкания и сопряженных операций.

Данные алгоритмы позволяют получить множество всех формальных понятий для заданного контекста. Однако время выполнения данных алгоритмов может быть экспоненциальным в зависимости от исходного контекста, что делает поиск формальных понятий высокозатратным по времени и ресурсам.

Высокая вычислительная сложность заставила исследователей искать пути ее снижения. Существует два направления разработок в данной области: уменьшения входа путем снижения размерности исходного формального контекста, уменьшение выхода путем отбора наиболее «важных» и информативных формальных понятий. Одним из методов первого направления является метод Монгуш [8], позволяющий разложить исходный формальный контекст на фрагменты, тем самым уменьшая вход для алгоритмов. В данной работе было реализовано однократное разложение исходного фор-

мального контекста методом Монгуш. В [8] доказано, что разложение контекста на фрагменты является «неискажающим», т.е. не происходит потери или добавления новых формальных понятий. Также множество формальных понятий исходного контекста может быть восстановлено объединением множеств формальных понятий фрагментов. Схема метода изображена на рис. 3.

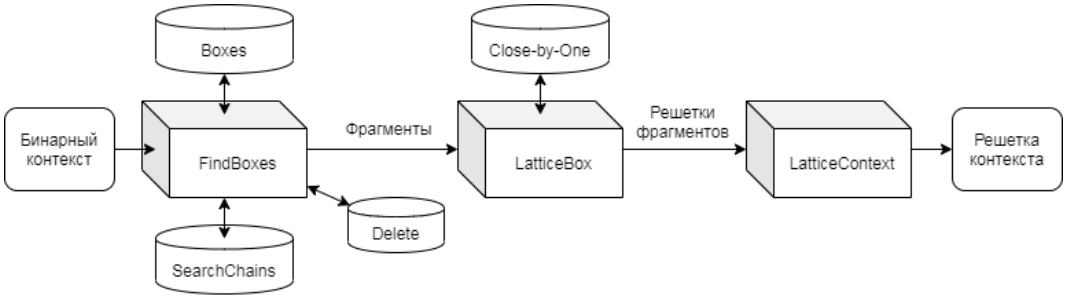


Рис. 3. Схема метода Монгуш

В основе метода лежат алгоритмы

– FindBoxes (получает на вход контекст $K=(G,M,I)$ и целое число $k>0$, разлагает исходный формальный контекст $K=(G,M,I)$ на фрагменты *k* раз), вычислительная сложность алгоритма – $O\left(|G|^{2k}\cdot|M|^{2k}\right)$,

– Boxes (разлагает фрагмент на более мелкие фрагменты), вычислительная сложность алгоритма – $O\left(|G|\cdot|M|\right)$,

– SearchChains (находит максимальный элемент максимальных взаимно непересекающихся цепей фрагментов), вычислительная сложность алгоритма – $O\left(|G|^{2}\cdot|M|^{2}\right)$,

– LatticeBox (строит решетку понятий для фрагмента), вычислительная сложность алгоритма – $O\left(|FC|\cdot|G|^{2}\cdot|M|\right)$,

– LatticeContext (собирает решетки фрагментов в решетку исходного контекста), вычислительная сложность алгоритма – $O\left(|\Omega|\cdot|FC|\cdot|G|^{2}\cdot|M|\right)$.

Реализация вышеописанных алгоритмов была проведена при помощи языка программирования Python версии 3.7 в среде разработки JetBrains PyCharm Community Edition 2018. Эксперименты проводились на ноутбуке с процессором Intel Pentium CPU B960 @ 2.20 GHz и ОЗУ 4 ГБ. Программа, реализующая данные алгоритмы, используя алгоритм Close-by-One, находит формальные понятия каждого фрагмента, полученного при однократном разложении, а после восстанавливает множество всех формальных понятий исходного контекста.

4. Вычислительные эксперименты

Было проведено три серии экспериментов. Результаты экспериментов показывают высокую вычислительную сложность задачи нахождения множества всех формальных понятий для бинарного и тем более нечеткого контекста: при увеличении контекста примерно в 5 раз, время поиска понятий возрастает в десятки раз быстрее. Высокая вычислительная сложность делает востребованными методы снижения размерности формального контекста и исследования в области нечеткого анализа формальных понятий и выдвигает гипотезу об использовании на нечетком формальном контексте метода фрагментации и дефрагментации бинарного контекста.

Первая серия экспериментов – анализ результативности алгоритмов Close-by-One, NextClosure и метода Монгуш на случайных бинарных контекстах. Исходные данные – случайно сформированные бинарные контексты размерности 50×10 и 100×20 различной плотности. Результаты первой серии экспериментов приведены в табл. 1.

Таблица 1

Результат работы программ на случайных бинарных контекстах

σ	$\lvert\Omega\rvert$	Алгоритм	$\lvert FC\rvert$	Время, с	σ	$\lvert\Omega\rvert$	Алгоритм	$\lvert FC\rvert$	Время, с
0.1	27	Close-by-One	25	0,196	0.6	98	Close-by-One	454	3,134
		NextClosure	25	0,142			NextClosure	454	2,223
0.2	43	Close-by-One	61	0,477	0.7	103	Close-by-One	755	4
		NextClosure	61	0,3			NextClosure	755	3
0.3	74	Close-by-One	93	0,38	0.8	105	Close-by-One	976	5,1
		NextClosure	93	0,477			NextClosure	976	4
0.4	100	Close-by-One	175	1,144	0.9	83	Close-by-One	896	2,3
		NextClosure	175	0,8			NextClosure	896	2,1
0.5	132	Close-by-One	314	2,1					
		NextClosure	314	1,5					

Усредненный результат по 10 экспериментам на случайных бинарных контекстах 100×20 приведен в табл. 2.

Таблица 2

Результат работы программ на десяти случайных бинарных контекстах 100×20

σ	$\lvert\Omega\rvert$	Алгоритм	$\lvert FC\rvert$	Время, с
0,5	862	Close-by-One	13334	484,6
		NextClosure	13334	278

Вторая серия экспериментов – анализ результативности алгоритмов Close-by-One, NextClosure и метода Монгуш на реальных контекстах. Исходные данные – девять бинарных контекстов, характеризующих рынки B2B (business to business, рынок коммерческих организаций), B2C (business to consumer, потребительский рынок) и B2G (business to government, рынок государственных и муниципальных закупок). Объектами контекстов являются 30 специально отобранных экспертов, а признаками – 13 критериев, выполнение которых оценивали эксперты.

В табл. 3 приведены результаты работы алгоритмов. С их помощью было найдено множество формальных понятий для всех девяти контекстов, которое может использоваться для анализа цифровизации предложенных рынков. Каждый контекст был однократно (при $k = 1$) разложен на фрагменты и также получено множество формальных понятий, полностью совпадающее с множеством, полученным алгоритмами NextClosure и Close-by-One.

Таблица 3

Результат работы программ на контекстах 30×13

№	Год	Рынок	Плотность контекста	$\lvert\Omega\rvert$	Алгоритм	$\lvert FC\rvert$	Время, с
1	2016	B2B	0.32	58	Close-by-One	98	0.312
					NextClosure	98	0.28
2		B2C	0.3154	50	Close-by-One	71	0.25
					NextClosure	71	0.2
3		B2G	0.454	42	Close-by-One	64	0.2184
					NextClosure	64	0.1872
4	2017	B2B	0.3282	71	Close-by-One	98	0.2964
					NextClosure	98	0.2
5		B2C	0.3513	62	Close-by-One	84	0.468
					NextClosure	84	0.3432

№	Год	Рынок	Плотность контекста	$\lvert\Omega\rvert$	Алгоритм	$\lvert FC\rvert$	Время, с
6		B2G	0.4641	51	Close-by-One	61	0.2184
					NextClosure	61	0.156
7		B2B	0.36154	57	Close-by-One	108	0.312
					NextClosure	108	0.2184
8	2018	B2C	0.41	69	Close-by-One	91	0.2496
					NextClosure	91	0.1872
9		B2G	0.495	51	Close-by-One	75	0.1872
					NextClosure	75	0.14

Третья серия экспериментов – проверка работы обобщенного алгоритма Гантера на нечетких формальных контекстах. Исходные данные – случайный нечеткий формальный контекст 15×5 с количеством степеней истинности $\lvert L\rvert=10$. Результат эксперимента приведен в табл. 4.

По результатам третьей серии экспериментов можно сделать вывод о высокой вычислительной сложности задачи нахождения нечетких формальных понятий из нечеткого контекста. Из таблицы 4 видно, что на сложность и конечный результат поиска нечетких понятий влияет выбор замыкания и сопряженных операций.

Таблица 4

Результат работы программы на случайном нечетком формальном контексте с замыканием в виде идентичности

Размерность контекста	Сопряженные операции	$\lvert FC\rvert$	Время, с
15×5	Лукасевич	2686	32,3
	Гедель	1671	10,3
	Произведение	16643	124,4

Заключение

В данной работе был проведен обзор соответствующей литературы, исследованы методы и алгоритмы решения задачи поиска формальных понятий для бинарных и нечетких контекстов, изучен метод Монгуш фрагментации и дефрагментации бинарных формальных контекстов. На основе изученных материалов были разработаны программы, реализующие алгоритмы Close-by-One, NextClosure и его обобщение на нечеткие контексты, частично реализован метод Монгуш. С помощью данных программ были проведены три серии экспериментов, демонстрирующих высокую вычислительную сложность задачи поиска формальных понятий для бинарных и тем более нечетких контекстов, что говорит о важности методов, позволяющих снизить сложность рассматриваемой задачи. В дальнейшем планируется более глубокое изучение метода Монгуш, а также его обобщение на нечеткие формальные контексты.

ЛИТЕРАТУРА

1. *Kuznetsov S.O.* Mathematical aspects of concept analysis / S.O. Kuznetsov // Journal of Mathematical Sciences. – 1996. – Vol. 80. – № 2. – P. 1654-1698.

2. *Игнатов Д.И.* Анализ формальных понятий: от теории к практике / Д. И. Игнатов, Р. Э. Яворский // Анализ изображений сетей и текстов. – Екатеринбург, 2012. – С. 1-12.

3. *Игнатов Д.И.* Анализ формальных понятий и его приложения / Д. И. Игнатов, Й. Пульманс // Инженерия знаний и технологии семантического ВЕБа / Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики. – Санкт-Петербург, 2011. – № 2. – С. 9-18.

4. *Ganter B.* Formal Concept Analysis: Mathematical Foundations / B. Ganter, R. Wille // Springer, 1999. – P. 284.

5. *Ganter B.* Formal Concept Analysis: Foundations and Applications / B. Ganter, G. Stumme, R. Wille // Springer-Verlag / Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence. – 2005. – № 3626. – P. 151-167.

6. *Биркгоф Г.* Теория решеток: пер. с англ. / К. Биркгоф. – М.: Наука. Главная редакция физико-математической литературы, 1984. – 568 С.

7. *Офицеров В.П.* Нечеткий анализ формальных понятий при разработке онтологий / В.П. Офицеров, С. В. Смирнов // Онтология проектирования. – 2017. – № 4 (26).
8. *Монгуш Ч.М.* Разработка метода и средств фрагментации и дефрагментации формальных контекстов: дис. … канд. физ.-мат. наук: 05.13.17 / Ч. М. Монгуш. – Красноярск, 2019. – 105 С.
9. *Ольшанский Д.Л.* Подбор алгоритма для параллельной реализации метода сходства в интеллектуальных ДСМ-системах // Научно-техническая информация. Серия 2. Информационные процессы и системы. – Москва, 2015. – С. 1-12.
10. *Kuznetsov S.O.* Comparing performance of algorithms for generating concept lattices / S. Kuznetsov, S. Obiedkov // Stability of a Formal Concept. – 2002. – № 15. – P.189-216.
11. *Belohlavek R.* Algorithms for fuzzy concept lattices. – 2002.
12. *Belohlavek R.* Galois connections with hedges / R. Belohlavek, V. Vychodil, T. Funiokova.
13. *Glodeanu C.V.* Attribute Exploration in a Fuzzy Setting / C. V. Glodeanu // CEUR Workshop Proceedings. – 2012. – P. 114-128.

GMDH СЕТИ С ОБРАТНОЙ СВЯЗЬЮ

Гузеев И.В., Кабанова Т.В.
Томский государственный университет
guzeev98@gmail.com, tvk@bk.ru

Введение

Вопрос о том, какую архитектуру нейронной сети выбрать для заданного набора данных для достижения наибольшего значения качества согласно заданной метрике [1] остается открытым ещё со времен их первого применения [2]. Выбор оптимальной модели осуществляется посредством произвольного перебора элементов счетного множества. Процесс перебора основывается на эмпирическом опыте и априорных умозаключениях исследователя, что является потенциальным источником грубых ошибок ещё на этапе предварительного анализа. Кроме того, не совсем ясно, когда стоит остановить перебор, максимальный показатель метрики качества зачастую является скорее поводом для подозрения в переобученности модели, чем стоп-сигналом, показывающим успех перебора, очевидно, что и низкие показатели недопустимы, но не очевидно, какие показатели считать таковыми. Такой подход может привести к необходимости перебора структур и определения допустимого интервала значений метрики, на основании субъективного опыта исследователя и, следовательно, к повышению конечной стоимости готовой модели и возможной потере её конечного качества. В итоге перебор всех гипер-параметров [3] и структур сводится к крайне дорогой с точки зрения вычислений задаче, а отсутствие внятного условия остановки и вовсе сводит задачу к нерешаемой. Откуда следует, что нам необходимо использовать способы, упрощающие сложность исходной задачи, а следовательно, количество итераций. Множество таких алгоритмов образуют так называемое семейство “жадных” алгоритмов или эвристик [4].

В данной статье будет описан эвристический алгоритм [4] поэтапного построения нейронной сети с обратной связью, где на каждой итерации алгоритма будет производится наиболее выгодный с точки зрения некой заданной метрики выбор, а само построение будет считаться завершенным когда рост показателя метрики качества модели остановится. Конечно же, данный подход не сможет гарантировать, что полученная таким путем модель действительно будет является наилучшей по некой метрике оценки качества. Однако, несомненным плюсом такого алгоритма будет очевидно меньшее количество начальных параметров, а также отсутствие необходимости решения проблемы выбора архитектуры.

Постановка задачи

Как говорилось ранее, выбор оптимальной сети чаще всего сводится к перебору параметров, основанном на эмпирическом опыте. В ходе своих исследований было изучено несколько видов нейронных сетей [5], а именно: глубокая нейронная сеть пря-

мого распространения (DFFNN), сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN). При использовании DFFNN исследователь вынужден заранее определить сложность искомой зависимости между входными образцами и выходными значениями, и на основе этих суждений искать необходимое количество слоев и узлов; CNN разбивают исходное сложное представление на множество более простых представлений, как это разбиение происходит – задается заранее, на этапе выбора архитектуры. Задача усложняется при попытке использовать комбинацию из CNN и DFFNN; Схожие проблемы встречаются и при использовании RNN.

Можно заметить, что все рассмотренные виды сетей принадлежат к множеству так называемых нейронных сетей глубокого обучения. Такой выбор был обоснован тем, что на сегодняшний день сети с глубоким обучением являются популярным решением в ряде проблем [6], а их размер усложняет задачу выбора оптимальной архитектуры для конкретного набора данных. Ведь чем больше узлов, тем больше времени на их обучение нам потребуется и тем сложнее добиться их полной оптимизации; в противовес, малое количество узлов может привести к упущению важных зависимостей, ориентируясь на которые, модель и определяет конечное предсказание. В итоге исследователь оказывается зажат двумя факторами: мы не хотим пропустить важные зависимости и мы не хотим обучать лишние веса (так как чаще всего это ухудшит оптимизацию всех весов сети, а следовательно ухудшит качество модели).

Большинство статей, связанных с улучшением показателей модели, работают с фиксированной архитектурой и нацелены на оптимизацию иных параметров сети посредством случайного или последовательного перебора или же с использованием эвристического подхода [7,8]. Хотя, исходя из нашего анализа моделей, следует, что наибольшей проблемой является именно выбор самой архитектуры, так как перебор всевозможных топологий является практически нерешаемой задачей.

Нашей целью будет попытка абстрагировать исследователя от этого выбора, а также существенно облегчить возможный перебор архитектур сети, путем избавления от необходимости задавать архитектуру напрямую, вместо этого сеть будет задаваться дискретными гипер-параметрами. В этих целях будет предложен алгоритм построения самоорганизующееся GMDH-сети с обратной связью [1].

Теоретическая часть

Анализ оригинального алгоритма

Для начала рассмотрим оригинальный алгоритм построения GMDH-сети с обратной связью за авторством Тадаши, используемый им для решения задачи поставленной в статье "GMDH-type Neural Networks with a Feedback Loop and their Application to the Identification of Large-spatial Air Pollution Patterns"[2] [9]. Основной алгоритм может быть разделен на две части:

1. Построение первого скрытого слоя

Сперва разделим данные на тренировочные и тестовые. Затем строится входной слой.

<Входной слой>

$$u_i = x_i,$$

где x_i, $i = \overline{1, p}$ – входные данные об образце, p – количество входных данных каждого наблюдения. Для входного слоя исходящий сигнал в точности равен входному.

<Скрытый слой>

[1] GMDH (Group method of data handling) – семейство индуктивных алгоритмов, предназначенных для решения задач автоматического компьютерного моделирования.

[2] Изначально алгоритм написан на английском языке, но для облегчения понимания, мы предоставим переведенную нами версию с сохранением всех введенных автором обозначений и максимально близкими терминами.

Сперва строятся всевозможные комбинации входных переменных без повторений и перестановок (Как правило при этом задается количество элементов в каждой комбинации – r, то есть всего будет C_p^r комбинаций) Затем, используя заранее заданное множество архитектур нейронов, для каждой из этих комбинаций входных переменных производится автоматический поиск оптимального нейрона с точки зрения метрики AIC [10]:

$$\mathrm{AIC} = n\log_e S_m^2 + 2(m+1) + C\ ,\ S_m^2 = \sum_{\alpha=1}^{\infty} \frac{(\varphi_\alpha - z_a)^2}{n}\ .$$

где φ_α – истинное выходное значение для образца α, z_α – предсказанное моделью значение для образца α, n – количество образцов в выборке, m – количество узлов в сети.

Также среди каждого оптимального нейрона производится отбор наилучших весов с помощью пошаговой регрессии. После чего с её же помощью производится и отбор оптимальных нейронов. Выходные сигналы отобранных нейронов обозначим как y_k и будем называть независимым переменными.

<Выходной слой>

Выходной слой содержит только один узел, который принимает сигналы от всех нейронов скрытого слоя, а активационная функция имеет вид:

$$\varphi^* = a_0 + \sum_{i=1}^{L} a_i y_i\ ,$$

где L – количество узлов скрытого слоя, а веса a_j были получены в процессе отбора узлов для скрытого слоя через пошаговую регрессию. Выходное значение $\varphi*$ задает обратную связь для всех последующих слоев и будет использоваться в комбинации с входными значениями в дальнейшем построении сети.

2. Построение второго и последующих скрытых слоев

На второй и последующих итерациях мы конкатенируем выходное значение уже полученной нами на прошлом этапе сети $\varphi*$ с входными переменными. Далее происходят аналогичные первому этапу вычисления, но теперь комбинации входных переменных должны включать в себя $\varphi*$.

Итерации продолжаются, пока показания метрики AIC улучшаются.

Используя этот эвристический алгоритм, мы можем полностью отойти от проблемы выбора архитектуры сети. Но, в тоже время, полностью лишаем себя какой-либо свободы выбора.

В ходе исследований данного алгоритма, мною предложены следующие изменения, призванные снизить уровень конкретизации с сохранением возможности использовать исходный алгоритм:

Предложенные параметры

1. Метрика качества узлов и их весов, метрика качества модели

В вопросе выбора оптимальных узлов и их весов, а также в определении текущего качества модели, автор использует AIC. Мною предлагается предоставить исследователю выбор метрики для каждого из этих процессов. Поэтому вводятся параметры ρ_{weights}, ρ_{nodes}, ρ_{model} – метрики, используемые при выборе весов, узлов и определения качества модели соответственно. Очевидно, что мы легко можем вернуться к исходному алгоритму при $\rho_{\text{weights}} = \rho_{\text{nodes}} = \rho_{\text{model}} = \mathrm{AIC}$.

2. Максимальная степень полинома, преобразующего входные сигналы

В [9] Тадаши определяет нейроны как пару функций, первая преобразует n-мерный вектор в скаляр (её мы будем называть функцией преобразования входного сигнала), вторая формирует выходной сигнал из этого скаляра (её мы будем называть активационной функцией).

Предложенные Тадаши нейроны всегда используют полиномы для преобразования входного сигнала, поэтому предлагается обобщить нейрон. Теперь узел будет задавать-

ся лишь максимальной степенью полинома *D* (общей для всех нейронов) и активационной функцией.

Данный параметр не позволит нам вернуться к исходному алгоритму ни при каких значениях *D*, но при идеальном отбирающем правиле[3] и достаточно больших *D* исходные полиномы будут использоваться в тех случаях, когда они являются оптимальными.

3. Множество активационных функций

В исходном алгоритме мы имеем лишь небольшое количество заранее заданных узлов, мною предлагается позволить исследователю самому задавать это множество. Причем, в силу введения параметра из п.2, узлы отличаются только активационной функцией и, соответственно, ей и задаются, поэтому исследователю достаточно лишь определить множество активационных функций *F*. Условие возвращения к оригинальному алгоритму практически аналогично п.2, но при этом *F* должно включать в себя предлагаемые автором функции.

4. Отбирающее правило для подбора наилучших весов и нейронов

Для подбора наилучших нейронов и их весов автор использует пошаговую регрессию, однако мы можем легко предоставить выбор отбирающего правило исследователю, для этого введем $<_{\text{nodes}}$,$<_{\text{weight}}$ − отбирающие правила для узлов и весов соответственно. Очевидно, что если мы зададим эти правила равные пошаговой регрессии, мы получим исходный алгоритм предложенный автором.

5. Граница изменения метрики качества модели

Также автор предлагает останавливать итерации при первом же ухудшении показателя качества модели, мною предлагается использовать иной подход и позволить метрике качества ухудшаться в пределах некого значения ε. Будем называть это значение допустимым отклонением, а условие остановки работы алгоритма будет изменено с $\text{AIC}_{\text{new}} < \text{AIC}_{\text{old}}$ на $\text{AIC}_{\text{new}} < \text{AIC}_{\text{old}} + \varepsilon$.[4] Очевидно, что исходный алгоритм может быть получен при $\varepsilon = 0$.

Полученный нами обобщенный алгоритм построения GMDH-сети с обратной связью может быть описан на псевдокоде следующим образом:

Исходные параметры: X - матрица $n \times m$, где каждая строка - информация о конкретном образце, столбец - конкретная характеристика образцов; Y - вектор выходных значений состоящий из n координат
Результат: Обученная GMDH-сеть с обратной связью
M - цепь подмоделей
$\rho_{model,new} = 0$ - качество модели на текущей итерации;
$\rho_{model,old} = 0$ - качество модели на предыдущей итерации;
$X^* = X$ - совокупность текущих характеристик системы;
L = m;
до тех пор, пока $\rho_{model,new}$ *не хуже* $\rho_{model,old}$ *больше чем на* ε **выполнять**
  C = всевозможные комбинации размером r текущих характеристик системы ;
  P = $\mathbb{C} \times \mathcal{F}$ - всевозможные пары из комбинаций характеристик и активационных функций (c_k, f_k);
  N = $\emptyset$ - Множество всех возможных узлов скрытого слоя ;
  цикл $i = 0;\ i < len(\mathbb{P});\ i = i + 1$ **:**
    $Node_i$ = Узел, формирующий сигнал на основании характеристик c_i посредством активационной функции f_i;
    Отбираем оптимальную с точки зрения $\rho_{weights}$ архитектуру нейрона с помощью $\Re_{weight}$;
    Добавляем $Node_i$ в N;
  конец
  H = Оптимальное с точки зрения ρ_{nodes} подмножество N полученное с помощью $\Re_{nodes}$;
  Добавляем в M новую подмодель со скрытым слоем H и выходных сигналом Ф*;
  $X^* = X \oplus \Phi^*$ - конкатенация каждого образца с его предсказанным моделью значением;
  L = m + 1;
  $\rho_{model,old} = \rho_{model,new}$;
  $\rho_{model,new}$ = качество новой модели относительно метрики качества ρ_{model};
конец

Введенные нами параметры 1, 2, 3 лишь расширяют множества возможных моделей, а для подбора параметров из пунктов 4 и 5 уже существуют готовые решения (как прямой перебор, так и вероятностный подход [11]).

[3] например, Прямой перебор всех комбинаций слагаемых

[4] Здесь мы используем конкретную метрику AIC, так как для разных метрик условия сравнения могут быть различны, в данном случае меньшее значение означает лучшее качество

Структура модели

Полученная модель может быть представлена как цепочка из подмоделей (рис. 1). Первый узел этой цепи формирует свои предсказания на основании лишь входных данных о наблюдении, следующие же узлы в свою очередь уже учитывают не только начальные входные данные, но и предсказания предыдущей модели в цепи. На каждой итерации к цепи прибавляется ещё один узел, представляющий собой подмодель – нейронную сеть с один скрытым слоем.

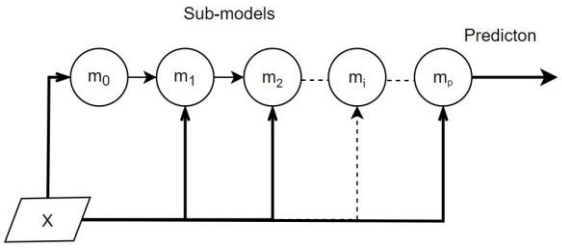


Рис. 1. Представление модели в виде цепочки нейронных сетей

Структура конкретной подмодели не может быть известной заранее. Но мы можем описать абстрактную структуру (рис. 2) для получения базового представления об устройстве изучаемой модели.

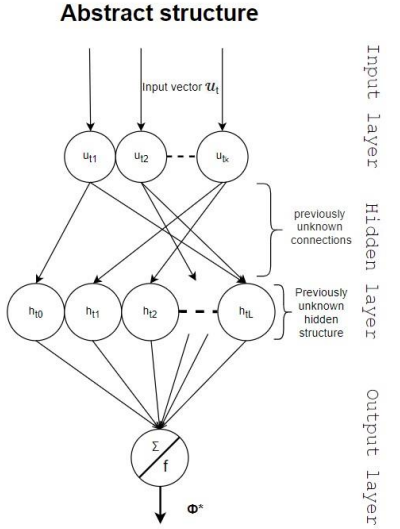


Рис. 2. Общая структура подмодели (нейронная сеть с одним скрытым слоем)

Как и было описано ранее, на вход нейронной сети поступает сигнал в виде вектора U: для первой подмодели он представляет собой вектор данных о наблюдении X; для второй и последующей подмодели вектор U получается путем конкатенации вектора X и выходного сигнала предыдущей подмодели m_{t-1}.

В скрытом слое H_t находятся нейроны $h_{t,i},\ i=\overline{1,L}$ (где L – количество узлов скрытого слоя), чья совокупность наилучшим образом описывает систему. Их связи автоматически определены на этапе построения всех возможных нейронов. Веса функции преобразования входных значений отобраны так, чтобы каждый отдельный нейрон описывал систему наилучшим образом. Узлы скрытого слоя всех кроме первой подмодели должны быть связаны с выходным сигналом предыдущей подмодели.

Выходной узел представляет собой линейную комбинацию выходных сигналов узлов скрытого слоя, веса линейной комбинации находятся автоматически при поиске наилушей совокупности нейронов скрытого слоя. Выходной сигнал подмодели обозначается как φ∗ и используется при построении следующей подмодели или как конечный результат работы всей модели.

Практическая часть

Описание параметров

Параметры модели были заданы согласно автору статьи, за исключением того, что метрикой качества модели был задан коэффициент детерминации (R^2)

$$R^2 = 1 - \sum_{i=1}^{N} \frac{\left(y_i - \hat{y}_i\right)^2}{\left(y_i - \bar{y}_i\right)^2}.$$

Также рассматриваются две модели: первая использует пошаговую регрессию для отбора весов GMDH(SR), вторая – полный перебор – GMDH(BF).

Информация о наборе данных

Для анализа работы алгоритма на реальных данных было решено использовать набор данных о продажах жилья в округе Кинг, США. Недвижимость описывается 21 столбцом содержащим как информацию о площади жилья, так и оценку внешнего вида[5].

Набор данных не содержит пропущенных значений и каждая строка полностью описана каждым из столбцов. В ходе изучения набора данных было решено исключить столбцы с информацией о почтовом адресе, дате продажи и номере в таблице данных. В итоге каждый образец описывается входным вектором из 17 координат (характеристик недвижимости) и одним выходным значением - ценой продажи. Для улучшения качества обучения была проведена нормировка данных по формуле

$$x_{\max}^{(k)} = \max_{1 \le i \le N}\left(x_{k,i}\right),\ x_{\min}^{(k)} = \min_{1 \le i \le N}\left(x_{k,i}\right),\ \hat{x}_{k,i} = \frac{x_{k,i} - x_{\min}^{(k)}}{x_{\max}^{(k)} - x_{\min}^{(k)}},$$

N – размер выборки.

Поведение модели на наборе данных

Процесс обучения моделей описывается с помощью двух рядом стоящих графиков, левый показывает значения метрики на каждой итерации алгоритма, правый отображает сложность системы (суммарное количество узлов сети). Для изучения процесса обучения модели, значение максимального ухудшения метрики ε было специально завышено и поэтому, работа алгоритма заканчивается не по стандартному критерию остановки, а по достижению максимального количества итераций.

При обучении на заданном наборе данных GMDH(SR) показывает большие скачки роста качества на первых итерациях, а затем лишь малое приращение. Рост сложности системы сохраняет практически линейный характер. Качественное падение при переходе на тестовую выборку достигает значений в 0.02%.

При реальной работе алгоритма, процесс бы был остановлен на четвертой итерации (рис. 3).

[5] Используемый набор данных может быть найден по этой ссылке https://www.kaggle.com/harlfoxem/housesalesprediction, описание всех столбцов: https://www.kaggle.com/harlfoxem/housesalesprediction/discussion/82135

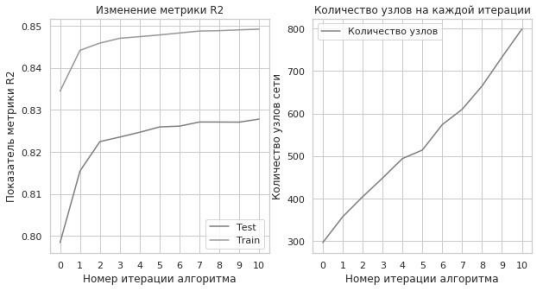


Рис. 3. Поведение GMDH(SR) на реальном наборе данных

Использование модели GMDH(BF) дает небольшие улучшения показателей метрики и снижение конечной сложности системы: 786 узлов у GDMH(BF) против 799 у GMDH(SW). При этом уже на первой итерации были практически достигнуты конечные результаты GMDH(SW). Рост сложности системы также сохраняет линейных характер, а падение при переходе на тестовую выборку лежит в предлах 0.02% (рис. 4).

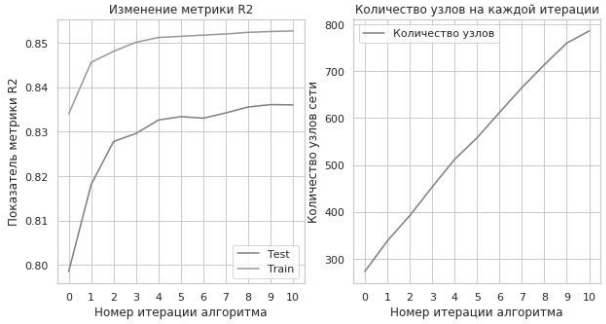


Рис. 4. Поведение GMDH(SF) на реальном наборе данных

Исходя из обоих графиков можно сделать вывод, что прямой перебор позволяет улучшить качество модели как в смысле точности, так и в смысле сложности системы. Ценой этого будет повышенное время построения модели, поэтому при наличии малых вычислительных мощностей рекомендуется использовать эвристические подходы, к примеру пошаговую регрессию.

На рис. 5 изображено четыре графика сравнения. Первый столбец содержит показатели моделей на обучающей выборке, второй на тестовой. Каждая строка соответствует заданной метрике: первая – коэффициент детерминации, вторая – средняя абсолютная ошибка. Сравнение производится с регрессионным методом опорных векторов (SVR), деревом регрессии(DecisionTreeRegressor) и линейной регрессией(LinearRegression). В работе используются готовые реализации на языке python библиотеки sklearn[6].

[6] Библиотека и подробное описание моделей могут быть найдены на официальном сайте: https://scikit-learn.org/

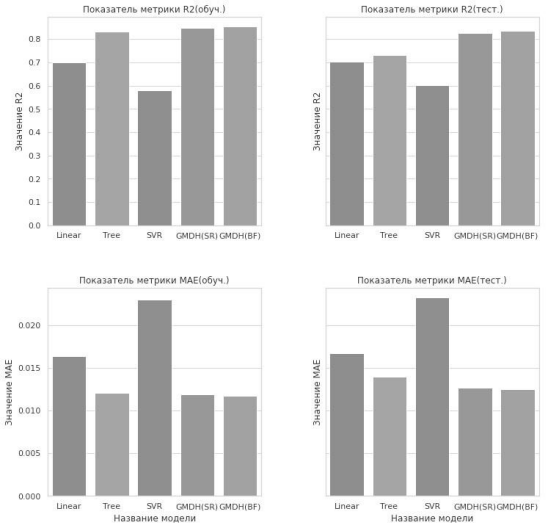


Рис. 5. Сравнение моделей на реальном наборе данных

Из графиков видно, что GMDH показывает наилучшие значения метрик качества. Также заметно значительно меньшая степень переобученности, т.к. показатели метрик на тестовой и обучающей выборке у моделей GMDH практически совпадают. Среди предложенных моделей конкурентоспособной оказалось лишь дерево регрессии, другие модели имеют значительно меньшее качество.

Заключение

В ходе проведенной нами работы был рассмотрен оригинальный алгоритм Тадаши по построению GMDH сети с обратной связью и разработан более общий алгоритм с расширенным количеством гипер-параметров. Полученный алгоритм в ходе своей работы строит сети с высоким качеством получаемых моделей даже при больших уровнях шума.

В результате проделанной нами работы мы свели задачу поиска оптимальной структуры сети к перебору параметров представляющих собой конечные множества и скаляры. Проблема перебора таких параметров, в отличии от перебора возможных структур, может быть решена уже существующими методами (как полный перебор всех вариантов, так и вероятностный подход). Использование GMDH подхода снижает уровень вкладываемой исследователем ошибки, так как избавляет его от проблемы выбора структуры сети на основании своих априорных суждений об исходном наборе данных и эмпирическом опыте.

ЛИТЕРАТУРА

1. *Hossin M. и M.N. Sulaiman.* “A review on evaluation metrics for data classification evaluations”. в: International Journal of Data Mining Knowledge Management Process (IJDKP) 5.2 (March 2015), с. 1—11.

2. *Clifford A. Pickover.* Artificial Intelligence: An Illustrated History: From Medieval Robots to Neural Networks (Sterling Illustrated Histories). Sterling Publishing Company, 2019.

3. *Kiam Choo.* “Learning Hyperparameters for Neural Network Models Using Hamiltonian Dynamics”. в: (2000).

4. *Tim Roughgarden.* Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming. Soundlikeyourself Publishing, LLC, 2019.

5. *Saptarshi Sengupta и др.* A Review of Deep Learning with Special Emphasis on Architectures, Applications and Recent Trends. 2019. eprint: arXiv:1905.13294.

6. *Vivienne Sze и др.* Efficient Processing of Deep Neural Networks: A Tutorial and Survey. 2017. eprint: arXiv:1703.09039.

7. *Afef Abdelkrim Sehla Loussaief.* “Convolutional neural network hyper-parameters optimization based on genetic algorithms”. в: International Journal of Advanced Computer Science and Applications 9.10 (2018), с. 252—266.

8. *Nils Reimers и Iryna Gurevych.* Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. 2017. eprint: arXiv:1707.06799.

9. *A.S. Pandya T. Kondo.* “GMDH-type neural networks with a feedback loop and their application to the identification of large-spatial air pollution patterns”. в: SICE 2000. Proceedings of the 39th SICE Annual Conference. International Session Papers (IEEE Cat. No.00TH8545) (Iizuka, Japan, 28—28 июля 2000). 2000.

10. *Tadashi Kondo и Junji Ueno.* “Revised gmdh-type neural network algorithm with a feedback loop identifying sigmoid function neural network”. в: International Journal of InnovativeComputing, Information and Control 2.5 (2006), с. 985—996.

11. *Jasper Snoek и др.* Scalable Bayesian Optimization Using Deep Neural Networks. 2015. eprint: arXiv:1502.05700.

ПРОГНОЗИРОВАНИЕ ОБЪЕМОВ ПОТРЕБИТЕЛЬСКОГО КРЕДИТОВАНИЯ В КОММЕРЧЕСКОМ БАНКЕ

Дарибаева Н.Т.
Филиал «Восход» Московского Авиационного Института
nurz_dar7@mail.ru

Введение

Кредитование физических лиц является одним из основных направлений деятельности коммерческих банков, а прогнозирование динамики объемов кредитования позволяет получить информацию о возможных состояниях кредитного сектора в будущем. Существуют различные методы и модели прогнозирования экономических процессов, и зачастую на практике бывает тяжело выбрать конкретную адекватную модель для анализа имеющихся данных. В данной статье рассмотрена адаптивная модель Брауна и трендовые модели на основе двух кривых роста для прогнозирования объемов кредитования.

1. Постановка задачи

Необходимо провести анализ объемов кредитования и осуществить их прогнозирование на два месяца вперед. Данные об объемах кредитования приведены в табл. 1.

Таблица 1

Объемы кредитования

Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
300	525	650	700	200	325	475	600	650	800	800	890

Схема экономико-математического исследования:

1. Построить график объемов продаж.
2. Построить адаптивную модель Брауна и трендовые модели на основе двух кривых роста.
3. Оценить адекватность построенных моделей и выбрать оптимальную из них. Построить доверительный интервал и оценить точность модели.
4. Осуществить прогнозирование объемов кредитования.

2. Трендовые модели на основе кривых роста

Прогнозирование на основе временных рядов экономических показателей относится к равномерным методам прогнозирования, базирующихся на экстраполяции. При этом учитываются два предположения:

1) временной ряд экономических показателей имеет тренд;
2) общие условия, определяющие развитие показателей в прошлом останутся без существенных изменений в течение периода упреждения – отрезок времени, на которой разрабатывается прогнозирование.

Аналитические методы выравнивания сводятся к выбору конкретных кривых роста и определения их параметров.

Кривая роста – функция, аппроксимирующая временной ряд. В экономике наиболее часто используются кривые роста, позволяющие описывать процессы 3-х видов:

1) без предела роста;

2) с пределом роста, без точки перегиба;

3) с пределом роста и точкой перегиба.

Для описания первого типа используются полиномиальные кривые роста, для описания процесса с пределом роста используются кривая Джонсона, функция Торнквиста и модифицированная экспонента. Для описания процессов третьего типа используются S-образные кривые: кривая логиста (функция Перла-Рида) и функция Гомперуа [1].

Параметр полиномиальных кривых оцениваются методом наименьших квадратов. Параметры экспоненты и S-образных кривых находят более сложными методами. Для простой экспоненты предварительно логарифмируют по некоторому основанию, затем применяют метод наименьших квадратов. Для получившейся линейной функции находят логарифмы параметров моделей, а затем и сами параметры.

3. Адаптивные методы прогнозирования

При краткосрочном прогнозировании, а также прогнозировании в ситуации изменения внешних условий, когда наиболее важными являются последние реализации исследуемого процесса, эффективными оказываются адаптивные методы.

Адаптивные модели способны быстро приспосабливать параметры и структуру к новым условиям. Наибольший вес при оценке параметров придается последнему наблюдению. Наиболее часто используются две модели, основанные на схеме скользящего среднего: метод Брауна и Хольта [2].

Эти модели представляют процесс развития, как линейную тенденцию с постоянно изменяющимися параметрами. Прогнозная оценка $y_p(t,k)$ уровня ряда $y(t+k)$ вычисляется в момент времени *t* на *k* шагов вперед:

$$y_p(t,k) = A_0(t) + A_1(t)k\,,$$

где $A_0(t)$ – оценка текущего *t*-го уровня; $A_1(t)$ – оценка текущего прироста.

Далее определяется величина их расхождения:

$$e(t+1) = Y(t+1) - Y_p(t,1)\,.$$

В соответствии с этой величиной корректируются параметры модели. В модели Брауна модификация осуществляется следующим образом:

$$A_0(t) = A_0(t-1) + A_1(t-1) + \left(1-\beta^2\right)\cdot e(t)\,,\ A_1(t) = A_1(t-1) + \left(1-\beta^2\right)\cdot e(t)\,,$$

где $0<\beta<1$, β – коэффициент дисконтирования данных, $e(t)$ – ошибка прогнозирования уровня $y(t)$, вычисленная в момент времени $(t-1)$ на один шаг вперед.

В моделях Хольта коэффициенты модифицируются следующим образом:

$$A_0(t) = A_0(t-1) + A_1(t-1) + \alpha_1\cdot e(t)\,,\ A_1(t) = A_1(t-1) + \alpha_1\cdot\alpha_2\cdot e(t)\,,$$

где α_1, α_2 – коэффициенты сглаживания или адаптации, изменяющиеся от 0 до 1. Параметры вычисляются последовательно от уровня к уровню, и их значения для последнего уровня определяют окончательный вид модели. Начальные значения параметров оцениваются методом наименьших квадратов на основе нескольких уровней ряда.

4. Сравнение численных результатов

Качество модели определяется ее адекватностью исследуемому процессу и точностью. Модель является адекватной, если ряд остатков обладает свойствами случайности, независимости последовательных уровней, нормальности распределения и равенства нулю средней ошибки.

Для этого выполняется проверка верности четырех свойств случайности для отклонений, полученных по линейной, степенной модели и модели Брауна.

Прогнозное значение изучаемого показателя включает точечный и интервальный прогноз.

На рис. 1 представлены результаты моделирования для линейной модели.

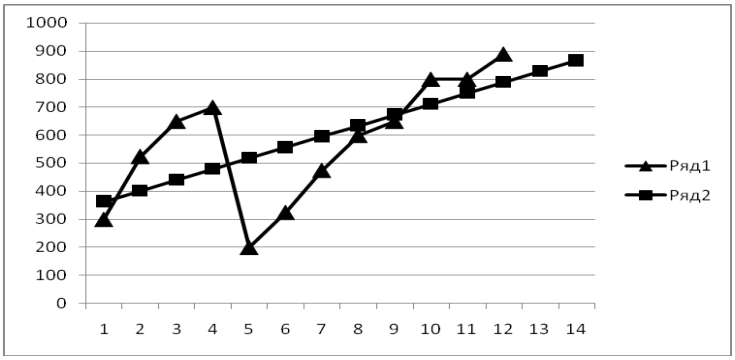


Рис. 1. Линейная модель

Ряд 1 представлен исходным рядом данных, ряд 2 – смоделированные значения.

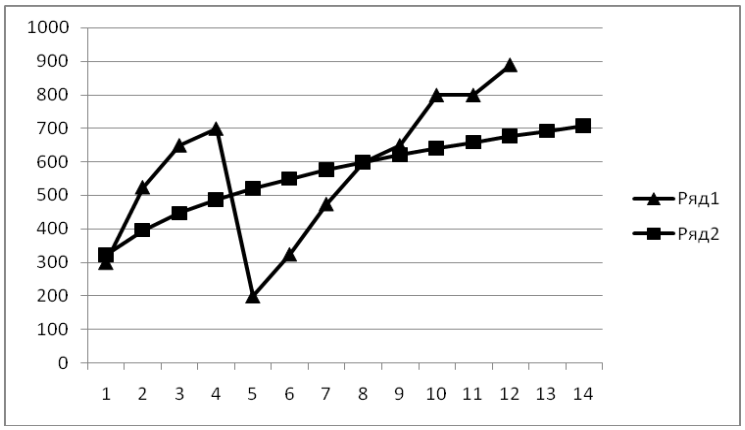


Рис. 2. Степенная модель

На рис. 3 представлены результаты численного эксперимента для модели Брауна.

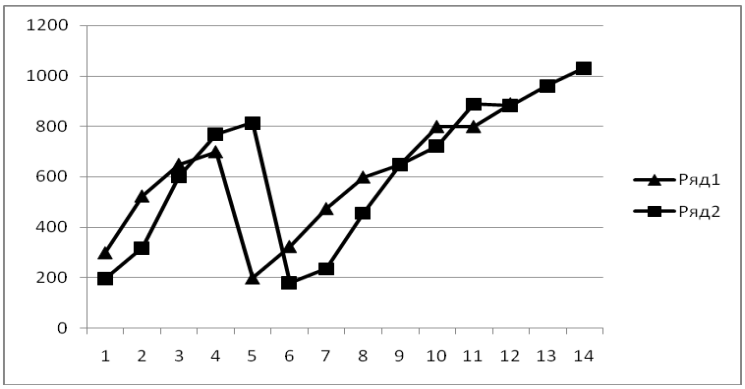


Рис. 3. Модель Брауна

Так как в модели Брауна не учитываются тренд и сезонные изменения, на графике можно заметить, что прогнозные значения на январь и февраль (точки 13 и 14 на оси абсцисса) увеличиваются, вместо спада. Фактически, январь-февраль являются не самыми эффективными с точки зрения потребительского спроса на кредит.

Для приведенных в постановке задачи модельных данных получены следующие значения (табл. 2).

Таблица 2

Сравнение численных результатов

Тип моддели	Свойство 1		Свойство 2	Свойство 3		Свойство 4	
Линейная	Да	5<7	Да	Да	0,006<0,01; 0,000125<2,365	Нет достаточных оснований	1,08≤1,274162≤1,36
Степенная	Да	5<7	Да	Нет	34,5>0,01; 0,676>0,01	Нет	1,035<1,08
Модель Брауна	Да	4<7	Да	Да	-0,53<0.01; -0.034<0.01	Да	1,71422785>1,36

Таким образом, из анализа полученных результатов, модель Брауна следует признать адекватной. Средняя относительная ошибка аппроксимации составляет 6,3%. Точность модели Брауна равна 93,7%. Степенная модель является неадекватной, так как не выполняются свойства 3 и 4, средняя относительная ошибка аппроксимации составляет 34,21%. Для линейной модели выполняются только два свойства, средняя ошибка равна 33,46%. Точности вычислений для линейной и степенной моделей ниже 70%.

Заключение

В настоящей работе рассмотрены адаптивная модель Брауна и трендовые модели на основе двух кривых роста для прогнозирования потребительского кредитования, оценена адекватность и точность построенных моделей. Приведен численный анализ полученных результатов. В ходе вычислений выявлено, что для представленных данных адекватной является только модель Брауна. Точность вычислений составляет 93,7%. Однако, стоит отметить, что модель работает только при небольшом горизонте прогнозирования.

ЛИТЕРАТУРА

1. *Лукашин Ю.П.* Адаптивные методы краткосрочного прогнозирования временных рядов. – М.: Финансы и статистика, 2003.

2. *Brown R.G.* Smoothing forecasting and prediction of discrete time series. - N.Y., 1963.

АЛГОРИТМ ОТБОРА ИНФОРМАТИВНЫХ ПАРАМЕТРОВ ПРОИЗВОДИТЕЛЬНОСТИ ДЛЯ ПРОАКТИВНОГО МОНИТОРИНГА СЕРВЕРА БАЗЫ ДАННЫХ

Дубровин М.Г.

Тюменский государственный университет

mikle1203@yandex.ru

Введение

Под сервером базы данных понимается специализированный программно-аппаратный комплекс, предназначенный для хранения, обработки и управления базой данных и обеспечивающий операции ввода-вывода при доступе клиентов к хранящейся информации [1]. Серверные системы относятся к сложным многомерным системам, обладающими свойствами гетерогенности, не стационарности и частичной стохастичности. Для анализа состояния и выявления событий в функционировании серверов баз данных используют различные автоматизированные средства мониторинга. Актуальным подходом является внедрение систем проактивного мониторинга, которые позволяют прогнозировать возникновение инцидентов в работе сервера [2] и генерировать соответствующие предупреждения, что дает возможность устранить возможные проблемы еще на этапе их зарождения.

Основным инструментом реализации системы проактивного мониторинга является задача обнаружения аномалий. Выявление инцидентов в таком случае строится на предположении, что при возникновении аномалий в работе сервера через некоторое время может возникнуть нарушение его функционирования. Методы решения задачи обнаружения аномалий, основанные на обучении с учителем, подходят в меньшей мере. Развитие состояния серверов БД приводит к тому, что понятие нормального состояния должно модифицироваться с течением времени и приводит к необходимости в постоянном переобучении классификатора [3].

Актуальной задачей при разработке системы проактивного мониторинга является определение минимально достаточного множества информативных параметров производительности сервера базы данных. С практической стороны, использование избыточного количества параметров может привести к тому, что сервер будет перегружен датчиками, а система мониторинга будет весьма громоздкой. К тому же, избыточное количество параметров снижает производительность используемых методов машинного обучения и может привести к переобучению моделей [4].

Целью работы является формирование алгоритма отбора информативных параметров для проактивного мониторинга сервера базы данных.

1. Материалы и методы

Рассмотрены существующие методологии по отбору параметров производительности информационных систем, нашедших применение на практике (USE [5], RED [6], The Four Golden Signals [7]). Метод The Four Golden Signals, используемый в компании Google, предназначен для высокоуровневых информационных систем и в большей степени подходит для определения множества параметров сервера базы данных.

В области машинного обучения для отбора информативных признаков существуют методы feature selection [3]. В контексте решения задачи обнаружения аномалий, рассмотрены только те методы фильтрации, которые не требуют размеченных данных и не рассматривают взаимосвязь с целевой функцией. Как правило, признаки с почти нулевым значением среднеквадратического отклонения не являются информативными и их можно исключить из множества [8]. Высокая корреляция между двумя признаками свидетельствуют о высокой взаимосвязи между ними и возможности удаления одного из них как неинформативного [3].

На основе рассмотренных методологий, предлагается алгоритм определения параметров для проактивного мониторинга серверов баз данных.

2. Алгоритм отбора информативных параметров производительности сервера базы данных

Пусть S – корпоративный сервер баз данных предприятия. Каждый сервер можно описать перечнем параметров, характеризующих его состояние $P_M = \{x_1,..,x_M\}$.

Обозначим $Y = \{Lattency, Traffic, Errors, Saturation\}$ – множество классов, соответствующих базовым группам, согласно методологии The Four Golden Signals [7].

Шаг 1. $P_M \to P_m$. Из множества P_M необходимо выбрать множество базовых параметров $P_m = \{x_1,..,x_m\}$, такое что $P_m \le P_M$, и каждому параметру $x_i \in P_m$ соответствует одна из групп методологии The Four Golden Signals: $f : P_m \to Y$.

Шаг 2. $P_m \to P_n$. Проверить множество P_m на информативность, отобрав множество информативных параметров P_n, $P_n \le P_m$.

Для определения информативности параметров определены следующие условия:

1. Среднеквадратическое отклонение s_x изменения значения каждого параметра $x \in P_n$ должно быть больше заданного порога p_1:

$$\bar{x} = \sum_{t=1}^{T} x_t \; ; \; s_x^2 = \frac{1}{T}\sum_{t=1}^{T}\left(x_t - \bar{x}\right)^2 \; ; \; s_x = \sqrt{s_x^2} \; ; \; \forall x \in P_n \, , \; x > p_1 \, ,$$

где x_t – значение параметра в момент времени t; $\bar{x}$ – среднее значение; s_x^2 – значение дисперсии.

2. Коэффициент корреляции Спирмена двух параметров $x_1, x_2 \in P_n$ должен быть меньше заданного порога p_2:

$$r_{x_1,x_2} = 1 - \frac{6}{T^3 - T}\sum_{t=1}^{T}\left(\text{rank}\left(x_{1,t}\right) - \text{rank}\left(x_{2,t}\right)\right)^2 \, , \; \forall x_1, x_2 \in P_n \, , \; r_{x_1,x_2} < p_2 \, ,$$

где x_1, x_2 – два исследуемых параметра сервера, $\text{rank}\left(x_{1,t}\right)$ – ранг, соответствующий значению параметра x_1 в момент t.

Для демонстрации практического применения алгоритма необходимо сформировать множество параметров согласно первому шагу алгоритма и провести вычислительный эксперимент для проверки сформированного множества на информативность.

3. Экспериментальные исследования

Экспертным путем, опираясь на существующие исследования по выбору параметров производительности серверов баз данных [9-11], отобрано 17 базовых параметров. Каждый параметр принадлежит к одной из групп метода The Four Golden Signals (табл. 1).

Таблица 1

Базовые параметры производительности сервера БД

№	Группа	Наименование	Обозначение	Ед. изм-я
1	Задержка (Latency)	Время выполнения операций	sql_time	с.
2		Время ожиданий по операциям	sql_waits	с.
3	Трафик (Traffic)	Количество операций в секунду	sql_per_sec	Ед./с.
4		Количество подключений к БД	db_logon_cnt	Ед.
5		Количество активных сессий	db_active_cnt	Ед.
6		Входящий сетевой трафик	srv_traffic_in	МБ./с.
7		Исходящий сетевой трафик	srv_traffic_out	МБ./с.
8	Ошибки (Errors)	Количество блокировок операций	sql_locks	Ед.
9	Насыщение (Saturation)	Размер пользовательских данных	db_data_size	МБ.
10		Размер файлов БД	db_file_size	МБ.
11		Процент чтений из буфера	db_cache_pct	%
12		Операции чтения файлов в секунду	srv_reads	Ед./с.
13		Операции записи файлов в секунду	srv_writes	Ед./с.
14		Загрузка центрального процессора	srv_cpu_load	%
15		Загрузка физической памяти	srv_mem_pct	%
16		Загрузка виртуальной памяти	srv_vmem_pct	%
17		Заполненность дискового пространства	srv_disk_pct	%

Для проведения эксперимента собраны статистические данные об изменении значений 17 отобранных параметров действующего сервера базы данных на производственном предприятии в г. Тюмень за недельный период с дискретностью в 10 минут. В качестве объекта исследования использован сервер под управлением операционной системы семейства Windows Server, на котором установлена база данных под управлением системы управления базой данных Oracle.

Для проверки первого условия информативности рассчитаны значения среднего $\bar{x}$ дисперсии s_x^2 и среднеквадратического отклонения s_x. Результаты расчётов представлены в табл. 2.

Таблица 2

Рассчеты среднеквадратического отклонения

№	Обозначение параметра	Среднее	Дисперсия	Среднеквадр. откл.
1	sql_time	95.74	17.57	4.19
2	sql_waits	4.68	19.73	4.44
3	sql_per_sec	1.06	0.89	0.94
4	db_logon_cnt	96.47	57.13	7.55
5	db_active_cnt	4.01	2.11	1.45
6	srv_traffic_in	101.72	2455.61	49.55
7	srv_traffic_out	348.00	9148.24	95.64
8	sql_locks	24.87	0.43	0.65
9	db_data_size	814841.78	2190234.92	1479.94
10	db_file_size	1450598.67	208384.09	456.49
11	db_cache_pct	99.40	0.32	0.57
12	srv_reads	40.14	3167.07	56.27
13	srv_writes	52.21	95629.02	309.24
14	srv_cpu_usage	60.89	36.88	6.07
15	srv_mem_pct	33.83	0.24	0.49
16	srv_vmem_pct	66.69	0.13	0.36
17	srv_disk_pct	24.55	2.08	1.44

В качестве порогового параметра принято $p_1 = 0.1$. В таблице видно, что все параметры удовлетворяют заданному условию, $\min(s_x) = 0.36$.

Для проверки второго условия построена корреляционная тепловая карта на основе корреляционной матрицы отобранных параметров (рис. 1).

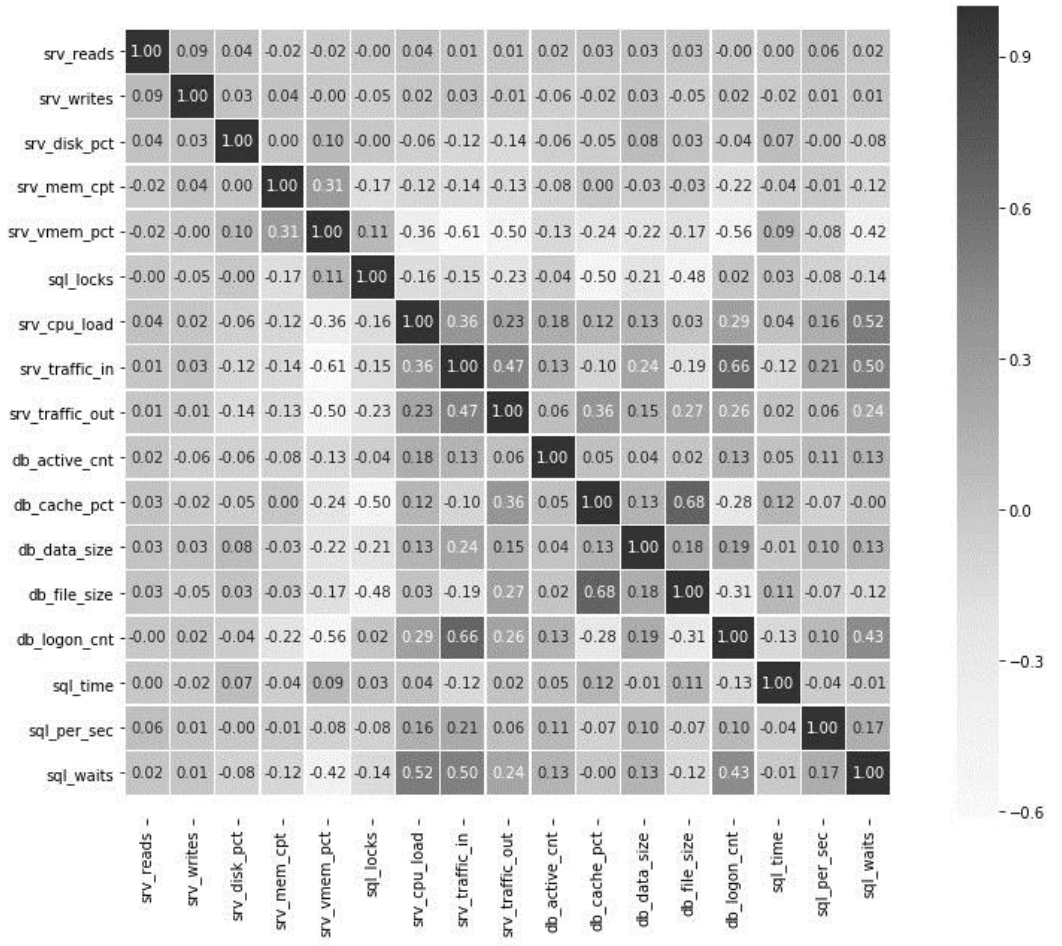


Рис. 1. Корреляционная тепловая карта параметров

Согласно шкале Чеддока [12], два признака имеют весьма высокую корреляционную связь при $|r| > 0.9$, поэтому в качестве порогового параметра принято $p_2 = 0.9$. На тепловой карте наглядно видно, что все параметры удовлетворяют заданному условию, значение коэффициента корреляции Спирмена лежит в следующем интервале: $r_{x_1,x_2} \in (-0.61; 0.68)$.

Параметры p_1 и p_2 могут быть варьированы с целью корректировки количества параметров искомого множества. При этом необходимо учитывать, что при уменьшении количества базовых параметров, система мониторинга, вероятно, сможет не распознать некоторые значимые аномалии, приводящие к возникновению инцидентов.

Результаты вычислительного эксперимента продемонстрировали, что все отобранные параметры являются в достаточной мере информативными.

Заключение

Рассмотрены подходы по отбору параметров для мониторинга и некоторые методы отбора информативных признаков. На основе рассмотренных методологий, предложен алгоритм отбора множества информативных параметров производительности сервера базы данных. Теоретическая значимость алгоритма заключается в новом способе формирования минимально достаточного множества параметров сервера, отличающегося комбинацией практической значимости и информативности искомого множества.

Практическая значимость заключается в возможности использования алгоритма при разработке систем проактивного мониторинга сервера базы данных и других сложных информационных систем.

ЛИТЕРАТУРА

1. *Ладыженский Г.М.* Системы управления базами данных-коротко о главном //СУБД. – 1995. – Т. 3192

2. Словарь терминов ITIL на русском языке, версия 2.0, 29 июля 2011 г. [Электронный ресурс]. URL: http://www.itsmforum.ru/reference/itil-glossary/ (Дата обращения: 25.02.2020)

3. *Шкодырев В.П. и др.* Обзор методов обнаружения аномалий в потоках данных //Second Conference on Software Engineering and Information Management (SEIM-2017)(full papers). – 2017. – С. 50.

4. *Chandrashekar G., Sahin F.* A survey on feature selection methods //Computers & Electrical Engineering. – 2014. – Т. 40. – №. 1. – С. 16-28.

5. *Gregg B.* Thinking methodically about performance //Communications of the ACM. – 2013. – Т. 56. – №. 2. – С. 45-51.

6. RED Method. [Электронный ресурс]. URL: https://www.weave.works/blog/the-red-method-key-metrics-for-microservices-architecture/ (Дата обращения: 26.02.2020)

7. *Beyer B. et al.* Site Reliability Engineering: How Google Runs Production Systems. – " O'Reilly Media, Inc.", 2016.

8. Overview of feature selection methods [Электронный ресурс]. URL: https://towardsdatascience.com/overview-of-feature-selection-methods-a2d115c7a8f7 (Дата обращения: 26.02.2020)

9. *Yoon D.Y., Niu N., Mozafari B.* Dbsherlock: A performance diagnostic tool for transactional databases //Proceedings of the 2016 International Conference on Management of Data. – ACM, 2016. – С. 1599-1614.

10. *Julian M.* Practical Monitoring: Effective Strategies for the Real World. USA: O'Reilly Media, 2018. 151 с.

11. *Корогодова А.О., Беленькая М.Н.* Проблемы производительности СУБД ORACLE под управлением сетевой ОС WINDOWS //T-Comm-Телекоммуникации и Транспорт. – 2013. – №. 7.

12. *Елисеева И.И.* Эконометрика: учебник / И. И. Елисеева [и др.]; под ред. И. И. Елисеевой. –2 - е изд., перераб. и доп. – М.: Финансы и статистика, 2007. –576 с.

ПРОГНОЗ ЗНАЧЕНИЙ ДЕБИТОВ СКВАЖИН С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Евсюткин И.В., Марков Н.Г.

Томский политехнический университет

ive1@tpu.ru, markovng@tpu.ru

Введение

При добыче углеводородного сырья (УВС) на промыслах происходит измерение большого числа технологических и геологических параметров. Значения этих параметров служат для оценки состояния скважин и продуктивных пластов, а также позволяют понять, какова технологическая и экономическая эффективность проведённых работ на фонде скважин, в том числе и геолого-технических мероприятий (ГТМ).

Задача прогноза значений параметров актуальна для управления фондом скважин, так как выявленный критический тренд может позволить заранее установить необходимость проведения определённого вида работ [1]. Однако существующие методы прогноза в лучшем случае позволяют выполнить экстраполяцию данных путём линейной регрессии [2]. Разумеется, столь простые методы не способны учесть множество аспектов, заложенных в данных и взаимовлияние их друг на друга в рамках временного ряда [3]. Соответственно, точность прогноза не всегда удовлетворяет специалистов нефтегазодобывающих предприятий.

Всё это указывает на актуальность разработки новых методов, позволяющих существенного повысить точность прогноза значений параметров скважин и продуктивных пластов.

В данной работе для прогноза значений дебитов скважин было предложено использовать глубокие искусственные нейронные сети (ИНС), показавшие свою эффективность при решении задач экстраполяции данных в других предметных областях. Проведено исследование эффективности различных архитектур ИНС прямого распространения при решении таких задач прогноза. Эксперименты показали, что использо-

вание таких ИНС позволяет значительно повысить точность прогноза значений дебитов скважин по сравнению с точностью, достигаемой с помощью известных методов.

1. Постановка задачи исследований

Целью планирования работы производства нефтегазодобывающего предприятия является определение таких технологических режимов работы фонда скважин и технологического оборудования, которые позволят достичь заданного объёма выпуска продукции (нефти, газа и газового конденсата). При этом наиболее важно планирование технологических режимов каждой добывающей скважины фонда и технологических режимов установок подготовки продукции из добываемого УВС, а также планирование режимов транспортировки продукции до магистральных нефте- и газопроводов.

Обычно на эксплуатируемых месторождениях прогнозу подлежит довольно большое число измеряемых параметров добывающих скважин и продуктивных пластов, среди них: дебит нефти, дебит газа, дебит жидкости (конденсата), дебит воды, давление забойное, давление пластовое, обводнённость и т.д. Однако наиболее востребованными являются прогнозы значений дебитов скважин.

Точное предсказание дебитов скважин является необходимой частью управления производством любого нефтегазодобывающего предприятия. Важно прогнозировать (планировать) дебиты не просто на конкретный день, а на определённый промежуток времени (например, равный месяцу), чтобы понимать, сколько в среднем УВС будет добываться из скважины в сутки. В этом случае говорят о планировании технологического режима скважины. Значение технологического режима скважины позволяет рассчитать экономический эффект от добычи УВС и понять рентабельность эксплуатации конкретной скважины, построить перспективные планы поставок продукции и учесть ограничения, наложенные органами надзора за пользованием недрами.

Для обеспечения высокой точности прогноза значений дебитов скважины было предложено использовать глубокие ИНС прямого распространения. Поэтому была поставлена задача исследования эффективности таких ИНС при решении задачи прогноза значений дебитов при использовании реальных данных с эксплуатируемого месторождения.

Для проведения исследований использовался архив данных одного из нефтегазоконденсатных месторождений Томской области, где за 6 лет эксплуатации накоплен значительный объём данных. Фонд добывающих скважин содержит 17 кустов скважин, в каждом кусте от 6 до 21 добывающей скважины, всего 142 добывающих скважины.

Для экспериментов были выбраны исключительно параметры типа дебитов (ежесуточные объёмы добычи компонентов УВС – нефти, воды и т. д.), так как они являются наиболее значимыми интегральными показателями добычи УВС. Для них в базе данных (БД) предприятия обычно имеется наибольший объём данных, не является исключением и предприятие, из архива которого были взяты данные для исследований.

2. Методика подготовки данных для обучения и тестирования ИНС

На предприятии использовалась методика прогноза (планирования) значений дебитов, основанная на методе экстраполяции по скользящей средней. Для каждого из дебитов скважин в анализируемой БД имеются прогнозное и фактическое значения. Прогноз геологической службой предприятия, судя по имеющимся данным, осуществлялся от –8 до 229 дней до измерения фактического значения дебита. То есть в БД имеются явно ошибочные значения параметров, по крайней мере, при планировании значений дебитов, так как нельзя планировать значение параметра уже после получения его фактического значения. Также вызывают сомнения прогнозы на большие временные промежутки (на 2 и более месяцев). Все такие данные отбраковывались и в экспериментах не использовались.

Все фактические измеренные значения дебитов на дату планирования и ранее подаются на входы ИНС (признак, англ. feature). Фактические значения дебитов через месяц после этой даты являются эталоном, то есть они служат эталоном выходным значениям ИНС (метка, англ. label) при её обучении и тестировании.

Данные, которые находятся между датой планирования и более поздней датой получения фактического значения дебита-эталона, не используются, несмотря на их наличие в БД, чтобы находиться в тех же самых условиях, что и находилась геологическая служба предприятия, планировавшая технологический режим работы скважины. Это позволит корректно осуществить сравнение полученных при исследовании результатов с имеющимися в БД значениями прогнозных (плановых) дебитов.

При проведении экспериментов было решено брать значения дебитов от даты планирования в течение трёх – пяти недель до получения фактического значения. Это практически важно при формировании технологических режимов скважин. Исходное число записей в БД 17117, после сужения периода планирования до заданного записей осталось 13032. Видно, что такие действия уменьшили объём обучающей выборки. Однако это позволило нам получить более точную оценку результатов прогноза значений дебитов на месяц по существующей на предприятии методике.

Число анализируемых записей и рассчитанные нами значения средних абсолютных ошибок [4] при прогнозе каждого параметра (вида дебита) по всему месторождению по методике предприятия указаны в табл. 1. При разработке наших моделей ИНС и при оценке их точности необходимо ориентироваться на уменьшение среднего абсолютного значения ошибки при прогнозе значений этих дебитов.

Текущий или капитальный ремонт скважин, остановка их по другим причинам или на проведение ГТМ вносят непредсказуемость в изменение параметров скважин. Это должно быть учтено при составлении обучающих примеров, чтобы в вектор признаков на входах ИНС по каждой скважине не попадали значения дебитов одновременно до и после проведения тех или иных ГТМ или остановок скважин. То есть планирование технологического режима скважины должно осуществляться исключительно с учетом периодов её эксплуатации.

Таблица 1

Оценки точности прогноза параметров по методике предприятия

Название параметра	Обозначение	Единица измерения	Число записей	Средняя абсолютная ошибка
дебит нефти	$Q_н$	т/сут	3557	13.010
дебит газа	$Q_г$	тыс. м3/сут	3458	10.290
дебит жидкости	$Q_ж$	м3/сут	3459	13.905
дебит воды	$Q_в$	т/сут	2558	1.741

Не ясно, какое оптимальное число значений дебита нужно подавать на входы ИНС, это предстоит определить экспериментально. Также открытым остаётся вопрос, стоит ли подавать на входы ИНС значения только одного какого-либо дебита или добавлять к ним ещё значения других параметров, например, дебита другой компоненты УВС или пластового давления продуктивного пласта. Так как число таких сочетаний очень велико, подобные эксперименты не выполнялись, и в качестве входных значений для ИНС использовались только фактические значения прогнозируемого дебита.

Суть предлагаемой методики подготовки данных в том, чтобы увеличивать при прогнозе число входных значений дебитов для ИНС за счёт расширения числа известных из БД фактических значений дебитов от 6 (6 дней назад от даты планирования) с постепенным добавлением таких значений за предыдущие дни до 15. Схематично процесс выбора данных изображён на рис. 1, где показано движение скважины во времени. Видно, что скважина была два раза остановлена на разные промежутки времени. Жирными штрихами отмечены взятые в эксперимент 13 фактических значений дебита на

скважине, получаемые ежедневно. Пусть дата прогноза значения дебита будет отмечена знаком «!» (метка для ИНС), и на эту дату в БД должно быть фактическое значение дебита. Дата, с которой планируется значение дебита, показана знаком «?». С даты планирования и по дату прогноза должен пройти один месяц (период планирования). Все фактические значения дебита (от 6 до 15 значений) до даты планирования становятся признаками, подаваемыми на входы ИНС (отмечены жирными штрихами).

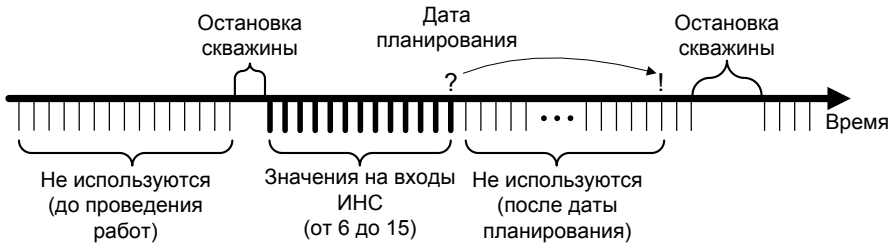


Рис. 1. Схема выбора значений измеряемых дебитов

Дате планирования может предшествовать остановка скважины для проведения каких-либо работ (например, ГТМ). Поэтому число признаков для ИНС, большее некоторого имеющегося числа, взять не получится. Например, из рис. 1 следует, что выбрать 14 признаков не удастся. Это значит, что в общем случае обучающие выборки будут не сопоставимы по объёму, так как для малого числа признаков на входах ИНС (например, 6) будет всегда больше обучающих примеров, чем для большого числа признаков (например, 15).

Если за какой-либо день в БД отсутствует фактическое значение дебита, и это не связано с проведением работ на скважине, то производится линейная интерполяция значения дебита по соседним известным значениям. Использование линейной интерполяции было продиктовано разработчиками БД.

Если между двумя остановками скважины есть только одно измерение дебита или оно вообще отсутствует, то в таких условиях провести интерполяцию нельзя, и прогноз на этом интервале становится невозможным. Если на скважине ГТМ не проводились, то берётся весь массив значений фактических дебитов для анализа без каких-либо ограничений. Аналогичным образом для анализа используются все значения параметров после последнего проведённого ГТМ на скважине.

Для извлечения данных о дебитах и остановках скважин, в том числе для ГТМ, из БД было создано специальное программное обеспечение (ПО). Данное ПО позволяет провести всю необходимую предварительную обработку данных, выбрать вид дебита (дебит нефти, газа, жидкости (газового конденсата) или воды), число признаков на входе ИНС (от 6 до 15), указать число блоков для кросс-валидации данных методом K блоков [5], а также создать файл для последующего обучения или тестирования ИНС. Пользователь видит при загрузке данных информацию о характеристиках выборки в текстовом поле. При формировании файла указывается, сколько было задействовано скважин при построении выборки и общее число обучающих и тестовых примеров.

Для проведения экспериментов с помощью этого ПО было построено 40 пар файлов (для обучения и тестирования ИНС) – по 10 пар на каждый вид дебита. Объёмы пар «признаки – метка» для разного числа входов ИНС и для всех дебитов указаны в табл. 2. При использовании метода K блоков выбрано $K = 5$. Это означает, что все вычисленные в ходе исследований итоговые значения средней абсолютной ошибки прогноза дебитов являются результатом усреднения по пяти значениям ошибок, полученным в ходе обучения и тестирования ИНС.

Таблица 2

Объёмы выборок для различных дебитов и разного числа входов ИНС

Дебиты	Число входов ИНС									
	6	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
$Q_н$	1248	1248	1246	1246	1244	1244	1243	1243	1243	1235
$Q_г$	1199	1199	1197	1197	1195	1195	1194	1194	1194	1188
$Q_ж$	1199	1199	1197	1197	1195	1195	1194	1194	1194	1188
$Q_в$	880	877	875	875	873	873	872	872	872	868

3. Эксперименты по прогнозу значений дебитов скважин

Программная реализация ИНС для проведения экспериментов проведена на языке C# с использованием библиотеки Microsoft CNTK [6]. С её помощью были автоматизированы операции по изменению архитектуры и гиперпараметров ИНС, обучены и тестированы ИНС на вышеуказанных выборках.

Число скрытых слоёв ИНС прямого распространения выбиралось равным 2, 3, 4, 5, 6, 7, 8, 10, 15. В ходе экспериментов было выявлено, что число слоёв, большее 8, не даёт прироста точности прогноза. Исследовалось также применение различных функций активации (TanH, LeakyReLU), позволяющих получать связи между нейронами с отрицательными воздействиями. Алгоритм оптимизации при обучении ИНС Adam. Скорость обучения динамически определялась по этому алгоритму. Каждый обучающий пример уникален и должен осуществлять воздействия индивидуально, но скорость обучения тем больше, чем меньше объём минивыборки, поэтому был выбран компромиссный вариант, равный 5. Число эпох выбиралось динамически в соответствии с достигаемой точностью обучения, если она несколько эпох существенно не менялась. Число нейронов в каждом скрытом слое выбиралось в соответствии с эвристическим правилом: $Число\ Нейронов = Число\ Входов * 2 + 2$.

На входы ИНС подаются фактические значения дебита за определённое число дней (от 6 до 15). Далее идёт ряд скрытых слоёв ИНС – от 2 до 8. В выходном слое ИНС один нейрон (прогнозируемое значение дебита). На рис. 2 в качестве примера приведена архитектура ИНС с 3 скрытыми слоями, 12 фактическими значениями дебита на входах ИНС и 26 нейронами в скрытых слоях, функция активации LeakyReLU.

Рис. 2. Схема архитектуры ИНС

В табл. 3 приведены в качестве примера результаты экспериментов с ИНС различных архитектур при решении на тестовых выборках задачи прогноза значений дебита нефти Qн переменными являются как функции активации и число скрытых слоёв ИНС,

так и число входов ИНС. Значение ошибки прогноза вычисляется как среднее по всем ошибкам, полученным по схеме кросс-валидации по 5 блокам, и по всем ошибкам для добывающих скважин в выборках. Жирным отмечается лучшая точность, результаты округлены до тысячных. Аналогичные эксперименты на тестовых выборках были проведены для дебитов газа, жидкости и воды, их результаты сведены в три таблицы.

Таблица 3

Результаты прогноза дебита нефти $Q_н$ с помощью ИНС на тестовых выборках

Параметр	Число входов ИНС	Функция активации / Число слоёв	Значение средней абсолютной ошибки прогноза, т/сут						
			2	3	4	5	6	7	8
$Q_н$, т/сут	6	TanH	8.243	8.310	8.771	7.482	8.238	8.829	8.631
		LeakyReLU	10.134	8.871	9.072	8.213	8.954	8.608	9.424
	7	TanH	8.979	8.745	8.564	8.693	7.609	8.442	9.164
		LeakyReLU	9.151	8.853	8.666	8.635	8.449	7.785	8.488
	8	TanH	8.034	8.578	8.458	8.461	9.977	9.415	8.413
		LeakyReLU	8.385	8.633	8.695	8.308	7.978	8.571	8.599
	9	TanH	8.167	8.576	8.564	9.295	8.723	7.801	8.728
		LeakyReLU	8.408	9.025	8.250	9.371	7.825	7.632	8.323
	10	TanH	7.098	6.759	6.409	6.957	6.526	7.250	6.631
		LeakyReLU	6.931	7.012	6.172	6.841	6.530	6.498	6.789
	11	TanH	6.531	6.313	7.194	6.994	6.732	8.254	7.631
		LeakyReLU	6.769	6.643	6.750	6.259	6.524	6.631	6.903
	12	TanH	7.682	7.898	7.475	8.400	8.372	8.099	8.451
		LeakyReLU	7.825	7.727	7.753	7.706	8.120	7.557	7.842
	13	TanH	8.324	7.490	7.588	7.956	7.895	8.107	8.051
		LeakyReLU	8.162	7.598	7.205	7.713	7.072	7.520	7.112
	14	TanH	7.716	7.490	7.876	7.879	8.028	7.944	10.256
		LeakyReLU	8.632	7.889	7.399	7.703	7.324	7.411	7.252
	15	TanH	7.579	7.512	7.383	8.487	7.427	7.462	8.388
		LeakyReLU	7.585	7.269	7.119	7.281	6.933	6.843	6.634

Такие же четыре таблицы содержат результаты прогноза значений дебитов в случае обучающих выборок. Отметим, что значения средней абсолютной ошибки на обучающих выборках меньше до 30% по сравнению с ошибками, полученными с помощью ИНС на тестовых выборках. Причём чем больше число входов и число слоёв ИНС, тем меньше разница средних абсолютных ошибок, полученных на обучающей и тестовой выборках.

Лучшие результаты из всех четырёх таблиц со значениями средних абсолютных ошибок при прогнозе дебитов, полученные при исследованиях ИНС с различными архитектурами на тестовых выборках, занесены в табл. 4. Сравнительный анализ лучших результатов из табл. 4 со средними абсолютными ошибками прогноза из табл. 1. показал, что для каждого из дебитов точность прогноза с помощью ИНС выше, чем полученная с помощью метода экстраполяции по скользящей средней. Прирост точности прогноза дебитов с помощью ИНС указан для тестовых выборок в сравнении с результатами из табл. 1 в последнем столбце табл. 4. Видно, что лучший результат прогноза каждого из дебитов, полученный с использованием ИНС, в два и более раза точнее, чем результат прогноза каждого из дебитов по методике, принятой на предприятии.

Таблица 4

Лучшие результаты при прогнозе значений дебитов скважин с помощью ИНС

Параметр	Средняя абсолютная ошибка	Особенности архитектуры ИНС	Прирост точности прогноза, %
Qн	6.172 т/сут	10 входов ИНС; 4 скрытых слоя; функция активации LeakyReLU	53
Qг	5.084 тыс. м3/сут	11 входов ИНС; 6 скрытых слоёв;	51

Параметр	Средняя абсолютная ошибка	Особенности архитектуры ИНС	Прирост точности прогноза, %
		функция активации LeakyReLU	
Qж	6.607 м3/сут	10 входов ИНС; 6 скрытых слоёв; функция активации LeakyReLU	52
Qв	0.654 т/сут	10 входов ИНС; 7 скрытых слоёв; функция активации TanH	62

Заключение

При планировании показателей производства нефтегазодобывающего предприятия важным является прогноз значений дебитов каждой добывающей скважины. Для повышения точности такого прогноза было предложено использовать глубокие ИНС.

Разработана методика подготовки реальных данных по дебитам скважин для обучения и тестирования глубоких ИНС прямого распространения. Проведены исследования различных архитектур таких ИНС при решении задач прогноза дебитов нефти, газа, жидкости (газового конденсата) и дебита воды. В результате этих исследований выявлены наиболее эффективные архитектуры ИНС, позволяющие увеличить точность прогноза в два и более раза по сравнению с точностью прогноза, даваемой традиционным методом экстраполяции по скользящей средней.

ЛИТЕРАТУРА

1. *Ankudinov A.A., Polyakova N.S., Radevich Y.E.* LUKOIL: How Data Mining Enhances Oilfield Development // ROGTEC Field Development [Электронный ресурс]. URL: https://rogtecmagazine.com/лукойл-мониторинг-разработки-местор/?lang=ru (дата обращения 03.04.2020).

2. *Марков Н.Г.* Информационно-управляющие системы для газодобывающего производства. – Томск: Изд-во Томского политехнического университета, 2016. – 261 с.

3. *Косков В.Н.* Геофизические исследования скважин [Электронный ресурс]. URL: https://pstu.ru/files/file/gnf/gis.pdf (дата обращения 03.04.2020).

4. Ошибка прогнозирования: как рассчитать и применять [Электронный ресурс]. URL: https://uppravuk.net/oshibka-prognozirovaniya-kak-rasschita/ (дата обращения 03.04.2020).

5. Кросс-валидация [Электронный ресурс]. URL: https://long-short.pro/post/kross-validatsiya-cross-validation-304/ (дата обращения 03.04.2020).

6. The Microsoft Cognitive Toolkit [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/cognitive-toolkit/ (дата обращения 03.04.2020).

ЗАДАЧИ ИССЛЕДОВАНИЯ ЗНАКОВЫХ ГРАФОВ

Ибрагимова Э.И., Семенова Д.В.
Сибирский федеральный университет
ibragimovaei@mail.ru, DVSemenova@sfu-kras.ru

Введение

Знаковые графы были впервые введены Фрэнком Харари при рассмотрении проблем социальной психологии [1]. Их можно интерпретировать как модель поведения исследуемого объекта или процесса, где вершинам графа сопоставляются переменные состояния объекта, а знак ребра представляет собой качественную оценку связи между смежными вершинами. В настоящее время знаковые графы применяются в таких областях, как анализ данных (классификация и кластеризация), моделирование социальных сетей, экономические модели, мультиагентные системы и системы адаптивного поведения. В системах поддержки принятия решений задача поиска оптимального решения может быть сформулирована как задача достижения знакового баланса в когнитивной карте, выражающей основные закономерности наблюдаемой ситуации в виде знакового графа. Также, знаковые графы используются при решении NP-трудных задач линейного программирования [2]. В связи с широким применением знаковых графов актуальны

исследования, направленные на разработку алгоритмов решения задач, связанных с ними.

Знаковым графом называется пара $\Sigma=(G,\sigma)$, где $G=(V,E)$ является неориентированным (n,m)-графом, на ребрах которого задана функция знака $\sigma\colon E\to\{+,-\}$. Здесь $n=|V|\geq 2$, $m=|E|\geq 1$. Знаковый граф называется сбалансированным, если каждый его простой цикл положительный, где знак цикла определяется как произведение знаков, входящих в него ребер. Необходимое и достаточные условие 2-сбалансированности (далее просто сбалансированности) знакового графа устанавливает следующая теорема Картрайта – Харари [1].

Теорема 1 (Картрайт – Харари). Знаковый граф $\Sigma=(G,\sigma)$ сбалансирован тогда и только тогда, когда множество его вершин *V* можно разбить на две доли *A* и *B*, одна из которых может быть пустой, таким образом, что $A\cup B=V$, $A\cap B=\varnothing$ и любое ребро, соединяющее вершины из одной доли, имеет знак «+», а ребро, соединяющее вершины из разных долей, имеет знак «–».

Целью данной работы является рассмотрение основных задач, связанных со свойством сбалансированности знаковых графов.

1. Распознавание сбалансированности

Задача распознавания сбалансированности знакового графа

Дано: знаковый граф $\Sigma=(G,\sigma)$, где $G=(V,E)$, $n=|V|\geq 2$, $m=|E|\geq 1$.

Требуется: проверить, является ли граф Σ сбалансированным.

Задача распознавания сбалансированности знакового графа может быть решена за полиномиальное время. Алгоритм основан на теореме Картрайта-Харари и поиске в графе в ширину или в глубину [3]. В разработанном алгоритме Signed_Balance в процессе обхода графа в ширину множество его вершин разделяется на две доли так, что если текущая вершина соединена с предыдущей положительным ребром, то данные вершины принадлежат одной доле, а если отрицательным, то разным [4,5]. После этого проверяется выполнение теоремы Картрайта-Харари.

Результаты вычислительных экспериментов распознавания сбалансированности знаковых графов с различной структурой приведены в табл. 1. Вычислительная сложность алгоритма Signed_Balance равна $O(n+m)$.

Таблица 1

Результаты вычислительных экспериментов для задачи распознавания сбалансированности

Структура	Кол-во вершин	Кол-ве ребер			Сбалансированность	Signed_Balance	Время работы, мс
		m	m^-	m^+			
дерево	5	4	4	0	да	да	2
двудольный	5	5	5	0	да	да	1
произвольный	6	7	4	3	нет	нет	1
полный	30	435	435	0	нет	нет	4
полный	30	435	225	210	да	да	3
полный	200	19900	19900	0	нет	нет	5
полный	200	19900	10000	9900	да	да	6

2. Поиск наибольшего по мощности сбалансированного подграфа (MBSP)

Задача поиска максимального сбалансированного подграфа (MBSP)

Дано: знаковый граф $\Sigma=(G,\sigma)$, где $G=(V,E)$, $n=|V|\geq 2$, $m=|E|\geq 1$.

Требуется: найти наибольшее по мощности множество ребер $E'\subseteq E$, которое будет сбалансировано в Σ.

Задача MBSP является NP-трудной. Действительно, в частном случае, когда в графе $\Sigma=(G,\sigma)$ все ребра отрицательны, MBSP сводится к задаче поиска в G двудольного подграфа с наибольшим числом ребер, для которой доказана NP-трудность [6].

Следует отметить, что задача поиска максимального сбалансированного подграфа эквивалентна задаче установления знакового баланса, которая заключается в том, что необходимо найти минимальный набор ребер, при изменении знаков которых граф станет сбалансированным. Число этих ребер называется индексом несогласованности и помогает понять, насколько граф не сбалансирован.

Для поиска точного решения была построена модель целочисленного программирования и использовался пакет программ CPLEX [4,5]. Поиск оптимального решения происходит в пространстве мощностью 2^{n+m} решений.

Для поиска приближенного решения был реализован алгоритм, предложенный в [2], основывающийся на свойстве знакового графа, вытекающем из теоремы Заславского [7]: знаковый граф сбалансирован тогда и только тогда, когда из него с помощью переключения вершин можно получить граф со всеми положительными ребрами. Вычислительная сложность данного алгоритма равна $O(m)$.

На графах с различной структурой, для которых заранее было известно, сбалансированы они или нет, были проведены вычислительные эксперименты с целью сравнения точного и приближенного решений, времени их поиска и вычисления индекса несогласованности. Из результатов, приведенных в табл. 2 видно, что эвристический алгоритм работает намного быстрее, чем CPLEX, при этом точное и приближенное решения совпадают. Кроме этого, в бесплатной версии программы CPLEX есть ограничение на размерность задачи. Также скорость работы CPLEX зависит от структурных особенностей графа и функции знака. Для сбалансированного графа время поиска точного решения гораздо меньше, чем для несбалансированного.

Также перед решением MBSP в некоторых случаях можно сократить размерность графа. В данной работе используется декомпозиционный подход, основывающийся на следующем правиле редукции.

Правило редукции [2]. Пусть граф $\Sigma=(V,E,\sigma)$ такой, что $V=V^1\cup V^2$ и $E=E\left[V^1\right]\cup E\left[V^2\right]\cup\{(i,j)\}$, $i\in V^1$, $j\in V^2$. Определим $\Sigma^1=\left(V^1,E\left[V^1\right],\sigma\right)$, $\Sigma^2=\left(V^2,E\left[V^2\right],\sigma\right)$. Решаем задачу независимо относительно Σ^1 и Σ^2. Получаем решения $H'=(V',E',\sigma)$, $H''=(V'',E'',\sigma)$. Тогда оптимальное решение относительно Σ будет $H=(V'\cup V''),E'\cup E''\cup\{(i,j),\sigma\}$.

При проведении вычислительных экспериментов рассматривались графы, которые при удалении ребер, не состоящих ни в одном из циклов, разбивались на несколько одинаковых полных подграфов, в которых все ребра отрицательные. Для проверки эффективности сокращения данных проверялось время поиска точного решения задачи MBSP в CPLEX для исходного графа и для сокращенного. Вычислительная сложность алгоритма по предобработке данных равна $O(nm+n+m)$. Результаты экспериментов, приведенные в табл. 3, показывают, что применение данного правила в некоторых случаях позволяет значительно уменьшить время поиска точного решения для MBSP.

Таблица 2

Результаты вычислительных экспериментов для MBSP

Структура и кол-во вершин	**Кол-во ребер**			**Сбалансированность**	**CPLEX**		**Эвристика**		**Индекс несогласованности**
	m	m^-	m^+		решение, кол-во ребер	время работы, мс	решение, кол-во ребер	время работы, мс	
дерево, 5	4	4	0	да	1	11	4	1	0

Структура и кол-во вершин	Кол-во ребер			Сбалансированность	CPLEX		Эвристика		Индекс несогласованности
	m	m^-	m^+		решение, кол-во ребер	время работы, мс	решение, кол-во ребер	время работы, мс	
двудольный, 5	5	5	0	да	5	10	5	1	0
произвольный, 6	7	4	3	нет	6	17	6	1	1
полный, 10	45	45	0	нет	25	32	25	1	20
полный, 10	45	25	20	да	45	12	45	1	0
полный, 15	105	105	0	нет	56	2670	56	1	49
полный, 15	105	56	49	да	105	14	105	1	0
полный, 30	435	435	0	нет	225	10800000	225	1	210
полный, 30	435	225	210	да	435	29	435	1	0

Таблица 3

Сокращение размерности графа

Кол-во вершин и ребер	Время работы CPLEX, мс	Время предобработки, мс	Кол-во удаленных ребер и полученных подграфов	Размерность каждого подграфа	Время работы CPLEX для каждого подграфа, мс
$n = 10$, $m = 21$	21	1	1 ребро, 2 подграфа	$n' = 5$, $m' = 10$	14
$n = 20$, $m = 91$	10095	3	1 ребро, 2 подграфа	$n' = 10$, $m' = 45$	32
$n = 15$, $m = 32$	33	1	2 ребро, 3 подграфа	$n' = 5$, $m' = 10$	13
$n = 24$, $m = 86$	26043	2	2 ребро, 3 подграфа	$n' = 8$, $m' = 28$	29
$n = 20$, $m = 43$	1029	1	3 ребра, 4 подграфа	$n' = 5$, $m' = 10$	14
$n = 28$, $m = 87$	43003	1	3 ребра, 4 подграфа	$n' = 7$, $m' = 21$	16

3. Мера сбалансированности

Одними из основных характеристик знакового графа являются мера и индекс сбалансированности. С их помощью можно определить степень несбалансированности графа.

Обозначим $C(\Sigma)$ – множество простых циклов, $C^-(\Sigma)$ – множество отрицательных циклов, $C^+(\Sigma)$ – множество положительных циклов графа Σ.

Мера сбалансированности $b(\Sigma)$ знакового графа $\Sigma = (G, \sigma)$, где $G = (V, E)$, есть доля простых положительных циклов среди всех циклов Σ:

$$b(\Sigma) = \frac{|C^+(\Sigma)|}{|C(\Sigma)|}.$$

Очевидно, что $0 \le b(\Sigma) \le 1$, причём $b(\Sigma) = 1$, если и только если все циклы графа положительны, т.е. граф сбалансирован. Определение меры сбалансированности некорректно, если граф не содержит ни одного цикла. Но по теореме Картрайта-Харари такой граф сбалансирован и в этом случае считается, что $b(\Sigma) = 1$.

Индекс сбалансированности $I(\Sigma)$ знакового графа $\Sigma = (G, \sigma)$, где $G = (V, E)$, есть сумма знаков всех простых циклов графа Σ:

$$I(\Sigma) = |C^+| - |C^-|.$$

Задача вычисления меры сбалансированности знакового графа

Дано: знаковый граф $\Sigma = (G, \sigma)$, где $G = (V, E)$, $n = |V| \ge 2$, $m = |E| \ge 1$.

Требуется: найти меру сбалансированности $b(\Sigma)$ графа Σ.

Задача вычисления индекса сбалансированности знакового графа

Дано: знаковый граф $\Sigma = (G, \sigma)$, где $G = (V, E)$, $n = |V| \ge 2$, $m = |E| \ge 1$.

Требуется: найти индекс сбалансированности $I(\Sigma)$ графа Σ.

Задача вычисления индекса сбалансированности знакового графа сводится к задаче вычисления меры сбалансированности, т.к мера и индекс сбалансированности связаны следующим соотношением:

$$b(\Sigma) = \frac{1}{2} + \frac{I(\Sigma)}{2\,|\,C(\Sigma)\,|}.$$

Эти задачи являются NP-трудными, поскольку общее число простых циклов в полном графе определяется формулой [4]:

$$|C| = \sum_{k=3}^{n} \frac{n!}{(n-k)!2k} = O\left(2^n\right).$$

В общем случае для графа порядка n вычислительная сложность нахождения всех простых циклов равна $O\left(n^2 \cdot 2^n\right)$.

Для решения данных задач был разработан алгоритм, основывающийся на поиске фундаментального множества циклов [3]. Сначала ищется само фундаментальное множество циклов, затем, с помощью симметрических разностей различных комбинаций найденных циклов, находится множество всех простых циклов. Далее, для каждого цикла вычисляется его знак и вычисляются мера и индекс сбалансированности знакового графа.

Вычислительная сложность поиска фундаментального множества циклов составляет $O\left(nm+n\right)$. Пусть мощность фундаментального множества циклов равна k. Тогда сложность генерации подмножеств будет $O\left(2^k\right)$. Следовательно вычислительная сложность всего алгоритма будет $O\left(\left(m-n+1\right)n+2^k\right)$.

В табл. 4 приведены результаты вычислительных экспериментов, проведенных на графах с различной структурой, для которых было заранее известно, сбалансированы ли они.

Таблица 4

Мера и индекс сбалансированности

Количество вершин и ребер графа	Сбалансированность	Мера сбалансированности	Индекс сбалансированности	Время работы алгоритма, мс
$n = 5$, $m = 10$	нет	0.4	–7	7
$n = 5$, $m = 10$	да	1	37	7
$n = 6$, $m = 15$	нет	0.56	23	166
$n = 6$, $m = 15$	да	1	207	166
$n = 7$, $m = 21$	нет	0.44	–157	8077
$n = 7$, $m = 21$	да	1	1347	8077
$n = 8$, $m = 28$	нет	0.56	1229	743033
$n = 8$, $m = 28$	да	1	10125	745092

Заключение

В данной работе сформулированы основные задачи знакового баланса и установлена их вычислительная сложность. Для решения данных задач были разработаны и реализованы точные алгоритмы. Для задачи MBSP также были реализованы приближенный алгоритм и правило редукции. В табл. 5 приведена вычислительная сложность всех разработанных алгоритмов.

Таблица 5

Разработанные алгоритмы

Задача	Алгоритм	Тип алгоритма	Сложность
Распознавание сбалансированности	Алгоритм 1. Signed_Balance	точный	$O\left(n+m\right)$
MBSP	Решение с помощью CPLEX	точный	$O\left(2^{n+m}\right)$

Задача	Алгоритм	Тип алгоритма	Сложность
MBSP	Алгоритм 2. Жадная эвристика	приближенный	$O(m)$
Предобработка данных	Алгоритм 3. Предобработка	точный	$O(nm+n+m)$
Вычисление меры и индекса сбалансированности	Алгоритм 4. Вычисление меры и индекса сбалансированности	точный	$O(mn+n+2^k)$

Разработанные алгоритмы были реализованы в виде пакета программ, схема которого представлена на рис. 1. Входными данными являются количество вершин и ребер графа, список ребер со знаком каждого ребра. Алгоритм распознавания сбалансированности, эвристика и предобработка данных для задачи MBSP и алгоритм поиска меры и индекса сбалансированности реализованы в среде разработки Visual Studio 2019 на языке C++. Точный алгоритм решения задачи MBSP был реализован в среде IBM ILOG CPLEX Optimization Studio. Для проведения вычислительных экспериментов использовался персональный компьютер со следующими характеристиками: Intel Core i7-3630QM CPU @ 2.40 GHz, 8ГБ оперативной памяти, 64-разрядная операционная система Windows 10.

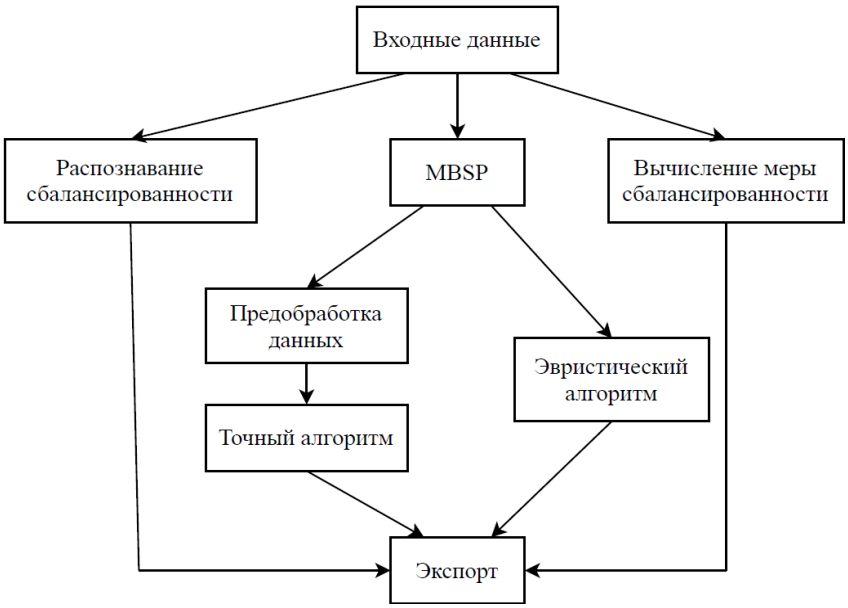


Рис. 1. Схема разработанного пакета программ

ЛИТЕРАТУРА

1. *Harary F.* Structural Balance: A Generalization of Heider's Theory // Psychological Review. 1956. Vol. 63 (5). P. 277–293.

2. *Figueiredo R., Frota Y.* The maximum balanced subgraph of a signed graph: Applications and solution approaches // European Journal of Operational Research 236. – 2014. – P. 473–487.

3. *Липский В.* Комбинаторика для программистов. Москва: «Мир», 1998

4. *Ибрагимова Э.И.* Сбалансированные графы и их применение / Э. И. Ибрагимова // Информационные технологии и математическое моделирование (ИТММ-2019) Материалы XVIII Международной конференции имени А. Ф. Терпугова. Часть 2, – 2019. – с. 15–20.

5. *Ибрагимова Э.И.* Исследование 2-сбалансированных знаковых графов / Э. И. Ибрагимова // Материалы VII Международной молодежной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем» 303, – 2019. – с. 276–281.

6. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

7. *Zaslavsky T.* Negative (and positive) circles in signed graphs: A problem collection / T. Zaslavsky // AKCE International Journal of Graphs and Combinatorics 15. –2018. – P. 31–48.

СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ ДЛЯ СЕМАНТИЧЕСКОЙ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ В РЕАЛЬНОМ ВРЕМЕНИ*

Игольников Н.А., Марков Н.Г.
Томский политехнический университет
nai6@tpu.ru, markovng@tpu.ru

Введение

В настоящее время актуальным научным и практически значимым направлением исследований является создание мобильных транспортных средств с интеллектуальными системами компьютерного зрения (СКЗ) на основе искусственных нейронных сетей. Такие СКЗ должны в автоматическом режиме в реальном времени анализировать изображения и распознавать объекты различной физической природы. Наиболее перспективной считается реализация мобильных СКЗ с использованием сверточных нейронных сетей (СНС) [1].

Для решения задачи семантической сегментации объектов на изображениях широко применяются СНС класса U-Net – классическая архитектура U-Net [2] и производные от нее архитектуры. Сегодня перспективной для решения задачи семантической сегментации ряд исследователей также считает СНС ICNet (англ. Image Cascade NETwork) [3].

Большая часть исследований при решении задачи семантической сегментации изображений с помощью СНС нацелена на достижение высокого качества сегментации и оставляет открытыми вопросы скорости обучения и исполнения модели СНС, сложности модели и затрат вычислительных ресурсов [3]. Однако в мобильных СКЗ существует необходимость решения задачи семантической сегментации с помощью аппаратно-реализованных СНС, например, с использованием программируемых логических интегральных схем (ПЛИС). При аппаратной реализации необходимо использовать модели СНС с минимальными требованиями к памяти и производительности вычислительного устройства СКЗ. Это делает актуальной задачу исследования по разным критериям эффективности наиболее распространенных моделей СНС, применяемых для семантической сегментации изображений.

Целью данной работы является исследование эффективности СНС класса U-Net и СНС ICNet и оценка возможности решения задачи семантической сегментации в реальном времени в случае последующей аппаратной реализации таких нейронных сетей.

1. Архитектура классической СНС U-Net

Рассмотрим особенности моделей СНС класса U-Net и СНС ICNet. Сначала будем анализировать архитектуру классической СНС U-Net.

Классическая СНС U-Net имеет архитектуру типа автоэнкодер и состоит из энкодера и декодера (рис. 1). Энкодер и декодер, в свою очередь, состоят из блоков, включающих сверточные слои, слои нормализации, активации и слои понижения (или повышения) размерности. Между энкодером и декодером производятся пробросы карт признаков (стрелки с ориентацией слева направо).

Основная идея этой архитектуры СНС заключается в том, чтобы охватить глобальный контекст изображения и в то же время сохранить пространственную информацию путем пробросов признаков от энкодера к декодеру, что позволяет сегментировать изображение с высокой точностью. В верхней части рис. 1 слева изображены составляющие блока энкодера и его условное обозначение, справа – составляющие блока декодера и его условное обозначение. В нижней части – схема архитектуры в общем виде.

На рис. 1 использованы следующие обозначения:

* Исследования были поддержаны грантом РФФИ №18-47-700010р-а.

– Conv – сверточный слой;
– ELU – функция активации ELU;
– Pool – слой понижения дискретизации;
– ConvTranspose – слой транспонированной свертки;
– En – блок энкодера;
– De – блок декодера.
Возле стрелок указаны размерности карт признаков.

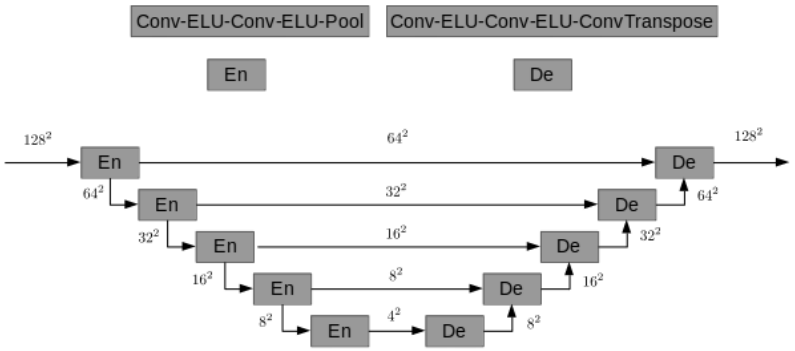


Рис. 1. Архитектура СНС U-Net

2. СНС U-Net с разреженной сверткой

В данной архитектуре СНС класса U-Net используются слои дилатационной свертки (англ. dilated convolution) [4], часто переводимой с английского как “разреженная свертка”.

Разреженная свертка отличается от классической свертки добавлением нулевых коэффициентов в сверточное ядро. Вводится параметр “фактор разреженности”, который показывает расстояние между соседними ненулевыми коэффициентами ядра. Сверточные ядра с различными значениями фактора разреженности приведены на рис. 2. Ненулевые коэффициенты сверточных ядер на рис. 2 обозначены точками.

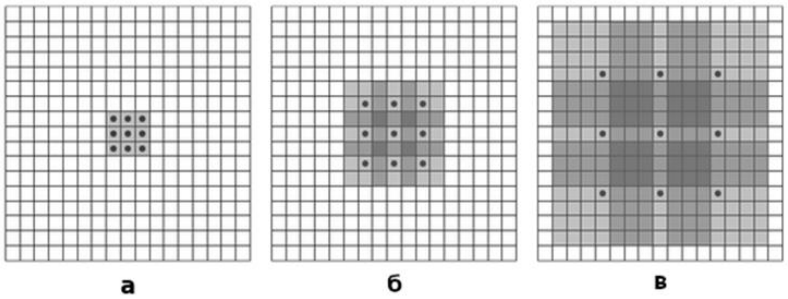


Рис. 2. Сверточные ядра с различным фактором разреженности: а – 1, б – 2, в – 4

Сверточное ядро с фактором разреженности, равным единице, эквивалентно ядру обычной свертки. Сверточное ядро размерностью 3x3 с фактором разреженности 2 имеет область восприятия размерностью 5x5, с фактором 4 – 9x9.

Архитектура СНС U-Net с разреженной сверткой приведена на рис. 3. Помимо ранее использованных обозначений, на рис. 3 применены следующие:
– DC – слой разреженной свертки;
– BN – батч-нормализация;
– Drop – слой dropout-нормализации.

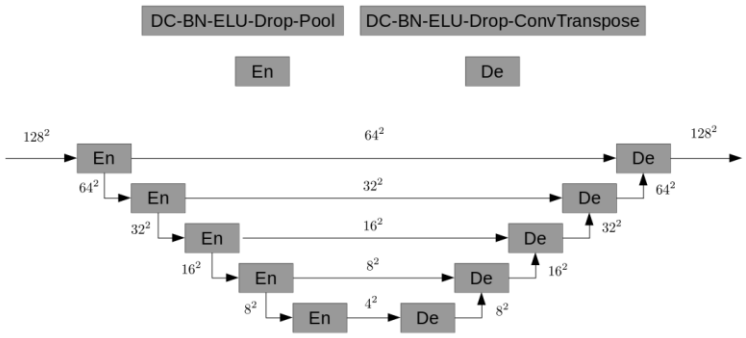


Рис. 3. Архитектура СНС U-Net с разреженной сверткой

Отличие СНС U-Net с разреженной сверткой от классической СНС U-Net состоит в том, что в блоках энкодера и декодера две свертки 3x3 заменены одной разреженной сверткой 3x3 с фактором 2.

Две последовательные свертки 3x3 эквивалентны одной свертке 5x5 [5]. При этом в первом случае количество весовых коэффициентов равно 18, а во втором – 25. Разреженная свертка 3x3 с фактором 2 также имеет область восприятия 5x5, а количество весовых коэффициентов в таком ядре свертки равно 9. Таким образом, в разреженной версии архитектуры СНС U-Net весовых коэффициентов сверточных фильтров в два раза меньше при неизменной области восприятия. Это приводит к значительному упрощению архитектуры СНС и, как следствие, к ускорению ее выполнения на вычислительном устройстве.

3. Архитектура СНС V-Net

Основное отличие архитектуры СНС V-Net [6] от ранее рассмотренных архитектур СНС класса U-Net заключается в наличии плотно-соединенных блоков в энкодере и декодере. Кроме этого добавлен блок скрытого представления между энкодером и декодером (обозначение – Bottleneck).

Архитектура СНС V-Net представлена на рис. 4. В верхней части рис. 4 изображены составляющие блоков энкодера и декодера, составляющие блока скрытого представления и условные обозначения этих блоков. В нижней части рис. 4. изображена схема архитектуры СНС V-Net с использованием условных обозначений составляющих ее блоков.

Возле стрелок на рис. 4. указаны размерности карт признаков. При проходе через блок скрытого представления размерность карт признаков остается неизменной. При передаче признаков в энкодере в нижележащие слои размерность карт признаков понижается (например, посредством субдискретизации), а при передаче признаков в декодере в вышележащие слои – повышается (например, посредством билинейной интерполяции).

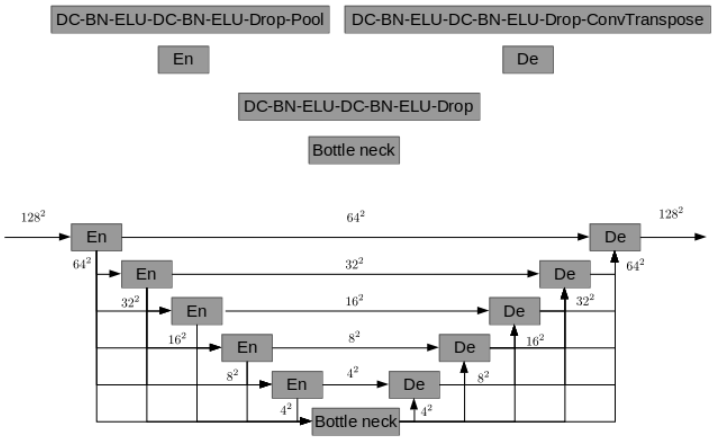


Рис. 4. Архитектура СНС V-Net

4. Архитектура СНС ICNet

Архитектура СНС ICNet состоит из трех ветвей, каждая из которых является самостоятельной СНС, обучаемой отдельно, помимо основного процесса обучения нейросети в целом.

Входное изображение и маска сегментации подаются в трех масштабах – исходном, уменьшенном в два и в четыре раза.

Первая ветвь является наиболее глубокой и обрабатывает изображения, уменьшенные в четыре раза.

Вторая ветвь обрабатывает изображения, уменьшенные в два раза.

Третья ветвь – простейшая из трех и обрабатывает изображение в исходном разрешении.

Схема архитектуры СНС ICNet приведена на рис. 5.

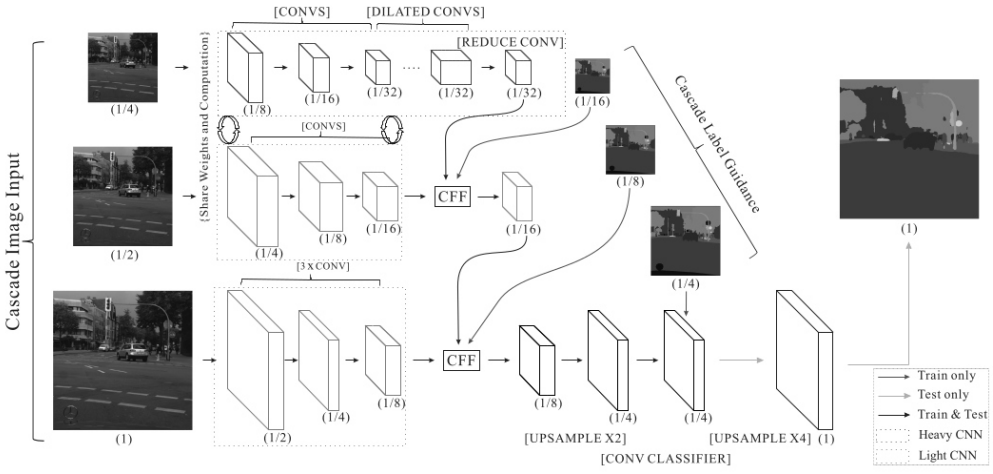


Рис. 5. Архитектура СНС ICNet [3]

5. Постановка задачи исследования эффективности моделей СНС

В качестве исследуемых моделей СНС выбраны четыре модели СНС с рассмотренными выше архитектурами:

– классическая СНС U-Net;

– СНС U-Net с разреженной сверткой;
– СНС V-Net;
– СНС ICNet.

В качестве обучающей и тестовой выборок использовался датасет TGS Salt Identification [7]. Пример изображения из обучающей выборки и маска сегментации для него показаны на рис. 6.

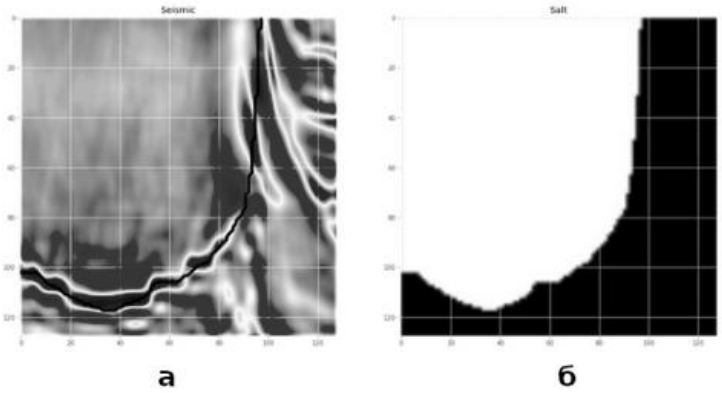


Рис. 6. Пример изображения и его маски из обучающей выборки: а – исходное изображение, б – маска сегментации

Размерность изображений выборки – 128x128x1. Обучение моделей СНС проводилось на подмножестве из этого датасета в виде выборки из 3600 изображений. Тестирование проводилось на 400 изображениях. Размер батча выбран равным 16, как наиболее удачный компромисс между временем обучения модели и скоростью сходимости процесса обучения СНС. В качестве алгоритма оптимизации выбран алгоритм Adam. Для обучения моделей использовался сервис Google Colaboratory [8].

Сравнение исследуемых моделей СНС производилось по следующим основным критериям:

– точность сегментации (метрика accuracy);
– время сегментации изображения;
– количество операций с плавающей запятой;
– количество хранимых чисел с плавающей запятой (объем промежуточных буферов и весовых коэффициентов СНС).

Исследования точности сегментации изображений для каждой из моделей СНС проводились отдельно для обучающей и тестовой выборок. Результаты для каждой из выборок должны получаться путем усреднения всех полученных значений метрики. Исследования моделей СНС по остальным трем критериям проводились на 4000 изображений. Результаты для каждого критерия также должны получаться усреднением всех полученных значений.

6. Результаты исследования моделей СНС

Результаты исследования точности сегментации изображений для всех моделей СНС приведены в табл. 1.

Таблица 1

Результаты исследования СНС с различной архитектурой по точности сегментации

Архитектура	Точность сегментации изображений на обучающей выборке, %	Точность сегментации изображений на тестовой выборке, %
U-Net	94.71	90.02
U-Net с разреженной сверткой	91.66	89.81
V-Net	94.94	92.36
ICNet	95.66	93.73

Из табл. 1 следует, что наилучший результат по точности сегментации изображений показывает модель СНС ICNet. Модель СНС U-Net с разреженной сверткой уступает по точности сегментации моделям СНС с более сложными архитектурами.

Результаты для сравнения исследуемых моделей СНС по среднему времени сегментации одного изображения приведены в табл. 2. Из нее следует, что модель СНС U-Net с разреженной сверткой показывает наилучший результат по времени сегментации изображений.

Таблица 2

Результаты исследования СНС с различной архитектурой по среднему времени сегментации одного изображения

Архитектура	Среднее время сегментации одного изображения, мс
U-Net	2.50
U-Net с разреженной сверткой	2.11
V-Net	3.89
ICNet	4.72

В табл. 3 приведены результаты исследования моделей СНС по количеству операций с плавающей запятой, а в табл. 4 – по суммарному объему промежуточных буферов и объему весовых коэффициентов. И в табл. 3, и в табл. 4 значения этих параметров указаны для одного входного изображения.

Таблица 3

Результаты исследования СНС с различной архитектурой по количеству операций с плавающей запятой

Архитектура	Операции		
	Умножения, MFLOP	Сложения, MFLOP	Сравнения, KFLOP
U-Net	64.98	64.98	666.62
U-Net с разреженной сверткой	31.80	31.80	428.54
V-Net	221.74	221.74	823.30
ICNet	221.48	221.48	837.63

Таблица 4

Результаты исследования СНС с различной архитектурой по количеству хранимых чисел с плавающей запятой

Архитектура	Суммарный объем промежуточных буферов, МБ	Объем весовых коэффициентов СНС, МБ
U-Net	5.52	1.97
U-Net с разреженной сверткой	3.61	1.08
V-Net	13.40	6.24
ICNet	7.40	26.72

Из табл. 3, 4 следует, что модель СНС U-Net с разреженной сверткой наименее требовательна к ресурсам памяти вычислительного устройства мобильной СКЗ.

Анализируя все полученные результаты исследования наиболее известных моделей СНС на предмет возможности решения задачи семантической сегментации в режиме реального времени, предпочтение следует отдать модели U-Net с разреженной сверткой. Она же наименее требовательна к памяти вычислительного устройства СКЗ при её аппаратной реализации. По точности сегментации изображений эта модель СНС незначительно уступает классической U-Net, однако её по точности на 4% превосходит лидер по этому критерию – модель нейросети ICNet.

Заключение

Сформулирована актуальная задача исследования эффективности наиболее известных моделей СНС для семантической сегментации изображений в реальном времени.

В результате исследований выявлено, что модель СНС U-Net с разреженной сверткой наиболее пригодна для решения задачи семантической сегментации изображений в реальном времени. Она же наименее требовательна к памяти вычислительного устройства мобильной СКЗ при её потенциальной аппаратной реализации. Все это весьма важно при создании вычислительного устройства мобильной СКЗ на ПЛИС.

Однако разработчикам мобильной СКЗ необходимо учитывать не очень высокую точность сегментации изображений, обеспечиваемую этой моделью СНС.

Если необходима высокая точность сегментации, то следует использовать модель СНС ICNet. Однако для её аппаратной реализации и функционирования в реальном времени требуются значительные ресурсы вычислительного устройства мобильной СКЗ.

ЛИТЕРАТУРА

1. *Mennatullah S.* RTSeg: Real-time Semantic Segmentation Comparative Study [Электронный ресурс] URL: arXiv:1803.02758 [cs.CV](дата обращения 28.04.2020).

2. *Olaf R.* U-Net: Convolutional Networks for Biomedical Image Segmentation [Электронный ресурс] URL: arXiv:1505.04597 [cs.CV] (дата обращения 10.02.2020).

3. *Hengshuang Z.* ICNet for Real-Time Semantic Segmentation on High-Resolution Images [Электронный ресурс] URL: arXiv:1704.08545 [cs.CV] (дата обращения 22.03.2020).

4. *Fisher Y.* Multi scale context aggregation by dilated convolutions [Электронный ресурс] URL: arXiv:1511.07122 [cs.CV] (дата обращения 23.02.2020).

5. CS231n Convolutional Neural Networks for Visual Recognition [Электронный ресурс] URL: http://cs231n.github.io/convolutional-networks/#layerpat (дата обращения 15.03.2020).

6. *Milletari F.* V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation [Электронный ресурс] URL: arXiv:1606.04797 [cs.CV] (дата обращения 07.04.2020).

7. TGS Salt Identification Challenge [Электронный ресурс] URL: https://www.kaggle.com/c/tgs-salt-identification-challenge/data (дата обращения 20.02.2020).

8. Google Colaboratory [Электронный ресурс]URL: https://colab.research.google.com (дата обращения 27.04.2020).

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РАЗЛИЧНЫХ ПОКАЗАТЕЛЕЙ ЦИТИРОВАНИЯ ДЛЯ ОЦЕНКИ И РАНЖИРОВАНИЯ КОНФЕРЕНЦИЙ*

Кочетков Д.М.[1], Бирюков А.А.[2], Ермолаева А.М.[2]

[1]*Национальный исследовательский университет «Высшая школа экономики», Москва, Россия*

[2]*Российский университет дружбы народов, Москва, Россия*

kochetkovdm@hotmail.com, birukou@gmail.com, ermolaevaanna@bk.ru

Аннотация

Значительная часть оригинальных результатов исследований публикуется в сборниках материалов конференций, что особенно характерно для таких предметных областей, как компьютерные науки и инженерия. К сожалению, материалы конференций часто недооцениваются с точки зрения оценки научных исследований. Кроме того, методология оценки и ранжирования конференций все еще находится в зачаточном состоянии. Наше исследование является попыткой восполнить этот пробел. Большинство методов оценки в наукометрии основаны на цитировании как относительной мере научного влияния (хотя ее часто путают с качеством). Оценка журналов использует простой подсчет цитирований, количество цитирований на одну статью, нормализованную цитируемость, h-индекс, период полужизни полученных/сделанных ссылок, а также производные, такие как импакт-фактор. В этой статье мы определяем основанные на цитировании метрики для конференций и вычисляем их по набору данных, содержащему список публикаций и цитирований из Scopus для топ-100 конференций по компьютерным наукам согласно Microsoft Academic. Затем мы сравним полученные метрики с метриками значимости (saliency) и престижа (prestige) от Microsoft Academic. Результаты показывают сильную корреляцию на нашем наборе данных. Эта работа представляет собой промежуточные результаты исследований,

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №18-00-01040 КОМФИ «Влияние новых технологий на городскую среду и качество жизни городских сообществ».

которые мы хотим развивать дальше как в академических исследованиях, так и в создании инструментов для оценки качества конференций.

Введение

Сборники материалов конференций и журналы являются важными площадками для публикации оригинальных материалов результатов исследований. Кроме того, материалы конференций играют значительную роль в ряде сообществ, такие как научные сообщества компьютерных наук и инженерии [1]. Несмотря на то, что уже разработано большое количество методов оценки и ранжирования журналов, процесс создания качественной системы оценки материалов конференций находится на очень ранней стадии.

В наукометрических исследованиях цитаты часто рассматриваются как показатель степени влияния статьи в научном сообществе. Неудивительно, что большинство показателей журналов также основаны на цитировании [2]. Помимо простого подсчета цитат и количества цитирований на статью, период полужизни полученных/сделанных ссылок (cited/citing half-life) является важным показателем, поскольку он указывает, как долго результаты оказывают влияние в научном сообществе. Период полужизни сделанных журнала - это средний возраст статей, цитируемых в данном журнале за данный год. По определению, половина ссылок в журнале будет старше периода полужизни; другая половина будет моложе. Дэвис и Кокран [3] проанализировали журналы, перечисленные в Journal Citation Reports с использованием статистических методов. Авторы пришли к выводу, что средневзвешенный период полужизни цитат для всех категорий журналов составил 6,5 лет, но это значение варьируется для отдельных категорий от 5,1 до 7,1 лет. Это показывает, что такие показатели, как классический импакт-фактор, который вычисляется для двухлетнего окна цитирования, не отражают продолжительность периода, в течении которого цитируется статья. Другой часто используемой метрикой является h-индекс, определяемый как максимальное значение h такое, что данный автор/журнал опубликовал h статей, каждая из которых цитируются как минимум h раз. Бар-Илан [4] сравнила h-индекс по всем трем показателям, используя разные базы данных: Web of Science, Scopus и Google Scholar. Результаты показывают, что выбор базы данных для расчета h-индекса играет важную роль. Схожие метрики цитируемости могут быть применены к конференциям, так как они проводятся периодически и имеют достаточно материалов для анализа.

Помимо метрик, основанных на цитировании, есть также метрики основанные на мнении сообщества или экспертов. Наиболее ярким примером является Computing Research and Education Association of Australasia (CORE). Кунгас и др. [5] провели исследование объективности CORE и двух рейтингов конференций, основанных на мнении сообщества, и пришли к выводу, что процент принятия (acceptance rate) статей является лучшим предиктором положения конференции в рейтинге. Тем не менее, использование дополнительных показателей, таких как количество ссылок на материалы конференции и количество цитат на 1 статью, улучшают предсказание рейтинга конференции. Эти метрики показали высокую точность для конференций высокого уровня, в то время для конференций среднего и более низкого уровня были продемонстрированы более слабые результаты.

Мехо показал важность включения конференций в оценку научных исследований [6], продемонстрировав, что 30% лучших публикаций в компьютерных науках - это конференции. Авторы попытались ответить на вопрос: является ли Scopus CiteScore инструментом, который будет достаточно точен для оценки качества конференция по компьютерным наукам? Ответ был положительным, кроме того, CiteScore имеет преимущество по сравнению с h-индексом, так как позволяет сравнивать конференции независимо от их возраста и размера, если им не менее четырех лет.

Следует также упомянуть здесь большую группу нормализованных метрик цитируемости, например, Field-Weighted Citation Impact (FWCI) и Source Normalized Impact per Paper (SNIP). Обе они не зависят от размера. FWCI используется для оценивания отдельных статей. Это отношение между фактическим числом цитат, полученных набором публикаций, и средним числом цитат, полученных всеми другими подобными публикациями в области. Последнее называется ожидаемым уровнем цитируемости или потенциалом цитирования. Подобные публикации относятся к одной и той же дисциплине одного и того же типа документа, и того же возраста [7]. SNIP рассчитывается путем деления количества цитирований на одну статью в журнале на ожидаемый уровень цитируемости в предметной области [8,9]. Исследование [10] подтверждает, что из различных журнальных метрик SNIP имеет наибольшую корреляцию с экспертными рейтингами такими как Excellence in Research for Australia (ERA), возможно, из-за его нормализованного характера.

В этой статье мы будем использовать две метрики для оценки воздействия конференции: суммарное нормализованное цитирование Total Normalized Citations (TNC) и индекс нормализованного цитирования Normalized Citation Index (NCI). Эти показатели основаны на количестве опубликованных работ и количестве ссылок (цитат) на материалы конференций по годам, а также среднее цитирование в мире на материалы конференций по годам. Более подробная информация о теоретической предыстории метрик, основанных на цитировании, можно найти в работе [11]. Мы вычислили метрики для топ-100 конференций по компьютерным наукам. Список основывался на Microsoft Academic. Было взято количество публикаций и цитирований за период времени из баз данных Scopus и SciVal. Мы также сравнили полученные метрики с показателями престижа (prestige) и значимости (saliency) от Microsoft Academic и выявили значимую корреляцию. Таким образом, можно предположить, что предложенные метрики могут быть использованы для оценки влияния конференций.

1. Данные и методы

Чтобы вычислить метрики, мы создали набор данных на основе Microsoft Academic Search (MAS) и данных Scopus. Во-первых, мы выбрали топ-100 конференций из MAS ранжированных по значимости (saliency) и престижу (prestige) за все годы. Во-вторых, для каждой конференции, мы получили общее количество статей и их цитирований за период 2005–2017 из Scopus. Это было сделано с помощью функции "расширенный поиск", для поиска конкретной аббревиатуры конференции с контролем источника в случае повторения аббревиатуры. Для трех конференций (Advances in Computing and Communications (ACC), International Symposium on Antennas and Propagation (ISAP), и Network and Distributed System Security Symposium (NDSS)) не было данных в Scopus за данный период. Мы также исключили из расчета конференции, которые не имели данных за последние три года (National Conference on Articial Intelligence (AAAI), Hawaii International Conference on System Sciences (HICSS) и Symposium on VLSI Circuits (VLSIC)). Это оставило нас с выборкой из 94 конференций. Используя количество публикаций и количество цитирований, мы вычислили среднее число цитат на одну статью для данного года. Такое число представляет собой количество цитат, накопленных с момента публикация до определенного года.

Нормализованная цитируемость за год *n* была рассчитана по следующей формуле:

$$NC_n = \frac{CCP_i}{CCP_{cp}},$$

где CCP_i – среднее число цитирований на одну статью данной конференции с момента публикации до текущего момента, CCP_{cp} – средний показатель количества цитирований на один доклад конференции того же года в мире, взятых по SciVal.

Нормализованный индекс цитирования (Normalized Citation Index) является кумулятивной метрикой:

$$NCI = \frac{\sum_{n=1}^{N} TNC_n}{\sum_{n=1}^{N} P_n},$$

где TNC_n – суммарное количество цитирований (Total Normalized Citations) материалов конференции за год n; P_n – количество публикаций в сборнике материалов конференции за год n; N – число лет. Для целей настоящего исследования мы взяли данные начиная с 2005 г. Суммарное нормализованное цитирование (Total Normalized Citations) является произведением нормализованных цитирований (Normalized Citations) года n и количества статей в материалах конференций года n:

$$TNC_n = NC_n \times P_n .$$

Мы сравнили полученные результаты с показателями "saliency" и "prestige", доступными в Microsoft Academic.

Согласно Ванг [12], значимость (saliency) – это аналитическая и прогностическая метрика, которая используется для ранжирования в Microsoft Academic. Эта метрика использует подход обучения с подкреплением к оценке важности субъектов, участвующих в научной коммуникации. Поскольку рассмотрение базовой сети происходит в соответствии с изменением временных условий, престиж (prestige) – это стохастический процесс. Чтобы смоделировать временные характеристики значимости, используется процесс авторегрессии скользящего среднего (ARMA), позволяющий аппроксимировать нестационарное распределение. Обратите внимание, что здесь также можно использовать простой процесс авторегрессии первого порядка (AR). Смешанная модель также используется для учета неоднородности сети, где взвешенная сумма значимости объектов, связанных с публикацией, является значимостью. Престиж – это соответствующая нормализованная версия значимости, определяемая как среднее значение значимости набора публикаций.

Для сравнения мы использовали корреляционный анализ, а именно – ранговую корреляцию Спирмена [13]. Коэффициент корреляции Спирмена является частным случаем общей корреляции и вычисляется как коэффициент корреляции Пирсона между ранжированными переменные. Необработанные баллы n преобразуются в ранги rgX_i и rgY_i, а коэффициент корреляции вычисляется следующим образом:

$$\rho = \frac{\operatorname{cov}\left(rg_X, rg_Y\right)}{\sigma_{rg_X}\sigma_{rg_Y}},$$

где ρ – обычный коэффициент корреляции Пирсона, но примененный к рангу переменных; $\operatorname{cov}\left(rg_X, rg_Y\right)$ является ковариацией ранжированных переменных; σ_{rg_X}, σ_{rg_Y} – стандартные отклонения ранжированных переменных.

2. Результаты

Результаты корреляционного анализа различных показателей материалов конференций отображены в табл. 1. Во-первых, мы подсчитали количество цитирований на одну статью и общее количество цитат (TS) как простые метрики, основанные на данных Scopus и SciVal. Тогда мы сравнили ее производные (вычисляемые TNS и NCI, а также значимость и престиж из Microsoft Academic).

Таблица 1

Корреляционный анализ метрик сборников конференций
(* – значимость на уровне 0,05, ** – значимость на уровне 0,01)

	TC Rank	CPP Rank	TNC Rank	NCI Rank	Saliency	Prestige
TC Rank	1.00	0.237*	0.939**	0.228*	0.737**	0.085
CPP Rank	0.237*	1.00	0.207*	0.985**	0.023	0.865**
TNC Rank	0.939**	0.207*	1.00	0.235*	0.694**	0.083
NCI Rank	0.228*	0.985**	0.235*	1.00	0.027	0.860**
Saliency	0.737**	0.023	0.694**	0.027	1.00	0.026
Prestige	0.085	0.865**	0.083	0.860**	0.026	1.00

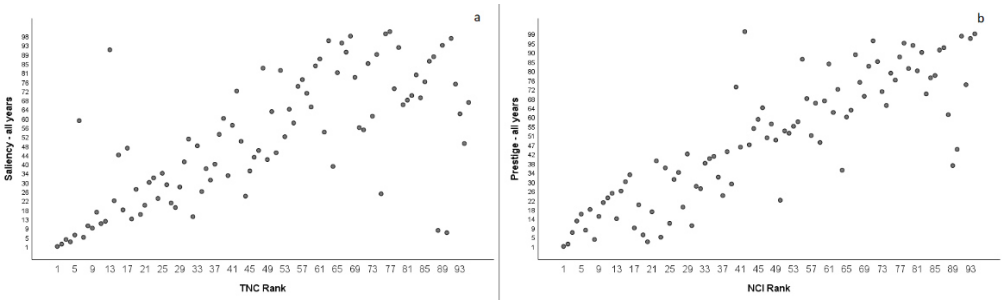


Рис. 1. Корреляционный анализ между a) Общее нормализованное количество цитирований (TNC) и Saliency; b) Нормализованный индекс цитирования (NCI) и Prestige.

Вполне естественно, что существует довольно сильная корреляция между нормализованными цитированием на публикацию и простым цитированием в расчете на одну публикацию, учитывая, что анализ происходит в рамках небольшой выборки "элитных" конференций. То же самое верно и для метрик, зависящих от размера (TC и TNC). В то же время существует очень сильная корреляция между размерно-зависимыми метриками на основе Scopus и Microsoft Academic (TNC и Saliency), а также независимыми от размера метриками (NCI и Prestige) (рис. 1). Полученные результаты свидетельствуют о том, что предложенные нами метрики конференций в этой статье сильно коррелируют с проприетарными метриками Microsoft Academic и может быть использован для оценки конференций.

3. Выводы и будущая работа

В этой статье мы определили показатели цитирования для конференций и протестировали их на данных Microsoft Academic и Scopus. Результаты показали значимую корреляцию между показателями импакта, рассчитанными на основе данных Scopus, и показателями значимости и престижа, которые используются для ранжирования конференций в Microsoft Academic. Таким образом, эти метрики могут быть использованы в качестве точного показателя импакта конференции. Предложенные нами метрики, зависящие от размера суммарно нормализованы цитат (TNC) и независимых от размера нормализованный индекс цитирования (NCI), может быть использован для крупномасштабного анализа и ранжирования конференций. Будущие исследования включают в себя тестирование метрик на большем наборе конференций и сравнение таких метрик с экспертными рейтингами и рейтингами основанные на мнении сообщества, такие как CORE или рейтинг конференций China Computer Federation (CCF).

ЛИТЕРАТУРА

1. *Shamir, L.* The effect of conference proceedings on the scholarly communication in computer science and engineering. Scholarly and Research Communication 1, 2 (2010).

2. *Mingers, J., and Leydesdorff, L.* A review of theory and practice in scientometrics. European journal of operational research 246, 1 (2015), 1-19.

3. *Davis, P.M., and Cochran, A.* Cited half-life of the journal literature. arXiv preprint arXiv:1504.07479 (2015).

4. *Bar-Ilan, J.* Which h-index? – a comparison of wos, scopus and google scholar. Scientometrics 74, 2 (2008), 257-271.

5. *Kungas, P., Karus, S., Vakulenko, S., Dumas, M., Parra, C., and Casati F.* Reverse-engineering conference rankings: what does it take to make a reputable conference? Scientometrics 96, 2 (2013), 651-665.

6. *Meho, L.I.* Using scopus's citescore for assessing the quality of computer science conferences. Journal of Informetrics 13, 1 (2019), 419-433.

7. Elsevier. Snowball metrics, 2020.

8. *Moed, H.F.* Measuring contextual citation impact of scientific journals. Journal of Informetrics 4, 3 (2010), 265-277.

9. *Waltman, L., van Eck, N.J., van Leeuwen, T.N., and Visser, M.S.* Some modifications to the snip journal impact indicator. Journal of Informetrics 7, 2 (2013), 272-285.

10. *Haddawy, P., Hassan, S.-U., Asghar, A., and Amin, S.* A comprehensive examination of the relation of three citation-based journal metrics to expert judgment of journal quality. Journal of Informetrics 10, 1 (2016), 162-173.

11. *Waltman, L., van Eck, N.J., van Leeuwen, T.N., Visser, M.S., and van Raan, A.F.* Towards a new crown indicator: An empirical analysis. Scientometrics 87, 3 (2011), 467-481.

12. *Wang, K., Shen, Z., Huang, C.-Y., Wu, C.-H., Eide, D., Dong, Y., Qian, J., Kanakia, A., Chen, A., and Rogahn, R.* A review of microsoft academic services for science of science studies. Frontiers in Big Data 2 (2019), 45.

13. *Myers, J.L., and Well, A.D.* Research Design and Statistical Analysis (2nd ed.). Mahwah, N.J. : Lawrence Erlbaum Associates, 2003.

АНАЛИЗ ОШИБОК БИНАРНОГО КЛАССИФИКАТОРА ТЕКСТОВ С ПРИМЕНЕНИЕМ МЕТА-ПРИЗНАКОВ

Павлюченко М.В., Кабанова Т.В.
Томский государственный университет
pavluchenkomaria@gmail.com, tvk@bk.ru

Введение

Благодаря техническому прорыву появилась возможность хранить и обрабатывать массивы данных, во много превосходящие по объему те, что общество могло позволить себе еще несколько десятилетий назад. В данной работе будет рассматриваться информация в текстовой форме, поэтому стоит уточнить, что в современном мире в режиме реального времени и ограниченного количества ресурсов происходят процессы создания, обработки и анализа огромного количества неструктурированной текстовой информации – новостей, записей в блогах, научных статей, обучающих материалов. Потребности общества в подобной информации растут с каждым годом, однако вместе с этим растет и необходимость в её классификации, т.к. это упрощает её поиск и делает его более быстрым.

Свойства исходных данных с течением времени меняются. Поэтому существует потребность обучения классификатора в режиме реального времени при постоянном поступлении обучающих примеров для более гибкой работы. Также различные входные данные приводят к неповторяющимся результатам, что затрудняет улучшение результатов работы классификатора. Еще одной из проблем является редкость некоторых данных в общем потоке, что затрудняет их обнаружение и в дальнейшем верную классификацию.

Поэтому актуальность работы заключается в том, что у современного общества есть необходимость качественной и, что важно, автоматической классификации огромного количества текстовой информации, однако на данный момент возникает необходимость в оценке результатов работы такого классификатора.

Мета-обучение представляет собой процесс самостоятельного увеличения системой своей эффективности с помощью получаемого ей опыта при решении задачи, оно направлено на обнаружение способов динамического поиска наилучшей стратегии обучения по мере увеличения количества задач [1]. Решение проблемы мета-обучения – вопроса о том, что позволяет алгоритму обучения доминировать в ограниченной области в пространстве всех возможных задач обучения – может указать, как согласовать алгоритмы обучения со свойствами задачи, что дает принципиальный подход к дина-

мическому выбору данных алгоритмов. Также мета-обучение позволит решить проблему учебных задач, лежащих за пределами доступных обучающих алгоритмов [2].

Существуют различные подходы к реализации мета-области и ввода некоторого набора правил:

1. Построение новой функции как дизъюнкции двух оригинальных функций, таким образом увеличивая пространство доступных гипотез [3].

2. Исключение функций, когда смещение, характеризующее способность модели алгоритмов настраиваться на целевую зависимость, считается неуместным [4].

3. Определения области мета-функций, являющихся релевантными производительности алгоритма обучения, над которой вводится набор правил для обнаружения условий, при которых алгоритм обучения превосходит другие [5]. В одной из статей для извлечения выбраны следующие характеристики: количество примеров, атрибутов и классов, коэффициент стандартного отклонения для каждого признака, асимметрия, отношение шум-сигнал [6].

Анализ результатов деятельности классификаторов в большинстве случаев остается на уровне подсчета метрик качества, однако и здесь существуют свои достаточно важные нюансы [7]:

1. Полученные эмпирические данные стоит оценивать только в проекции на особенности используемых в процессе обучения и работы модели текстовых данных, т.к. это сделает дальнейший анализ более понятным и прозрачным.

2. Для работы бинарного классификатора существует возможность “ловушки” при использовании метрик точности из-за того, что данные метрики не обладают чувствительностью к наличию или отсутствию эффективности классификатора относительно текста в случае, когда количество категорий велико, а среднее отношение количества категорий к числу текстовых документов мало.

Вопрос использования мета-функций для текстовой информации уже затрагивался в рамках задачи автоматического присвоения документов заранее определенному набору категорий, как общий индуктивный процесс, автоматически создающий классификатор для категории A1, анализируя характеристики набора документов, которые были разделены на категории A1 и A2 экспертом в данной области; исходя из этих характеристик, индуктивный процесс позволял выявить характеристики, которые должен иметь новый документ, чтобы его можно было классифицировать принадлежащим категории A1. Однако, этот подход к реализации требует большого количества размеченных данных [8].

1. Постановка задачи

Данная работа связана с построением мета-функций согласно статье [9]. X и C обозначают пространство исходных наблюдений и пространство классов соответственно. Мета-функции на основе метода k ближайших соседей предназначены для замены исходного пространства ввода X новым информативным и компактным пространством ввода M. Пространство M – мета-пространство – представляет собой конкатенацию числовых векторов мета-функций, определенных для каждого наблюдения $x \in X$ и категории $c_j \in C$ для $j = \overline{1,|C|}$.

Задача данной работы построить заданные мета-функции и проанализировать их действие для бинарного классификатора текстов с целью увеличения его точности за счет уменьшения количества сделанных им ошибок.

2. Построение библиотеки и описание данных

В рамках данной работы создана библиотека функций, реализующая согласно их описанию функции из статьи Кануто [9]. Добавочными требованиями к ней являются наличие тестов для каждой функции и совместимость с библиотекой sklearn. Данная

библиотека реализована на языке программирования Python с использованием дополнительных библиотек анализа данных.

Для нашей задачи был взят датасет новостей из открытого источника Kaggle, который содержит около 200 тысяч новостей на английском языке с 2012 по 2018 гг., полученных от HuffPost. Данные были подготовлены к дальнейшему анализу путем парсинга страниц, удаления нерелевантных символов, перевода спецсимволов в словесный формат, лемматизации и дальнейшей векторизации.

Особенностью данных была их несбалансированность. Эта проблема была решена путем составления балансирующей функции, сохраняющей однородность данных.

3. Описание и анализ мета-функций

В данном подразделе будут рассмотрены некоторые из мета-функций, предложенных в статье Кануто [9] и будет проведен их анализ с точки зрения улучшения результатов классификатора. Для каждой из них будут представлены описание и сделанные предположения.

Centroid мета-функция

Здесь используется понятие центроида класса – вектора, построенного путем усреднения каждой координаты среди векторов класса.

1. Диагональ означает положение, равноудалённое от обоих центроидов, которое нельзя классифицировать, как однозначно определенное. На графике (рис. 1) по оси абсцисс расстояние до центроида 0 класса, а по оси ординат – до центроида 1 класса.

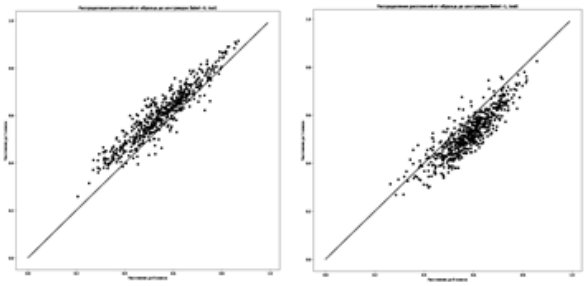


Рис. 1. Распределение расстояний от образца до центроидов, класс 0 слева и класс 1 справа, test

2. Модуль расстояния экземпляра до диагонали учетом знака соответствующего центроида должен превышать эмпирически заданное значение epsilon для уменьшения вероятности ошибки в работе классификатора.

На левом графике (рис. 2) верхняя горизонтальная линия обозначает диагональ, а нижняя показывает значение epsilon, задаваемое исследователем, между этими линиями выделяется участок под диагональю, где распределение объектов для данного класса еще велико, и поэтому в данном промежутке будет велика вероятность ошибки для определения другого класса. На правом графике (рис. 2) нижней является линия диагонали, а верхняя обозначает значение epsilon. Задача исследователя найти такое число, чтобы между линиями было достаточно много образцов, однако область не была бы слишком велика, потому что далее подразумевается её исключение для дальнейшего анализа.

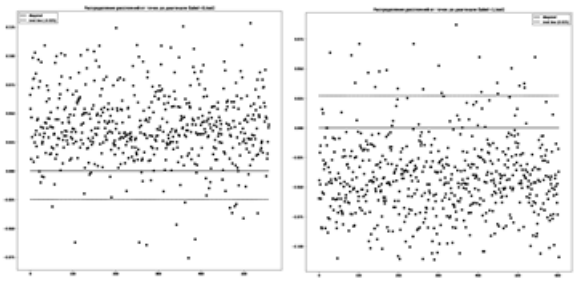


Рис. 2. Распределение расстояний от образца до диагонали, класс 0 слева и класс 1 справа, test

3. Корректность выбранного значения epsilon проверяется путем построения гистограммы распределения неправильных образцов. Пунктирными линиями на графике (рис. 3) отмечены значения epsilon для каждого из классов (значения могут совпадать по модулю), линии отделяют промежуток с высокой плотностью неправильных результатов классификатора, что говорит о грамотном выборе числа epsilon.

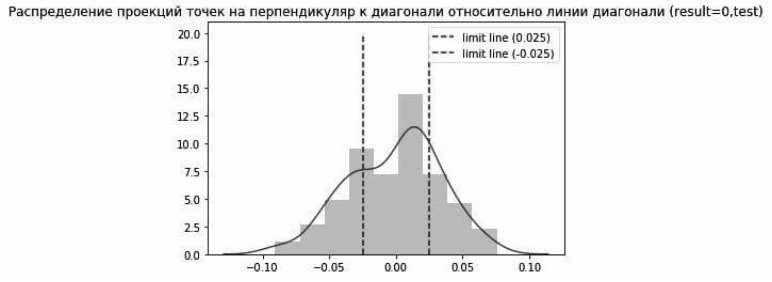


Рис. 3. Распределение проекций точек на перпендикуляр к диагонали относительно линии диагонали, неправильный результат, test

Вывод: для датасета могут быть найдены область, где классификатор допускает большое количество ошибок, и область, где определение класса заранее сводится к событию, не зависящему от расположения координаты на графике относительно центроидов.

Cnt мета-функция

Рассматривается вектор, полученный путем подсчета числа n_j соседей (среди k) для целевого вектора x, которые принадлежат классу c_j. Т.е. он показывает количество объектов из данного класса среди k ближайших соседей.

1. Существует некоторое число Ф соседей класса 1, такое, что при выполнении неравенства $H \geq \Phi$, где H – число соседей первого класса для данного экземпляра среди k, классификатор значительно улучшит точность поиска класса 1. Данное число Ф определяется на графике (рис. 4) правильно классифицированных экземпляров – это область резкого возрастания плотности, отделенная пунктирной линией для Ф.

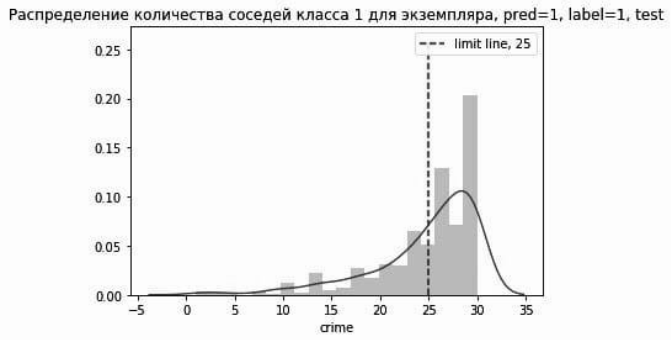


Рис. 4. Распределение числа соседей класса 1 при правильном предсказании криминальных новостей, test

2. Для улучшения качества классификации необходимо найти такое количество соседей, при котором будет достигнуто наибольшее количество верно классифицированных образцов при наименьшем количестве неверно классифицированных. Данное число получается путем построения количества правильно и неправильно предсказанных соседей нужного нам класса в зависимости от количества соседей в правой части неравенства на графике (рис. 5). Но в данном случае оценка слишком приблизительна, поэтому для дальнейшего анализа строятся левый и правый графики (рис. 6) резких скачков для правильно классифицированных объектов и для неправильно классифицированных объектов одного класса соответственно. Скачки получаются за счет разделения общего числа соседей на равные группы и рассмотрения разностных отношений уже для групп.

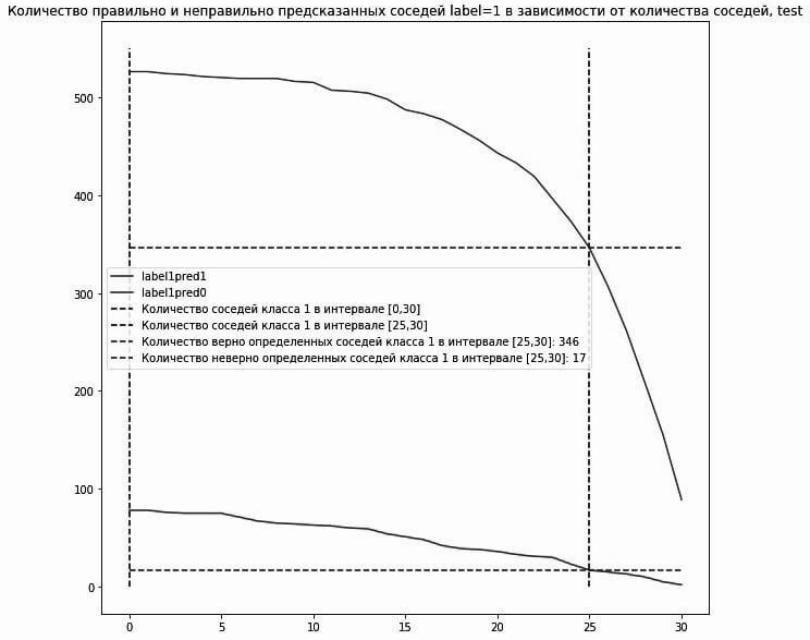


Рис. 5. Количество правильно и неправильно предсказанных соседей класса 1
в зависимости от количества соседей, test

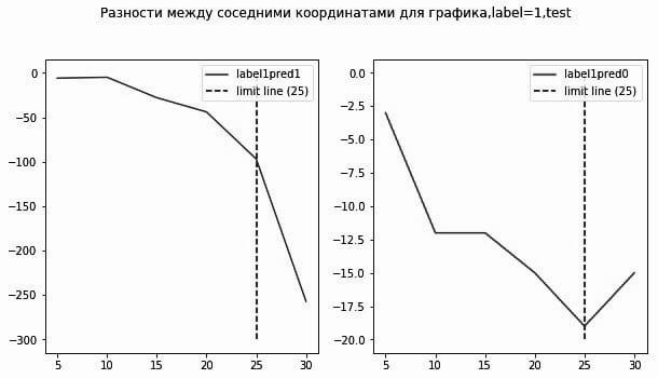


Рис. 6. Количество правильно и неправильно предсказанных соседей класса 1
в зависимости от количества соседей для групп, test

Вывод: с помощью данной мета-функции можно найти пороговое значение соседей нужного класса, результаты для которого заметно превосходят полученные для не превосходящего пороговое значений.

Ncnt мета-функция

Вектор, который показывает распределение классов вокруг объекта через сумму дистанций k ближайших объектов, которые принадлежат классу c_j, до данного x. Представляет информацию как о количественном распределении, так и о степени близости с объектами класса.

1. Если обозначить исходные суммы за A и B, а за A' и B' – нормированные с условием, что за B всегда берется большая сумма, то можно сделать следующие предположения: чем больше значение A' (при условии, что $B'=1$), тем ближе были расположены исходные классы, а для исходных расстояний (A,B) малая разность по модулю означает ситуацию, когда два класса невозможно разделить. Чем больше $B'-A'$ (при условии, что $B'=1$), тем дальше друг от друга находятся классы, т.е. тем больше вероятность, что объект X будет определен однозначно. Однако, существует "случай с выбросами", когда сумма (не равная 1) получается достаточно большой за счет одиночных выбросов, входящих в число соседей.

2. При графическом анализе необходимо выделить промежуток с повышенной плотностью ошибок. Таким образом, при исключении данного интервала доля верных ответов возрастает, что тоже необходимо проверить графически. Границы промежутка не являются точными.

Вывод: с помощью данной мета-функции можно определить промежуток, исключив который, можно значительно улучшить качество классификации.

Qrt мета-функция

Вектор получается путем рассмотрения пяти величин, которые характеризуют распределение k соседей x относительно соседей из класса c_j. Величины: наименьшее и наибольшее расстояния до объекта, первый, второй и третий квартили. Отражает плотность концентрации наблюдений класса возле нового объекта. Полезен для определения областей с низкой плотностью объектов из заданного класса.

1. Для разности между минимальными расстояниями 1 и 0 классов необходимо найти значение с наименьшей абсциссой, но так, чтобы слева от этой линии оставалось некоторая часть выборки (не только выбросы).

На левом графике (рис. 7) представлено пересечение двух распределений: левее для первого класса, правее для нулевого класса, и можно заметить, что минимальное расстояние до объектов первого класса несколько меньше, чем до нулевого. На правом графике – дельты для двух классов соответствующих образцов. При смещении пунк-

тирной линии на правом графике влево будет увеличиваться по модулю возможная разность, оставляя ту область, где результат классификатора более точный.

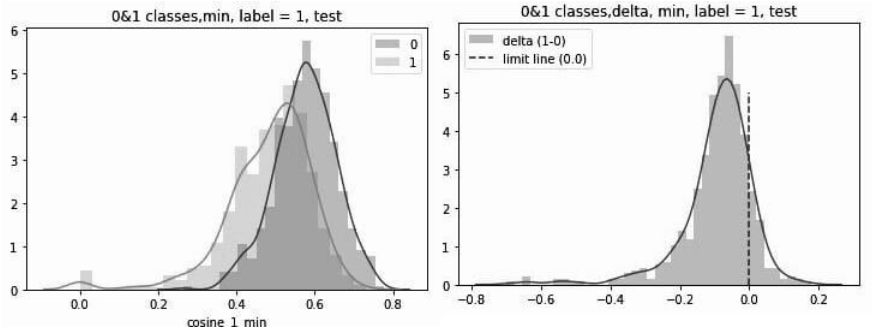


Рис. 7. Распределение минимальных расстояний от объекта для первого и нулевого классов и их разности, test

2. Для разности между максимальным и минимальным расстоянием для 1 класса необходимо найти значение с наибольшей абсциссой, но так, чтобы справа от этой линии оставалось некоторая часть выборки (не только выбросы).

На левом графике (рис. 8) представлено пересечение двух распределений: левее для минимумов класса 1, а правее – для максимумов, и стоит отметить, что чем меньше расстояние между ними, тем ближе друг к другу находятся образцы, они как бы сжаты вдоль оси расстояний. На правом графике – дельты для минимума и максимума класса 1 соответствующих образцов. С помощью перемещения пунктирной линии вправо определяется значение разности, не превышающей некоторое положительное число, при котором точность классификатора как для первого класса, так и точность в целом получается выше.

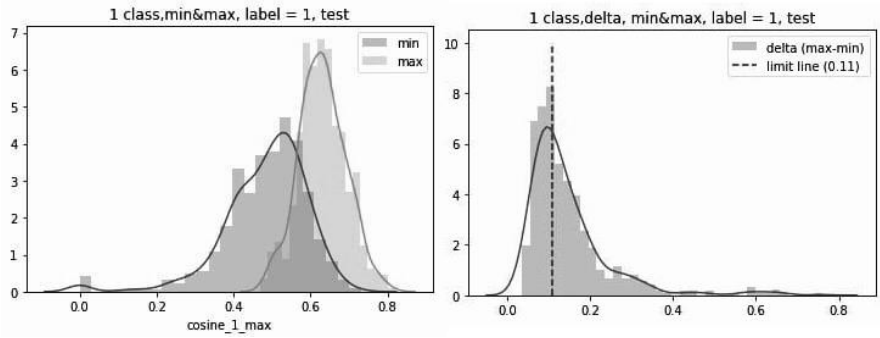


Рис. 8. Распределение разности минимальных и максимальных расстояний от объекта для первого класса и их разности, test

Вывод: данная мета-функция показывает плотность концентрации наблюдений класса возле нового объекта, позволяет оценить распределение соседей относительно друг друга.

Sum_cent мета-функция

Вектор содержит среднее сходство между центроидой класса c_j и всеми x^* из множества соседей x, принадлежащих к категории c_j. Предоставляет нам информацию о степени неоднородности пространства в подобласти образца. Полезен для определения наличия в области аномально расположенных образцов.

1. Если представить некоторую область (для примера, окружность) объектов класса c_j, сосредоточенных вокруг центроиды этого класса, тогда радиусом этой окружности будет значение среднего расстояния от объектов до соответствующей центроиды. Если данный радиус мал, то рассмотрение данной мета-функции можно свести к рассмотрению Centroid функции.

2. В малой своей окрестности центроиды окружены экземплярами обоих классов практически в равных долях.

3. Для правильно определенных криминальных новостей характерно заметно преобладание образцов данного класса вокруг центроиды первого класса, более чем в два раза превышающее количество образцов вокруг объектов класса 0. Принимая во внимание результаты для класса 1 (без разделения на корректные и нет), можно предположить, что для объекта, имеющего перевес по количеству объектов заданного класса более чем в два раза в малой области центроиды этого класса, классификатор верно определит класс.

4. Статистическое обоснование экспериментов

Одним из важнейших вопросов при использовании предложенных в п. 3 мета-функций является их применимость в одинаковой степени как к тренировочной выборке, так и к тестовой. Таким образом, возникает первое требование эквивалентности test и train выборки.

Для правильного выбора параметрических или непараметрических критериев для дальнейшего анализа необходимо проверить данные на принадлежность нормальному закону распределения. Поэтому первым этапом была произведена проверка на отклонение от нормального закона для данных, выбранных с использованием мета-функции Centroid. Было использовано два подхода: а. проекции экземпляров на соответствующие оси; б. проекции экземпляров на перпендикуляр к диагонали. Но т.к. оба подхода дали одинаковый результат, здесь будет описан только первый.

Для дальнейших экспериментов будет использовано значение уровня значимости $\alpha = 0.05$. По критерию Шапиро-Уилка были получены значения *p*-value для класса 1 равное $0.012 < 0.05$ и для класса 0 равное $0.08 > 0.05$. По критерию Андерсона-Дарлинга было получено значение статистики, для класса 1 равное $1.003 > 0.782$ и для класса 0 равное $0.421 < 0.782$, где 0.782 – критическое значение статистики для заданного уровня значимости α.

Вывод: согласно критериям Шапиро-Уилка и Андерсона-Дарлинга гипотеза о нормальности распределения отклоняется для класса 1 и не отклоняется для класса 0.

Для проверки эквивалентности выборок сформулируем нулевую гипотезу H_0: $M(V_{\text{test}}) = M(V_{\text{train}})$, где V_{train} и V_{test} – это выборки, построенные с помощью мета-функции Centroid.

Для нулевого класса данные выборки не противоречат гипотезе о нормальности на заданном уровне значимости, поэтому для проверки гипотезы H_0 можно применить критерий Стьюдента. Тогда *p*-value будет равно $0.87 > 0.05$ и на уровне значимости α по критерию Стьюдента гипотеза H_0 для нулевого класса не отклоняется.

Поскольку для первого класса гипотеза о нормальности была отклонена, то для проверки вышеизложенной гипотезы H_0 можно применить непараметрический критерий Манна-Уитни. Значение *p*-value будет равно $0.376 > 0.05$, следовательно, гипотеза H_0 по критерию Манна-Уитни на уровне значимости α для первого класса не отклоняется.

Таким образом, можно сделать вывод: рассмотренные для train выборки результаты можно считать применимыми для test выборки, и наоборот.

Второй вопрос касается возможности разделить нулевой и первый классы, что является важным условием адекватной работы классификатора. С помощью мета-функции Centroid проверим данные обоих классов на различие между ними. На рис. 9 представлен график распределения нулевого и первого классов, имеющих достаточно малую площадь пересечения, т.е. данные в целом разделены. В обоих случаях заметна левосторонняя асимметрия. Нормальность по критерию Лиллиефорса для заданного уровня значимости $\alpha = 0.05$ не подтверждается: *p*-value для нулевого класса –

$1.67 \times e^{-19}$, для первого – $1.34 \times e^{-21}$, что в обоих случаях много меньше 0.05. Следовательно, можно использовать только непараметрические критерии. По критерию Манна-Уитни выявляется сильная несогласованность гипотезы о равенстве значений параметров в выборках (значение *p*-value близко к нулю). Таким образом, получаем, что данные двух классов имеют сильное разделение и их можно использовать для классификатора.

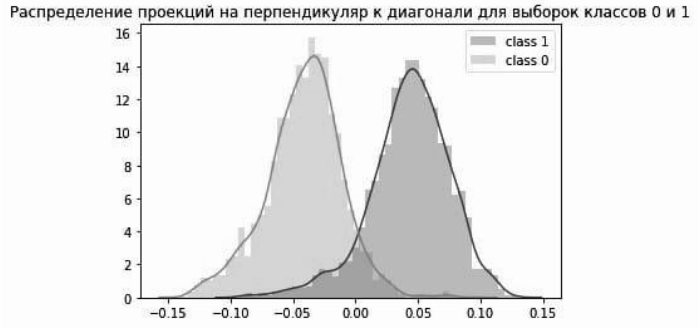


Рис. 9. Распределение проекций на перпендикуляр к диагонали для train и test выборок классов 0 и 1

Заключение

В рамках данной работы была проведена реализация некоторых мета-функций из статьи Кануто [9] в виде кода на языке программирования Python. Было реализовано построение бинарного классификатора текстов и обучение его на открытом датасете новостей.

Был проведен графический и аналитический анализ мета-функций для используемого классификатора на исходном наборе данных, в результате чего были выдвинуты предположения о дальнейшем применении использованных мета-функций и областях, в которых стоит ожидать ошибку или правильный результат работы классификатора.

Предварительные результаты применения мета-функций (без sum-cent мета-функции): точность (метрика accuracy) для train выборки увеличилась с 0.903 до 0.965, для test выборки точность увеличилась с 0.88 до 0.94. Тренировочная выборка сократилась в 3.47 раза, а тестовая – в 3.78 раза.

С помощью статистического анализа было продемонстрировано, что тестовая и тренировочная выборки схожи между собой, и возможно применение полученных для одного типа выборок результатов мета-функций к другому типу выборки.

Также предлагается два варианта дальнейшей работы:

1. Вариант "внедрения" мета-пространства вместо исходного пространства векторизованных текстов новостей.
2. Вариант создания "надмодели", которая работает извне и на основе полученных характеристик модели будет выдавать свой вердикт о том, является ли предсказание самого классификатора верным.

ЛИТЕРАТУРА

1. *Thrun S.* Lifelong Learning Algorithms. Learning to Learn 8, 1998, c.181-209

2. *Vilalta R., Drissi Y.* A Perspective View And Survey Of Meta-Learning. Artificial Intelligence Review 18(2), 2002, P.75-95

3. *Utgoff P.* Shift of Bias for Inductive Concept Learning. Machine Learning: An Artificial Intelligence Approach, 1986, P. 107-148.

4. *Gordon D.* Queries for Bias Testing. Proceedings of the Workshop on Change of Representation and Problem Reformulation, 1992, P. 53-65.

5. *Aha D.* Generalizing from Case Studies: A Case Study. Proceedings of the Ninth International Workshop on Machine Learning, 1992, P. 1-10.

6. *Gama J., Brazdil P.* Characterization of Classification Algorithms. Proceedings of the seventh Portuguese Conference on Artificial Intelligence, 1995, P. 189-200.

7. *Yang Y.* Information Retrieval. Proceedings of the seventh Portuguese Conference on Artificial Intelligence, 1999, P. 69-90.
8. *Sebastiani F.* Machine Learning in Automated Text Categorization. ACM Computing Surveys, 2002, P. 1-47.
9. *Canuto S.* On Efficient Meta-Level Features for Effective Text Classification. CIKM Conference Paper, 2014, P. 1709-1718.

ФОРМИРОВАНИЕ ЦИФРОВОГО ДВОЙНИКА ГОРОДА ПРИ УЧАСТИИ ГОРОЖАН НА ПРИМЕРЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ВИДЕОНАБЛЮДЕНИЯ

Седун Д.А., Гончарова Н.А.
Томский политехнический университет
tsksedoy@gmail.com, natg@tpu.ru

Введение

Статья дает понимание понятия цифрового двойника, опираясь на различные источники, как с точки зрения объекта, так и сточки зрения процесса. Осуществляется попытка определения цифрового двойника города, как сложнейшего комплекса данных. Рассматривается система интеллектуального видеонаблюдения, с определением характерных черт, как части цифрового двойника города. Освещается практика участия горожан в формировании систем интеллектуального видеонаблюдения. Оцениваются возможные синергетические эффекты такого участия.

1. Определение понятия «цифровой двойник города»

Промышленность и научные круги определяют понятие цифрового двойника несколькими различными способами. Однако, возможно, ни одна из этих групп не уделяет должного внимания аспектам цифрового двойника, как процесса. К примеру, в некоторых источниках, цифровой двойник представляет собой интегрированную модель готового продукта, которая призвана отражать все производственные дефекты и постоянно обновляться, чтобы демонстрировать износ в течение всего срока эксплуатации [1].В других широко распространенных определениях цифровой двойник описывается как цифровая модель физического объекта с поддержкой датчиков, которая имитирует объект в режиме реального времени [2].

По сути, цифровой двойник можно определить как развивающийся цифровой профиль исторического и текущего поведения физического объекта или процесса, который помогает оптимизировать эффективность бизнеса. Цифровой двойник основан на массивных, кумулятивных измерениях данных в реальном времени в реальном масштабе по множеству измерений. Эти измерения могут создать развивающийся профиль объекта или процесса в цифровом мире, который может дать важную информацию о производительности системы, что приведет к действиям в физическом мире, таким как изменение дизайна продукта или производственного процесса.

Реальная сила цифрового двойника, а также важность его значения заключается в том, что он может обеспечить почти полную связь в реальном времени между физическим и цифровым мирами. Вероятно, из-за этой интерактивности между реальным и цифровым миром продукта или процесса, цифровые двойники могут обещать более богатые модели, которые дают более реалистичные и целостные измерения непредсказуемости. А благодаря более дешевым и более мощным вычислительным возможностям эти интерактивные измерения можно анализировать с помощью современных архитектур массивной обработки и усовершенствованных алгоритмов для получения прогнозирующей обратной связи в реальном времени и выполнения автономного анализа. Это может привести к фундаментальным изменениям в проектировании, изменениям технологических процессов, которые почти наверняка будут недостижимы с помощью существующих методов [3].

Город – это крупный населённый пункт, жители которого заняты, как правило, не сельским хозяйством. Имеет развитый комплекс хозяйства и экономики, является скоплением архитектурных и инженерных сооружений, обеспечивающих жизнеобеспечение населения [4].

Исходя из вышесказанного, можно дать следующее определение понятию «цифровой двойник города»: Цифровой двойник города – это сложнейший комплекс данных характеризующих множество показателей всех сфер города в их историческом развитии, актуальном состоянии, взаимодействии этих показателей между собой и моделировании перспективы изменения показателей для получения качественно новых стратегий и направлений развития сфер города не доступных для получения традиционными способами проектирования.

2. Термин «интеллектуальные системы видеонаблюдения»

Термин «Интеллектуальные системы видеонаблюдения» (ИСВН) стал употребляться к системам видеонаблюдения оснащенным дополнительным функционалом. Функционал этот подразумевает аналитику изображения фиксируемого камерой и выполнение алгоритмов связанных с такой аналитикой. В РФ можно четко определить природу внедрения функций аналитики в системах видеонаблюдения. Требования к обязательному оснащению аналитическими функциями системы видеонаблюдения прописаны в федеральных законах и подзаконных актах регламентирующих системы видеонаблюдения на некоторых объектах различного функционального назначения. Среди таких объектов можно выделить:

• Объекты транспортной инфраструктуры (16-ФЗ «О транспортной безопасности», Постановление Правительства РФ от 28 июля 2018 г. N 886; Постановление Правительства РФ от 26 апреля 2017 г. N 495, Постановление Правительства РФ от 5 апреля 2017 г. N 410, Постановление Правительства РФ от 14 сентября 2016 г. N 924, Постановление Правительства РФ от 16 июля 2016 г. N 678);

• Места массового пребывания людей (Федеральный Закон от 06.03.2006 35-ФЗ «О противодействии терроризму», Постановление Правительства РФ от 25 марта 2015 г. №272);

• Образовательные учреждения(СП 251.1325800.2016 Здания общеобразовательных организаций. Правила проектирования (с Изменением N 1));

• Промышленные предприятия топливно-энергетического комплекса и атомной промышленности (Атомная промышленность — ФЗ от 21 ноября 1995 г. N 170 Предприятия топливно-энергетического комплекса — ФЗ от 21.07.2011 N 256;);

• Опасные производства (ФЗ от 21 июля 1997 г. N 116).

Строго говоря, Постановление Правительства РФ от 26.09.2016 № 969:п. 32. Четко определяет, что «к техническим системам и средствам интеллектуального видеонаблюдения относятся:

а) технические системы и средства идентификации физических лиц;

б) технические системы и средства обнаружения тревожных ситуаций».

Стоит отметить, что аналитические функции производителями представлены в различных вариантах исполнения. Можно классифицировать разнообразие программного обеспечения (ПО) по различиям аппаратной платформы [5]. В табл. 1 приведены характерные для такой классификации примеры.

Таблица 1

Виды ПО с аналитическими функциями по различиям используемой аппаратной платформы

Описание	Преимущества	Недостатки
ПО и аналитика выполняется прямо на камере.	Стоимость ПО уже заложена в камеру; Экономия трафика; Простота настройки и эксплуатации	Низкая интеграция с другими системами; функционал ограниченный производителем;
ПО и аналитика выполняется на спе-	Меньшая стоимость по сравнению с	Зачастую жесткие требования к коли-

Описание	Преимущества	Недостатки
циальном устройстве или видеорегистраторе	серверным вариантом, Простота настройки и эксплуатации. Клиентское ПО от производителя идеально принимает, обрабатывает и выводит данные об аналитике	честву и производителю оборудования внутри одной системы; функционал ограниченный производителем. Сложная интеграция с другими системами
ПО и аналитика выполняется на удаленных облачных сервисах	Для работы подойдут практически любые камеры, нужен только видеопоток. Высокая масштабируемость. Постоянно совершенствующиеся алгоритмы аналитики.	Высокие требования к пропускной способности канала; регулярные платежи, которые, в конечном счете, могут стать существенной статьей издержек; не большой выбор среди вендоров и типов аналитики; сложность в настройке и интеграции с другими системами.
ПО и аналитика выполняется на серверах	Практически любые камеры. Низкая нагрузка на камеры и видеорегистраторы.	Высокая стоимость; сложность настройки и интеграции.

Представленные на рынке решения не только выполняют требования прописанные в вышеуказанных нормативных документах, но и предлагают ряд других аналитических функций. Результатом такого функционала ИСВН становятся массивы данных пригодных для дальнейшей передачи, приема и обработки другими системами. Такие данные при достаточном масштабировании пригодны для использования, как элементы цифрового двойника города.

3. Участие горожан

При формировании городской среды (ГС) нельзя не упомянуть о сфере безопасности и комфорта. Системы видеонаблюдения являются неотъемлемой составляющей такой сферы городской среды. Практика участия горожан в формировании городской среды описана зарубежными и отечественными исследователями. В своих работах зарубежные и отечественные исследователи и практики обосновывают безусловную значимость участия граждан в развитии ГС, предлагают конкретные инструменты для обеспечения этого участия, а также раскрывают обусловленность самого городского сообщества тем пространством, в котором оно живёт и удовлетворяет свои потребности [6].

Приведем пример такого участия в модернизации действующей системы видеонаблюдения горожанами. На рис. 1 проиллюстрирован ситуационный план системы цифрового видеонаблюдения на одном из многоквартирных домов (МКД) г.Томска. Используя инструменты по обеспечению участия жителей в обсуждении модернизации системы видеонаблюдения такой системы, был выявлен ряд потребностей, который должен существенно улучшить процесс эксплуатации и обслуживания:

1) в зону обзора не будут попадать автоматические ворота, которые в скором времени планируется ввести в эксплуатацию, это нужно учесть;
2) наличие «слепых зон» в придомовой территории;
3) необходимость вести видеосъемку входов в подъезды;
4) необходимость вести видеосъемку в большом и малом пассажирских лифтах каждого подъезда;
5) необходимость более детально вести видеосъемку для предоставления информации силовым структурам в случае противоправных действий или ДТП.

Ряд мер был сформирован не единовременно, многократно обсуждался с собственниками помещений и лицами в них проживающими. При этом работа с горожанами производилось несколькими способами, чтобы охватить максимально большую по возможности целевую аудиторию. Использовались личные опросы, анкетирование, проведение голосований через социальные сети и мессенджеры.

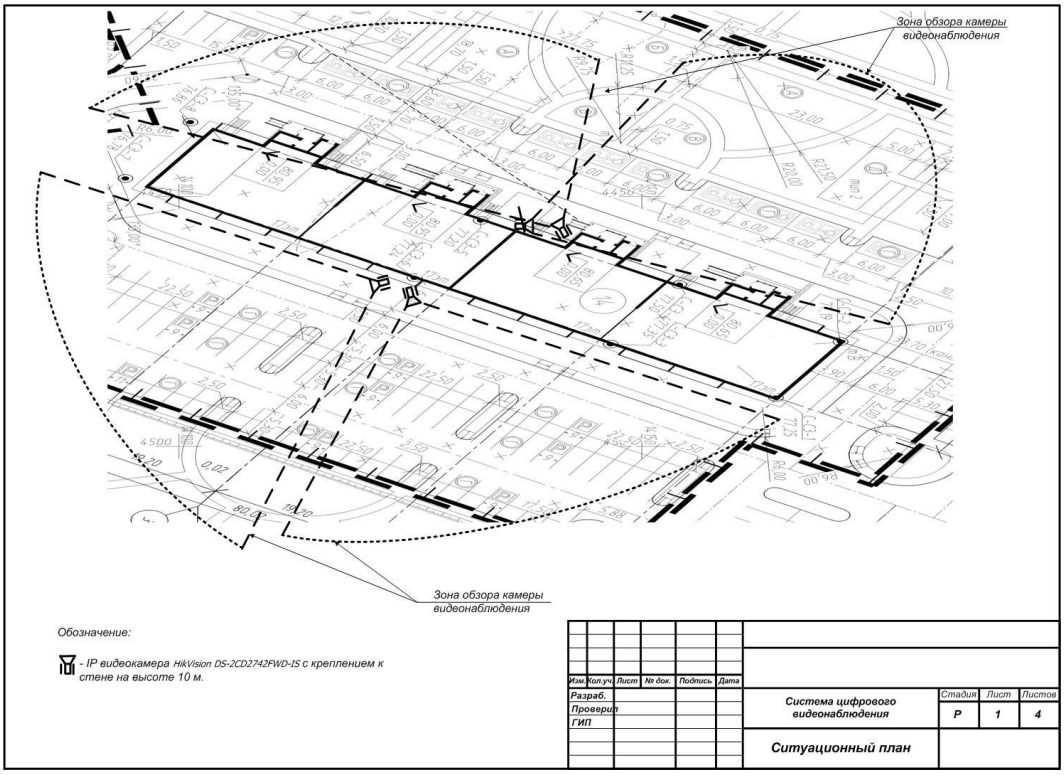


Рис. 1. Ситуационный план с изначальной конфигурацией системы видеонаблюдения на объекте

Рис. 2 иллюстрирует конечный вариант модернизированной системы, которая получилась в результате участия горожан. Стоит обратить внимание, что для выполнения пятого пункта в ряде требований к модернизации было предусмотрено решение на основе аналитики. Система дооборудована купольными камерами с возможностью ужаленного управления (PTZ) и многократным оптическим зумом. Обслуживающей организацией был установлен сервер с программным обеспечением, выполняющим аналитику видеопотоков со стационарных обзорных камер. ПО выполняет функции «трекинга» или выражаясь иначе функции определения потенциально важных движущихся на видеопотоке обзорных камер объектов и удаленного управления PTZ камеры с отображением уже на ее видеопотоке крупным планом этих объектов. Такое решение позволяет добиться высокой детализации изображения с значительно меньшими затратами.

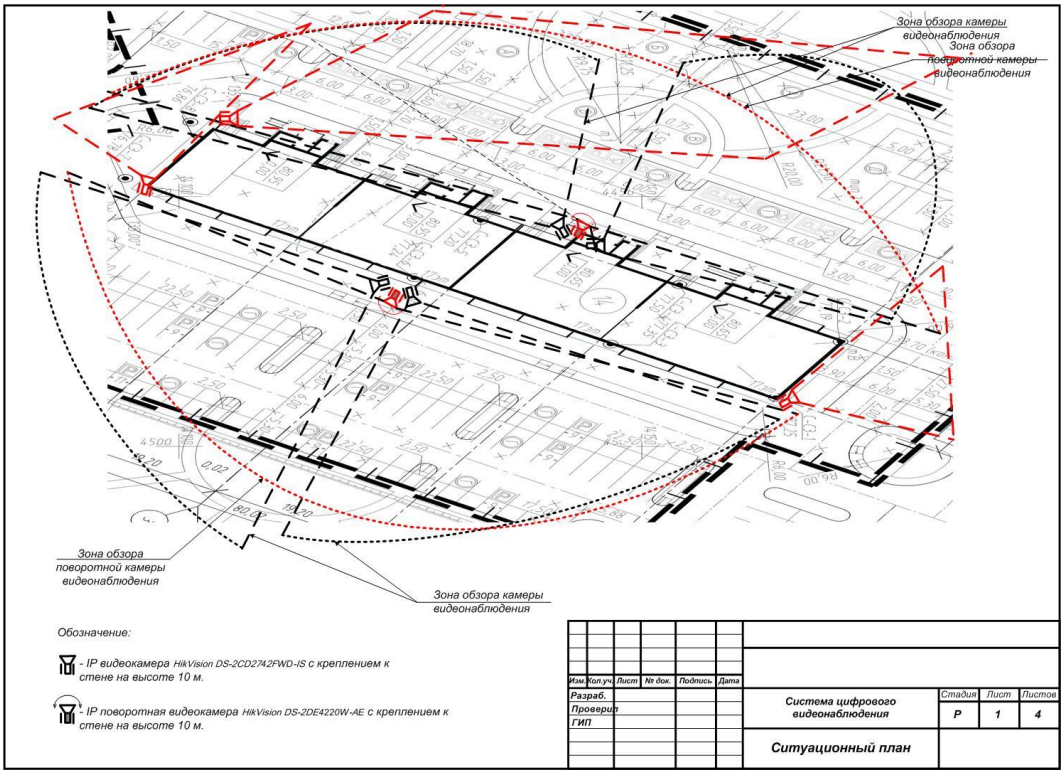


Рис. 2. Ситуационный план с окончательной конфигурацией системы видеонаблюдения на объекте

Приведенный пример лишь один из множества реализованных за последние годы в г. Томске. Способы взаимодействия с горожанами, а также внедрение ИСВН при проведении таких работ существенно повышает безопасность и комфорт городской среды.

4. Синергетический эффект выбранных решений

Следующие неочевидные дополнительные положительные эффекты можно увидеть при реализации таких мероприятий:

– Аналитика в ИСВН создает дополнительные массивы данных пригодных для анализа и внедрения в цифровой копии города. Это могут быть данные видеосемантики, данные о тревогах, данные об идентификации и распознавании и т.д.
– Горожане получают не только практический опыт участия по улучшению городской среды, но и более социализированную среду их обитания.
– Организация, практикующая такой подход, получает большее количество заявок, увеличивает число заключенных договоров, а как следствие прибыли, растет популярность ее услуг, бренд организации становится более узнаваемым.

Заключение

Процесс формирования цифрового двойника города представляется процессом сложным, требующим множества ресурсов, технологий и данных всех сфер города. Тем не менее, использование методов привлечения к формированию городской среды граждан и участие в таком формировании квалифицированных специалистов позволяет создавать дополнительные полезные массивы данных. Чем больше информации о различных сферах города будет обрабатываться в цифровом двойнике, тем больше он будет соответствовать реальным объекта и процессам в городе, и тем точнее можно будет делать прогнозы и проектировать дальнейшее развитие.

ЛИТЕРАТУРА

1. *Jack Reid and Donna Rhodes*, Digital system models: An investigation of the non-technical challenges and research needs, Conference on Systems Engineering Research, Systems Engineering Advancement Research Initiative, Massachusetts Institute of Technology, 2016. http://seari.mit.edu/documents/preprints/REID_CSER16.pdf, C. 2

2. *Michael Grieves*, Digital twin: Manufacturing excellence through virtual factory replication, 2014, https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication,. – С. 1.

3. *Aaron Parrott, Lane Warshaw.* Industry 4.0 and the digital twin technology Deloitte Insigts (12-05-2017). https://www2.deloitte.com/us/en/insights/focus/industry-4-0/digital-twin-technology-smart-factory.html#endnote-3 С. 5–9.

4. *Моргунов Е.В., Мамаев С.М.* Развитие городов через призму качества жизни населения // Вестник Томского государственного университета. Экономика. – 2017. – № 38. – 1 с.

5. Интеллектуальная видеоаналитика . https://securityrussia.com/blog/videoanalitika.html.

6. *Деменев А.Г., Шубина Т.Ф., Шубина П.В., Ненашева М.В., Макулин А.В., Тарасов И.А.* Опыт общественного участия в планировании комфортной городской среды на примере Архангельской области // Арктика и Север. 2018. № 33. С. 91-117. DOI: 10.17238/issn2221-2698.2018.33.91.

РАСПОЗНАВАНИЕ НОТ В ВОКАЛЬНОМ ИСПОЛНЕНИИ С РЕЗКИМ ИЗМЕНЕНИЕМ ЧАСТОТ ОСНОВНОГО ТОНА

Якимук А.Ю., Головчинер М.Н.
Томский государственный университет
yay@keva.tusur.ru, golovchiner@mail.ru

Введение

Применение программных средств при обучении распространено во многих областях знаний. Однако в сфере обучения вокальному мастерству существующие программы не всегда точно определяют спетую исполнителем ноту. Преподаватель, осуществляя занятия в классе, поочередно осуществляет прослушивание каждого из учеников и дает рекомендации. При таком подходе на начальном этапе обучение проходит длительный период вырабатывания музыкального слуха. Эффект биологической обратной связи полезен в сфере обучения музыкантов, что подводит к необходимости разработки системы распознавания нот с высокой точностью [1,2]. Существующие коммерческие программы, реализующие функцию обучения пению, отличаются низкой точностью распознавания спетых исполнителем нот. Кроме того, описанный выше подход требует регулярного посещения занятий учениками в музыкальной школе.

В данной работе рассматривается подход к распознаванию нот, спетых с вибрато или глиссандо и не воспринимаемых алгоритмом распознавания нот, настроенным на работу с чисто исполненными нотами.

1. Постановка задачи

Разработанный программный комплекс [1] включает в себя алгоритм распознавания нот. Принцип работы алгоритма распознавания нот в вокальном исполнении, применяемый в программном комплексе, заключается в следующем. Для каждого момента времени определено значение частоты основного тона, поэтому возможно определить, какая нота «прозвучала» в данный отсчет. Определение частоты основного тона осуществляется по алгоритму, основанному на модификации математической модели слуховой системы человека, представленной в [2]. Предварительное тестирование программного комплекса показало, что повышение качества идентификации нот возможно при расширении диапазона охватываемых частот и определении ситуаций, снижающих точность работы алгоритма.

Проведенные испытания алгоритма показали, что при обработке аудиозаписей, включающих различные подходы к исполнению нот в диапазоне от 70 до 800 Гц, программный комплекс позволяет распознать более 95% спетых диктором нот. Было определено, что при пении арпеджио, крещендо и декрещендо отсутствует влияние на качество работы алгоритмов в программном комплексе. Отсутствие влияния на качество

идентификации нот заключается в том, что в алгоритме при анализе учитывается только частота основного тона. Спецификой работы одного из этапов алгоритма идентификации нот, отвечающего за определение принадлежности вокализованного участка к ноте, заключается в ориентированности на чистое исполнение.

Исполнить ноту так, чтобы все частоты в момент пения находились в пределах полосы, отведенной под нее, достаточно сложно. По этой причине учитываются соседние ноты выше и ниже исполняемой. На рис. 1–3 спектр частот разделен на участки, соответствующие границам нот. Каждый участок выделен горизонтальной полосой на фоне графика частот. По оси *X* – время, по оси *Y* – частоты основного тона.

Как можно видеть на рис. 1, сегмент, спетый в пределах одной ноты, был распознан программой. Количество моментов, попавших в соседние ноты, незначительно и не оказало влияние на результат идентификации. Преобладающее число распознанных частот основного тона оказалось в диапазоне спетой ноты. С учетом прописанных в алгоритме требований по точности исполнения, программа смогла сделать вывод о спетой ноте. В случае, если количество фрагментов в каждом из 3-х участков оказалось приблизительно одинаковым, алгоритм воспринимает весь вокализованный сегмент как шум.

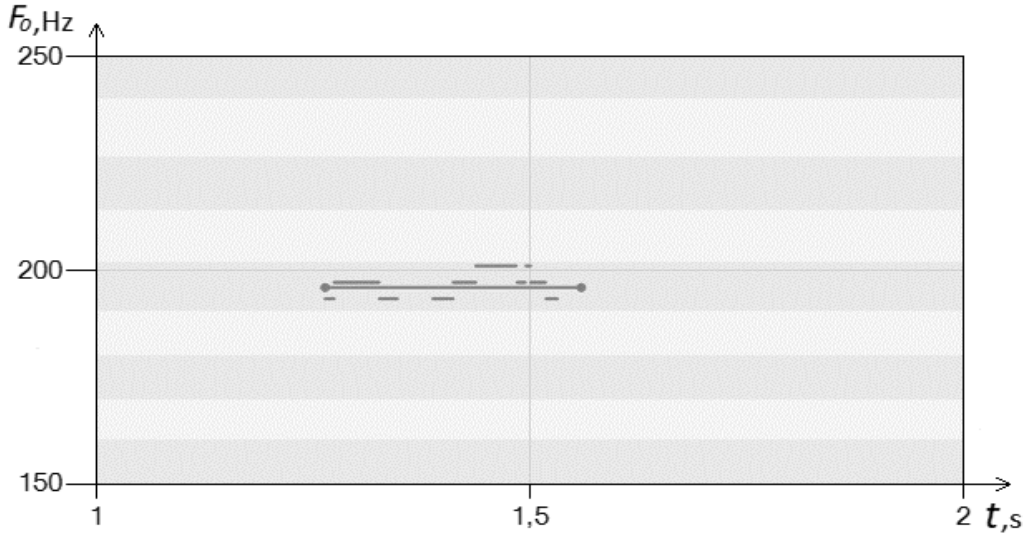


Рис. 1. Сегмент с пением в пределах одной ноты

Было определено, что ошибки возникают для ситуаций с подходами, подразумевающими изменение значения частоты звучания [3] в процессе исполнения: вибрато (рис. 2) и глиссандо (рис. 3).

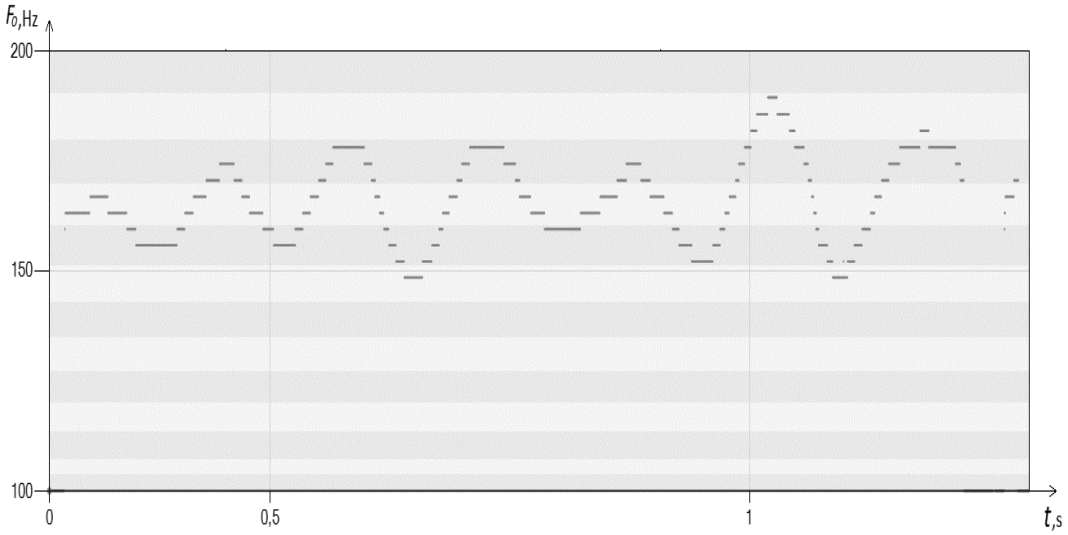


Рис. 2. Сегмент с нотой, спетой с вибрато в голосе

При пении вибрато происходят колебания в частоте звучания относительно какой-то главной ноты в пределах полутона. Следует отметить, что существуют разновидности вибратоподобных колебаний. Сюда относят тремоляцию и «качание голоса». В связи с тем, что алгоритм настроен на определение чистых нот, подобные сегменты воспринимаются как шум. Как можно заметить, колебания во время исполнения основной ноты (в данном случае – нота «ми малой октавы») охватывают сразу 4 ноты. Учет минимальной длительности звучания ноты приводит к тому, что для каждой из 4-х нот недостаточное количество времени происходило пение в ее границах. При этом возврат к этой ноте происходит после перехода к другим нотам, что снижает долю частот, относящихся к главной ноте.

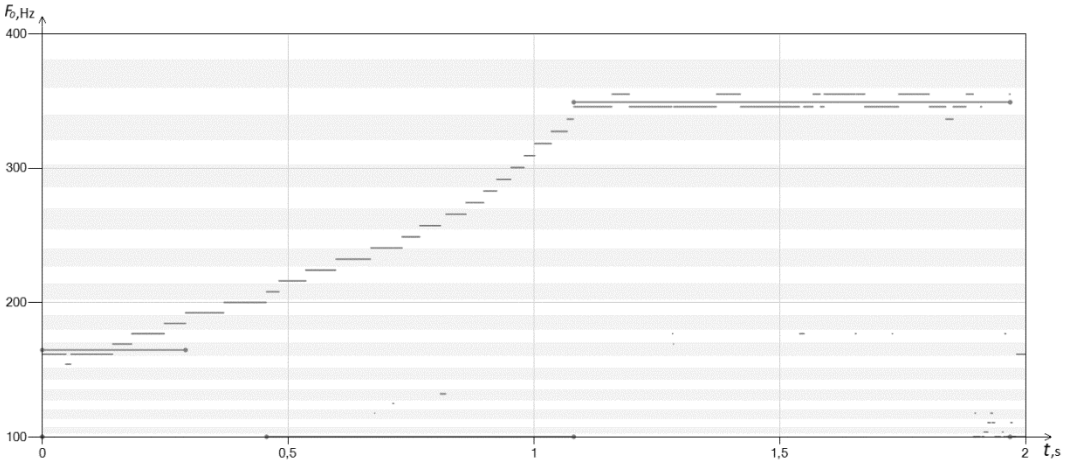


Рис. 3. Сегмент, спетый с восходящим глиссандо

Как известно из теории музыки, при пении глиссандо происходит плавное скольжение от одной ноты к другой. При таком пении переход происходит слишком быстро для того, чтобы суметь идентифицировать каждую отдельную ноту в момент скольжения, поскольку на каждый сегмент из охваченных нот приходится отводится менее 0.1 секунды. Глиссандо бывает, как восходящим (как представлено на рис. 3), так и нисходящим. Как можно заметить на рисунке, алгоритм способен определить начальную и

конечную ноты, между которыми осуществляется скольжение. Однако охват 12-ти нот в промежутке между ними воспринимается как шум.

Таким образом, возникает необходимость в идентификации участков, воспринимаемых программой как шум. Будем считать, что направленный переход от одной ноты к другой – это глиссандо, а колебания относительно одной ноты в рамках единого сегмента – это тремоляция. На данном этапе не будет рассматриваться определение типа тремоляции.

2. Анализ исполнения нот с резким изменением частот основного тона

Поскольку массив найденных частот основного тона, в рамках которого не удалось идентифицировать чистую ноту, можно воспринимать как временной ряд, было принято решение провести предварительный анализ данных. Среди процедур предварительного анализа выделяют: поиск аномалий, сглаживание временных рядов, проверку наличия тренда и расчет показателей динамики процессов.

С точки зрения рассматриваемой предметной области аномальными наблюдениями можно воспринимать всплески частот, определяемыми за пределами звучания основной мелодии. Сюда относятся любые шумы, отсеиваемые алгоритмами на этапе сегментации нот. Соответственно, данную процедуру можно считать выполненной.

В рамках сглаживания временного ряда происходит вычисление истинных уровней ряда расчетными значениями, имеющими более гладкую динамику, чем исходные данные. Было решено сконцентрировать внимание на методах механического сглаживания, поскольку в исследуемой задаче необходимо оценивать каждый отдельный уровнять ряда с учетом фактических значений соседних с ним уровней. Метод взвешенной скользящей средней не применим для исследуемой задачи, поскольку в нем не может содержаться квадратичный или кубический тренд. Также нет необходимости в применении метода экспоненциального сглаживания, т.к. он используется в задачах прогнозирования развития процесса после исследуемой области. В связи с этим будет использован метод простой скользящей средней. В качестве количества наблюдений P, входящих в интервал сглаживания, было решено использовать величину минимальной длительности звучания ноты, применяемой в алгоритме сегментации и идентификации нот. Выбор обусловлен тем, что при пении вибрато и глиссандо длительность пения в рамках подобных участков существенно выше данного параметра, но при этом идентификация отдельных чистых нот не была осуществлена базовым алгоритмом. Недостатком метода является то, что первые и последние P наблюдений останутся несглаженными, чем можно пренебречь ввиду общей продолжительности оцениваемого участка.

Проверка наличия тренда по своей сути является проверкой гипотезы о неизменности среднего значения временного ряда. Для проверки ряд будет разбит на n частей, каждая из которых рассматривается как отдельная совокупность. Полученные средние для каждой совокупности будут сравнены. В случае, если средние значения возрастают или убывают, будем считать, что исследуемый отрезок может содержать глиссандо (восходящее или нисходящее соответственно).

3. Анализ вибратоподобных вокальных исполнений

Все вычисления были осуществлены в Microsoft Office Excel. Встроенный функционал содержит функцию «Анализ данных», в котором для выбранного ряда можно построить скользящее среднее. На рис. 4 представлен результат применения данного метода. В встроенном методе происходит анализ только предшествующих значений без учета следующих после оцениваемого, что позволяет получить огибающую, неинформативную в рамках проводимого исследования.

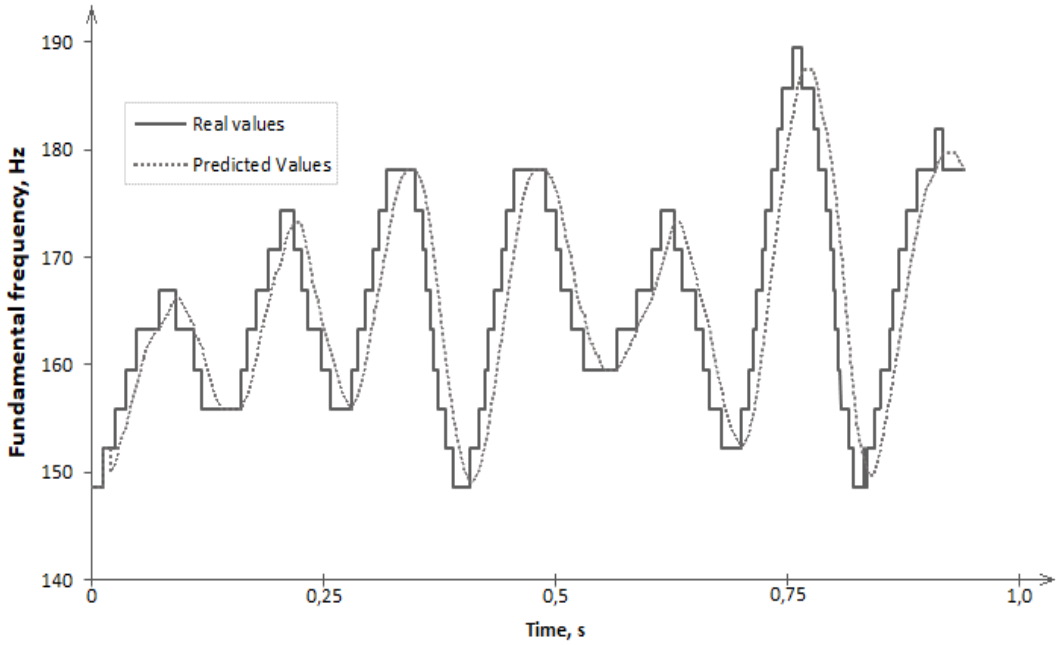


Рис. 4. Скользящее среднее для вибратоподобного участка в Excel

С целью решения данной проблемы была использована классическая формула скользящего сглаживания с учетом измерений до и после сглаживаемого, в интервал сглаживания были включены 0,5*P* значений, предшествующих сглаживаемому, и столько же последующих значений (рис. 5). В результате на отрезке от 0 до 0,5*P* значения рядов будут совпадать, но, уже начиная со следующего значения, мы сможем увидеть усреднение. Сплошной линией обозначены значения анализируемых частот основного тона, а прерывистой колеблющейся – сглаженные значения.

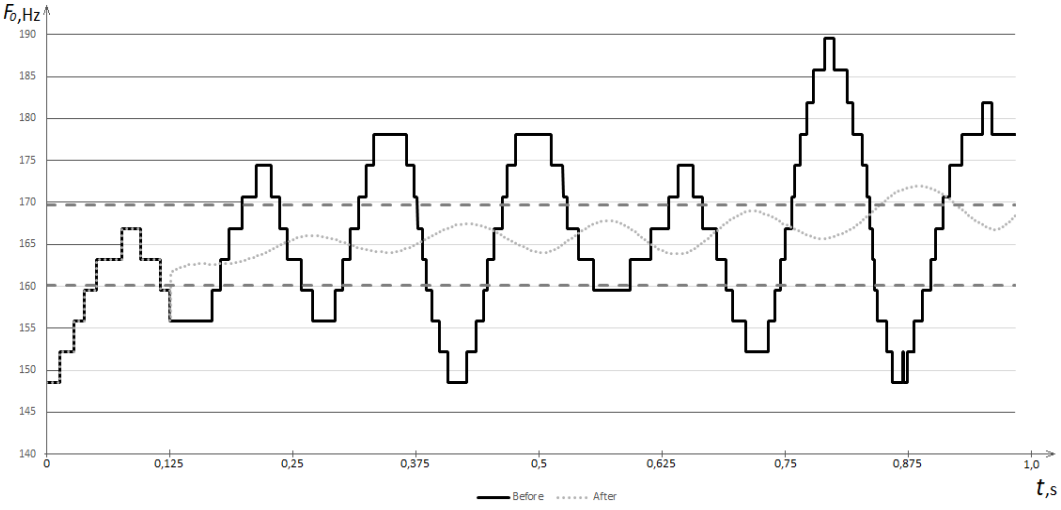


Рис. 5. Скользящее среднее для вибратоподобного участка

На представленном рисунке для ноты «ми малой октавы» параллельными пунктирными линиями указаны нижняя (160.121 Гц) и верхняя (169.643 Гц) границы звучания, определенные с помощью среднего логарифмического [4]. Отметим, что в исследуемом отрезке присутствует 7 колебаний в частоте основного тона, если судить по

графику. При дальнейшем исследовании типа колебаний в пении данный аспект может быть применен как критерий отличия вибрато от тремоляции.

Как было сказано ранее, наличие тренда определяется сравнением полученных средних значений для каждой из совокупностей исследуемого ряда. Также средствами Excel на графике частот основного тона для ряда можно задать построение линейного тренда (рис. 6). На этапе анализа применимости метода к исследуемой задачи будем использовать встроенную функцию в Excel, а на этапе реализации в программном комплексе – сравнение средних значений для отрезков исследуемого диапазона.

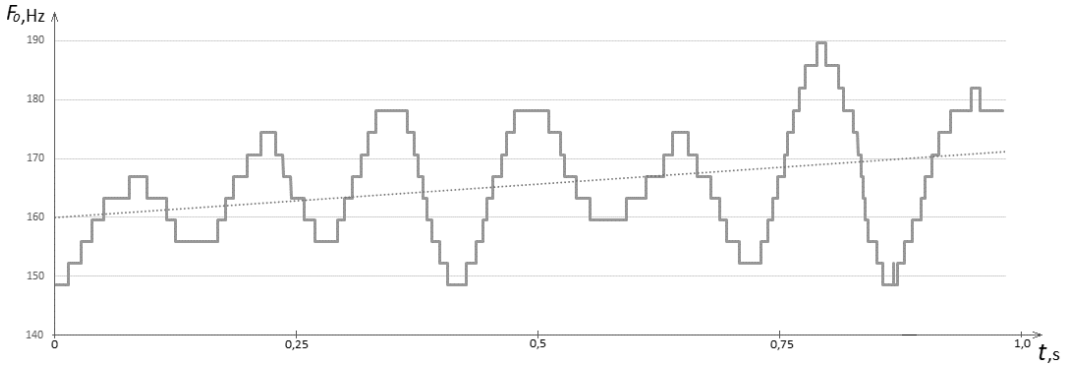


Рис. 6. Линейный тренд для вибратоподобного участка

Как можно заметить из рисунка, представленного выше, линейный тренд позволяет определить, что на большей части исследуемого участка средние значения частот относятся к одной ноте. Однако, по линейному тренду невозможно оценить наличие колебаний в частотах основного тона. В связи с этим использовать только линейный тренд для оценки участков с тремоляцией не представляется возможным. Тренд позволяет дать оценку количества нот, охватываемых на исследуемом участке, но не тип исполнения. В свою очередь скользящее среднее одновременно интерпретирует и наличие колебаний, и охватываемые ноты, что делает данный метод оценки более предпочтительным для вибратоподобных участков.

4. Анализ глиссандирующих переходов в пении

Для определения глиссандо в исследуемом участке необходимо выполнить другую задачу. Базовый алгоритм сегментации и идентификации нот определяет 2 ноты в подобных ситуациях: ту, с которой началось скольжение, и ту, на которой скольжение закончилось. Соответственно в качестве первой решаемой задачи для определения наличия глиссандо – сравнение нот на границах исследуемого участка. Если ноты окажутся идентичными, то участок между ними не может соответствовать глиссандо и должен быть рассмотрен на предмет отнесения к вибрато. В иных случаях мы будем наблюдать линию тренда, проходящую через несколько нот. Чем больше разница между нотами и чем быстрее был произведен переход между ними, тем больше будет угол у полученной линии тренда относительно оси времени.

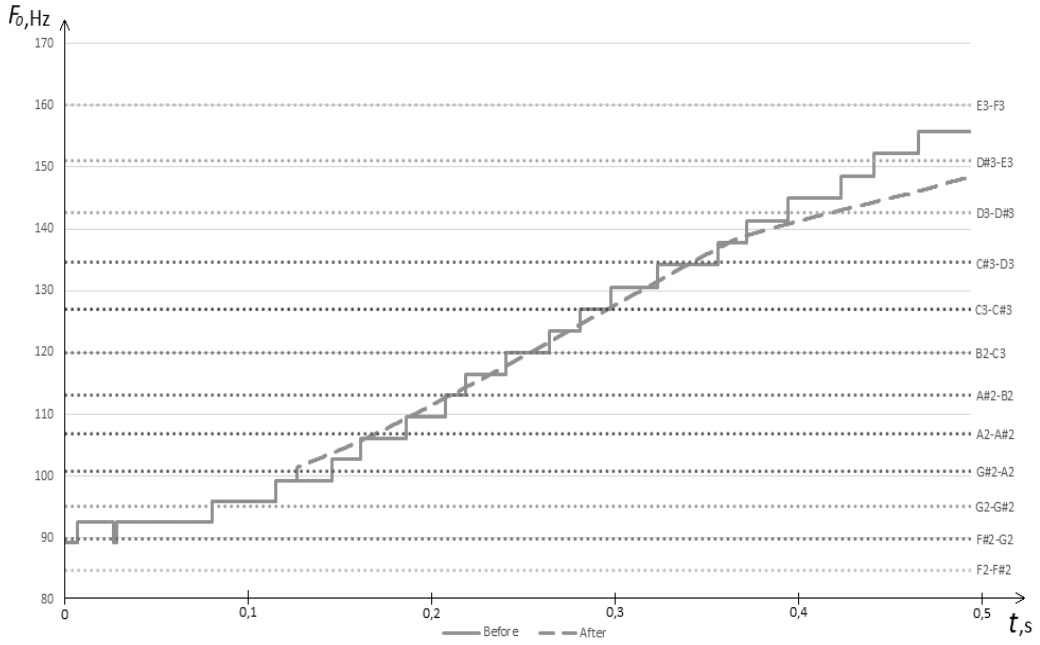


Рис. 7. Скользящее среднее для участка с восходящим глиссандо

Как и для участка с тремоляцией, рассмотрим скользящее среднее для исследуемого диапазона. Как можно заметить на рис. 7, в переходе от ноты «соль малой октавы» к ноте «ми 1 октавы» отсутствуют явные колебания в частоте основного тона. Сплошной линией на графике обозначены частоты основного тона, а пунктирной линией – скользящее среднее для исследуемого диапазона. Параллельные точечные линии соответствуют границам нот. Полученная линия скользящего среднего проходит через 9 нот. Таким образом, полученные данные свидетельствуют о восходящем глиссандо. Ниже (рис. 8) представлен график для линейного тренда, полученного на исследуемом диапазоне. Фактически, линейный тренд для исследуемого диапазона близок к результату, полученному при построении скользящего среднего.

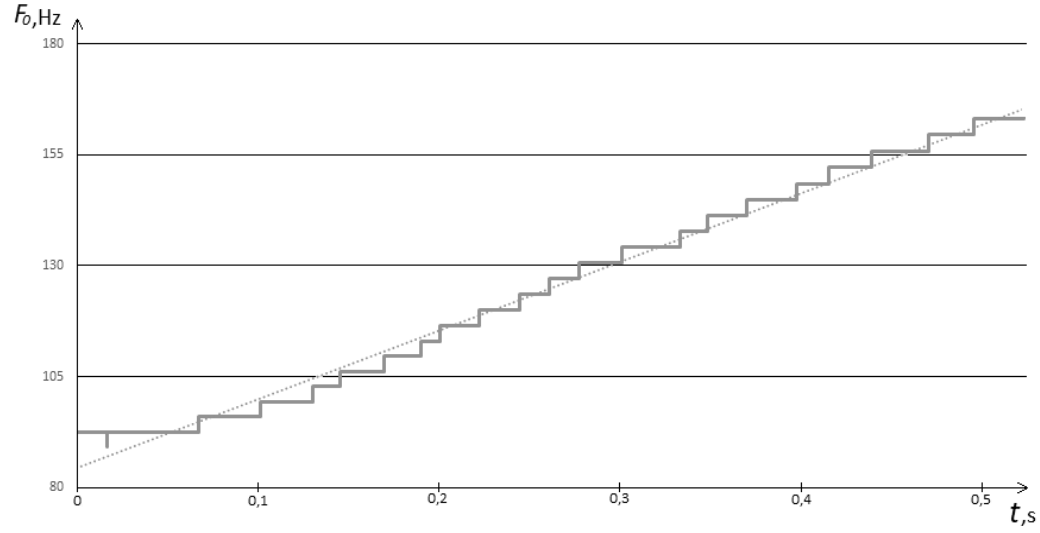


Рис. 8. Линейный тренд для участка с восходящим глиссандо

Заключение

Как можно заметить из полученных результатов, при использовании скользящего среднего для участков с тремоляцией наблюдается эффект колебания, отсутствующий в линейном тренде для данного участка. С другой стороны, для глиссандо скользящее среднее и линейный тренд ведут себя схожим образом. Это позволяет нам применять оба метода комплексно для определения текущей ситуации.

Предлагаемая идея заключается в получении обеих оценок для исследуемого отрезка. Полученные наборы данных будут сравниваться на предмет величины сходства между их оценками некоторому параметру *N*. Размер величины данного параметра требует отдельного эксперимента. В рамках данного эксперимента должны быть рассмотрены смешанные ситуации, в рамках которых временные ряды ведут себя неоднозначно, с целью снижения числа ошибок 1-го и 2-го рода при определении типа исследуемых диапазонов частот основного тона.

В случае, если в пении было глиссандо, результаты скользящего среднего и линейного тренда для данного временного ряда будут близки. Для данной ситуации необходимо будет осуществлять проверку на предмет количества нот, через которые был осуществлен переход, и направление перехода (восходящее или нисходящее глиссандо). В результате будет возможно отсеивать участки с шумом между соседними сегментами, соответствующие одной ноте, и участки с всплеском шумов, находящимся за пределами звучания диапазона между нотами.

Для ситуаций, в которых разница в оценках временного ряда будет превышать величину критерия *N*, будем считать, что происходило пение с вибрато в голосе. Как было замечено ранее, скользящее среднее обладает волнообразной структурой в случае наличия колебаний в исследуемом участке. Подсчет количества колебаний в единицу времени может быть использован в задаче классификации типа тремоляции. Данный параметр будет полезен при обучении вокальному исполнению в случае слишком частых или редких колебаний в исполнении ноты. Кроме того, в случае определения вибрато в пении также будет необходим контроль количества нот, через которые проходил тренд. Как можно заметить на рис. 9, колебания для спетой ноты проходят с скольжением вниз. Фактически, в указанном примере происходит смешение глиссандо и вибрато. Однако, ни к одному из типов данный участок отнести не представляется возможным по двум причинам: для глиссандо не характерно наличие колебаний в голосе, а для вибрато не должно быть скольжений на другую ноту.

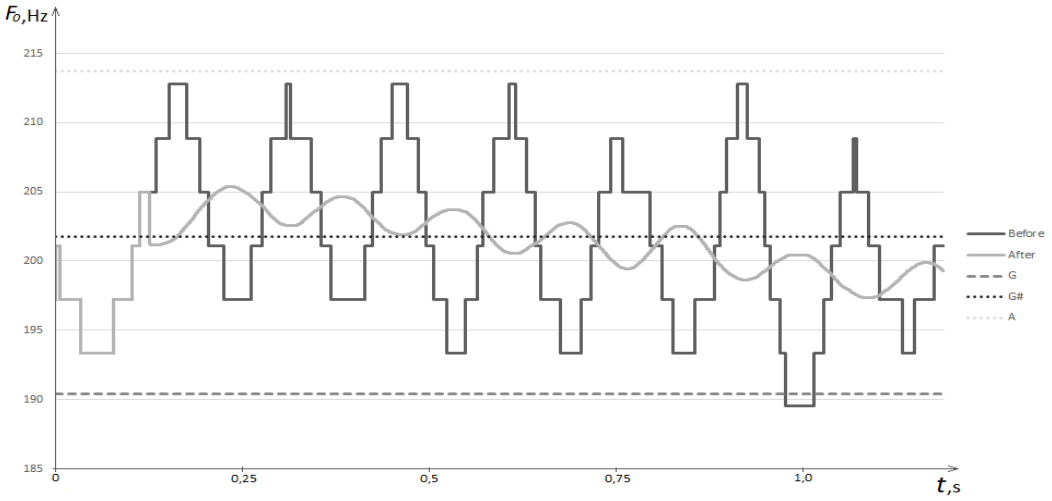


Рис. 9. Скользящее среднее для комбинированного участка

Одним из вариантов сравнения между собой полученных оценок является применение метода выделения синхронности [5], рассмотренный ранее в рамках оценки сходства вокальных исполнений [6].

ЛИТЕРАТУРА

1. *Leydon C., Bauer J.J., Larson C.R.* The role of auditory feedback in sustaining vocal vibrato. Acoustical Society of America. Vol. 114(3). 2003. P. 1575-1581.

2. *Ferguson S., Moere A.V., Cabrera D.* Seeing sound: real-time sound visualisation in visual feedback loops used for training musicians. Ninth International Conference on Information Visualisation (IV'05). London, UK. 2005. P. 97-102.

3. *Якимук А.Ю.* Распределенный программный комплекс по распознаванию нот / А.Ю. Якимук, М.Д. Холопов // Перспективы развития фундаментальных наук: сборник трудов XVI Международной конференции студентов, аспирантов и молодых ученых. Том 7. IT-технологии и электроника, 2019. – С. 125-127.

4. *Конев А.А.* Модель и алгоритмы анализа и сегментации речевого сигнала: автореф. дис. … канд. техн. наук. – Томск, 2007.

5. *Катаева Е.С.* Применение алгоритма выделения синхронности для метеорологических временных рядов/ Е. С. Катаева, Г.М. Кошкин // Известия вузов. Физика. — Т.56, № 9/2. — С.229–231.

6. *Катаева Е.С.* Применение выделения синхронности для оценки сходства вокальных исполнений / Е.С. Катаева, Ю.Р. Свешникова, А.Ю. Якимук // Информационно-коммуникационные технологии в педагогическом образовании. 2019. № 4 (61). С. 54-58.

II. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ИХ ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ

СОЗДАНИЕ ОБУЧАЮЩЕЙ ПРОГРАММЫ ДЛЯ ФОРМИРОВАНИЯ НАВЫКА ОПРЕДЕЛЕНИЯ ХАРАКТЕРИСТИК КРИВЫХ ВТОРОГО ПОРЯДКА И ИХ ПОСТРОЕНИЯ

Багдалов П.Д., Пахомова Е.Г.
Томский государственный университет
pablobagdal@gmail.com, peg@tpu.ru

Введение

Компьютерные технологии занимают всё большее место в жизни человека. В частности, в области образования. Мы все свидетели того, как люди, несмотря на пандемию, стремятся делиться знаниями и развиваться. Но не всем удаётся возобновить полноценный учебный процесс. В частности, проверку знаний студентов и анализ их ошибок. В привычном формате лицом к лицу преподаватель тратит время на каждого студента отдельно, сталкиваясь с типичными ошибками и объясняя их причину. В дистанционном формате по аудио/видео связи объяснить то же самое гораздо сложнее, а если приходится писать – у преподавателя и вовсе не останется личного времени. Что, в свою очередь, негативно влияет на продуктивность.

Компьютер – это, в первую очередь, быстрый доступ к огромному объёму информации, её обработке, обмене и хранении. Так, имея доступ к сети общего пользования или загрузив данные с внешних носителей, любому человеку предоставляется возможность самостоятельно развивать свои навыки в той или иной области.

На данный момент, обладая базовыми навыками работы с компьютером, человек может найти в Интернет ответ практически на любой вопрос и этот факт на первый взгляд обещает блестящие перспективы дистанционному образованию. Но более пристальное изучение вопроса показывает, что информация в сети часто противоречива, в том числе в разделе точных наук, например в математике. Поэтому полностью самостоятельное самообразование посредством онлайн-платформ и тренажёров крайне затруднительно. Обучающийся может встретить в сети не вполне корректный алгоритм, или по неопытности переложить алгоритм, применимый для одного типа задач на другие, где он вовсе недопустим.

И всё-таки обучение с использованием программ и онлайн ресурсов допустимо в разумных пределах. На мой взгляд, если мы не подвергаем сомнению авторитетности университетского образования, то и знания преподавателей принимаем как достоверные. Поэтому допустимо использование, в первую очередь, тех ресурсов, которые рекомендуются непосредственно преподавателями.

Предлагаемое решение призвано улучшить жизнь студентов и преподавателей. Много времени уходит впустую. Почему так происходит? Ответ довольно простой: мы не пользуемся теми возможностями и технологиями, которые у нас есть. Речь об обучающих программах, которые могут если не заменить преподавателя, то частично делегировать его обязанности по проверке знаний студентов, избавляя от рутиной работы с однотипными ошибками.

Именно желание делегировать компьютеру частично или полностью такие функции преподавателя, как проверка полученных знаний и анализ ошибок обучающегося, привело к появлению программ, которые называются обучающими.

Существуют несколько видов обучающих программ: контролирующие, наставнические, имитационные и моделирующие, а также развивающие игры [1].

Контролирующие программы предназначены для проверки усвоенного учащимся материала. Как правило, такие программы в случайном порядке генерирует вопросы и задачи из некоторого банка заданий и сверяет введенный ответ с верным ответом.

Наставнические программы, это программы, которые предоставляют учащемуся теоретический материал, разбитый на блоки, и проверяют усвоение этого материала, предлагая в конце каждого блока ответить на контрольные вопросы. Ответы учащегося анализируются, и на основе этого анализа определяется следующий шаг в обучении. Таким образом эти программы осуществляют управление ходом обучения. Среди наставнических программ выделяют линейные, разветвленные, адаптивные и комбинированные. Линейная программа состоит из небольших последовательно сменяющихся блоков информации с контрольными вопросами. По результатам ответов учащийся либо возвращается к началу блока (среди ответов есть неверные), либо переходит к изучению следующего блока (все ответы верны). Разветвленные программы отличаются от линейных тем, что обучающемуся при неправильном ответе даётся возможность воспользоваться дополнительной информацией и исправить ответ. Адаптивные программы предполагают выбор уровня сложности учебной информации, возможность при необходимости обращаться к источникам информации. Комбинированные включают в себя компоненты линейных, разветвленных и адаптивных программ. Среди коммерческих обучающих программ наиболее распространены именно комбинированные программы.

Главное отличие контролирующей программы от наставнической: контролирующая программа способна только определить правильность ответа обучающегося, наставническая программа – способна проанализировать ответ учащегося и дать рекомендацию по дальнейшему ходу обучения.

Имитационные и моделирующие – позволяют использовать компьютер для эксперимента. Вводя команды с клавиатуры, обучающийся управляет ходом эксперимента.

Развивающие игры – создают определенную виртуальную среду и возможность влиять на эту среду со стороны пользователя, тем самым изучая эту среду. Чаще всего виртуальная среда частично или полностью моделирует наш реальный мир. Среди всех обучающих программ, развивающие игры самые сложные в реализации. Их создание требует совместной работы программистов, специалистов в предметной области, психологов [2].

В интернете для студентов, изучающих математику, предлагается множество онлайн-калькуляторов, дающих готовое решение. И они прекрасно выполняют свою задачу, если интересует только ответ, но они не решают главной задачи обучающей программы – формирование устойчивого навыка, который формируется только в том случае, когда обучающийся сам решает задачу, а не разбирает готовое решение. Поскольку процесс решения задачи, как правило, сопровождается ошибками, особенно на начальном этапе изучения темы, и эти ошибки для студента не очевидны, то наиболее эффективными будут обучающие программы наставнического типа, которые проанализируют ответ студента и дадут ему рекомендацию по дальнейшим действиям.

В области математики, по сравнению с количеством онлайн-калькуляторов, к использованию которых студенты часто прибегают, крайне мало наставнических программ. В частности, в области формирования навыка определения характеристик кривых второго порядка и их построения программ обнаружено не было ни в русскоязычном, ни в англоязычном интернете, в связи с чем было принято решение о разработке такой программы.

Следует заметить, что не для любого раздела математики можно создать эффективную обучающую программу. Это возможно только для тех разделов, где приобретение навыка связано с реализацией вычислительных алгоритмов. К таким разделам относится линейная алгебра, векторная алгебра, аналитическая геометрия и некоторые разделы математического анализа.

1. Постановка задачи

Цель работы – создать обучающую программу наставнического типа, которая бы позволяла приобрести навык определения характеристик кривых второго порядка и их построения.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Провести анализ типовых ошибок, допускаемых при определении характеристик и построении.

2. Реализовать класс SecondOrderCurve, определяющий базовую структуру кривых.

3. Реализовать классы наследники Ellipse, Circle, Hyperbola и Parabola, отражающие специфику каждого типа кривой и генерирующие соответствующие значения параметров уравнения.

4. Реализовать выявление следующих ошибок студента:

а) ошибка определения названия кривой;

б) ошибка определения формы кривой;

в) ошибка соответствия введённого названия кривой к выбранной графической форме;

г) ошибка знака координат центра или вершины;

д) ошибка выделения полного квадрата;

е) арифметическая ошибка;

ж) ошибка определения полуосей или других параметров.

5. Реализовать пользовательский интерфейс включающий в себя:

а) главное меню;

б) справочный материал;

в) разобранный пример;

г) тренажер (генерация уравнения, поля для ввода ответа).

2. Анализ типовых ошибок студента

Поскольку определение характеристик кривых второго порядка – это определенный алгоритм, то если студент совершает ошибку, это значит, что он выполнил неверный шаг алгоритма или несколько шагов. Комбинаций ошибок может быть большое количество и все проанализировать крайне сложно, но, как показывает практика, наиболее часто студенты совершают следующие ошибки:

а) неправильно определяет тип кривой;

б) ошибается в знаках координат центра (вершины);

в) путает оси;

г) неправильно выделяет полный квадрат.

После того, как студент вводит ответ, программа сравнивает ответ студента с данными, полученными из сгенерированного объекта и, если ошибок нет, то сообщает об этом пользователю. Если ошибки есть, то для выявления каждой из ошибок используется специальный метод. Рассмотрим каждый из них подробнее.

1) Для выявления ошибки названия кривой, идёт сравнение выбранного пользователем варианта с тем, что прикреплён к сгенерированному объекту. Если форма кривой при этом будет определена правильно, то студент получит уведомление: «Неверное название».

2) Для выявления ошибки определения формы кривой используется аналогичный элемент объекта. Если пользователь правильно определил название, но ошибся в определении формы кривой, то получит уведомление «Неверная форма кривой», а если ошибся в обоих случаях, то идёт проверка соотнесения введённого названия с указанной формой кривой. Если соответствие установлено, но пользователь получит уведомление: «Неверная кривая», а если указанная форма кривой не будет соответствовать

названию кривой, то получит уведомление: «форма кривой не соответствует указанному названию».

3) Для выявления ошибки знака координаты введённое значение сравнивается с член-данным объекта, хранящим координаты центра (фокусов). Студент получает соответствующее уведомление: «Ошибка знака в координатах центра (фокусов)».

4) Маркером ошибки выделения полного квадрата является не совпадение абсолютных величин координат центра (вершины). При наличии этого маркера, программой генерируется «неверные ответы» и сравнивает их с введенными студентом данными. Их совпадение означает, что такого рода ошибка действительно допущена и пользователь получает уведомление «Ошибка выделения полного квадрата». Ошибка выделения полного квадрата наиболее частая арифметическая ошибка, но возможны и другие типы арифметических ошибок. Поэтому, если «неверные ответы» не совпали с ответами студента, то выводится сообщение об арифметической ошибке, хотя в перспективе возможно внедрить и другие более редкие, но тоже возможные ошибки.

3. Интерфейс программы

В текущей реализации рассматривается обработка только эллипса. Все варианты допускаемых ошибок и последствий учитываются в построенной блок-схеме (рис. 1, 2)

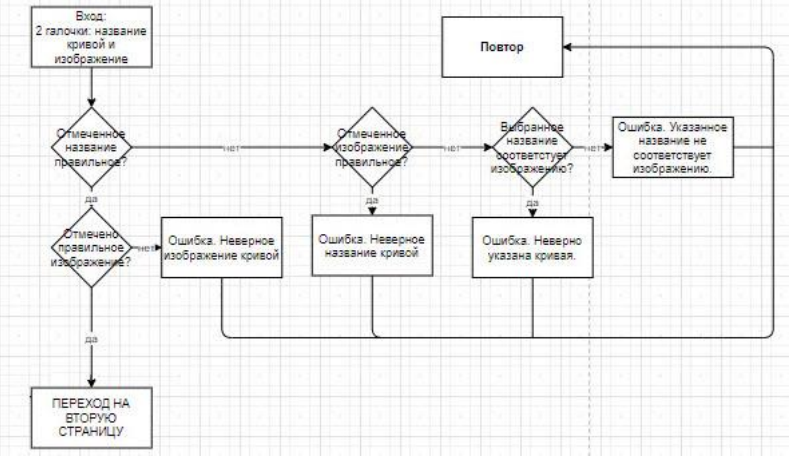


Рис 1. Блок-схема логики обработки ошибок первой страницы упражнений

Интерфейс программы интуитивно понятен. Это важно для обучающей программы и влияет на ее эффективность [3]. Работа программы начинается с открытия главного меню (рис. 3). В главном меню студенту предлагается выбрать следующие действия:

1) Начать
2) Теория
3) Пример
4) Выход

Выбор действия осуществляется нажатием соответствующей кнопки.

Кнопки «Теория» и «Пример» однотипные и предоставляют возможность ознакомления с теоретическим материалом и разобранным примером. По окончании работы с соответствующими окнами происходит возврат в главное меню. Такая организация сделана для того, чтобы во время выполнения собственного задания студент не переписывал решение по образцу, а думал самостоятельно.

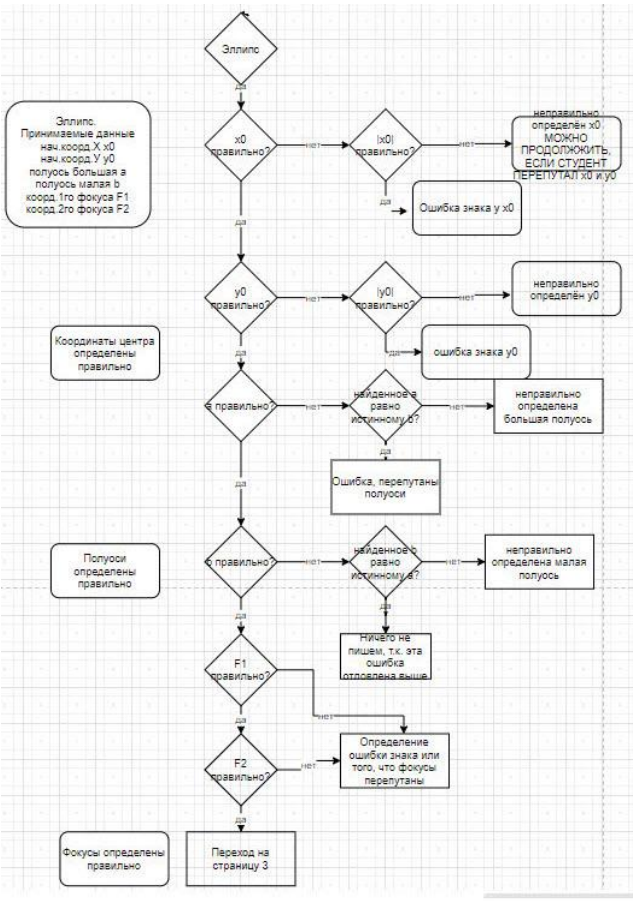


Рис 2. Блок-схема логики обработки ошибок второй страницы упражнения (для эллипса)

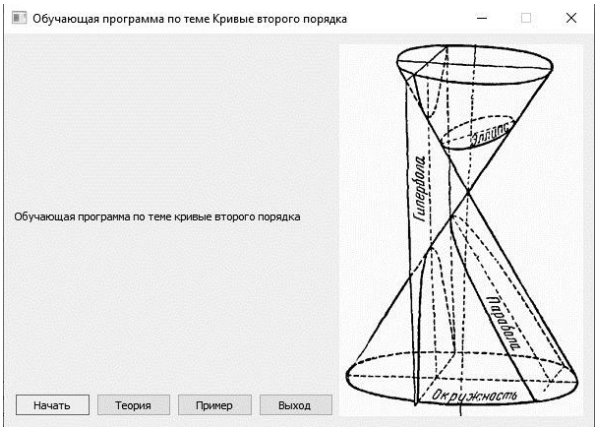


Рис 3. Главное меню обучающей программы

Кнопка «Начать» приводит на единый для всех типов кривых экран: выбор названия кривой и её формы (рис. 4,5) Уже на этом этапе, по опыту преподавателей, у обучаемых возникают сложности. При допущении ошибки мы дадим понять, какого рода ошибка допущена и предложим изучить теорию.

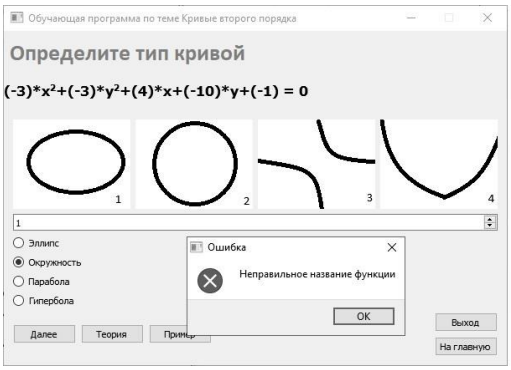


Рис 4. Первое окно программы. Ошибка названия

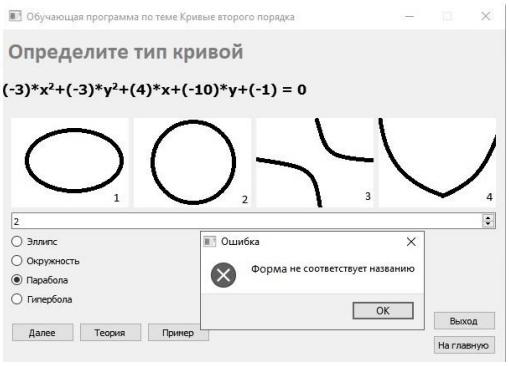


Рис.5. Первое окно программы. Ошибка соответствия

Кнопка «Далее» переводит на специализированный для каждой кривой экран (рис. 6). В нём студент задаёт все запрашиваемые параметры и характеристики. Нередки случаи, когда решение правильно, но в начале допущена ошибка знака или перепутаны полуоси, вследствие чего перепутаны координаты фокусов. Или, наоборот, оси определены правильно, но при дополнении до полного квадрата – арифметическая ошибка (рис. 7)

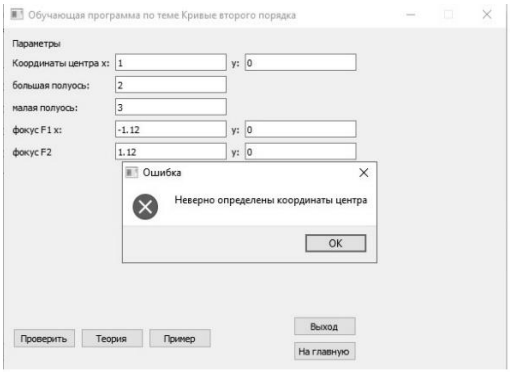


Рис 6. Ошибка знака координат

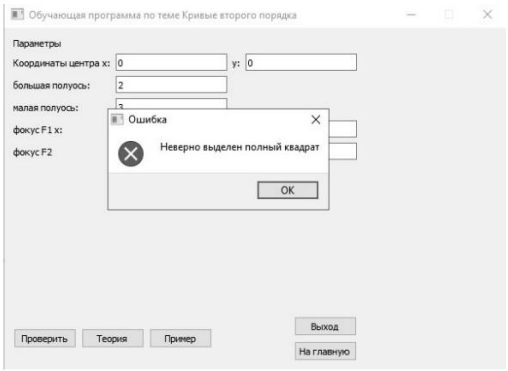


Рис.7. Ошибка выделения полного квадрата

Преподаватели с большим стажем знают все типичные ошибки студента в своей области. Обрабатывая в программе такие сценарии, мы поможем как студенту, так и преподавателю. Студент сможет быстрее понять, в чём он ошибается и на что следует обратить внимание. Это позволит быстрее сформировать необходимый ему навык, т.к. в случае неудачного решения задачи место ошибки и ее тип определяется немедленно, а значит можно исправить ошибку и продолжить дальнейшую работу. А преподаватель избавляется от необходимости выявлять в работах студентов типовые ошибки. Это уменьшает нагрузку на преподавателя, ведь значительную часть работы преподавателей занимают проверка работ студентов и работа со студентами над их ошибками.

При этом надо отметить, что полностью заменить преподавателя программа не сможет. Она выявляет именно типичные ошибки. Но существуют и другие ошибки. Например, ошибки, которые связаны с непониманием студентом теоретического материала. И здесь не поможет программа, здесь необходим человек, который поймет, в чем проблема и сможет помочь эту проблему преодолеть.

Заключение

В результате выполненной работы по созданию обучающей программы был проведен анализ существующих обучающих программ, анализ типовых ошибок, допускаемых при определении характеристик кривых второго порядка и их построении, реали-

зован класс SecondOrderCurve, Ellipse, создана работающая обучающая программа. Продолжается разработка классов Circle, Hyperbola, Parabola, будет произведена доработка существующего алгоритма, дизайна, а также адаптация по сложности. Такие программы нужны, их нужно развивать и популизировать.

ЛИТЕРАТУРА

1. *Гефан Г.Д., Кузьмин О.В.* Методика построения контрольно-обучающих программ и их использование в преподавании математических дисциплин // Вестник Бурятского государственного университета. 2013. № 15. С. 23–28.

2. *Аязбаев Т.Л., Галагузова Т.А.* Технология создания компьютерных обучающих программ // Международный журнал экспериментального образования. 2015. – № 3 (часть 2) – С. 76-78.

3. *Афанасьев В.В., Тыщенко О.Б., Афанасьева И.В.* Анализ показателей эффективности обучающих программ // I Всероссийская научно-техническая конференция 'Компьютерные технологии в науке, проектировании и производстве'. Тезисы докладов, часть V, Нижний Новгород, 1999, 43с., стр. 15.

ДВУХКОМПОНЕНТНОЕ РАЗЛОЖЕНИЕ КАРДИОЛОГИЧЕСКОЙ КРИВОЙ

Безходарнов Н.И., Самохина С.И.
Томский государственный университет
nblaaa@mail.ru

Введение

Ещё с давних времён людей беспокоил вопрос о работе сердца. Оно является жизненно необходимым органом, по этой причине исследование его работы очень важно для всего человечества. Одним из аспектов является изучение сердечных мышц, которые выполняют основную работу. При обследовании таких мышц строят кардиологическую кривую, для анализа которой необходимо разложение на две компоненты [1,2], имеющие экспоненциальный вид. Данная статья и повествует об одном из способов решении данной проблемы.

1. Постановка задачи

Рассматривается кардиологическая кривая, которая задаётся последовательностью точек на плоскости. Чтобы провести её анализ, требуется получить разложение на две компоненты. Ставится задача получения аппроксимации данной кривой двумя компонентами экспоненциального вида [1–3].

Целью данного проекта является разработка приложения, реализующего разложение кардиологической кривой в функцию следующего вида:

$$y = a_1 e^{-b_1 (x-c_1)^2} + a_2 e^{-b_2 (x-c_2)^2} . \quad (1)$$

Для достижения данной цели были сформулированы следующие подзадачи:

1. Изучение теоретического материала по аппроксимации функций и нахождению точек минимума и максимума.
2. Выбор метода аппроксимации функции для реализации алгоритма.
3. Реализация алгоритма разложения

После выполнения данных пунктов была написана программа с удобным графическим интерфейсом, которая решает поставленную задачу.

2. Функциональные возможности программы

Реализация была выполнена в среде разработки Qt Creator с использованием библиотек Qt [4]. Данная среда является удобной в разработке графического интерфейса, а также способна обеспечить кроссплатформенность. Библиотеки Qt предоставляют широкий спектр возможностей для разработки различного рода приложений.

Функционал разрабатываемой программы включает в себя следующее: чтение кардиологической кривой из устройства, получающее данные при проведении опыта, или из файлов, содержащих результаты прошлых опытов; отображение графиков на основе полученных данных; нахождение разложения кардиологической кривой, её построение и вывод подробной информации о результатах. Для реализации данного функционала был написан алгоритм, позволяющий найти аппроксимирующую кривую к кардиологической и составляющий основную часть программы.

3. Описание алгоритма

Алгоритм, предложенный в этой статье, решает задачу аппроксимации функции определённого вида. Начальными данными для него является кардиологическая кривая, которая задаётся с помощью последовательности точек на плоскости. Ставится задача получения из такого набора точек функции вида (1).

В начале алгоритма среди заданных точек ищутся две с максимальным значением функции кардиологической кривой, по которым строится начальное приближение следующим образом: коэффициенты c_1 и c_2 приравниваются аргументам заданной функции в этих точках, a_1 и a_2 – значениям функции, а b_1 и b_2 находятся с помощью близлежащих к найденным точек из следующего уравнения:

$$-b_i\left(x-c_i\right)^2=\ln\left(\frac{y}{a_i}\right),$$

где $\left(x,y\right)$— координаты близлежащей точки.

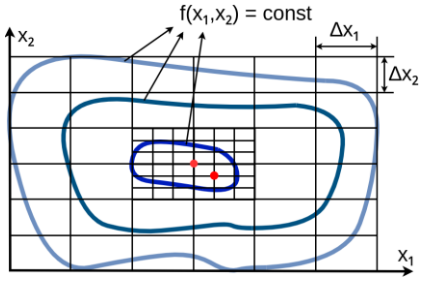


Рис. 1. Изображение работы метода сетки в двухмерном пространстве

После нахождения начального приближения используется, так называемый, метод сетки [5], являющийся методом нахождения минимума функции, который заключается в следующем: задаётся некоторый шаг, на который будет осуществляться отступ по каждой координате, затем вычисляются значения функции в точках, отстоящих от текущей на заданный шаг, причём отступ происходит во все возможные стороны. Из всех вычисленных значений выбирается минимальное и запоминаются координаты точки, при которых это значение достигается. Далее данные действия повторяются до тех пор, пока в найденной точке функция не достигнет требуемого минимума.

Так как задачей является аппроксимация функции, а метод сетки предназначен для нахождения точки минимума, то было необходимо построить такую функцию, точка минимума которой позволила бы найти требуемую нам аппроксимацию. Функция для данного метода была составлена следующим образом: из заданной кривой берутся 50 равноотстоящих друг от друга точек (если точек задано меньше, то тогда используются все) и вычисляется сумма модулей разности координат значений кривой на значения аппроксимирующей функции в заданной точке:

$$f\left(a_1,b_1,c_1,a_2,b_2,c_2\right)=\sum_{i=1}^{50}\left|y_i-a_1e^{-b_1\left(x-c_1\right)^2}-a_2e^{-b_2\left(x-c_2\right)^2}\right|.$$

Бывают случаи, когда метод нашёл локальную точку минимума, которая не подходит, то есть функция аппроксимирует исходную недостаточно хорошо. Тогда алгоритм специально ухудшает результат, чтобы покинуть окрестность этой точки и попасть в окрестность нужной.

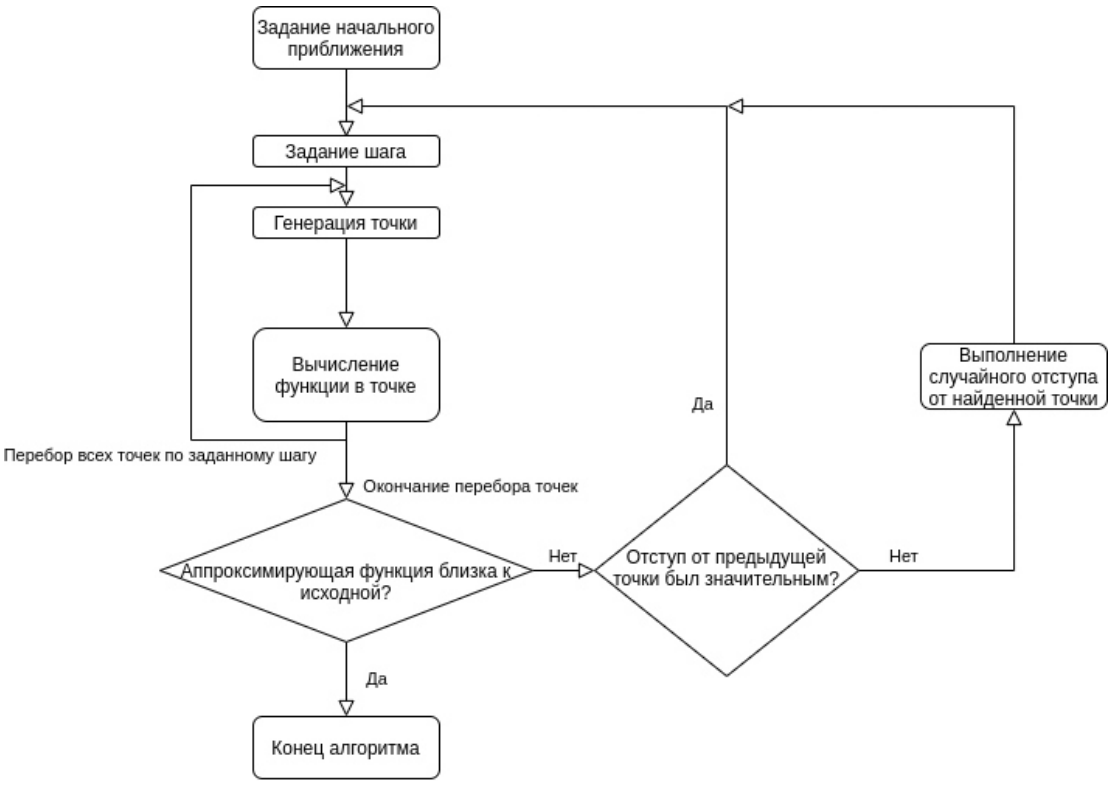


Рис. 2. Схема алгоритма

Данный алгоритм находит требуемую аппроксимацию кардиологической кривой, но для удобного взаимодействия с ним был написан графический интерфейс, который обеспечивает простоту работы.

4. Интерфейс программы

Графический интерфейс реализован в виде окна, содержащего кнопки, текстовые поля и инструменты для построения графиков. При запуске программы появляется одно главное окно. Для начала работы требуется открыть файл, из которого нужно загрузить данные о кардиологической кривой. Это можно сделать, нажав кнопку «Открыть файл» и в появившемся диалоговом окне выбрав файл для открытия. После этого станут активными текстовые поля «Эксперимент», «Канал» и кнопка «Построить график». Для дальнейшей работы нужно выбрать нужные номера эксперимента и канала и затем нажать на кнопку «Построить график». Тогда в окне появится специальное поле, в котором отобразится график и с которым можно взаимодействовать следующим образом: колесом мыши можно приближать и отдалять график; движением мыши с зажатой левой клавишей можно двигать график; с помощью кликов левой клавиши мыши можно задавать промежуток, выбирая начальную и конечную точки, для аппроксимации. Для удобства задания промежутка на курсоре мыши в поле с графиками отображается чёрная вертикальная линия, помогающая ориентироваться. После задания промежутка активируется кнопка «Начать аппроксимацию», нажав которую начнутся вычисления. Во время вычислений главное окно потемнеет, станет неактивным, и на нём появится окно

с просьбой подождать. Когда все вычисления закончатся, то это маленькое окно закроется, а на главном отобразятся графики аппроксимации, а именно: аппроксимирующая кривая и кривые её компонент по отдельности.

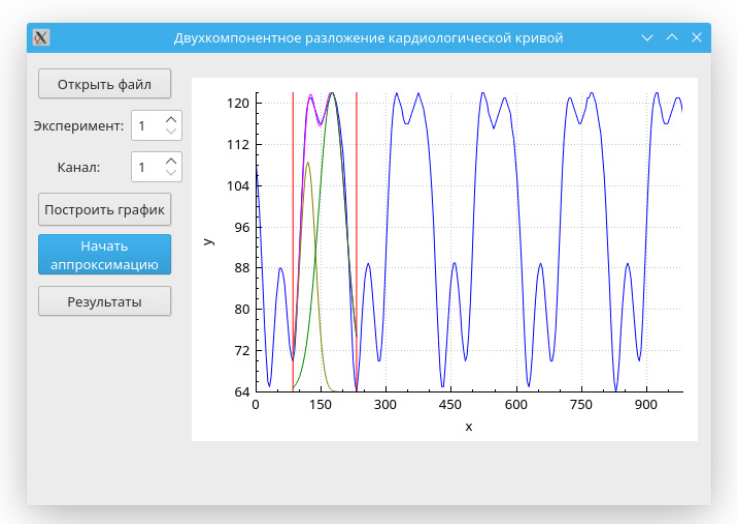


Рис. 3. Главное окно программы после вычислений

Также после вычислений станет активной кнопка «Результаты», нажав которую можно получить коэффициенты получившейся функции и ещё вид первой и второй производных.

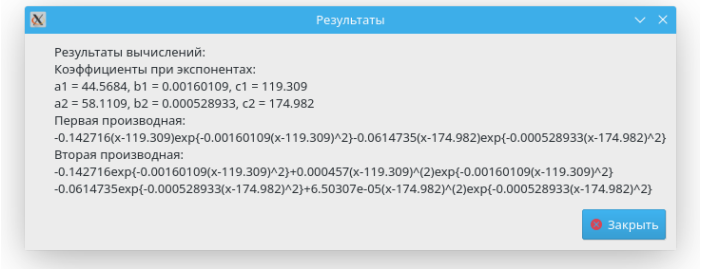


Рис. 4. Окно информации о результатах

Таким образом, программа выполняет свою поставленную задачу и наглядным образом отображает получившиеся результаты вычислений.

Заключение

Как показывает практика, данный алгоритм успешно справляется со своей задачей. Он был протестирован на множестве разных входных данных, и на выходе получаются достаточно хорошие результаты.

Надеемся, что в будущем данный алгоритм окажет немалое влияние в нашей жизни и поможет сделать её лучше и безопаснее.

ЛИТЕРАТУРА

1. *Богомаз С.А.* Исследование действия кардиологических веществ на пулы кальция с помощью анализа компонентов структуры сокращения миокарда: дис. … канд. мед.наук /С.А. Богомаз – Томск, 1991. –184с.

2. *Тарасенко В.Ф., Лаптев Б.И.* Способ компонентного анализа сокращения миокарда // Сборник трудов молодых ученых ТНЦ АМН. 1991. – С.44-47

3. *Самохина С.И., Шишкин М.М.* Программный комплекс моделирования двухкомпонентного разложения кардиологической кривой // Математическое и программное обеспечение информационных, технических и эконо-

мических систем : материалы VI Междунар. молодежной науч. конф. Томск, 24-26 мая 2018 г. Томск: Издательский Дом ТГУ, 2018. С. 83-88.
4. Qt [Электронный ресурс]: Qt Reference Pages – URL: https://doc.qt.io/qt-5/reference-overview.html
5. Pandia [Электронный ресурс]: 2.4 Методы поиска экстремума функций многих переменных – 2013 – URL: https://pandia.ru/text/80/091/32983.php

АЛГОРИТМ И ПРОГРАММА РАСЧЕТА ЭЛЕКТРОМАГНИТНОГО РАССЕЯНИЯ НА ТОНКИХ ОРТОГОНАЛЬНЫХ ИДЕАЛЬНО ПРОВОДЯЩЕМ И ДИЭЛЕКТРИЧЕСКОМ ЦИЛИНДРАХ

Дмитренко А.Г., Балашова О.М.
Томский государственный университет
dmitr.tsu.202@mail.ru, balashovajkz@mail.ru

Введение

Значительный интерес для исследователей представляет изучение рассеяния электромагнитных волн в резонансной частотной области на структурах, состоящих из одного или нескольких тонких цилиндров конечной длины. Этот интерес обусловлен необходимостью решения таких практически важных проблем как проблемы радиолокационной заметности, идентификации объектов, оценки рассеяния диэлектрическими или металлическими цилиндрическими деталями различных геометрически сложных тел и др.

Под тонким цилиндром обычно понимается цилиндрическое тело, поперечные размеры которого много меньше его длины и длины падающей волны. Анализ имеющейся в распоряжении авторов литературы показывает, что известны работы, например [1–4], в которых рассмотрено рассеяние электромагнитной волны на одиночном тонком прямолинейном идеально проводящем цилиндре, а также работы, например [5–7], в которых рассмотрено рассеяние электромагнитной волны на одиночном тонком однородном прямолинейном диэлектрическом цилиндре. Некоторый вклад в решение проблемы рассеяния электромагнитной волны на тонких цилиндрах внесен и работами одного из авторов данной статьи. Так, например, в работе [8] рассмотрено рассеяние на структурах, состоящих из нескольких тонких прямолинейных идеально проводящих цилиндров, в работе [9] предложен новый метод решения задачи электромагнитного рассеяния на тонком диэлектрическом цилиндре. Однако в известной литературе до сих пор отсутствуют работы, посвященные рассеянию на структурах, состоящих как из идеально проводящих, так и из диэлектрических тонких цилиндров, кроме работ авторов данной статьи, например [10–11]. В работе [10] предложен численный метод решения задачи электромагнитного рассеяния на структурах, состоящих из одного тонкого диэлектрического цилиндра и одного тонкого идеально проводящего цилиндра. В работе [11] этот метод применен к анализу рассеяния электромагнитной волны на структуре, состоящей из параллельных идеально проводящего и диэлектрического цилиндров.

Данная работа продолжает цикл работ авторов, посвященных электромагнитному рассеянию на структурах, содержащих как тонкие идеально проводящие цилиндры, так и тонкие диэлектрические цилиндры. В ней рассмотрено рассеяние электромагнитной волны на структуре, состоящей из двух тонких ортогональных цилиндров, один из которых является диэлектрическим, а другой – идеально проводящим. Такая структура является моделью реальной ситуации, когда прямолинейная тонкая металлическая антенна размещена на тонкой диэлектрической цилиндрической подставке и ориентирована вдоль поверхности Земли.

1. Постановка задачи и ее решение

Геометрия задачи показана на рис. 1. Рассматривается стационарная (зависимость от времени выбрана в виде $e^{-i\omega t}$ задача дифракции электромагнитного поля $\vec{E}_0, \vec{H}_0$ на

структуре, состоящей из двух тонких прямолинейных ортогональных цилиндров, один из которых является диэлектрическим, а другой – идеально проводящим. Диэлектрический цилиндр имеет длину l_d, радиус r_d и характеризуется электродинамическими параметрами ε_i, μ_i. Идеально проводящий цилиндр имеет длину l_p, радиус r_p. Для цилиндров выполняются условия $2r_d << \lambda$, $r_d << l_d$; $2r_p << \lambda$, $r_p << l_p$, где λ – длина падающей волны. Декартова система координат *Oxyz* выбрана таким образом, что ее начало *O* совпадает с серединой осевой линии диэлектрического цилиндра, а ось *z* направлена вдоль его осевой линии. Идеально проводящий цилиндр расположен так, что его осевая линия ориентирована вдоль оси *x*. Структура размещена в однородной среде D_e с электродинамическими параметрами ε_e, μ_e. Требуется найти рассеянное поле $\{\vec{E}_e, \vec{H}_e\}$ в среде D_e.

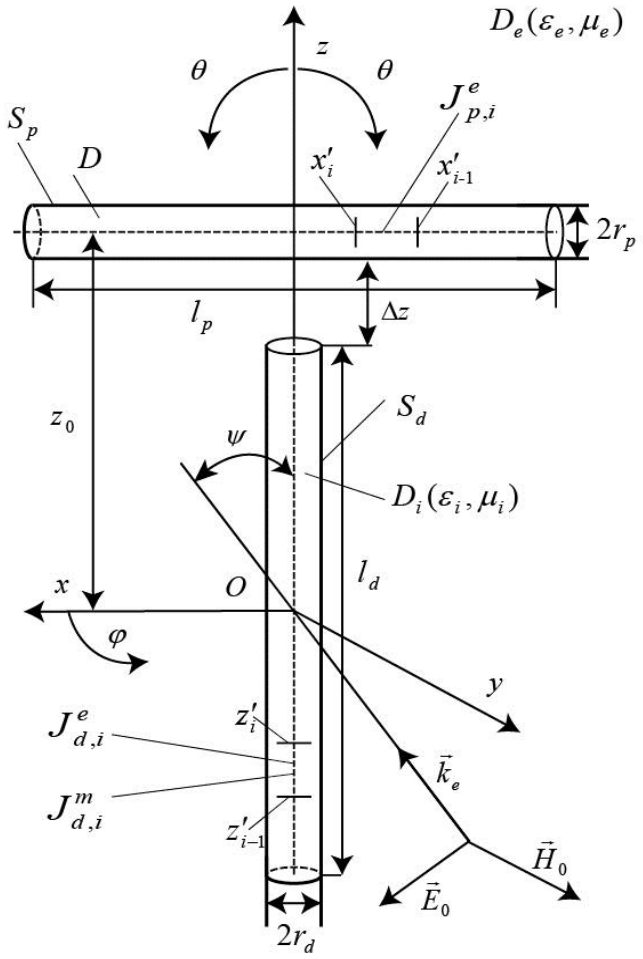


Рис. 1. Геометрия задачи

Кроме поля $\{\vec{E}_e, \vec{H}_e\}$ в D_e, существует поле $\{\vec{E}_i, \vec{H}_i\}$ внутри диэлектрического цилиндра (в области D_i). Поля $\{\vec{E}_e, \vec{H}_e\}$ и $\{\vec{E}_i, \vec{H}_i\}$ должны удовлетворять уравнениям Максвелла

$$\operatorname{rot} \vec{E}_e = i\omega\mu_e \vec{H}_e, \ \operatorname{rot} \vec{H}_e = -i\omega\varepsilon_e \vec{E}_e \text{ в } D_e, \quad (1)$$

$$\operatorname{rot}\vec{E}_i = i\omega\mu_i\vec{H}_i,\ \operatorname{rot}\vec{H}_i = -i\omega\varepsilon_i\vec{E}_i \text{ в } D_i, \tag{2}$$

граничным условиям

$$\left[\vec{n}_i, \left(\vec{E}_i - \vec{E}_e\right)\right] = \left[\vec{n}_i, \vec{E}_0\right],\ \left[\vec{n}_i, \left(\vec{H}_i - \vec{H}_e\right)\right] = \left[\vec{n}_i, \vec{H}_0\right] \tag{3}$$

на поверхности S_d диэлектрического цилиндра и граничным условиям

$$\left[\vec{n}, \vec{E}_e\right] = -\left[\vec{n}, \vec{E}_0\right] \tag{4}$$

на поверхности S_p идеально проводящего цилиндра.

Кроме того, поле $\left\{\vec{E}_e, \vec{H}_e\right\}$ в D_e должно удовлетворять условиям излучения

$$\left[\sqrt{\varepsilon_e}\vec{E}_e, \vec{R}/R\right] + \sqrt{\mu_e}\vec{H}_e = O\left(R^{-1}\right),\ \left[\sqrt{\mu_e}\vec{H}_e, \vec{R}/R\right] - \sqrt{\varepsilon_e}\vec{E}_e = O\left(R^{-1}\right),\ R \to \infty. \tag{5}$$

В выражениях (3)–(5) $\vec{n}_i$ – единичный вектор нормали к поверхности S_d диэлектрического цилиндра; $\vec{n}$ – единичный вектор нормали к поверхности S_p идеально проводящего цилиндра; $R = \sqrt{x^2 + y^2 + z^2}$; $\left[\vec{a}, \vec{b}\right]$ – векторное произведение. Отметим, что при записи граничных условий (3)–(4) учтено, что поле во внешней среде D_e представлено в виде суммы рассеянного $\vec{E}_e, \vec{H}_e$ и возбуждающего $\vec{E}_0, \vec{H}_0$ полей.

Метод решения задач рассеяния на структурах, содержащих один тонкий идеально проводящий цилиндр и один тонкий диэлектрический цилиндр, предложен и подробно описан в работе [10]. Основные идеи этого метода следующие. Поле $\left\{\vec{E}_e, \vec{H}_e\right\}$ во внешней среде D_e представлено в виде суммы полей неизвестных вспомогательных токов, непрерывно распределенных вдоль осей цилиндров. Поле $\left\{\vec{E}_i, \vec{H}_i\right\}$ внутри диэлектрического цилиндра представлено в виде суммы полей пар элементарных электрических диполей с неизвестными дипольными моментами, дискретным образом расположенных на вспомогательной поверхности, охватывающей диэлектрический цилиндр и также имеющей форму цилиндра. Такие представления для полей удовлетворяют уравнениям Максвелла (1)–(2) и условиям излучения (5). Для того, чтобы удовлетворить граничным условиям (3)–(4), необходимо соответствующим образом выбрать неизвестные распределения осевых токов и неизвестные дипольные моменты.

Для этого первоначально осевая линия диэлектрического цилиндра разбивается на N_d участков, а осевая линия идеально проводящего цилиндра – на N_p участков, в пределах которых величины вспомогательных токов можно считать постоянными; на рис.1 это отрезки $\left[z'_{i-1}, z'_i\right]$ и $\left[x'_{i-1}, x'_i\right]$, соответственно. Токи на этих участках $J^e_{d,i}$, $J^m_{d,i}$ и $J^e_{p,i}$ называются «элементами токов». Далее, с использованием метода коллокаций, то есть путем поточечного удовлетворения граничным условиям (3)–(4) в точках j_d на поверхности S_d диэлектрического цилиндра и в точках j_p на поверхности S_p идеально проводящего цилиндра получают систему линейных алгебраических уравнений для определения неизвестных элементов токов и дипольных моментов. После решения этой системы необходимые компоненты рассеянного поля получаются с использованием представления для поля $\left\{\vec{E}_e, \vec{H}_e\right\}$ во внешней среде D_e.

С учетом специфики рассматриваемой задачи этим методом получены следующие выражения для компонент рассеянного поля в дальней зоне в сферической системе координат:

$$E_{e,\theta}(M)=\sqrt{\frac{\mu_e}{\varepsilon_e}}H_{e,\varphi}(M)=\frac{e^{ik_eR}}{R}D_\theta(\theta,\varphi)+O\left(R^{-2}\right),$$
$$E_{e,\varphi}(M)=-\sqrt{\frac{\mu_e}{\varepsilon_e}}H_{e,\theta}(M)=\frac{e^{ik_eR}}{R}D_\varphi(\theta,\varphi)+O\left(R^{-2}\right). \quad (6)$$

Компоненты диаграммы рассеяния $D_\theta(\theta,\varphi)$ и $D_\varphi(\theta,\varphi)$ определяются выражениями

$$D_\theta(\theta,\varphi)=\frac{i\omega\mu_e}{4\pi}\left[-\sin\theta\sum_{i=1}^{N_d}J_{d,i}^e\int_{z_{i-1}'}^{z_i'}e^{-ik_ez'\cos\theta}dz'+\right.$$
$$\left.+\cos\theta\left\{\sin\theta\sin^2\varphi\sin(\theta-\varphi)+\cos\varphi\right\}e^{-ik_ez_0\cos\theta}\sum_{i=1}^{N_p}J_{p,i}^e\int_{x_{i-1}'}^{x_i'}e^{-ik_ex'\sin\theta\cos\varphi}dx'\right], \quad (7)$$
$$D_\varphi(\theta,\varphi)=\frac{k_e}{4\pi}\left[i\sin\theta\sum_{i=1}^{N_d}J_{d,i}^m\int_{z_{i-1}'}^{z_i'}e^{-ik_ez'\cos\theta}dz'-\sin\varphi e^{-ik_ez_0\cos\theta}\sum_{i=1}^{N_p}J_{p,i}^e\int_{x_{i-1}'}^{x_i'}e^{-ik_ex'\sin\theta\cos\varphi}dx'\right],$$

где z_0 – соответствующая координата середины осевой линии идеально проводящего цилиндра; θ и φ – общепринятые угловые сферические координаты; элементы токов $J_{d,i}^e$, $J_{d,i}^m$ ($i=\overline{1,N_d}$), $J_{p,i}^e$ ($i=\overline{1,N_p}$), являющиеся кусочно-постоянной аппроксимацией осевых токов цилиндров (рис. 1), находятся из системы линейных алгебраических уравнений

$$\left[\vec{n}_i^{j_d},\left(\vec{E}_i^{j_d}-\vec{E}_e^{j_d}\right)\right]=\left[\vec{n}_i^{j_d},\vec{E}_0^{j_d}\right],\ \left[\vec{n}_i^{j_d},\left(\vec{H}_i^{j_d}-\vec{H}_e^{j_d}\right)\right]=\left[\vec{n}_i^{j_d},\vec{H}_0^{j_d}\right],\ j_d=\overline{1,L_d},$$
$$E_{e,x}^{j_p}=-E_{0,x}^{j_p},\ j_p=\overline{1,L_p}, \quad (8)$$

где j_d – точки коллокации на поверхности S_d диэлектрического цилиндра, их количество равно L_d; j_p – точки коллокации на поверхности S_p идеально проводящего цилиндра, их количество равно L_p; $\vec{n}_i^{j_d}$ – единичный вектор нормали в точках коллокации j_d; $\vec{E}_e^{j_d},\vec{H}_e^{j_d}$, $\vec{E}_i^{j_d},\vec{H}_i^{j_d}$ и $\vec{E}_0^{j_d},\vec{H}_0^{j_d}$ – компоненты внешнего, внутреннего и возбуждающего полей в этих же точках; $E_{e,x}^{j_p},E_{0,x}^{j_p}$ – ориентированные вдоль оси x составляющие рассеянного и возбуждающего полей в точках коллокации j_p на поверхности идеально проводящего цилиндра. Для вычисления компонент векторов $\vec{E}_e^{j_d},\vec{H}_e^{j_d}$, $\vec{E}_i^{j_d},\vec{H}_i^{j_d}$, а также компоненты $E_{e,x}^{j_p}$ использованы выражения (6) и (7) работы [10].

2. Описание компьютерной программы

Изложенные выше соотношения реализованы в виде программы для расчета компонент рассеянного поля и контроля точности получаемых результатов по критерию невязки граничных условий. Программа написана в среде Mathcad14, блок-схема программы представлена на рис. 2–4.

В блоке 1 вводятся геометрические параметры структуры и параметры метода, а также строится система (8). К геометрическим параметрам относятся параметры l_p,r_p и l_d,r_d, $\varepsilon_i'=\varepsilon_i/\varepsilon_e$, $\mu_i'=\mu_i/\mu_e$, характеризующие идеально проводящий и диэлектрический цилиндры, расстояние Δz между цилиндрами, а также координаты y_0,z_0 осевой линии идеально проводящего цилиндра. К параметрам метода относятся параметры, определяющие разбиение осевых линий цилиндров, количества точек коллокации на поверхностях цилиндров и другие вспомогательные параметры. Подробная схема блока 1 показана на рис. 3.

Рис. 2. Общая блок-схема компьютерной программы

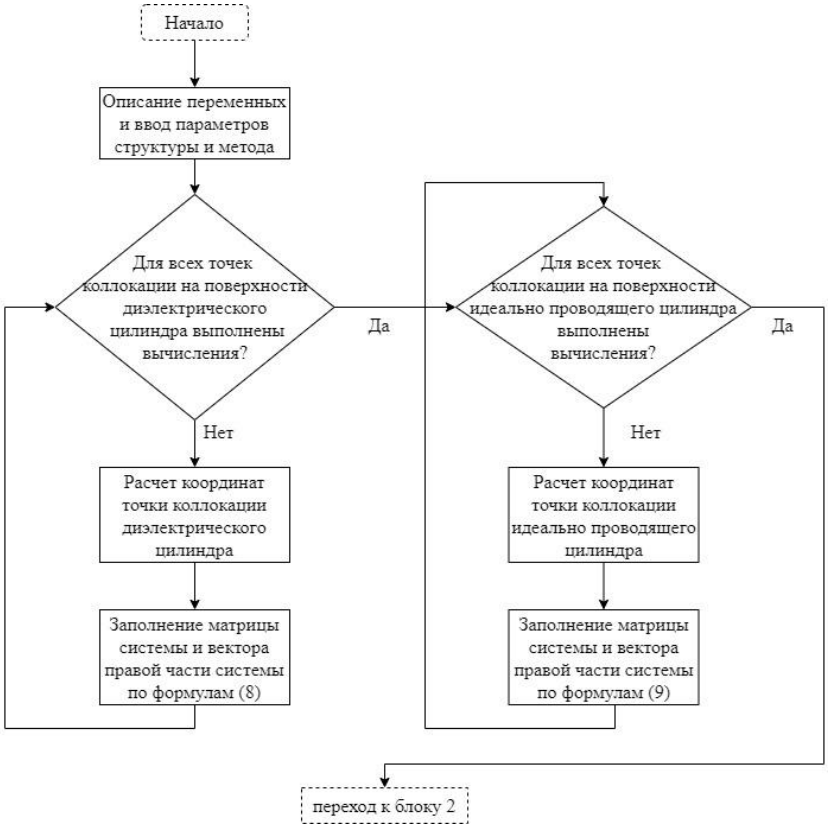


Рис. 3. Блок 1 – ввод данных и построение СЛАУ

В блоке 2 выполняется поиск решения СЛАУ методом сопряженных градиентов. Подробная схема блока 2 показана на рис. 4.

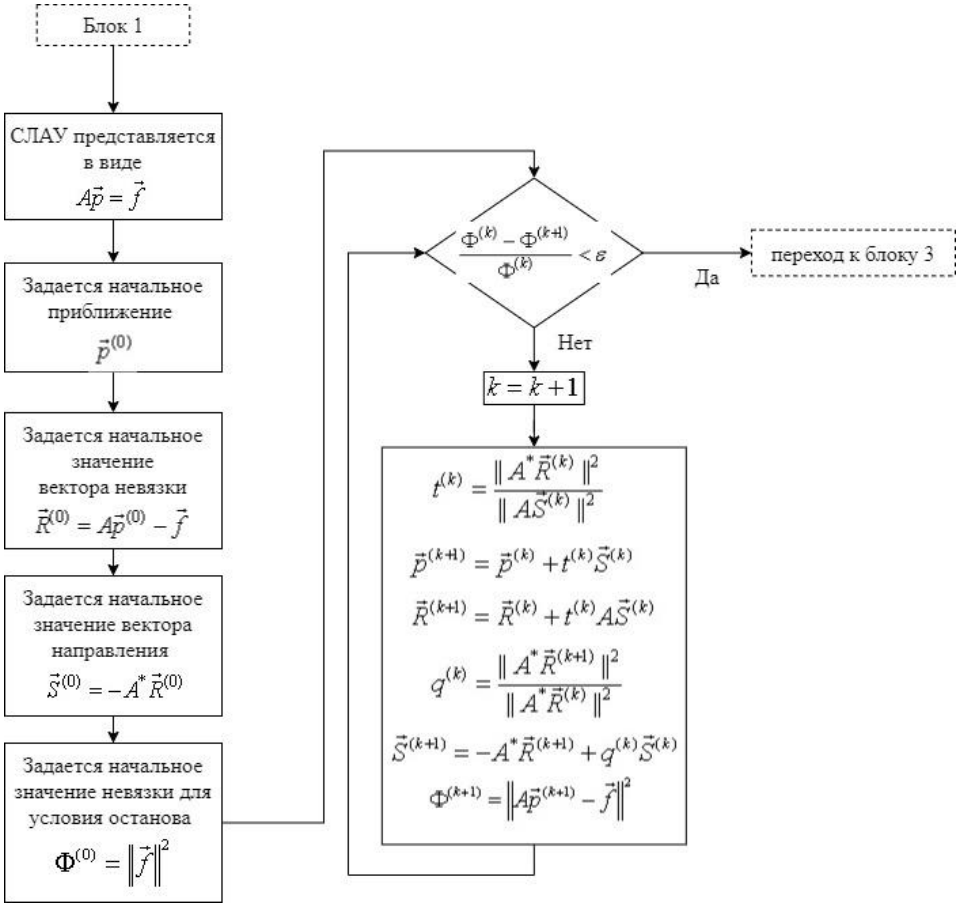


Рис. 4. Блок 2 – решение системы

Блок 3 осуществляет вычисление компонент диаграммы рассеяния по формулам (7), а также бистатических сечений рассеяния по формуле

$$\frac{\sigma}{\lambda^2} = \frac{1}{\pi}\left(\left|D_\theta(\theta,\varphi)\right|^2 + \left|D_\varphi(\theta,\varphi)\right|^2\right). \qquad (9)$$

Результаты расчетов представляются на экране в виде таблиц и графиков.

Блок 4 осуществляет контроль точности решения задачи по величине относительной нормы невязки граничных условий $\Delta = \sqrt{\frac{\Phi'}{\Phi_0}}$, где Φ' – норма невязки системы (8) в точках, промежуточных по отношению к точкам коллокации, а Φ_0 – норма правых частей системы (8) в этих же точках.

3. Численные результаты

Разработанная программа была использована для сравнения получаемых с ее помощью результатов с известными результатами, а также для расчета сечений рассеяния (9) конкретных структур. Ниже приведены некоторые результаты этих исследований.

Рис. 5 иллюстрирует сравнение результатов расчета, выполненных предлагаемым методом, с результатами других авторов. Поскольку решенная нами задача другими авторами ранее не решалась, мы не имеем возможности сравнить наши результаты с такими же результатами других авторов для рассматриваемой задачи. Однако при малых значениях относительной диэлектрической проницаемости диэлектрического цилиндра получаемые нами результаты должны быть очень близки соответствующим ре-

зультатам для одиночного идеально проводящего цилиндра. Последняя задача является решенной, и результаты ее решения содержатся, в частности, в книге [4]. Это обстоятельство и было использовано для контроля правильности получаемых нами результатов. По оси абсцисс на рис. 5 отложен угол θ в градусах (см. рис. 1), по оси ординат – сечение рассеяния, нормированное на квадрат длины волны. Кривая 1 – сечение рассеяния для одиночного идеально проводящего цилиндра, взятое из работы [4], кривая 2 – сечение рассеяния для рассматриваемой структуры при относительной диэлектрической проницаемости диэлектрического цилиндра $\varepsilon_i' = 4$ в *E*-плоскости (плоскости *xOz*).

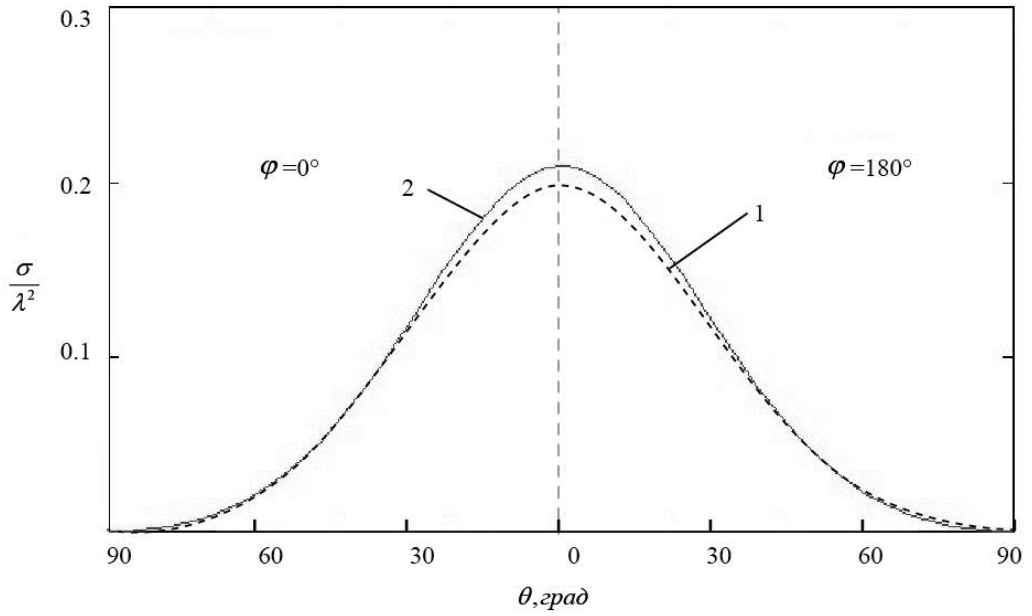


Рис. 5. Сравнение бистатических сечений рассеяния, полученных предлагаемым методом, с сечениями рассеяния, представленными в работе [4]. Кривая 1 – сечение рассеяния, взятое из работы [4], кривая 2 – сечение рассеяния, полученное предлагаемым методом при $\varepsilon_i' = 4$.

При получении кривой 2 длина и радиус идеально проводящего цилиндра выбраны такими же, как в работе [4]: $k_e l_p = 4.71$ ($l_p = 0.75\lambda$), $k_e r_p = 0.03$; длина диэлектрического цилиндра $k_e l_d$ выбрана также равной 4.71; радиус $k_e r_d$ диэлектрического цилиндра выбран равным 0.1; расстояние между цилиндрами $k_e \Delta z = 3$ ($\Delta z = 0.48\lambda$). Предполагается, что структура возбуждается плоской волной, падающей вдоль оси *z*, вектор $\vec{E}_0$ падающей волны направлен вдоль оси *x*.

Как показывает рис. 5, результаты, полученные предложенным методом, близки к результатам работы [4]. Это говорит как о правильности самого метода решения задачи, так и о правильности выбора параметров метода.

На рис. 6 показаны полученные в результате выполненных расчетов зависимости сечений обратного рассеяния исследуемой структуры от угла ψ падения плоской волны при различных длинах идеально проводящего цилиндра. Параметры диэлектрического цилиндра предполагались фиксированными и равными $\varepsilon_i' = 20$, $k_e l_d = 4.71$, радиусы цилиндров были выбраны одинаковыми $k_e r_d = k_e r_p = 0.1$, расстояние между цилиндрами было взято $k_e \Delta z = 3$. Плоская волна падает на структуру так, что векторы $\vec{k}_e$ и $\vec{E}_0$ лежат в плоскости *xz*, ψ – угол между положительным направлением оси *z* и вектором $\vec{k}_e$ (см. рис. 1). Угол падения отложен на оси абсцисс, а по оси ординат отложено сече-

ние обратного рассеяния, нормированное на квадрат длины волны. Кривая 1 на рис. 6 относится к случаю, когда длина идеально проводящего цилиндра $k_e l_p = 3.14$ ($l_p = 0.5\lambda$), кривая 2 – к случаю, когда $k_e l_p = 4.71$ ($l_p = 0.75\lambda$), кривая 3 – к случаю, когда $k_e l_p = 6.28$ ($l_p = \lambda$).

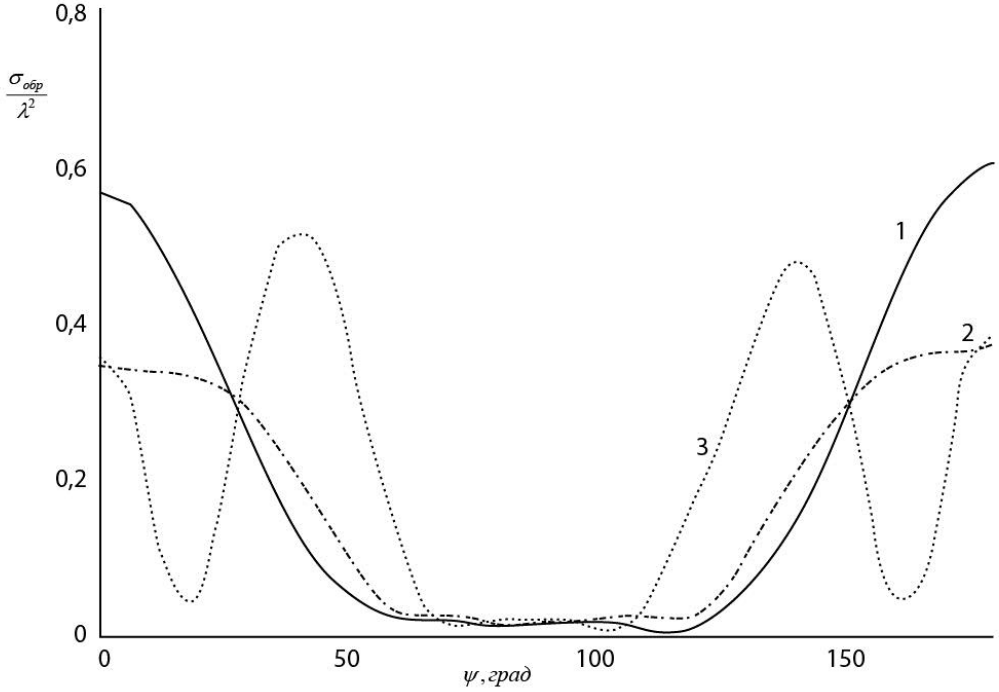


Рис. 6. Сечения обратного рассеяния при различных значениях длины идеально проводящего цилиндра. Кривая 1 соответствует длине $k_e l_p = 3.14$, кривая 2 – длине $k_e l_p = 4.71$, кривая 3 – длине $k_e l_p = 6.28$.

Заключение

Таким образом, в данной работе представлены алгоритм и программа расчета электромагнитного рассеяния на структуре, состоящей из двух ортогональных тонких цилиндров, один из которых является диэлектрическим, а другой – идеально проводящим. Созданная программа позволяет определять характеристики рассеяния широкого класса структур, отличающихся взаимным расположением цилиндров и их геометрическими и электродинамическими параметрами. Подробно исследована одна из возможных структур. Осуществлено сравнение полученных результатов с известными, исследована зависимость сечений обратного рассеяния от длины идеально проводящего цилиндра.

ЛИТЕРАТУРА

1. *Tavis M.T.* Total radar cross section of thin wires at all angles of incidence // Journal of Applied Physics. 1975. Vol. 46, No. 7. P. 3213-3215.

2. *Hatamzadeh-Varmazyar S.* New numerical method for determining the scattered electromagnetic fields from thin wires // Progress in Electromagnetic Research B. 2008. Vol. 3. P. 207-218.

3. *Bogerd J.C., Tijhuis A.G., Klaasen J.J.A.* Electromagnetic excitation of a thin wire: a travelling – wave approach // IEEE Transactions on Antennas and Propagation. 1998. Vol. 46, No. 8. P. 1202-1211.

4. *Bowman J.J., Senior T.B.A., Uslengli P.L.E.* Electromagnetic and acoustic scattering by simple shapes. Amsterdam: North-Holland Publ. Company, 1969. P.488.

5. *Yan W.Z., Du Y., Wu H., Liu D.W.* EM scattering from a long dielectric circular cylinder // Progress in Electromagnetics Research, PIER 85. 2008. P. 39-67.

6. *Santalla del Rio V., Abalde-Lima L., Christodoulou C.G.* Electromagnetic scattering from vegetation cylindrical components // IEEE Geoscience and Remove Sensing Letters. 2015. Vol. 12, No. 4. P. 751-755.

7. *Sarabandi K., Senior T.B.A.* Low-frequency scattering from cylindrical structures at oblique incidence // IEEE Transactions on GRS. 1990. Vol. 28, No. 5. P. 879-885.

8. *Дмитренко А.Г.* Численный метод исследования электромагнитного рассеяния структурами, содержащими тонкие проводники / Дмитренко А.Г., Колчин В.А. // Радиотехника и электроника. 2003. Том. 48, № 5. С. 545-551.

9. *Дмитренко А.Г.* Решение задачи электромагнитного рассеяния на тонком диэлектрическом цилиндре методом вспомогательных источников / Дмитренко А.Г., Гольцварг Е.П. // Радиотехника и электроника. 2011. Том.56, № 5. С. 600-607.

10. *Дмитренко А.Г.* Численный метод решения задачи электромагнитного рассеяния на тонких параллельных идеально проводящем и диэлектрическом цилиндрах / Дмитренко А.Г., Балашова О.М. // Материалы VI Международной молодежной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем». Томск, 24-26 мая 2018 г.

11. *Дмитренко А.Г.* Программное обеспечение для решения задачи электромагнитного рассеяния на тонких параллельных идеально проводящем и диэлектрическом цилиндрах / Дмитренко А.Г., Балашова О.М. // Материалы VII Международной молодежной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем». Томск, 23-25 мая 2019 г.

UX В НОВЫХ КАНАЛАХ ВЗАИМОДЕЙСТВИЯ С ПРИЛОЖЕНИЕМ: ГОЛОСОВОЕ УПРАВЛЕНИЕ И УПРАВЛЕНИЕ ЧЕРЕЗ ЧАТ

Киреев Д.А., Литвинова Н.И., Попов Н.С., Морозова А.С., Шкуркин А.С.

Томский государственный университет

dima30051999@mail.ru, natgraham1864@gmail.com, popovns99@mail.ru

Введение

В настоящее время наблюдается рост популярности голосового взаимодействия с приложениями среди пользователей. Голосовые ассистенты от ряда крупных компаний всё сильнее закрепляются в жизни массового потребителя. Смарт-динамики или умные колонки, позволяющие упростить управление системами умного дома, сильнее подогрели интерес людей к голосовым помощникам.

1 Постановка задачи

Рассмотрим функционал банковских голосовых помощников, так как они расширяют потенциальный функционал ассистента в областях, связанных с финансами. Одной из основных проблем работы пользователя с ассистентами является скудный набор команд, которые они могут выполнять.

Задачей работы является рассмотрение имеющихся решений на рынке голосовых банковских помощников, а также расширение спектра возможного взаимодействия пользователя с приложениями, создание ряда паттернов для управления и навигации при помощи управления голосом и командами через чат.

2. Направления развития

Рассмотрим три области, что выделяются среди вариантов развития данного направления [1]. Первая из них, это финансовая информация. На данный момент, огромное количество финансовых учреждений становятся источником информации о рынке или перспективных фирмах. Яркий этому пример банк Morgan Stanley, который создал голосовое приложение, что позволяет клиентам легко получить доступ к их информации о рынке.

Вторая из рассматриваемых областей – банковское дело. Эта область является одной из наиболее распространенных для использования голосовых технологий в финансах. Сюда относятся: осуществление банковских операций, проверка остатка на счете, получение платежной информации, оплата счетов и т.д. Ярким примером данного направления развития является Erica от Bank of America [2], имеющая широкий спектр функций для помощи клиенту в работе с личными финансами.

И заключительная область – консультативная. В финансовой индустрии голосовые ассистенты начали использоваться для автоматизации задач обслуживания клиента и

повышения удобства. Замена операторов в колл-центрах, ответ пользователю на предусмотренные темы и вопросы, предоставление интересующей информации. Это сокращает затраты банка, а также увеличивает скорость принятия входящих заявок от клиентов и их обработку. Голосовой ассистент способен быстро и эффективно решить возникший вопрос. На случай, если потребитель не знает, как начать диалог с ботом, разработчики обучают искусственный интеллект задавать наводящие вопросы, работать в активном режиме, предугадывая желания пользователя, или рекомендуя потенциально интересные для него предложения. Из-за скудных, на данный момент, возможностей финансовых ассистентов, бот может не распознать запрос клиента. В таком случае голосовой помощник переводит обратившегося клиента на сотрудника call-центра.

Несмотря на большой прорыв в развитии голосовых помощников, эта технология все еще имеет скудный набор функционала.

3. Исследование потребностей потенциальных пользователей

Для глубокого понимания потребностей пользователей, в работе было проведено дневниковое исследование. Тридцать человек на протяжении недели записывали свои возникающие запросы, с которыми они сталкивались ежедневно, к гипотетически идеальному голосовому ассистенту.

В результате был получен внушительный объём запросов – от простых запросов о погоде, до сложных, требующих обработки крупного пласта информации.

Классификация запросов по их сложности:

- Простейшие запросы – в них входят вопросы о времени, погоде, просьба установить напоминание или будильник.
- Многошаговые запросы – требуют несколько действий для исполнения, включая взаимодействие с другими приложениями. Сюда входят: поиск места на карте, денежные переводы, покупка билетов, осуществление звонков, отправка сообщений, запуск и конфигурация приложений.
- Аналитические запросы – их успешное исполнение подразумевает обработку большого объёма данных, аналитику и ответ, основанный на полученной информации. К данной категории относятся: сбор информации об аккаунте, расчет бюджета на выбранный определённый период, составление списка покупок на основе продуктов в холодильнике.

На первый взгляд, в разных группах имеются схожие запросы, но участники исследования добавляли различные уточнения, которые указывали на определенную классификацию и усложняли работу ассистенту. Пользователь может просто попросить заказать пиццу. Но если он захочет заказать самую дешёвую, с определёнными ингредиентами, и в пиццерии с хорошими отзывами, то ассистенту придётся провести небольшое исследование, чтобы успешно выполнить запрос.

На основе результатов исследования был составлен перечень типов запросов, расположенных в зависимости от их популярности (табл. 1).

Таблица 1

Перечень типов запросов

Напоминание	Напоминание о событиях, подписках, выплатах со сроком, расписание.	89%
Вывод информации	Вывод интересующей информации: запрос в поисковик, заметки, новости.	69%
Транзакции	Финансовые операции как заказ, бронирование, оплата и т.д.	69%
Общение	Звонок, сообщение	55%
Сбор информации	Формирование текстового файла, содержащего интересующую информацию об аккаунте. (Пример: выписка по счету, последние транзакции)	48%
Заметки	Запись идей, составление списка.	46%
Управление устройством	Запуск приложения на устройстве, обновление, перезагрузка.	12%
Интернет вещей	Взаимодействие с другим устройством	16%

Помимо поисковых запросов, установки будильника и вопросов о погоде, был зафиксирован интерес пользователей в финансовой сфере, а также интернете вещей. Ожидаемо среди запросов участников доминируют напоминания. Поисковые запросы оказались в равной степени востребованы, как и простые операции, связанные с финансовой сферой. Далее, с относительно небольшим разрывом, идут запросы на коммуникацию с другими пользователями, сбор определенной информации, такой как отчет о выплатах, тратах или статистика пользователя, и заметки. Наименее популярными оказались запросы, связанные с управлением устройством и интернет вещей, что возможно обусловлено отсутствием предметов быта для управления.

4. Функциональные требования и нефункциональные

На основе анализа имеющихся решений на рынке голосовых банковских ассистентов, и результата дневникового исследования, были определены следующие функциональные требования. Они предлагаются как основа для проектирования конкурентоспособного голосового помощника не только в финансовой сфере.

Функциональные требования:

1. Приложение должно предоставить возможность ведения естественного диалога с пользователем в текстовых и голосовых каналах.
2. Приложение должно определять тематику обращения.
3. Ассистент должен поддерживать контекст диалога.
4. Приложение должно содержать удобный интерфейс администрирования.
5. Наличие уведомлений о возможных тратах, благодаря прогнозной аналитике и обработке массивных пластов информации о пользователе.
6. Приложение должно оповещать о выплатах по кредиту, ипотеке и др.
7. Напоминания о повторяющихся или просроченных счетах/выплатах различного рода.
8. Приложение должно уточнять информацию у пользователя в случае неудачи в распознавании запроса.
9. Сбор информации об аккаунте для возможности предоставления отчетов о последних выплатах или расчётах.
10. Определение неожиданно высоких трат, благодаря детекции аномалий.
11. Блокировка/разблокировка карт по запросу пользователя.
12. Совершение покупок, с подтверждением оплаты от пользователя, голосовыми/текстовыми командами.

Помимо функциональных требований, также были выявлены и нефункциональные:

1. Приложение должно поддерживать интеграцию с телефонным каналом.
2. Интеграция с web-чатами.
3. Приложение всегда должно выдавать ответ на запрос пользователя.

Заключение

Если говорить о перспективе развития UX во взаимодействии с приложением через голосовое управление и чат, то цифровые ассистенты станут наиболее подходящей сферой для их реализации. Одной из основных проблем, которая была выделена при анализе предметной области - скудный набор функций, которые они могут выполнять.

При проектировании следует делать упор на обработку сложных команд с учётом множества информации, такой как контекст и данные о пользователе. Ассистент должен работать в активном режиме, зачастую предугадывая потребности пользователя, предлагать свой функционал. Потенциальное множество запросов гораздо обширнее, чем реализованное на данный момент.

В результате работы был проведен анализ рынка финансовых голосовых помощников. На основе анализа были выделены одни из лучших приложений, на которые стоит ориентироваться при создании собственного приложения [3,4].

Также было проведено исследование, которое заключалось в определении ежедневных потребностей потенциальных пользователей. В ходе данного исследования были выделены наиболее актуальные направления для разработки паттернов взаимодействия через голосовое управление или управлением через чат.

ЛИТЕРАТУРА

1. *Matt Lang*. How Voice Assistance Will Change The Financial Services [Электронный ресурс] URL: https://medium.com/voice-tech-podcast/how-voice-assistance-will-change-the-financial-services-industry-6448d060e486 (дата обращения: 20.08.2020)

2. Голосовой ассистент Erica: [Электронный ресурс] // Bank of America. URL: https://promo.bankofamerica.com/erica/ (дата обращения: 20.08.2020).

3. *Киреев Д.А., Литвинова Н.И., Попов Н.С.* UX в новых каналах взаимодействия с приложением: голосовое управление и управление через чат // Информационные технологии: Материалы 58-й Междунар. науч. студ. конф. 10–13 апреля 2020 г. / Новосиб. гос. ун-т. – Новосибирск : ИПЦ НГУ, 2020. С. 66.

4. *Rodionov A., Shkurkin A.* Multiplayer Blackjack //Труды Томского государственного университета. – Т. 304. Серия физико-математическая: Математическое и программное обеспечение информационных, технических и экономических систем: материалы VII Междунар. молодежной науч. конф. Томск, 23-25 мая 2019 г. Томск: Издательский Дом Томского государственного университета, 2019. С. 342-345.

РАЗРАБОТКА ГРАФИЧЕСКОГО КРОССПЛАТФОРМЕННОГО ПРИЛОЖЕНИЯ «UNIGAME»

Лихоманов Т.Д., Безходарнов Н.И., Буторина Н.Б.
Томский государственный университет
lihomanov.t@mail.ru, nnatta07@mail.ru

Введение

В настоящее время многие люди очень загружены работой, и у них возникает потребность время от времени отдыхать, переключать своё внимание. Этому способствует несложные и недолгие игры. Но по сей день на компьютеры таких приложений почти нет, что создаёт неудобства. По этой причине было решено реализовать собственное приложение, которое позволяло бы отвлечься от повседневных забот.

1. Постановка задачи

Целью работы является реализовать графическое приложение для компьютеров, являющееся небольшой развивающей игрой.

Требования к игре должны быть следующими: игра должна запускаться на компьютерах, она должна быть несложной, должна быть возможность закрытия игры в любой момент.

Для реализации данного приложения были поставлены следующие подзадачи:

1. Принять концепцию игры, которая бы удовлетворяла требованиям.
2. Разработать удобный интерфейс приложения для взаимодействия с пользователем.
3. Реализовать игру в виде компьютерной программы.

Для выполнения поставленной задачи было решено воспользоваться концепцией игры под платформу Android – «Пифагория». Она заключается в том, что пользователю даётся размеченная область для построения отрезков, на котором требуется решать какие-либо геометрические задания. Такая концепция была выбрана по той причине, что её можно реализовать по поставленным ранее требованиям.

2. Варианты построения приложений

После того, как концепция была принята, возникает вопрос о реалицазии в виде компьютерной программы. Прежде всего, перед каждым разработчиком стоит выбор: разрабатывать приложение только под одну платформу или делать его кроссплатформенным. На сегодняшний день существует немало платформ, под которыми работают

устройства. Из самых известных можно выделить следующие: Windows, Linux, MacOS, Android, IOS. Из-за большого количества таковых данный вопрос является актуальным.

Чтобы разрешить данный вопрос, рассмотрим плюсы и минусы каждого подхода. Каждый вариант по-разному сказывается на ситуациях разработки приложения, всегда существуют нюансы в выборе способа разработки приложения. Например, приложение, написанное непосредственно только для Windows будет сложно адаптировать под другие операционные системы, хоть и является возможным.

Рассмотрим подход разработки под одну платформу, или же нативный подход.

Нативный подход – это разработка приложений под конкретную аппаратно-программную платформу. В основном эти приложения пишутся на языках, специально созданных для данной платформы. Такой подход позволяет выполнить оптимизацию приложения для получения более стабильной и быстрой работы, что является существенным достоинством для разработки больших и сложных программ. Из минусов можно выделить то, что перенос реализованного приложения под другие платформы будет очень трудоёмким [1].

Теперь нужно рассмотреть подход разработки под несколько платформ, или же кросплатформенный подход.

Кроссплатформенный подход предполагает унификацию большей части программного кода приложения, его независимость от последующего выбора платформы. Как правило, под кроссплатформенной разработкой подразумевают использование специальных фреймворков, которые представляют собой реализации основных команд для взаимодействия с компьютером для разных платформ. Этот инструмент способен, в большинстве случаев, сделать приложение независимым от выбора платформы, но в случае добавления в программу платформо-зависимых компонентов он утратит эту способность. Плюсом этого подхода является возможность быстрого и лёгкого переноса приложения под разную платформу, что позволяет охватить большее количество устройств. Из минусов можно выделить следующее: для разработки такого приложения все компоненты не должны иметь зависимости от какой-либо платформы, большая сложность оптимизации приложений [1].

Рассмотрев все предложенные походы, было решено использовать кроссплатформенный. Так как требуется разработать несложную игру, то минусы нативной разработки являются несущественными. При этом, возможность создания приложения под разные платформы позволяет запускать его на разных устройствах, что делает его доступнее. Такая доступность является очень большим плюсом для нашей игры.

3. Фреймворк QT

После того, как был выбран кроссплатформенный подход, требуется выбрать подходящий инструментарий. Как говорилось ранее, выбранный подход подразумевает собой использование фреймворков. Прежде, нужно сказать, что было решено реализовывать приложение на языке C++, так как он обеспечивает достаточно быструю работу приложений и допускает разработку кроссплатформенных программ [2]. Следовательно, дальнейшим шагом является выбор такового инструмента.

Фреймворк – это программная платформа, которая определяет структуру программной системы. Облегчает разработку и объединение разных компонентов большого программного проекта.

На сегодняшний день существует множество фреймворков для разработки программ, но было решено остановиться на QT.

QT – это библиотека для создания кроссплатформенных графических приложений на языке программирования C++, при этом она поддерживает и другие языки программирования. Он направлен на быстрое прототипирование и разработку кроссплатформенных приложений [3].

Плюсы данного фреймворка:

- Существует множество хороших инструментов для разработки приложений с использованием библиотеки QT. Например, IDE QT Creator, QT Designer, QT Linguist и QT Assistant.
- QT содержит интуитивно понятные интерфейсы и классы для разработки. К тому же, в ней присутствуют мощные инструменты для работы с графикой. К примеру, анимация, работа с сетями и т.д.

Недостаток данного фреймворка:

- В QT очень сложен входной порог для начинающих. Чтобы с уверенностью использовать фреймворк QT, нужно знать и понимать ключевой механизм взаимодействия объектов – сигналы и слоты.

В момент написания данной работы у нас уже был опыт работы с данной библиотекой, поэтому недостаток никак не оказывает влияния на разработку приложения.

4. Разработка уровней

Теперь, когда был выбран весь необходимый инструментарий, требовалось начать разработку самого приложения, а прежде всего уровней.

Так как концепция игры основана на геометрии, то, соответственно, нужно было связать уровни игры с данной областью [4]. Прежде всего, любые фигуры строятся из отрезков, поэтому было решено создать задачи именно для них. Задачи на тему отрезков заключается в том, чтобы разделить некоторый отрезок на определённые части, причём у пользователя ограничениями являются точки, по которым можно строить разделительные отрезки. После отрезков было решено придумать задачи на треугольники, так как они обладают множеством свойств и классификаций. Примером задач на эту тему является построение равнобедренного треугольника, имея в распоряжении только определённые точки:

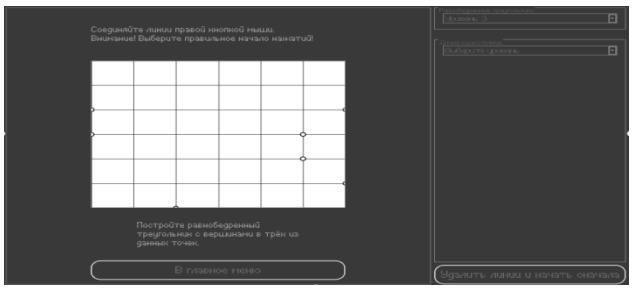


Рис. 1. Окно приложения

После разработки концепций уровней требуется создать их реализацию. Сначала нужно было организовать отрисовку уровней на компьютере. Так как приложение у нас небольшое, то было решено отрисовывать их в центральном окне. Ещё в правой части окна был создан выпадающий список, где можно выбрать уровень.

При выборе уровня происходит следующее:

1. Очистка игрового окна.
2. Отрисовка матрицы (сетка в игровом окне).
3. Инициализация нужного уровня и логики.

Также существует дополнительная кнопка в игровом окне, которая отображается не для всех уровней, а только для тех, которым требуется принудительная очистка из-за сложности прохождения уровня. Вид окна для такого уровня отображён на рис. 1.

В нижней части центрального окна появляется текст с заданием уровня, а в верхней части подсказка для прохождения уровня (подсказываем способ взаимодействия пользователя с игрой).

Уровень начинается со следующей функции:

```
void Level_Triangle_3::startLevel()
{
    // Сразу показываем подсказку / инструкцию к уровню
    this->showHint();

    // Показываем задание
    this->showTooltip();

    // Отрисовываем уровень
    this->paintLevel();
}
```

Рис. 2. Исходный код запуска уровня

Самой отрисовкой точек для уровня занимается функция *«paintLevel»*. В этой функции существует массив точек, который заполняется следующим образом:

```
for (qint32 i = 0; i < 7; ++i)
{
    if (i == 0)
    {
        this->paintPoint(QPoint(0, height * 2));
        points.push_back(QPoint(0, height * 3));
    }

    if (i == 2)
        points.push_back(QPoint(width * i, height * 6));

    if (i == 5)
    {
        this->paintPoint(QPoint(width * i, height * 3));
        points.push_back(QPoint(width * i, height * 4));
    }

    if (i == 6)
    {
        this->paintPoint(QPoint(width * 6, height * 2));
        this->paintPoint(QPoint(width * 6, height * 5));
    }
}
```

Рис. 3. Исходный код заполнения уровня

Мы вычисляем нужные места в матрице, где нужно произвести отрисовку точек с помощью функции *«paintPoint»*.

Самая интересная часть та, когда пользователь рисует сами линии. Обработка нажатий клавиш мыши происходит в двух слотах:

```
void Level_Triangle_3::paintPointOnGraphicView(QMouseEvent* event)
{
    double rad = 5;
    _scene->addEllipse(QRectF(event->x() - rad, event->y() - rad, rad * 2.0, rad * 2.0), QPen(), QBrush(Qt::yellow));

    _previousPos = event->pos();
}

void Level_Triangle_3::paintLineOnGraphicView(QMouseEvent* event)
{
    if (event->button() == Qt::MouseButton::RightButton)
    {
        if (_previousPos != QPoint())
            _scene->addLine(QLineF(QPointF(_previousPos), QPointF(event->pos())), QPen(QBrush(Qt::yellow), 3));

        _previousPos = event->pos();
    }
}
```

Рис. 4. Исходный код рендера отрезков пользователя

Первый слот добавляет на сцену окружность и сохраняет место нажатия. Второй слот занимается непосредственно отрисовкой линии на сцене. Отрисовывается это следующим образом:

Рис. 5. Отрисованные отрезки пользователя

Здесь можно заметить правильное решение поставленной задачи, а именно те отрезки, которые образуют равнобедренный треугольник. Для проверки правильности решения, пользователь должен нажать клавишу «Space» на клавиатуре. Для выбранного уровня вызывается функция проверки, которая проходится по всем точкам и проверяет, связывают ли линии нужные точки на сцене. В случае успеха функция вызывает завершение уровня, которая изменяет статус уровня в статистике на «Пройден».

Таким образом были организованы уровни в нашем приложении.

5. Интерфейс приложения

После разработки основной части игры, а именно уровней, требуется разработать интерфейс взаимодействия пользователя с программой. Как известно, практически все хорошие игры начинаются с главного меню, поэтому нужно было сперва организовать именно его. Окно при запуске моего приложения принимает следующий вид:

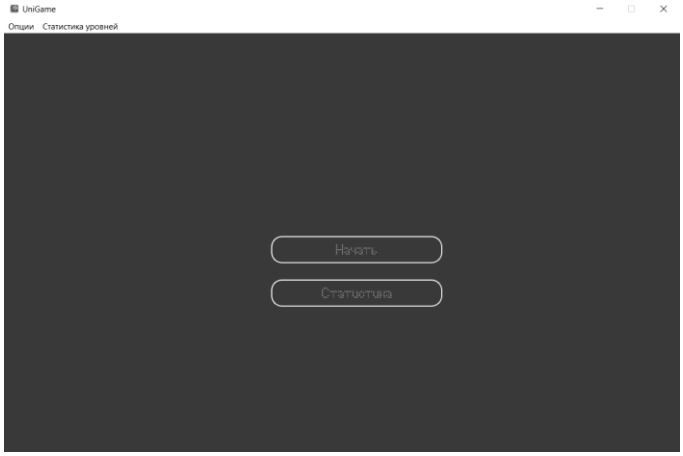


Рис. 6. Главное меню

В главном меню находится 2 кнопки, через которые можно управлять приложением. Кнопка «Начать» позволяет пользователю перейти непосредственно в основную игровую область, где находятся игровые уровни, окно отображения и т.д. Кнопка «Статистика» имеет свойство отображения пройденных уровней в отдельном окне, чтобы пользователь мог наблюдать за своими результатами (сколько уровней он прошел и какие уровни у него ещё не пройдены).

Также в этом окне существует панель инструментов, которое находится сверху приложения. В области этой панели находится 2 вкладки, которые, в свою очередь, являются тоже некими вариациями кнопок, через которые пользователь взаимодействует с приложением.

По нажатию на графу «Статистика уровней» перед пользователем появляется дополнительное меню с двумя кнопками:

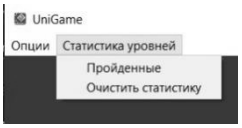


Рис. 7. Панель инструментов, подменю «Статистика уровней»

Кнопка *«Пройденные»* открывает перед пользователем окно с информацией о пройденных и не завершённых уровнях. В этом окне находятся три столбца:

1. В первом столбце находится имя раздела геометрии, по которой создана задача.
2. Во втором столбце находится номер уровня.
3. Для третьего столбца устанавливается статус, пройден уровень или нет. Если пользователь успешно прошёл уровень в разделе, то его статус будет выделен зелёным цветом, в ином случае, когда он ещё не прошёл уровень, клетка статуса будет выделена красным цветом.

Рис. 8. Окно статистики уровней

Вся статистика уровней хранится в файле на компьютере и, в случае необходимости, подгружается в оперативную память.

Кнопка *«Очистить статистику»* является функциональной в том смысле, что при нажатии на неё пользователь командует приложению очистить файл статистики, то есть приложение полностью обнулит статистику и все уровни будут иметь статус «Не пройден».

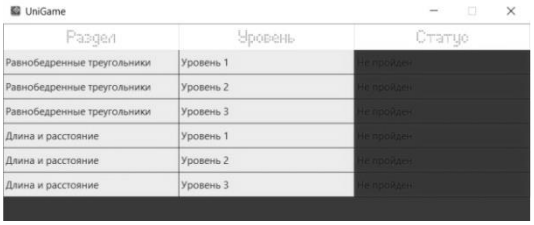


Рис. 9. Окно статистики уровней после обнуления

В интерфейсе остаётся только вид окна во время прохождения уровня, которое описано ранее и представлено на рис. 1.

Таким образом, был реализован достаточно удобный интерфейс для нашего приложения, что позволяет улучшить взаимодействие пользователя с компьютером.

Заключение

В ходе выполнения нашей работы был изучен и применён кросплатформенный способ разработки приложения на языке C++ с использованием библиотеки QT. В про-

цессе разработки возникали разные ситуации, которые приводили к ошибкам, но в итоге, все они были решены.

В результате, было создано приложение, которое позволяет отвлечься от своих забот и размять свою голову несложными задачками. В дальнейшем планируется дополнять приложение всевозможными уровнями. Надеемся, наша программа сделает наш мир лучше и комфортнее.

ЛИТЕРАТУРА

1. Artjoker. [Электронный ресурс]: Нативная или кроссплатформенная разработка? – 2020. – URL: https://artjoker.ua/ru/blog/nativnaya-ili-kross-platformennaya-razrabotka/

2. *Страуструп Б.* Язык программирования C++ // Издательство Бином. 2012 – 1136 с.

3. QT. [Электронный ресурс]: Официальная документация QT – 2020. – URL: https://doc.qt.io/

4. *Арутюнян Г.В., Марчевская Е.В., Маркчевский И.К.* Элементарная геометрия. методы решения задач // Учебная литература МГТУ им. Н.Э. Баумана. 2014. – 222 с.

СОЗДАНИЕ ОБУЧАЮЩЕЙ ПРОГРАММЫ ПО ТЕМЕ «РЕШЕНИЕ СЛАУ МЕТОДОМ ГАУССА»

Прилепова И.Д., Пахомова Е.Г.

Томский государственный университет

prilepovaid@mail.ru, peg@tpu.ru

Введение

В стремительно развивающемся мире компьютеры стали чем-то большим, чем просто электронные вычислительные машины. Они нашли применение в разных сферах человеческой деятельности: медицина, промышленность, банковское дело. Постепенно эти электронные устройства внедряются и в образование: процесс обучения вышел за рамки учебников, стал более ярким и интересным благодаря наглядности получаемой информации (презентации, видео-уроки).

Сейчас существует множество онлайн-курсов, приложений, обучающих программ для получения новой информации вне стен обучающего заведения. Однако нельзя сказать, что компьютер сможет полностью заменить преподавателя, потому что в процессе обучения важна вовлечённость обеих сторон, важно критическое мышление и анализ проделанной учеником работы не только с точки зрения правильности: недостаточно просто отточить навык, но важно понять, для чего этот навык нужен.

Как правило, изучив материал и приступая к решению практических задач, ученики совершают одни и те же типичные ошибки: часто, по невнимательности, нежели по незнанию. Однако отработка умений занимает большое количество времени как у ученика, так и у преподавателя, которому из раза в раз нужно выискивать сделанную ошибку. Поэтому для формирования и отработки практических навыков отлично подходят специальные обучающие программы, указывающие те места в решение, где чаще всего совершаются ошибки.

Обучающая программа – это система, которая обеспечивает непрерывное обучение, предоставляет теоретический материал и комплекс упражнений для закрепления изученного материала, а также осуществляет контроль деятельности ученика. Обучающие программы можно разделить на несколько типов [1]:

- тренировочные;
- наставнические;
- имитационные;
- развивающие игры.

Тренировочные программы нужны для закрепления приобретённых навыков, т.е. предполагается, что студент изучил теоретический материал. Программы этого типа предлагают учащемуся вопросы и задачи в случайной последовательности, а затем подсчитывают количество верных ответов, а также количество правильно решённых задач

(часто в случае правильного ответа может выдаваться поощряющая фраза). Если ученик даёт неправильный ответ, он может получить подсказку.

Наставнические программы предлагают студентам изучить теоретический материал, а задачи и вопросы служат для управления обучением и организации диалога между человеком и машиной. Если ученик даёт неверный ответ, программа делает “шаг назад” для повторного изучения материала.

Имитационные или моделирующие программы позволяют осуществлять своего рода эксперимент, используя графические и вычислительные возможностикомпьютера. Программы этого типа предоставляют студенту возможность наблюдать на экране некий процесс и влиять на него путём подачи команд.

Развивающие игры строят для ученика воображаемую среду, набор возможностей и средств их реализации. Использование возможностей, которые игра предоставляет ученику, для изучения мира игры развивает у ученика познавательные навыки, умение находить закономерности.

Итак, основанием для классификации обучающих программ по четырём типам служит организация учебной деятельности студентов при работе с программами. Наиболее популярными, благодаря своей относительной простоте и унификации многих частей, являются тренировочные и наставнические программы. В отличие от первых двух, имитационные программы и развивающие игры требуют работы не только программистов и специалистов в области изучаемого предмета, но также методистов и психологов [2].

1. Постановка задачи

Задача данной работы заключается в создании тренировочной обучающей программы для решений систем линейных алгебраических уравнений (СЛАУ) методом Гаусса. Процесс решения разбит на этапы. На каждом этапе программа отслеживает действия студента и в случае ошибки выводит соответствующее сообщение. Предполагается, что студент уже изучил материал и ему требуется отработать навыки, необходимых для решения СЛАУ методом Гаусса. Однако в программе предусмотрена возможность в случае необходимости обратиться к теоретическому материалу.

2. Интерфейс программы

Интерфейс разработанной программы интуитивно понятен. Это важное качество обучающей программы, от которого, как показывает анализ [3] существенно зависит качество обучающей программы. При входе в главное меню (рис. 1) студент видит приветственную надпись, а также три кнопки: “Теория” – для получения теоретической информации о методе Гаусса, “Образец решения” – для рассмотрения примеров решений систем линейных алгебраических уравнений, “Решите самостоятельно” – для непосредственного перехода к решению СЛАУ. Предполагается, что к началу работы с программой студент уже знаком с методом Гаусса и ему требуется только сформировать навык практического решения. Однако возможность повторения теоретического материала предусмотрена. Также полезно иметь решения похожих задач. Это позволит, опираясь на готовые примеры, составлять план решения своей.

Главная часть обучающей программы - «Решите самостоятельно». При нажатии на эту кнопку на экране появляется система линейных алгебраических уравнений и матрица системы с пустыми ячейками, которую студент должен заполнить соответствующими коэффициентами при неизвестных (рис. 2). После нажатия кнопки “Проверить”, программа сверяет введённую информацию с правильным ответом, и в случае ошибки студент получает просьбу о повторном заполнении матрицы, а ячейки очищаются. Характерная ошибка студентов на данном этапе - потеря знака коэффициента, что делает весь дальнейший процесс решения неверным. Именно поэтому необходим контроль на

этапе ввода данных. Если ответ введён верно, программа переходит к следующему этапу. Также студент может очистить матрицу самостоятельно, нажав кнопку “Сначала”.

Рис 1. Главное меню обучающей программы

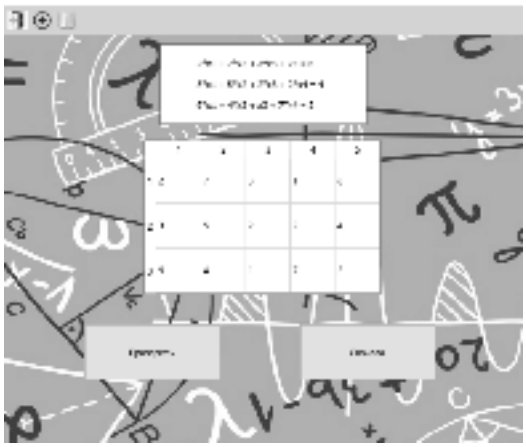


Рис. 2. Получение матрицы системы

На втором этапе решения СЛАУ (рис. 3) студенту необходимо привести заполненную ранее матрицу к ступенчатому виду, т.е. получить нули под главной диагональю путём прибавления строки, умноженной на число, к другой строке или же перестановкой строк местами. Поскольку основная задача обучающей программы – сформировать навык нахождения общего решения, выполнение арифметических действий на данном этапе перепоручаются компьютеру. Чтобы достигнуть цели данного этапа, в ходе выполнения преобразований студенту нельзя терять нули, т.е. количество нулей под главной диагональю должно либо увеличиваться, либо не изменяться, и программа отслеживает соблюдение этого условия (рис. 4). Количество действий студента не ограничено, как и время на решение любого примера из представленных в программе.

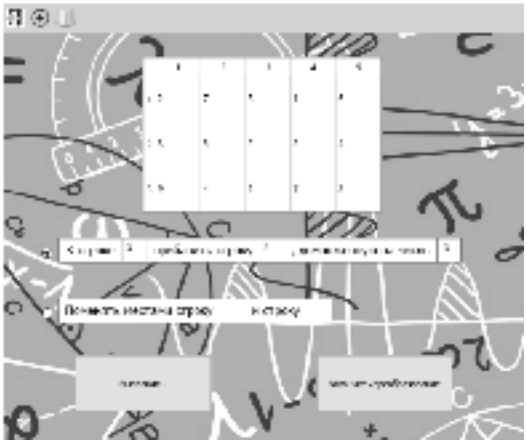


Рис. 3. Преобразование матрицы системы

Рис. 4. Предупреждение о потере нуля

Закончив преобразование матрицы, необходимо определить, совместна ли система (рис. 5). В случае несовместности системы, работа программы прекращается и студент возвращается в главное меню. Если же система совместна, студенту необходимо записать ранг матрицы системы и определить количество решений (рис. 6).

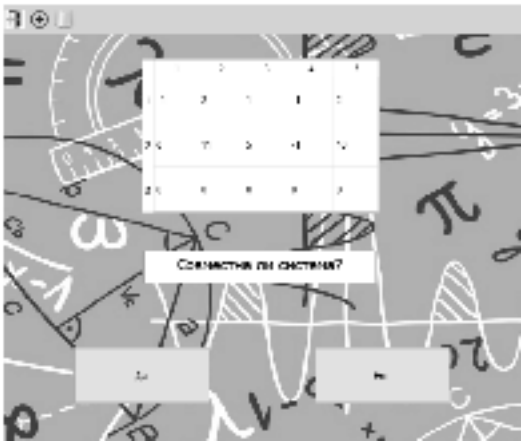


Рис. 5. Совместность системы

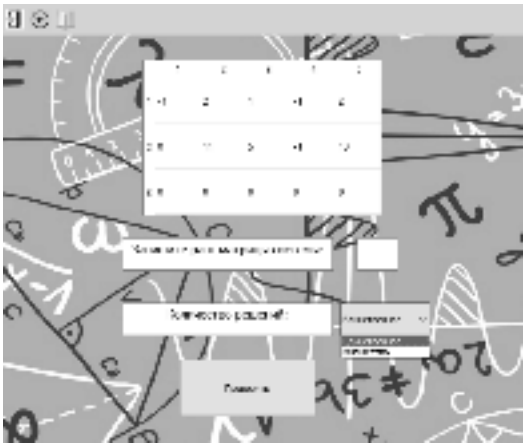


Рис. 6. Ввод ранга и количества решений

Решений может быть множество, а может – только одно, и программа предусматривает оба варианта. В случае единственного решения, появляется окно, в котором студенту необходимо ввести значения соответствующих неизвестных (рис. 7). В случае правильного ответа студент возвращается в главное меню, иначе – получает сообщение с просьбой проверить введённый ответ.

Если решений множество (рис. 8), необходимо выбрать зависимые и свободные переменные. Согласно методу Гаусса, зависимые переменные выбраны верно, когда определитель, составленный из соответствующих столбцов матрицы системы, не равен нулю и в программе предусмотрена проверка этого требования. Но недостаточно выбрать правильно зависимые переменные, важно также максимально облегчить поиск решения, поэтому выбранный определитель должен быть треугольным. В случае, когда зависимые переменные выбраны верно, но определитель не является треугольным, программа советует выбрать другие переменные, однако при желании студент может работать с уже выбранными.

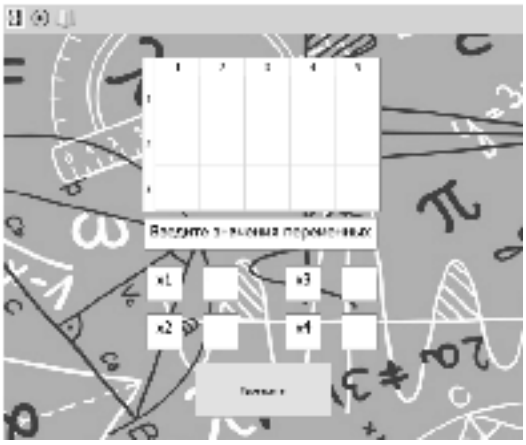


Рис. 7. Единственное решение

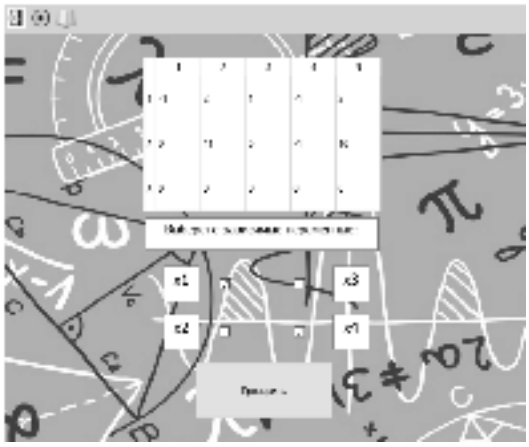


Рис. 8. Множество решений

На последнем этапе работы с программой (рис. 9) студенту необходимо самостоятельно выразить зависимые переменные через свободные. Если ответ записан верно, программа выводит соответствующее сообщение, и студент может вернуться в главное меню (рис. 10). Иначе - программа просит перепроверить решение.

В коде программы предусмотрен алгоритм, который выражает зависимые переменные через свободные так же, как это делает студент, поэтому нет необходимости создавать базу ответов для каждого примера, представленного в программе.

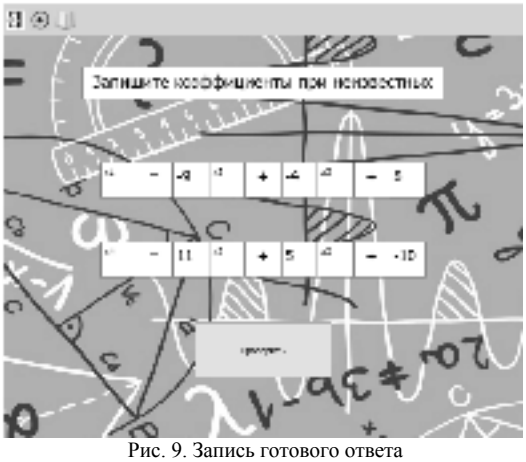


Рис. 9. Запись готового ответа

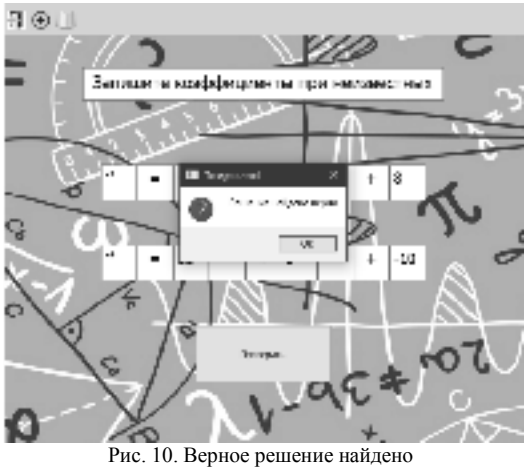


Рис. 10. Верное решение найдено

Иными словами, на всём пути программа решает систему вместе со студентом, подобно преподавателю, но заметно экономит время, которое может потратить преподаватель на поиск ошибок. Также студент учится самостоятельно работать с информацией, представленной в теоретической части, и применять её в ходе решения СЛАУ.

Заключение

Таким образом, поставленная на первом этапе задача по созданию программы, которая следит за ходом решения студентом СЛАУ и указывает на допущенные ошибки, была выполнена. В дальнейшем предполагается создать алгоритм, который генерирует систему таким образом, чтобы она была совместной и имела заранее заданный ранг. Как показали эксперименты, задание коэффициентов системы случайным образом не является эффективным, т.к. в большинстве случаев ранг системы оказывается на единицу меньше числа неизвестных, что не позволяет сформировать навыки правильного выбора зависимых и свободных неизвестных. Кроме того, возможно появление несовместной системы, что не позволяет сформировать необходимые навыки по нахождению общего решения. Более эффективным видится подход, при котором случайным образом генерируется ранг системы и исходя их этого параметра формируется сама система. Кроме того, планируется предусмотреть выбор студентом уровня сложности задачи (на этапе перехода к блоку решения задачи).

ЛИТЕРАТУРА

1. *Гефан Г.Д., Кузьмин О.В.* Методика построения контрольно-обучающих программ и их использование в преподавании математических дисциплин // Вестник Бурятского государственного университета. 2013. № 15. С. 23-28.

2. *Аязбаев Т.Л., Галагузова Т.А.* Технология создания компьютерных обучающих программ // Международный журнал экспериментального образования. 2015. – № 3 (часть 2) – С. 76-78.

3. *Афанасьев В.В., Тыщенко О.Б., Афанасьева И.В.* Анализ показателей эффективности обучающих программ // I Всероссийская научно-техническая конференция 'Компьютерные технологии в науке, проектировании и производстве'. Тезисы докладов, часть V, Нижний Новгород, 1999, 43с., стр. 15.

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА АНАЛИТИЧЕСКОЙ ПОДСИСТЕМЫ ДЛЯ ПРОГРАММНОГО КОМПЛЕКСА ПОДДЕРЖКИ РАБОТЫ ПРЕПОДАВАТЕЛЯ ВУЗА

Славянова Я.И., Лагерев Д.Г.
Брянский государственный технический университет
yanchos7@gmail.com, LagerevDG@mail.ru

Введение

Ведение учета успеваемости и посещаемости студентов вуза является неотъемлемой частью образовательного процесса в высшем учебном заведении. Зачастую это предполагает ведение преподавателем бумажных лекционных и лабораторных карточек групп. Существует ряд программных средств, позволяющих автоматизировать данный процесс [1,2]. Одним их таких программных средств является программный комплекс поддержки работы преподавателя вуза [3]. Данный программный комплекс позволил преподавателям перейти от устаревшего формата ведения учета в виде бумажных карточек групп к системе с единой базой данных, доступной как в веб-версии, так в виде мобильных клиентов для операционных систем «iOS» и «Android». Мобильные клиенты позволяют вести учет успеваемости и посещаемости, не привязывая преподавателя к стационарному рабочему месту и поддерживая актуальность данных путем синхронизации локальных баз данных, хранящихся на устройствах, с глобальной, в то время как веб-версия расширяет данные функциональные возможности за счет возможности управления курсами дисциплин и выдаваемыми работами, а также проведения внутренних и промежуточных аттестаций.

1. Подходы к выявлению группы риска

Помимо учета успеваемости и посещаемости студентов вуза, деятельность структурных подразделений вуза направлена улучшение качества обучения и своевременное выявление возникающих проблем или же предупреждение их возникновения. Так, при попытке предупреждения возникновения проблем как таковых речь идет о выявлении так называемой «группы риска», которая для каждой потенциальной проблемы своя. Это могут быть как студенты, которые несвоевременно закроют сессию, так и те студенты, которые и вовсе не будут допущены до защиты выпускной квалификационной работы. Выделение группы риска подразумевает выявление формирующихся в ходе обучения студентов закономерностей, знание которых позволило бы проводить превентивные меры еще до того, как потенциальные проблемы станут реальными.

Одними из методов выявления группы риска являются прогнозирование и поиск последовательностей, относящиеся к интеллектуальному анализу данных, который представляет собой направление информационных технологий, охватывающее всю область проблем, связанных с извлечением знаний из массивов данных. В данном случае в качестве массива данных выступает база данных программного комплекса, хранящая в себе данные об успеваемости и посещаемости студентов начиная со своего внедрения. Проведение интеллектуального анализа аккумулируемых в ходе обучения студентов данных об их успеваемости и посещаемости позволит выявить те закономерности, которые определят необходимую в рамках поставленной задачи группу риска, что позволит произвести превентивные меры по устранению потенциальных проблем.

Кроме того, выявление закономерностей становится возможным при наличии качественной визуализации данных. Грамотно составленные визуальные интерактивные отчеты, включая каждый содержащийся на них график, решают поставленную бизнес-задачу, что позволяет быстро выявлять закономерности при взгляде на них даже неподготовленному пользователю.

2. Описание возможностей пользователей

Будучи заинтересованными в улучшении качества предоставляемого образования и своевременном выявлении проблем и являясь пользователями программного комплекса, преподаватель и куратор группы также преследуют и собственные цели.

Преподаватель кафедры заинтересован в том, чтобы преподаваемая им дисциплина была интересна и актуальна для студентов. Будучи заинтересованными дисциплиной, студенты с большим вниманием отнесутся к прослушиванию лекций и выполнению лабораторных работ. Кроме того, особо заинтересованные студенты могут в дальнейшем связать свою будущую карьеру с данной областью. Однако, будучи интересной и актуальной, дисциплина также должна быть ориентирована на средний показатель уровня сложности. Это подразумевает составление методических указаний таким образом, чтобы изучение дисциплины средним студентом не вызывало у него непреодолимых трудностей. Преподаватель не заинтересован в том, чтобы большинство из студентов не понимало излагаемый на занятиях материал. Также следует уделять должное внимание расчету времени выполнения студентами работ во избежание установки таких сроков сдачи, в которые не может уложиться большая часть средней группы.

Куратор группы, будучи ответственным за учебно-воспитательные процессы в группе, заинтересован в выявлении проблемных студентов, отстающих по большинству преподаваемых группе дисциплин, и своевременном проведении разъяснительных работ, а также мероприятий по их ликвидации. Кроме того, куратор группы заинтересован в поощрении выдающихся студентов. Если же выявляются дисциплины, вызывающие сложности у курируемой группы в целом, куратор заинтересован в проведении бесед с преподавателями о возникших у студентов сложностях, а также непосредственно с самими студентами и затем, при необходимости, с руководством кафедры. Это необходимо для представления общей картины сложившейся проблемной ситуации.

Действия, осуществляемые пользователями программного комплекса в лице преподавателя и куратора группы, которые направлены на выявление группы риска, изображены в виде диаграммы вариантов использования в нотации «UML» на рис. 1.

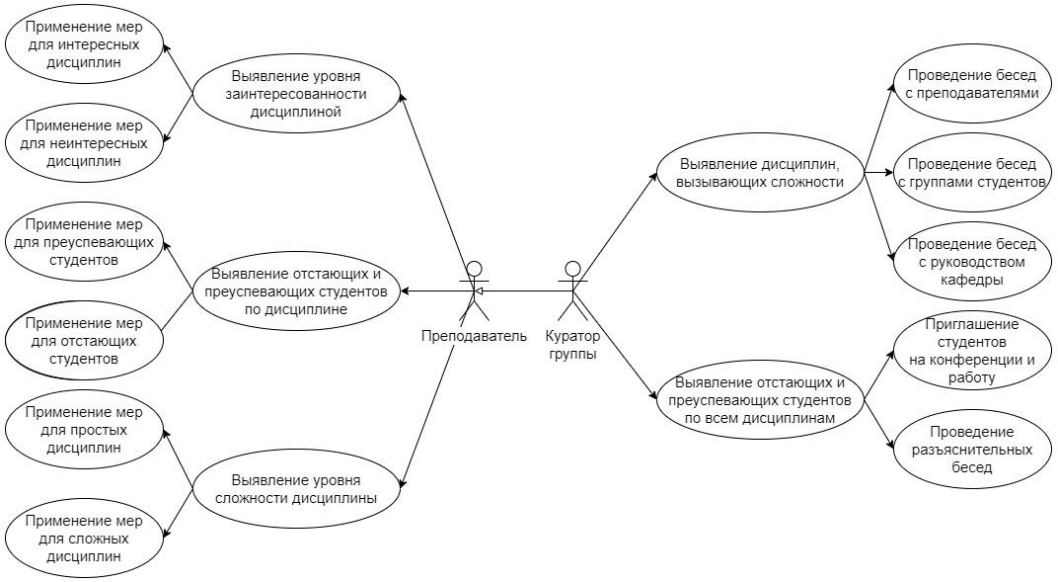


Рис. 1. Диаграмма вариантов использования аналитической подсистемы

В данном случае под группой риска может подразумеваться не только группа студентов, но и группа преподаваемых им дисциплин в зависимости от поставленной пользователем задачи.

3. Архитектура аналитической подсистемы

Существующий программный комплекс располагает веб-клиентом и веб-сервисом, с помощью которых реализуются учет успеваемости и посещаемости студентов вуза. В качестве программной платформы в программном комплексе используется технология «ASP.NET MVC» с основным языком программирования «C#». Для эффективной работы веб-приложения также используется скриптовый язык «JavaScript» и его фреймворк «AngularJS». Выбор технологии, основанной на .NET Framework влечет за собой однозначный выбор веб-сервера «IIS». В качестве СУБД используется Microsoft SQL Server, поскольку данная система управления базами данных наилучшим образом взаимодействует с ASP.NET-приложениями. Верстка страниц приложения осуществляется с помощью фреймворка «Bootstrap», включающего готовые HTML5- и CSS-шаблоны, однако это не исключается использования HTML5- и CSS-компонентов в чистом виде.

Разработка аналитической подсистемы требует доработки существующих веб-клиента и веб-сервиса путем реализации подсистемы визуализации аналитики в веб-клиенте, а также создания базы данных аналитики и реализации подсистемы аналитики в веб-сервисе. Кроме того, доработки требуют существующие в веб-сервисе подсистемы работы с базой данных и передачи данных. Для реализации механизма скоринга была выбрана аналитическая платформа «Loginom» [4], тогда как в качестве средства разработки аналитических отчетов была выбрана платформа бизнес-анализа «Power BI». Архитектура разрабатываемой подсистемы аналитики представлена на рис. 2.

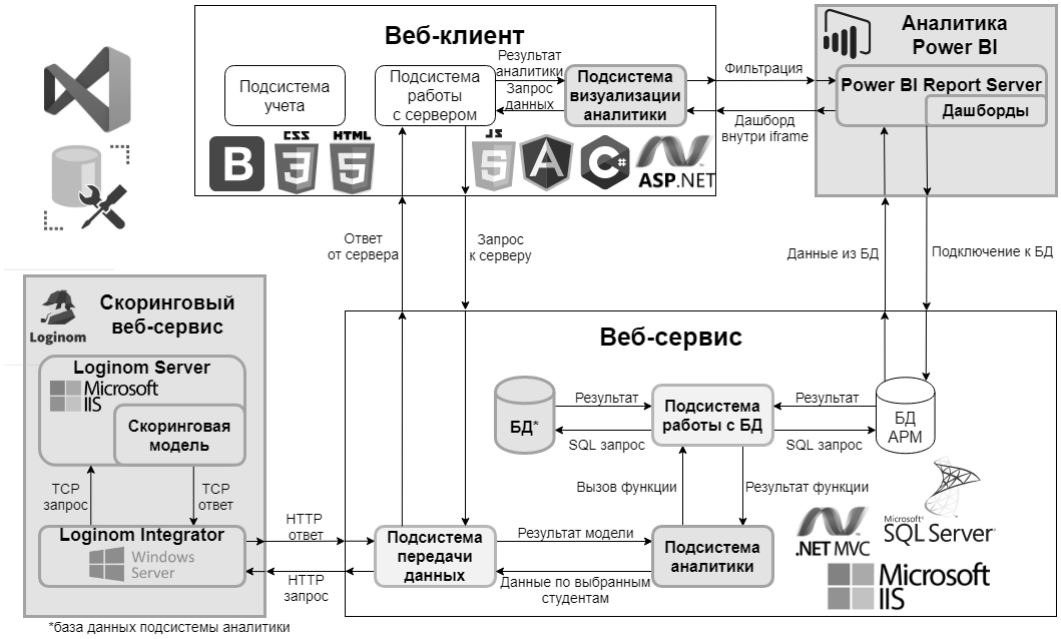


Рис. 2. Архитектура аналитической подсистемы

Механизм скоринга реализуется за счет разработки сценария, содержащего скоринговую модель и хранящегося на сервере «Loginom Server», и его дальнейшей публикации в виде веб-сервиса. Связь подсистемы аналитики и опубликованного сценария осуществляется путем отправки подсистемой передачи данных HTTP-запросов и получения HTTP-ответов от модуля «Loginom Integrator». Разработанные аналитические отчеты (дашборды), в свою очередь, публикуются на сервере «Power BI Report Server», получая данные с помощью шлюза из базы данных программного комплекса. Встраивание аналитических отчетов осуществляется с использованием iframe в подсистеме визуализации аналитики на стороне веб-клиента.

4. Построение скоринговой модели

Выявление группы риска на примере студентов, которые могут быть отчислены из вуза, осуществляется с помощью скоринговой модели, построенной в аналитической платформе «Loginom». Данная модель рассчитывает вероятность наступления события, отчисления студента из вуза и на основании заданного для логистической регрессии [5] порога отсечения однозначно определяет исход (рис. 3).

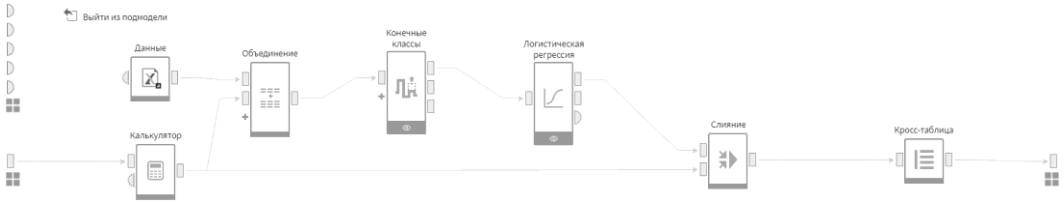


Рис. 3. Скоринговая модель

В качестве характеристик студентов, подаваемых на вход модели, выступают:

- средняя оценка, полученная за расчетно-графические и курсовые работы;
- процент лабораторных работ, полученный балл за которые ниже «хорошо»;
- средняя продолжительность выполнения лабораторных работ в минутах;
- средняя продолжительность выполнения расчетно-графических и курсовых работ в минутах;
- процент пропусков лекционных занятий по неуважительной причине;
- количество групп, в которых побывал студент за время своего обучения в вузе.

Выбор интересующего студента осуществляется в интерфейсе путем фильтрации студентов по уровню образования и группе. Интерфейс веб-клиента, реализующий взаимодействие веб-сервиса программного комплекса со скоринговой моделью, опубликованной в виде веб-сервиса, представлен на рис. 4.

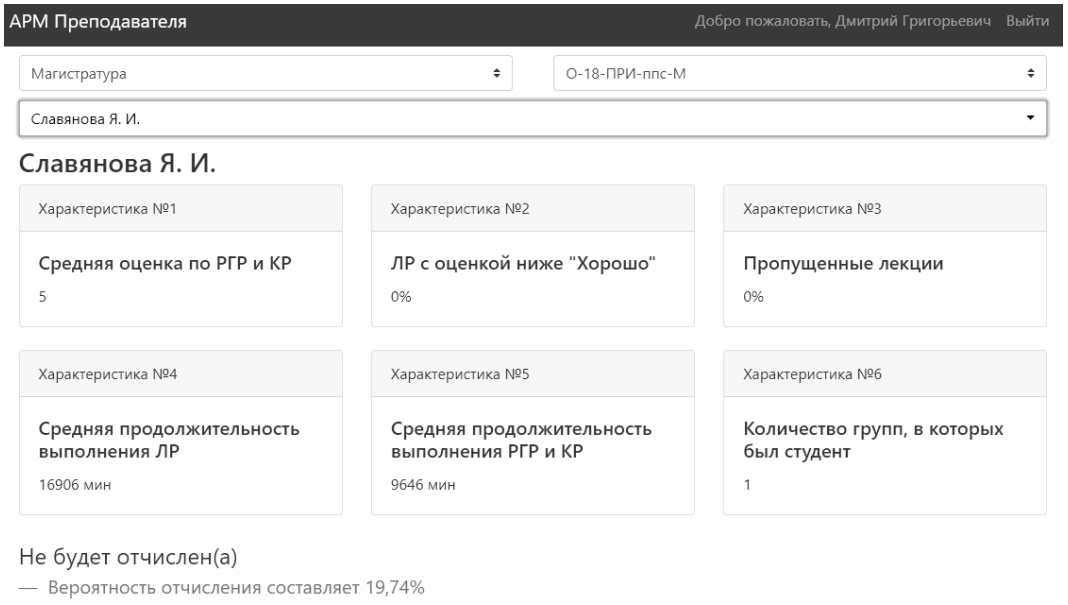


Рис. 4. Применение скоринговой модели для определения вероятности отчисления студента

Для выбранного студента выводятся агрегированные характеристики, на основании которых получаются вероятность его отчисления и исход, определенный путем сравнения порог отсечения и данной вероятности.

5. Визуальные интерактивные отчеты

Разработка визуальных интерактивных отчетов для пользовательских ролей преподавателя и куратора группы осуществлена в платформе бизнес-анализа «Power BI» [6], выбор которой обусловлен используемым стеком технологий. Визуальный интерактивный отчет преподавателя, позволяющий оценивать уровень заинтересованности студентами преподаваемыми им дисциплинами, уровень их сложности, а также выявлять отстающих и преуспевающих студентов по дисциплинам, представлен на рис. 5

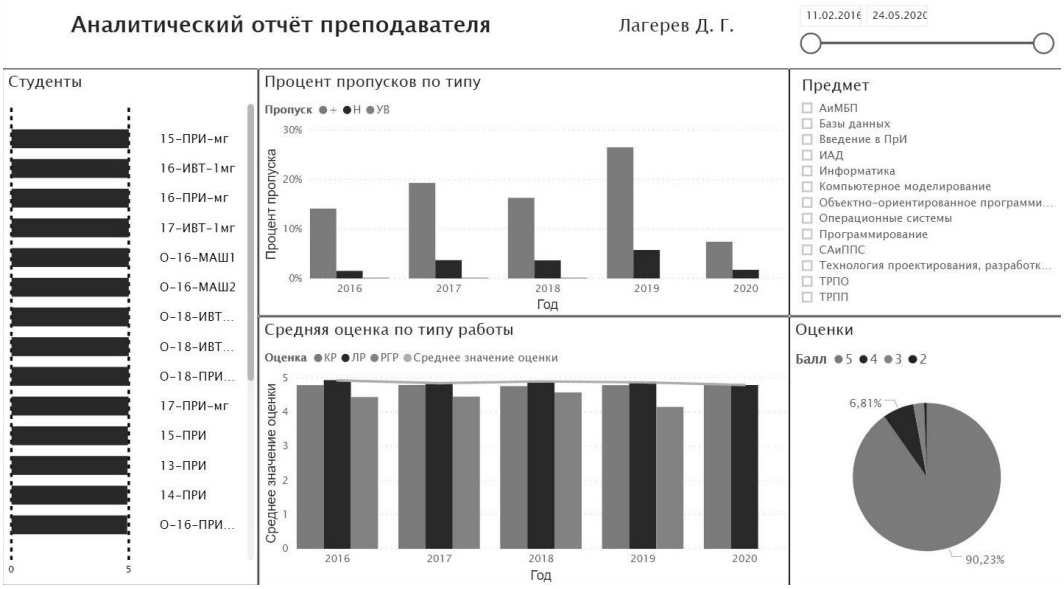


Рис. 5. Визуальный интерактивный отчет, разработанной для пользовательской роли «преподаватель»

Визуальный интерактивный отчет куратора группы, позволяющий выявлять дисциплины, вызывающие сложности у студентов курируемых групп, а также отстающих и преуспевающих студентов этих групп по всем дисциплинам, представлен на рис. 6.

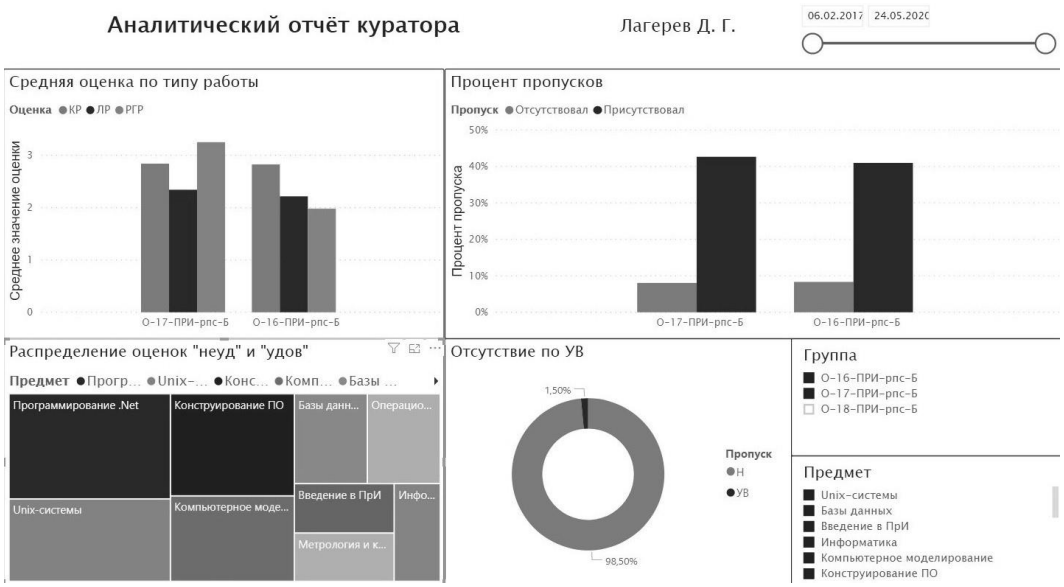


Рис. 6. Визуальный интерактивный отчет, разработанный для пользовательской роли «куратор группы»

Авторизованный пользователь системы может одновременно выступать как в роли преподавателя ряда дисциплин, так и в роли куратора ряда групп. Это означает, что авторизованному пользователю, который преподает в системе хотя бы одну дисциплину, должен быть доступен аналитический отчет преподавателя. Аналогично, авторизованному пользователю, который курирует в системе хотя бы одну группу, должен быть доступен аналитический отчет куратора группы.

Заключение

В рамках проделанной работы была спроектирована и реализована аналитическая подсистема для программного комплекса поддержки работы преподавателя вуза. Использование данной подсистемы позволяет преподавателям определять уровень заинтересованности студентов дисциплиной, уровень её сложности и выявлять отстающих и преуспевающих студентов по дисциплине. Куратору группы, в свою очередь, использование подсистемы позволяет выявлять дисциплины, вызывающие у студентов сложности, а также отстающих и преуспевающих студентов по всем дисциплинам. Кроме того, данная подсистема после обучения на имеющихся данных за несколько лет может обеспечить заблаговременное выявление зарождающихся проблем за счет выявления студентов, находящихся в группе риска, что делает возможным принятие превентивных мер куратором до того, как потенциальные проблемы реализуются.

ЛИТЕРАТУРА

1. Функциональные возможности программного продукта «1С: Университет ПРОФ». URI: https://solutions.1c.ru/catalog/university-prof/features.

2. Функциональные возможности системы управления обучением «Moodle». URI: https://moodlearn.ru/mod/page/view.php?id=174/.

3. *Калевко В.В.* Программный комплекса «Автоматизированное рабочее место преподавателя» / В. В. Калевко, Д. Г. Лагерев, А. Г. Подвесовский // Современные технологии и ИТ-образование / Сборник научных трудов II Международной научной конференции «Конвергентные когнитивно-информационные технологии» и XII Международной научно-практической конференции «Современные информационные технологии и ИТ-Образование» / под редакцией В. А. Сухомлина, Факультет вычислительной математики и кибернетики МГУ имени М. В. Ломоносова. – Москва: Лаборатория открытых информационных технологий факультета ВМК МГУ имени М. В. Ломоносова, 2017. – С. 197–205.

4. Аналитическая платформа «Loginom». URI: https://loginom.ru/.

5. *Hosmer D.W.* Applied Logistic Regression / David W. Hosmer, Stanley Lemeshow. – Second edition. – Wiley Publishing, Inc., 2000. – 375 p.

6. *De Jonge K.* Dashboarding & Reporting with Power BI: How to Design and Create a Financial Dashboard with Power BI – End to End / Kasper De Jonge. – Kindle Edition. – Holy Macro! Books, 2018. – 235 p.

СОЗДАНИЕ ОБУЧАЮЩЕЙ ПРОГРАММЫ ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ "РАЗЛИЧНЫЕ ПРЕДСТАВЛЕНИЯ БУЛЕВОЙ ФУНКЦИИ"

Стародубцева М.О., Буторина Н.Б.
Томский государственный университет
manyaioan@mail.ru, nnatta07@mail.ru

Введение

В настоящем постиндустриальном обществе традиционное обучение не утратило свою значимость, но в современном мире его заметно потеснили альтернативные оригинальные методы обучения. В частности, обучающие программы. Постепенно они начали внедряться в различные сферы деятельности человека.

Компьютерная обучающая программа – это программное средство, предназначенное для решения определенных педагогических задач, имеющее предметное содержание и ориентированное на взаимодействие с обучаемым. Они рассматриваются как предмет изучения или выступают в качестве инструментария при решении образовательных задач.

Конечно, обучающие программы не в силах заменить преподавателей, но они помогают усовершенствовать и дополнить деятельность преподавателя, таким образом они помогают улучшить процесс обучения, а также сделать его более ярким и запоминающимся и ещё позволяют проконтролировать знания, умения и навыки учащихся.

В поддержку курса Дискретная математика было решено написать обучающую программу по теме: «Обучающая программа по дискретной математике. Различные представления булевой функции» с интуитивно понятным интерфейсом для учащихся с подсказками и теоретическими материалами.

1. Виды обучающих программ

Обучающая программа – это описание процесса обучения, которое включает в себя материал для изучения, задания необходимые для овладения материала и указания для решения определенных типов задач, все это составляет основу программированного обучения. Различают следующие типы обучающих программ [1,2].

Тренировочно-закрепительные программы необходимы для обучения определенных навыков, то есть данный комплекс программ используется, когда теоретический материл усвоен и предопределяется для закрепления полученных навыков. Программы данного типа в случайной последовательности предоставляют учащемуся вопросы и предлагают определенные задачи, а затем подсчитывают число правильных и неправильных ответов, если учащийся дает неправильный ответ, то он сможет получить подсказку.

Методика тьюторского сопровождения – это программы, которые включают в себя теоретический материал и определенные типы заданий, которым педагог не может уделять достаточное время для усвоения материала. Данный комплекс помогает индивидуализировать обучение учеников, каждый сам распределяет время для изучения материала это является качественным обучением дополнительного обучения.

Наставнические программы предлагают теоретический материал, при усвоении которого задаются вопросы и предлагается определённые типы задач для управления полного обучения. Если учащийся дает неправильный ответ, то программа может «откатить назад» для повторного изучения теоретического материала.

Учебные игры – комплекс программ, который в игровом режиме тренирует и обучает ученика помогает усвоить материал, а также закрепить его на практике, тем самым формируя необходимые навыки.

Моделирующие или имитационные программы основаны на иллюстративных, графических и вычислительных возможностях компьютера, таким образом данный тип программ позволяет реализовать интересный компьютерный эксперимент. В данной реализации такой программы учащимся предоставляется возможность для проведения исследований на который они сами могут повлиять командами, введенными с внешних устройств, которые позволяют изменять значение параметров.

Таким образом, выбор одной из программ зависит от теоретической базы образовательного учреждения, подготовки преподавательского и ученического состава и его творческого желания и возможности.

Из всех возможных вариантов нами были выбраны две классификации, а именно наставническая программа и тьюторское сопровождение. Такое совмещение удобно, так как можно одновременно и наглядно изучить, и повторить теоретический материал данного раздела, сразу проверить свои знания на практике, понять и разобрать ошибки. Данные виды программ наиболее популярны из-за своей простоты и доступности, потому как в сравнении с развивающими играми и моделирующими программами, не требуют включения специалистов из других отраслей знания.

2. Достоинства и недостатки обучающих программ

Компьютерные обучающие программы в данный период занимают важную часть жизни людей, они так же помогают поддерживать учебный процесс наравне с традиционными методами обучения, однако по сравнению с традиционным учебно-методическими средствами обучающие программы обеспечивают новые возможности [3].

Достоинства	Недостатки
+ Возможность создания наглядного представления информации	– Необходимость иметь доступ к компьютеру
+ Обучение пользователя с работой	– Недостаточная интерактивность обучающих программ
+ Возможность создания алгоритма проверки знаний учащихся	– Нет возможности задать возникшие вопросы
+ Уменьшение затрат на преподавание	– Отсутствие регулярного контроля над выполнением учебного плана
+ Представляет возможность проверки задач	
+ Предоставляет функцию «подробного» ответа или небольшие подсказки	

Таким образом, можно сделать вывод, что компьютерные обучающие программы помогают вывести обучение на новый этап, они обладают большими преимуществами над традиционными способами обучения. При наличии обучающих программ в учебных заведениях появляется возможность внедрения дистанционных технологий обучения, тем самым обучающие программы являются существенным средством удаленного обучения.

3. Программное обеспечение

Разработанная нами обучающая программа «Различные представления булевой функции» была создана в среде разработки Qt Creator. Qt – кроссплатформенный Фреймворк для разработки программного обеспечения на языке программирования C++, так же имеется разработка ПО и для других языков программирования: Python – PyQt, PySide, Ruby – QtRuby, Java – Qt jambi, PHP– PHP-Qt и другие. Данная среда представляет широкие возможности и является очень удобной для разработки графического интерфейса. Некоторые достоинства из Qt Creator: удобный конструктор GUI-форм, кроссплатформенность, быстрый компилятор Qt, визуализация данных Qt, поддержка отладки, компиляции, профилирования, авто заполнение кода и рефакторинга [4].

4. Функционал программы

Разработанная нами обучающая программа «Различные представления булевой функции» позволяет пользователю наблюдать преобразования одной и той же функции в различные логические функции, из одного вида в другой. В настоящее время данная программа может принимать булевы функции в виде ДНФ, КНФ и вектора и преобразовывать их в логические функции вида: КНФ [5], ДНФ [6], СКНФ [7], СДНФ [8], Таблица истинности, Полином Жегалкина [9], а также способна отобразить полную классификацию Поста (Теоремы Поста) [10] для введенной функции.

Весь функционал реализован графическим интерфейсом, поэтому данная программа оснащена кнопками. Учащемуся предоставляется выбор, как именно он захочет ввести строку в специальное поле ввода: в форме ДНФ, КНФ или вектора. Чтобы получить результат, после ввода строки необходимо нажать кнопку «Выполнить»; для того, чтобы очистить строку, необходимо нажать кнопку «Очистить». При нажатии кнопки «Помощь» появляется новое диалоговое окно, где пользователь может посмотреть, как корректно ввести строку, а также посмотреть результат логических операций. При

нажатии на кнопку «Теория» также появляется диалоговое окно, где предоставлены определения логических функций. В функционале программы есть кнопка «Достоинства и недостатки», где можно посмотреть плюсы и минусы выбранной логической функции (ДНФ, Таблица истинности, КНФ, СДНФ и СКНФ, полином Жегалкина).

Обучающая программа предусматривает неправильный ввод функции, поэтому при допущенной ошибки у пользователя появится предупреждение «Некорректно введенная строка», а при неверно написанном векторе, появляется подсказка «Вектор должен состоять из 2^n значений».

5. Интерфейс программы

При входе в обучающую программу перед пользователем появится интерфейс (рис. 1).

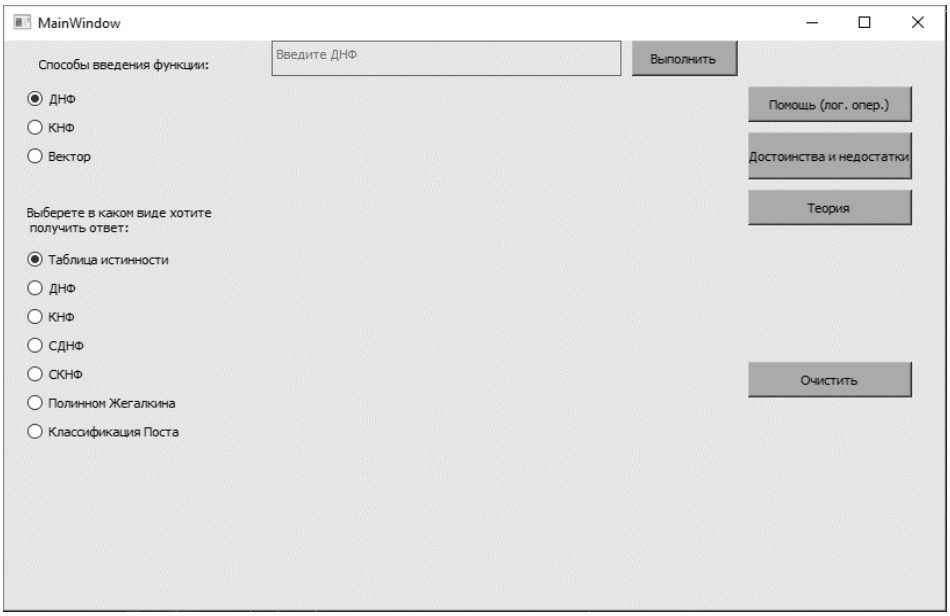


Рис. 1. Интерфейс программы

Пример преобразования из ДНФ в таблицу истинности (рис. 2).

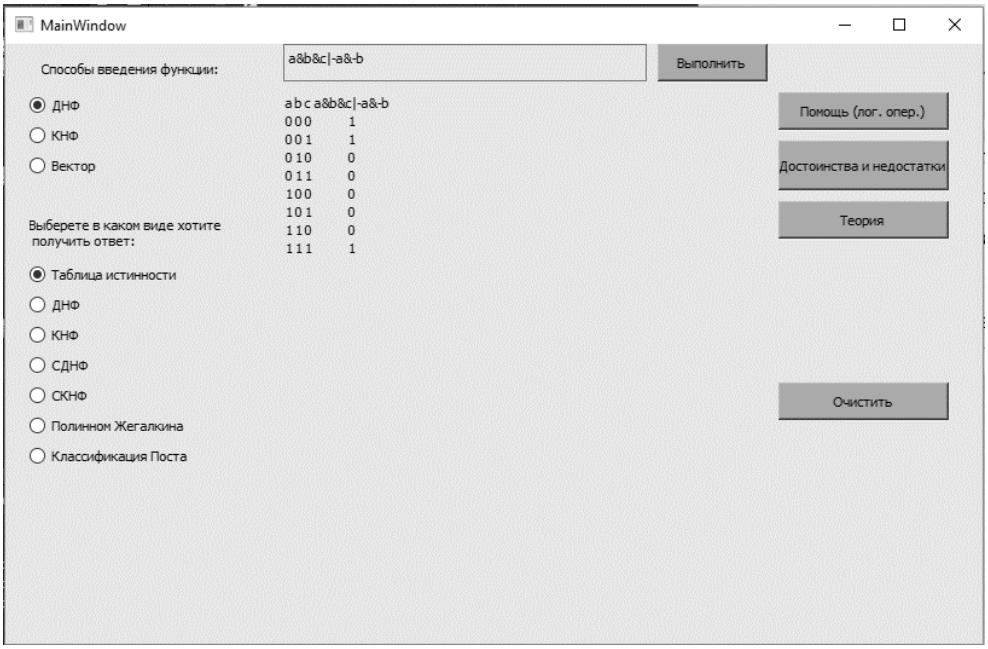


Рис. 2. Пример преобразования ДНФ в таблицу истинности

Пример вывода и использования теорем Поста (рис. 3).

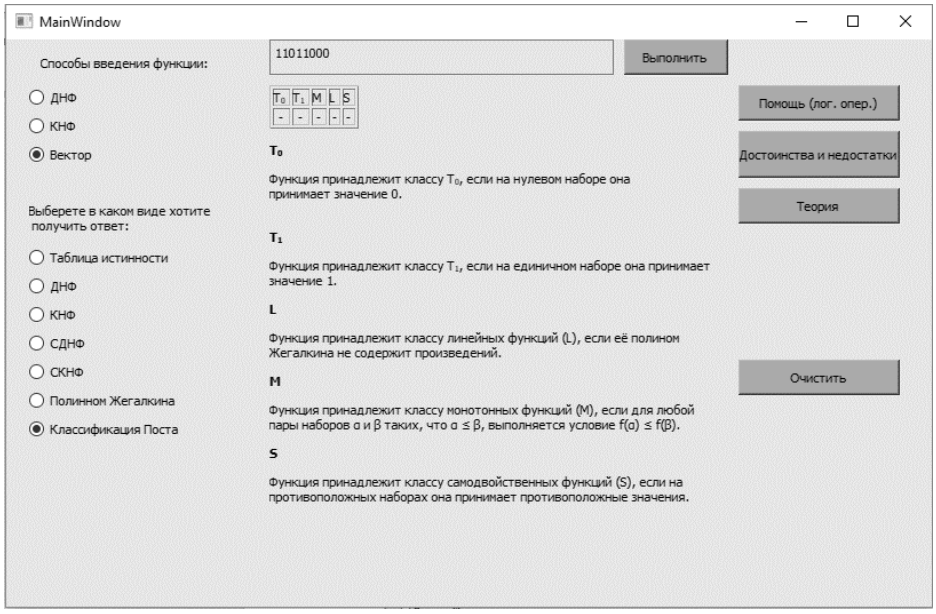


Рис. 3. Вывод и использование теорем

Примеры вспомогательных диалоговых окон (рис. 4–6).

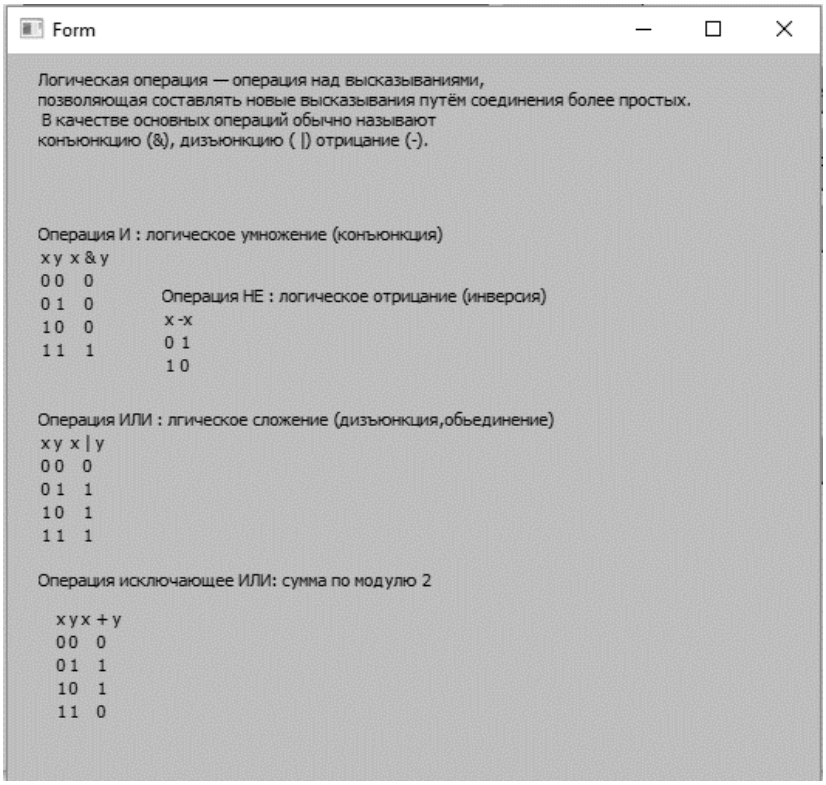


Рис. 4. «Помощь»

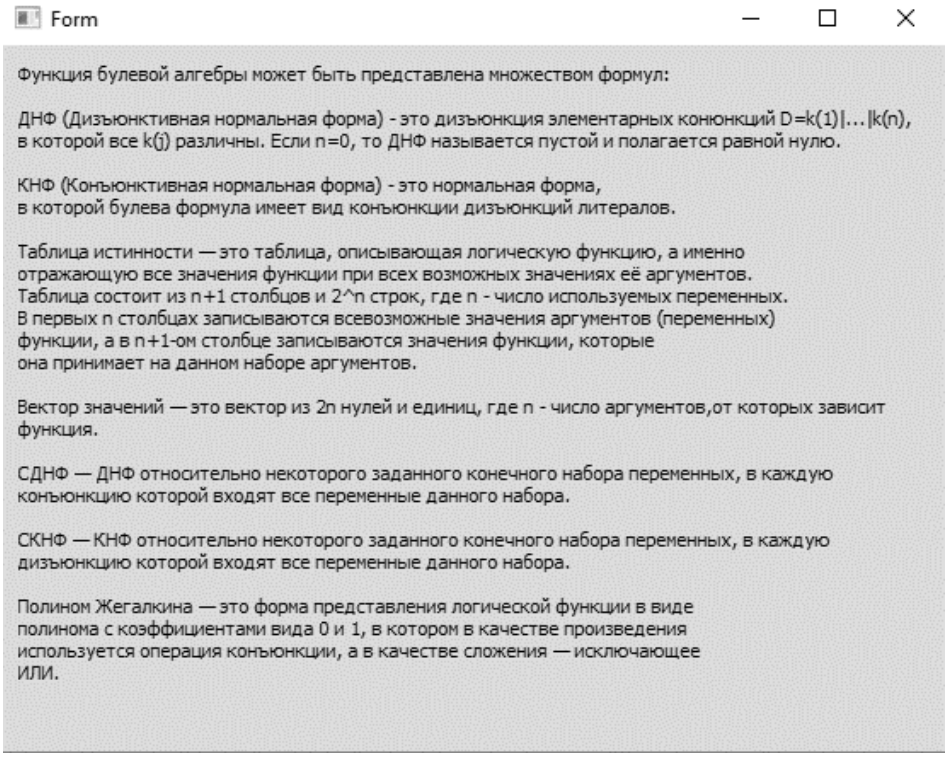


Рис. 5. «Теория»

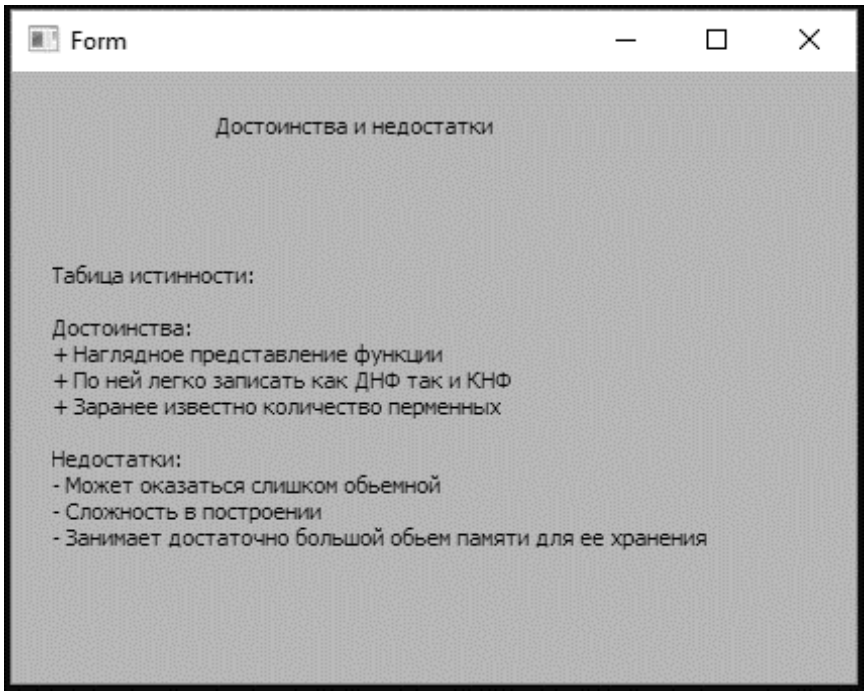


Рис. 6. «Достоинства и недостатки»

Заключение

Обучающую программу «Различные представления булевой функции» можно применить как контроль остаточных знаний как по теоретической части курса, так и по решению практических задач. Возможна подготовка как по отдельной теме, так и по всему курсу. Программа может быть использована в курсе «Дискретная математика» как средство обобщения материала, при повторении, а может быть и для изучения новых понятий.

ЛИТЕРАТУРА

1. Студопедия [Электронный ресурс]: Типы обучающих программ – 2015. – URL: https://studopedia.ru/7_3929_tipi-obuchayushchih-programm.html.

2. Справочник [Электронный ресурс]: Типы обучающих программ – 2020. – URL: https://spravochnick.ru/informacionnye_tehnologii/setevye_informacionnye_sistemy/tipy_obuchayuschih_programm/

3. Обучающие компьютерные программы [Электронный ресурс]: Достоинства и недостатки компьютерных обучающих программ в сетевом и локальном вариантах – URL: https://sites.google.com/site/obuchkomprog/home/informacionnye-razdely/dostoinstva-i-nedostatki-komputernyh-obucausih-programm-v-setevom-i-lokalnom-variantah.

4. Записки информационщика [Электронный ресурс]: Краткий обзор кроссплатформенного фреймворка Qt – 2015. – URL: https://nicknixer.ru/programmirovanie/kratkij-obzor-krossplatformennogo-frejmvorka-qt/.

5. Wikipedia [Электронный ресурс]: Конъюнктивная нормальная форма – 2019. – URL: https://ru.wikipedia.org/wiki/Конъюнктивная_нормальная_форма.

6. Wikipedia [Электронный ресурс]: Дизъюнктивная нормальная форма – 2019. – URL: https://ru.wikipedia.org/wiki/Дизъюнктивная_нормальная_форма.

7. Wikipedia [Электронный ресурс]: Совершенная конъюнктивная нормальная форма – 2018. – URL: https://ru.wikipedia.org/wiki/Совершенная_конъюнктивная_нормальная_форма.

8. Wikipedia [Электронный ресурс]: Совершенная дизъюнктивная нормальная форма – 2020. – URL: https://ru.wikipedia.org/wiki/Совершенная_дизюнктивная_нормальная_форма.

9. Habr [Электронный ресурс]: Что нам стоит полином Жегалкина построить – 2016. – URL: https://habr.com/ru/post/275527/.

10. Wikipedia [Электронный ресурс]: Критерий Поста – 2018. – URL: https://ru.wikipedia.org/wiki/Критерий_Поста.

СОЗДАНИЕ ОБУЧАЮЩЕЙ ПРОГРАММЫ ДЛЯ ФОРМИРОВАНИЯ НАВЫКА ВЫЧИСЛЕНИЯ ОБРАТНОЙ МАТРИЦЫ

Сыч М.Б., Пахомова Е.Г.
Томский государственный университет
peg@tpu.ru

Введение

Современные компьютерные технологии повлияли почти на все сферы жизни человека. Не остается в стороне и процесс образования. С момента появления первого персонального компьютера стало понятно, что информационные технологии имеют большие перспективы в образовательном процессе.

Прежде всего, они обеспечивают доступ к информации и быстрый обмен данными. Это делает информацию более доступной и удобной в хранении, а поисковые системы позволяют найти информацию практически по любому вопросу.

Однако доступность информации это не единственное влияние информационных технологий на образование. Уже давно появились идеи переложить на компьютер некоторые функции преподавателя. Например, проверку полученных учеником знаний и навыков. Существуют области, в которых компьютер с этим вполне успешно справляется. Например, первую часть ЕГЭ проверяется именно компьютером.

Но функция преподавателя это не только проверка правильности ответа студента. Помимо того, что преподаватель может указать на ошибку, он также способен ее проанализировать и, возможно, помочь в ней разобраться. Но время у преподавателя ограничено, и не всегда есть возможность помочь студенту, например, в период дистанционного обучения. Именно желание делегировать компьютеру частично или полностью такие функции преподавателя, как проверка полученных знаний и анализ ошибок обучающегося, привело к появлению программ, которые называются обучающими.

Существуют несколько видов обучающих программ: контролирующие, наставнические, имитационные и моделирующие, а также развивающие игры [1].

Контролирующие программы – это программы, которые используются, когда учебный материал уже изучен и предназначаются для проверки того, как студент усвоил этот материал. Программа в случайной последовательности ставит ученику вопросы и предлагают задачи, подсчитывая количество ошибок.

Наставнические программы – это программы, которые предоставляют ученику теоретический материал, при изучении которого задаются вопросы и предлагаются управление ходом обучения. Программа анализирует ответ ученика и, исходя из ответа, определяет следующий шаг в обучении. В своей основе наставнические программы – алгоритмы программированного обучения, которые были выдвинуты одним из известных психологов XX века Скиннером. Наставнические программы подразделяются на линейные, разветвленные, адаптивные и комбинированные. Линейная программа состоит из небольших последовательно сменяющихся познавательных блоков учебной информации с контрольными вопросами и вариантами ответов. Разветвленные программы отличаются от предыдущих тем, что ученику при неправильном ответе даётся возможность воспользоваться дополнительной информацией, после чего исправить ответ. Адаптивные программы позволяют выбрать уровень сложности учебной информации, при необходимости обращаться к источникам информации. Комбинированные включают в себя компонент линейных, разветвленных и адаптивных программ. Именно комбинированные программы имеют наибольшее распространение среди коммерческих обучающих программ.

Отличием контролирующей программы от наставнической является то, что в первом случае программа способна только определить правильность ответа обучающегося,

а во втором – программа способны проанализировать ответ учащегося и дать рекомендацию по дальнейшим действиям.

Имитационные и моделирующие – позволяют использовать компьютер для эксперимента, студенту предоставляется возможность вводить некоторые команды с клавиатуры и влиять на ход эксперимента.

Развивающие игры – создают определенную виртуальную среду и возможность влиять на эту среду со стороны пользователя, тем самым изучая эту среду. В основном виртуальная среда частично моделирует реальный мир.

Анализ обучающих программ, которые находятся в открытом доступе в Интернете показывает:

1. Существует множество контролирующих программ, ярким представителем которых является приложение «Билеты ПДД». Наверняка Вы знаете людей, которые использовали его в той или иной форме, перед сдачей экзамена ГИБДД.

2. Широко распространены приложения для обучения иностранным языкам, такие как «Duolingo», «Lingualeo» и другие. Они относятся к комбинированным обучающим программам наставнического типа. Также существуют подобные приложения для изучения языков программирования, например «SoloLearn».

3. К имитационным и моделирующим можно отнести приложение «Chemist», которое моделирует химическую лабораторию и позволяет проводить эксперименты, не тратя реагенты и безопасно. Примером таких программам являются и те программы, которые студенты ТГУ используют в ходе курса «Методы оптимизации» и «Теория оптимального управления».

4. Существует также большое количество обучающих игр. Они, как правило, сложнее в реализации, т.к. требуют совместной работы программистов, специалистов в области изучаемого предмета и психологов [2]. Существующие обучающие игры рассчитаны, в основном, на детей. Одним из представителей этой группы является серия игр «Заработало», эти игры позволяют изучить в игровой форме основы механики и термодинамики.

Для студентов в Интернете по математике можно найти в основном онлайн-калькуляторы, которые нельзя отнести к обучающим программам. Современные онлайн-калькуляторы во многих случаях предоставляют довольно полное решение задачи, но они не решают главной задачи обучающей программы – формирование устойчивого навыка, который формируется только в том случае, когда обучающийся сам решает задачу, а не разбирает готовое решение. Поскольку процесс решения задачи нередко сопровождается ошибками, особенно на начальном этапе изучения темы, и эти ошибки обычно для студента не очевидны, то наиболее эффективными будут обучающие программы наставнического типа, которые проанализируют ответ студента и дадут ему рекомендацию по дальнейшим действиям.

Отдельно хотелось бы отметить образовательный комплекс «Линейная алгебра и аналитическая геометрия» от компании «1С», поскольку разрабатываемая программа относится к тому же разделу математики, что и указанная. Как утверждают сами разработчики – «... Образовательный комплекс сочетает в себе важные достоинства. Он содержит фундаментальный теоретический материал и предоставляет пользователю широкие возможности современных информационных технологий, позволяющих эффективно изучать курс самостоятельно…». Курс соответствует программе технического вуза и содержит в себе следующие темы:

1) матрицы и определители;
2) линейные пространства;
3) системы линейных уравнений;
4) евклидовы и унитарные пространства;
5) линейные операторы;
6) квадратичные и билинейные формы;

7) точки, прямые, плоскости;
8) эллипс, гипербола, парабола;
9) кривые второго порядка;
10) элементы теории поверхностей второго порядка.

Знакомство с программой показывает, что в ней отсутствует тема «Обратные матрицы», а ведь это широко применяемый инструмент во многих областях математики и ее приложениях. Данный комплекс помимо теоретического материала содержит «Задачи и упражнения для самостоятельной работы», в котором обучающимся даются различные задания из задачника, и ответ учащегося сравнивается с правильным. При этом, если ответ учащегося неверен, ему просто сообщается верный ответ. Это существенно отличается от того, что реализовано в нашей программе. Во-первых, для обучающихся каждый раз создается новая матрица, для которой необходимо вычислить обратную, во-вторых, программа способна проанализировать ответ, а не просто сообщить о том верный он или нет.

Следует отметить, что не для любого раздела математики можно создать эффективную обучающую программу. Это возможно только для тех разделов, где приобретение навыка связано с реализацией вычислительных алгоритмов. К таким разделам относится линейная алгебра, векторная алгебра, аналитическая геометрия и некоторые разделы математического анализа.

1. Постановка задачи

Цель работы – создать обучающую программу адаптивного и наставнического типа, которая бы позволяла приобрести навык вычисления обратной матрицы.

Для достижения поставленной цели необходимо выполнить следующие задачи.

1. Провести анализ типовых ошибок, допускаемых при вычислении обратной матрицы.

2. Реализовать класс Matrix, в котором происходит вычисление обратной матрицы и реализуется анализ введенного студентом ответа.

3. Реализовать выявление следующих ошибок студента:

а) ошибка знаков алгебраических дополнений;

б) ошибка транспонирования матрицы;

в) ошибка вычисления определителя исходной матрицы;

г) неверные алгебраические дополнения второго порядка (для матриц порядка 3).

5. Реализовать пользовательский интерфейс включающий в себя:

а) главное меню;

б) справочный материал;

в) разобранный пример:

г) тренажер (генерация исходной матрицы, поля для ввода ответа и поле с анализом результата);

д) руководство пользователя;

е) меню выбора сложности.

2. Анализ типовых ошибок студента

Поскольку вычисление обратной матрицы это определенный алгоритм, то если студент совершает ошибку, это значит, что он выполнил неверный шаг алгоритма или несколько шагов. Таких комбинаций может быть большое количество и все проанализировать крайне сложно, но, как показывает практика, наиболее часто студенты совершают следующие ошибки:

а) забыл транспонировать матрицу;

б) забыл про знаки алгебраических дополнений;

в) неверный найден определитель исходной матрицы;

г) потерян множитель перед союзной матрицей;

д) ошибка в формуле вычисления алгебраических дополнений второго порядка.

Для того чтобы понять, допустил ли студент ошибки, программа сначала сравнивает ответ студента с правильно вычисленной обратной матрицей и если ошибок нет, то сообщает пользователю об этом. Если ошибки есть, то для выявления каждой из ошибок используется специальный метод. Рассмотрим каждый из них подробнее.

1. Для выявления ошибки транспонирования, считается вспомогательная матрица к текущей, которая находится как обратная матрица, но без шага транспонирования, затем вспомогательная матрица сравнивается с ответом студента, если совпали все элементы, то студент наверняка пропустил шаг транспонирования, если совпало хотя бы 70 % ответов, то студент с большой долей вероятности допустил ошибку транспонирования и еще несколько арифметических.

2. Для выявления ошибки знаков алгебраических дополнений, элементы с нечетной суммой индексов обратной матрицы умножаются на –1 и сравниваются с соответствующими элементами матрицы, введенной студентом. Если все или большая часть из них совпали с ответом студента, то он явно забыл про знаки алгебраических дополнений.

3. Для выявления ошибки определителя считалось отношение каждого элемента обратной матрицы, к соответствующему элементы ответа студента. Если все отношения совпадают, но не равны единице, то студент умножал союзную матрицу на неверный коэффициент, что возможно только при ошибке вычисления определителя.

4. Для выявления ошибок вычисления алгебраических дополнений второго порядка, находилась матрица, в которой использовалась неверная формула для вычисления определителей второго порядка и проводилось сравнение этой матрицы с матрицей введенной студентом. Если большая часть элементов совпадала, значит студентом действительно допущена эта ошибка.

3. Интерфейс программы

Интерфейс программы интуитивно понятен. Как показывает анализ [3], это важно для обучающей программы и влияет на ее эффективность. Работа программы начинается с открытия главного меню (рис. 1). В главном меню студенту предлагается выполнить следующие действия:

1) выбрать уровень сложности;
2) ознакомиться с руководством пользователя;
3) изучить пример;
4) изучить теорию;
5) выйти из программы.

Выбор действия осуществляется нажатием соответствующей кнопки «руководство пользователя», «теория», «пример», «упражнение», «выход».

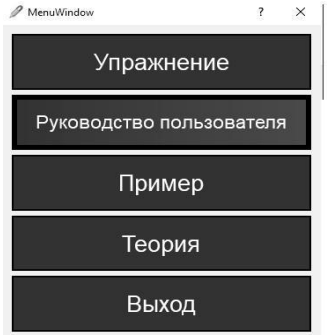


Рис 1. Главное меню обучающей программы

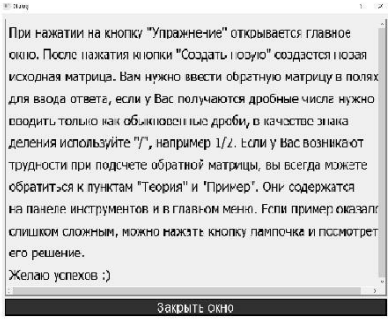


Рис. 2. Руководство пользователя

При нажатии на кнопку «руководство пользователя» открывается окно с кнопкой возврата в главное меню и текстовым полем, содержащим краткое описание и инструкцию по работе в программе (рис. 2). Кнопки «Теория» и «Пример» аналогичны преды-

дущей, но текстовые поля содержат теоретический материал и разобранный практический пример соответственно.

Кнопка упражнения переводит пользователя в меню выбора сложности (рис. 3). Предполагается три уровня сложности: легкий, средний, сложный. Легкий уровень предполагает работу с матрицей порядка два, средний – с матрицей порядка три, сложный – с матрицей порядка четыре. После выбора сложности открывается главное окно программы.

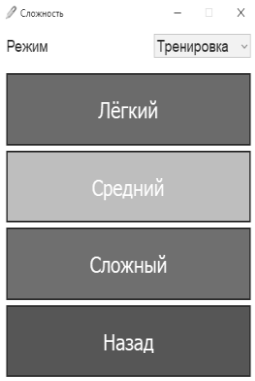


Рис. 3. Меню выбора сложности

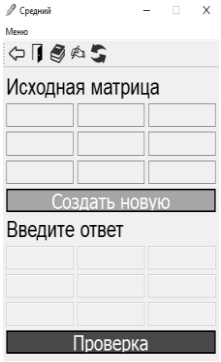


Рис. 4. Главное окно программы

Главное окно программы (рис. 4) содержит поле для размещения исходной матрицы, которая генерируется случайным образом при нажатии кнопки «Создать новую». После того, как исходная матрица будет сгенерирована, становятся доступными поля ввода ответа. Вводимый студентом ответ может быть целым или дробным. В последнем случае ответ вводится как неправильная дробь (числитель и знаменатель разделяются знаком «слэш»). Реализация ввода дробного ответа была сделана помощью регулярных выражений.

После того, как все поля для ответа будут заполнены, становится доступна кнопка проверки. По ее нажатию выводится окно с сообщением о правильности выполнения задания. В случае, если был введен неверный ответ в сообщении выводится тип ошибки или, если ошибка не соответствует ни одному из типов, то выводится количество ошибок (рис. 5,6).

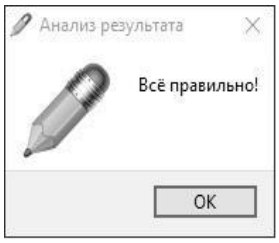


Рис. 3. Ответ верен

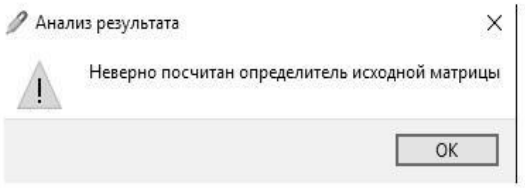


Рис. 4. Введен неверный ответ

После закрытия окна с сообщением студент может обратиться к справочным материалам, очистить все поля соответствующей кнопкой или сгенерировать новую матрицу.

Заключение

В результате выполненной работы по созданию обучающей программы был проведен анализ существующих обучающих программ, анализ типовых ошибок, допускаемых при вычислении обратной матрицы, реализован класс Matrix, создана работающая обучающая программа. В дальнейшем планируется доработать программу, а именно:

1) ввести проверку комбинации ошибок (неверно посчитан определитель и потеряны знаки алгебраических дополнений, не транспонирована матрица и потеряны знаки алгебраических дополнений);
2) сделать версию для Linux;
3) добавить выбор языка (русский или английский);
4) добавить возможность посмотреть решение текущего примера, после нескольких неудачных попыток.

ЛИТЕРАТУРА

1. *Гефан Г.Д., Кузьмин О.В.* Методика построения контрольно-обучающих программ и их использование в преподавании математических дисциплин // Вестник Бурятского государственного университета. 2013. № 15. С. 23-28.

2. *Аязбаев Т.Л., Галагузова Т.А.* Технология создания компьютерных обучающих программ // Международный журнал экспериментального образования. 2015. – № 3 (часть 2) – С. 76-78.

3. *Афанасьев В.В., Тыщенко О.Б., Афанасьева И.В.* Анализ показателей эффективности обучающих программ // I Всероссийская научно-техническая конференция 'Компьютерные технологии в науке, проектировании и производстве'. Тезисы докладов, часть V, Нижний Новгород, 1999, 43с., стр. 15.

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЖЕСТКИХ СИСТЕМ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ПРИ МОДЕЛИРОВАНИИ КИНЕТИКИ ПЛАСТИЧЕСКОЙ ДЕФОРМАЦИИ

Трифонов С.А., Самохина С.И.
Томский государственный университет
stasta956@gmail.com, sv.sam.tsk@gmail.com

Введение

Изучение металлов, а именно такого явления как пластическая деформация, является сложной задачей для математического моделирования и экспериментального исследования. Пластическая деформация металлов – это сложный динамический процесс. Его сложность зависит от материала, который подвергается деформации, от его свойств, а также способов внешнего воздействия на него. Наиболее эффективным способом описания сложных систем на языке математики является математическое моделирование. С помощью достаточно полных математических моделей, отражающих процессы образования и аннигиляции деформационных дефектов, становится возможным проведение исследований явлений, трудно или невозможно осуществляемых в реальном физическом эксперименте.

1. Моделирование процессов образования и аннигиляции деформационных дефектов

В исследовательских работах, посвященных процессам пластической деформации применяются математические модели, базирующиеся на уравнениях баланса дефектов деформации. Данные модели разрабатывались в многочисленных работах [1–7]. В моделях рассматриваются разные наборы деформационных дефектов и механизмы их возникновения и исчезновения (аннигиляции). Одной из более поочередно и отчетливо продуманных моделей, основанных на балансных уравнениях деформационных дефектов, считается математическая модель пластической деформации скольжения [8]. В базе представленной модели лежит принцип упрочнения как процесса скопления деформационных дефектов и отдыха в результате термически активируемого залечивания дефектов деформации.

В основе математических моделей пластической деформации, базирующихся на уравнениях баланса дефектов деформации, лежат частные модели простейших механизмов и процессов пластичности. Именно выбор типов деформационных дефектов, определяющих закономерности пластической деформации, и механизмов их рождения,

аннигиляции и взаимной трансформации определяет адекватность и возможности математической модели.

Математическая модель пластической деформации скольжения, базирующаяся на балансных уравнениях деформационных дефектов [8], разработана в трудах Л.Е. Попова с сотрудниками. В основу этой модели легла концепция упрочнения и отдыха. Данная концепция описывает весь процесс пластической деформации как совокупность двух противоположных процессов: упрочнения, связанного с возникновением искажений кристаллической решетки и отдыха, связанного с исчезновением этих искажений в зависимости от изменения температуры и времени [9]. Таким образом, упрочнение кристаллов в процессе пластической деформации происходит вследствие накопления дефектов деформации в кристаллической решетке, а разупрочнение связано с их исчезновением в результате образования термических процессов, происходящих во время деформации.

Изначально модель пластической деформации кристалла была сформулирована в виде:

$$\sigma = \Phi(\rho), \tag{1.1}$$

$$\varepsilon = \Psi(\rho), \tag{1.2}$$

где σ – напряжение, ε – деформация, ρ – плотность дислокаций.

Уравнения (1.1), (1.2) представляют в параметрической форме кривую деформации σ–ε. Данная формулировка модели кривой деформационного упрочнения не является удобной математически т.к. часто скрывает физическое содержание микроскопических процессов пластичности, однако именно с ней связаны первые успешные реализации модели.

Математическая модель кинетики пластической деформации скольжения в общем случае, содержит три типа уравнений, описывающих: 1) наружное влияние, которое является предпосылкой пластической деформации; 2) отклик деформируемого кристалла на деформирующее воздействие; он представляет уравнение или систему уравнений, связывающих сдвиговые, диффузионные и другие компоненты макроскопической деформации с переменными, характеризующими внешнее воздействие и деформационную дефектную среду; 3) кинетику деформационных дефектов в кристалле. Система уравнений кинетики деформационных дефектов описывает производство деформационных дефектов под внешним воздействием и их аннигиляцию в результате различных релаксационных процессов. Система уравнений, описывающая процесс пластической деформации, в общем виде в работах [8,10] была представлена как:

$$\frac{d\rho_i}{dt} = \frac{d\rho_i}{da}\dot{a} + \frac{d\rho_i}{dt}, \tag{1.3}$$

$$\frac{dc_k}{dt} = \frac{dc_k}{da}\dot{a} + \frac{dc_k}{dt}, \tag{1.4}$$

$$\dot{a} = f\left(\tau, \rho, c_k, T\right), \tag{1.5}$$

$$F\left(\tau, a, t\right) = 0 \tag{1.6}$$

Здесь ρ_i – плотность дислокаций *i*-го типа, c_k – концентрация точечных дефектов *k*-го типа. Первое слагаемое уравнений баланса деформационных дефектов (1.3), (1.4) описывает интенсивность их генерации, второе слагаемое – скорость аннигиляции.

Уравнение (1.6) описывает внешнее воздействие, которое вызывает деформацию. Характеристикой отклика твердого тела на приложение деформирующего напряжения является скорость деформации (1.5).

2. Жесткие дифференциальные уравнения. Основные понятия

Описанные выше математические модели упрочнения и разрушения деформационных дефектов при пластической деформации, представлены в виде систем дифференциальных уравнений первого порядка, в общем случае может быть записана как:

$$\frac{dy_i(x)}{dx} = f_i\left(y_1(x),...,y_N(x),x\right),\ i=\overline{1,N}. \tag{2.1}$$

Известно, что система N дифференциальных уравнений вида (2.1) имеет множество решений, которые в общем случае зависят от N параметров. Чтобы определить значения этих параметров, необходимо наложить N дополнительных условий на функции, т.е. поставить задачу Коши. Для задачи Коши начальное условие будет задаваться в виде:

$$y_i(x_0) = y_{i,0},\ i=\overline{1,N}. \tag{2.2}$$

Решением системы обыкновенных дифференциальных уравнений (далее ОДУ) (2.1) является любая упорядоченная совокупность $y_1(x),...,y_N(x)$ обращающая каждое уравнение (2.1) в тождество [11,12].

В ходе численного решения систем ОДУ математических моделей пластической деформации появляются вспомогательные трудности, связанные с тем, что процессы возникновения и аннигиляции дефектов деформации являются разными по скорости действия, кроме этого порядок переменных системы значительно различается. В этом случае, как правило, приходится иметь дело с жесткой системой ОДУ.

Исследования жестких систем начались в начале XX-го века в сфере радиотехники. Затем в середине 50-х годов XX века новая волна интереса возникла при исследовании химической кинетики, в которых находились как очень медленно, так и очень быстро протекающие химические процессы. В этот момент выяснилось, что считавшиеся самыми надежными методы типа Рунге – Кутты перестали корректно работать в этих условиях, а именно стали давать сбой в расчетах этих задач.

В 1970-х годах исследователи Шампайн и Гир (Shampine L.F., Gear W.C.) проводили большое количество вычислительных экспериментов с системами уравнений с разнопорядковыми величинами. При помощи этих экспериментов исследователи дали свое определение жесткой системы: исходная задача для системы (2.1) считается жесткой, в случае если хотя бы у одного из собственных чисел якобиана

$$J_{i,j} = \partial f_i \big/ \partial y_j,\ i,j=\overline{1,N}$$

вещественная часть меньше нуля и велика, а решение на большей части отрезка интегрирования изменяется медленно [13].

Для удобства систему уравнений (2.1) и начальное условие (2.2) можно представить в компактной векторной форме, которая будет иметь вид:

$$\begin{cases} Y' \equiv \dfrac{dY}{dx} = F\left(Y(x),x\right), \\ Y(x_0) = Y_0, \end{cases} \tag{2.3}$$

где Y, Y_0, Y' и F это вектора размерности $N \geq 1$, x – это независимая переменная.

Еще одним определением жесткой системы является следующее: система ОДУ вида $Y' = A \cdot F(Y)$ называется жесткой, когда вещественные части всех собственных значений матрицы A меньше нуля и отношение $s = \dfrac{\max\left\{\left|\mathrm{Re}(\lambda_i)\right|,\ 1 \leq i \leq N\right\}}{\min\left\{\left|\mathrm{Re}(\lambda_i)\right|,\ 1 \leq i \leq N\right\}}$ велико.

Число *s* является числом жесткости системы. Единственная проблема данного определения жесткой системы в том, что не понятно в какой момент можно считать, что число *s* велико, т.е. не указана явно граница.

3. Методы численного решения жестких систем ОДУ

При численном решении задачи Коши (2.3) происходит вычисление значений $Y_1, Y_2, ..., Y_n$ в некоторой последовательности точек $x_1, x_2, ..., x_n$. Существует большое множество способов вычисления, можно привести основные подходы их построения.

Наиболее простым способом нахождения решения в точке x_{n+1}, если оно известно в x_n, является способ, который основан на разложении функции в ряд Тейлора

$$Y(x_{n+1}) = Y(x_n) + hF(Y_n, x_n, h), \tag{3.1}$$

где $F(Y, x, h) = Y'(x) + \frac{h}{2}Y''(x) + \frac{h^2}{3!}Y'''(x) + \dots$.

Если данный ряд прервать затем заменить $Y(x_n)$ приближенным значением Y_n, то получим приближенную формулу, которая является вычислительной схемой явного метода Эйлера [11,12]: $Y_{n+1} = Y_n + hF(Y_n, x_n)$.

Однако применение формулы (3.1) ограничивается задачами, в которых вычисление производных высших порядков не затруднительно, а зачастую это не так.

В начале XX века исследователи Рунге, Хойн и Кутта предложили подход, основанный на построении формулы для Y_{n+1} вида

$$Y(x_{n+1}) = Y(x_n) + h\Phi(Y_n, x_n, h), \tag{3.2}$$

в которой функция Ф близка к *F*, однако отличается отсутствием производных от функций правой части уравнения. Таким образом были получены явные и неявные методы, требующие s-кратного вычисления функции правой части на каждом шаге интегрирования. Эти методы отлично подходят для практических расчетов: они дают возможность просто заменять шаг интегрирования, считаются одношаговыми. Наиболее известная формула четырехэтапного метода Рунге – Кутты четвертого порядка.

Наиболее главной проблемой в использовании методов Рунге – Кутты и других явных методов является выбор величины шага интегрирования h, которая обеспечивает устойчивость вычислительной схемы [11,12].

С помощью усовершенствования данных формул был выведен класс неявных методов, у которых обозначенная проблема преимущественно снята. Неявные методы представляются алгебраическими нелинейными уравнениями относительно значений Y_{n+1}. К примеру, неявный метод Эйлера

$$Y(x_{n+1}) = Y(x_n) + hF(Y_{n+1}, x_{n+1}).$$

Вычислив приближение к решению в точках $x_1, x_2, ..., x_n$ можно использовать их для нахождения решения в точке x_{n+1}. Данная идея приводит к классу многошаговых методов, например, к методам Адамса. Формулы неявных методов Адамса-Мултона имеют вид [11,12]:

$$Y_{n+1} = Y_n + h\sum_{i=1}^{q}\beta_i F_{n-i+1}. \tag{3.3}$$

Создание многошаговых методов основано на применении полинома степени *q*. Приближенное значение решения $Y(x)$ в точке x_{n+1} представляется в виде линейной комбинации нескольких приближенных значений решения и его производной в данной и предыдущих *q* точках. Внедрение многошаговых формул ставит задачу вычисления *q*

штук исходных значений $Y_1, Y_2, ..., Y_q$, точность задания которых обязана быть не меньше точности соответствующей формулы. В общем виде полином, можно представить

$$Y_{n+1} - \sum_{j=1}^{q} \alpha_j Y_{i+1-j} - h \sum_{j=0}^{q} \beta_j F\left(x_{i+1-j}, Y_{i+1-j}\right) = 0 . \quad (3.4)$$

В отличие от одношаговых методов, определенных выражением (3.2), многошаговые методы, определенные выражением (3.4), не являются самоначинающимися. В работе [12] показано, что существуют многошаговые методы, с помощью которых можно начинать интегрировать систему ОДУ без использования в начале одношаговых методов в качестве разгона. Такие методы называются методами переменного порядка.

Описанная ранее сложность выбора шага интегрирования *h*, обеспечивающего численную устойчивость метода, делает неявные методы предпочтительнее для практического использования.

Существуют неявные методы Рунге – Кутты, основанные на формуле (3.3). Простейшим неявным методом Рунге – Кутты является модифицированный метод Эйлера. Он задается формулой:

$$y_{n+1} = y_n + h \frac{f\left(x_n, y_n\right) + f\left(x_{n+1}, y_{n+1}\right)}{2} . \quad (3.5)$$

Для реализации данного метода используются два вычисления функции на каждом шаге, прогноз и коррекция:

$$\tilde{y}_{n+1} = y_n + hf\left(x_n, y_n\right), \; y_{n+1} = y_n + h \frac{f\left(x_n, y_n\right) + f\left(x_{n+1}, \tilde{y}_{n+1}\right)}{2} .$$

Для увеличения точности итерации коррекции нужно сделать несколько раз. Улучшенный метод Эйлера имеет второй порядок точности. Преимуществом неявных методов Рунге – Кутты считается их большая устойчивость, что очень важно при решении жестких уравнений.

Еще одно семейство неявных методов, которое хорошо подходит для решения жестких задач, это семейство (m,k)-методов [17].

Пусть заданы целые положительные числа *m* и *k*, $k \le m$. Обозначим через M_m множество целых чисел *i*, удовлетворяющих условию $1 \le i \le m$, а через M_k и J_i – подмножества из M_m следующего вида:

$$M_k = \left\{ m_i \in M_m \,|\, 1 = m_1 < m_2 < ... < m_k \le m \right\},$$

$$J_i = \left\{ m_{j-1} \in M_m \,|\, j > 1, m_j \in M_k, m_j \le i \right\}, \; 2 \le i \le m.$$

Тогда семейство (m,k)-методов записывается в виде:

$$y_{n+1} = y_n + \sum_{i=1}^{m} p_i k_i, \; D_n = E - \alpha\tau f_n',$$

$$D_n k_i = \tau f\left(y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j \right) + \sum_{j \in J_i} a_{ij} k_j, \; i \in M_k, \; D_n k_i = k_{i-1} + \sum_{j \in J_i} a_{ij} k_j, \; i \in M_m \setminus M_k,$$

где τ – шаг интегрирования решаемой задачи; y_n –приближенное решение при $t = t_n$; *E* – единичная матрица размерности *n*; $f_n' = \dfrac{\partial f\left(y_n\right)}{\partial y}$ – матрица Якоби векторной функции $f\left(y\right)$; *a*, p_i, α_{ij} и β_{ij} – вещественные коэффициенты, определяющие свойства точности и устойчивости (m,k)-метода. Рассмотрим конкретный метод из этого семейства, а именно (4,2) – метод четвертого порядка точности. Он выглядит следующим образом:

$$y_{n+1} = y_n + \sum_{i=1}^{4} p_i k_i,\ D_n = E - \alpha\tau f_n',$$

$$D_n k_1 = \tau f\left(y_n\right),\ D_n k_2 = k_1,\ D_n k_3 = \tau f\left(y_n + \beta_{31}k_1 + \beta_{32}k_2\right) + \alpha_{32}k_2,\ D_n k_4 = k_3 + \alpha_{42}k_2,$$

а числовые коэффициенты имеют следующие значения:

$$a = 0.57281606248213,\ p_1 = 1.27836939012447,$$
$$p_2 = -1.00738680980438,\ p_3 = 0.92655391093950,$$
$$p_4 = -0.33396131834691,\ \beta_{31} = 1.00900469029922,$$
$$\beta_{32} = -0.25900469029921,\ \alpha_{32} = -0.49552206416578,$$
$$\alpha_{42} = -1.28777648233922.$$

Именно при таких значениях числовых коэффициентов схема обладает 4-м порядком точности.

Осуществление неявных вычислительных методов не является единственно возможной. Для поиска решений нелинейных уравнений используется метод Ньютона. При этом трудность нахождения начального приближения и общая мысль метода прогноз-коррекции остается без изменений. Переход к неявным методам в значимой степени убрал проблему выбора величины шага интегрирования *h* как фактора-стабильности метода. С другой стороны, это привело к проблемам выбора начального приближения и величины шага *h*, которые обеспечивают сходимость итерационного процесса решения нелинейных алгебраических уравнений. Понятно, что для линейных задач таких проблем нет и использование неявных вычислительных методов для них существенно проще и надежней.

Однако и у неявных схем есть свой недостаток. На каждом шаге итерации приходится решать приведенную систему нелинейных алгебраических уравнений, а это не всегда является возможным.

Розенброком был предложен класс неявных методов, в котором не решается система нелинейных уравнений. Рассмотрим основные идеи этого популярного алгоритма [18], основанного на приведении жестких дифференциальных уравнений к разностной форме типа:

$$\left(I - \alpha\Delta A - \beta\Delta^2 A^2\right)\frac{y_{i+1} - y_i}{\Delta} = f\left(y_i + \gamma\Delta f\left(y_i\right)\right) \tag{3.6}$$

Здесь Δ – шаг интегрирования, α, β, γ – некоторые параметры, а $A = \dfrac{\partial f\left(y_i\right)}{\partial y}$.

Если рассматривать не одно уравнение, а систему *N* уравнений, то *A* является матрицей размера *N*x*N*, составленной из частных производных (матрица Якоби). Конкретные значения параметров α, β, γ можно вычислить после разложения решения в ряд Тейлора в точке y_i, затем подставить его в (3.5). От значений этих параметров зависит максимально возможный порядок точности. Например, для схемы третьего порядка точности значения этих параметров будет следующим: $\alpha = 1.077$, $\beta = -0.372$, $\gamma = -0.577$.

Алгоритм Розенброка основывается на следующих действиях, которые выполняются на каждом *i*-м шаге интегрирования:

1. Вычисляется матрица производных в точке y_i.
2. Следующая точка y_{i+1} находится из матричного уравнения (3.6) с данными коэффициентами.

Метод Розенброка является одношаговым и явным. Впрочем, как видно из формулы метода, пересчет каждого шага требует: во-первых, численного определения производных функции f_y, и во-вторых, решения системы линейных уравнений.

Класс методов Розенброка является очень распространенным и востребованным, потому что имеет очень высокие вычислительные качества, которые позволяют найти решение за довольно быстрое время и с достаточно высокой точностью.

Подведем итоги по рассмотренным методам. Так, для решения задач с малым числом жесткости подойдут простейшие неявные методы: Эйлера, трапеций, прямоугольников. Лучший результат показывает (4,2)-метод т.к. он обладает четвертым порядком точности.

Для решения задач умеренной жесткости подойдут неявные методы Рунге – Кутты, а именно – неявный метод второго порядка.

Для решения задач высокой жесткости, если допустимо использование малых шагов интегрирования, предпочтительнее выбрать (4,2)-метод. При необходимости использования более крупных шагов следует применять комплексную схему Розенброка.

Заключение

В данной работе был описан процесс моделирования кинетики пластической деформации, основанный на процессах образования и аннигиляции точечных дефектов. Были представлены модели кинетики упрочнения и разрушения дальнего атомного порядка в случае сверхдислокационных источников и источников, испускающих одиночные дислокации. Эти модели представляют собой системы обыкновенных дифференциальных уравнений. Исходя из определений жесткой системы был сделан вывод что данные системы являются жесткими. Для нахождения численного решения жестких систем ОДУ были сформулированы два класса методов: явные и неявные. Наиболее предпочтительными в решении жестких систем являются неявные методы. В данной работе были рассмотрены наиболее востребованные семейства неявных методов, были представлены рекомендации по их выбору.

ЛИТЕРАТУРА

1. *Акулов Н.С.* Дислокации и пластичность.– Минск: Издательство АН БССР, 1961.– 109 с.

2. *Гилман Дж.* Микродинамическая теория пластичности // Микропластичность.–1972.– С. 18–37.

3. *Orlov A.K.* Kinetics of dislocations // Theory of crystals defects.– Prague: Publishing House of the Czechoslovak Academy of Sciences, 1966.– 317–338 pp.

4. *Lagneborg R.* Dislocation mechanisms in creep // Intern. Metals. Rev.– 1972.– Pp. 130–146.

5. *Ханнанов Ш.Х.* Кинетика дислокаций и неоднородная деформация кристаллов при одиночном скольжении // Математические модели пластичности.– Томск: Издательство ТПУ, 1991.– С. 11–16.

6. *Bergstrom J.* A dislocation model for the stress strain behaviour of polycrystalline

-Fe with special emphasis on the variation of the densities of mobile and immobile dislocations // Mater. Sci. and Eng.– 1970.– no. 4.– Pp. 193–200.

7. *Essmann V., Mughrabi H.* Annihilation of dislocations during tensile and cyclic deformation and limits of dislocation densities // Phil. Mag. (a).– 1979.– no. 6.– Pp. 731–756.

8. *Попов Л.Е., Кобытев В.С., Ковалевская Т.А.* Концепция упрочнения и динамического возврата в теории пластической деформации // Известия вузов. Физика.– 1982.– № 6.– С. 56–82.

9. *Большанина М.А., Большанина Н.А., Горелов И.К.* Влияние скорости деформации на механические свойства олова // ЖЭТФ.– 1934.– С. 1084–1089.

10. Математическое моделирование пластической деформации / Л. Е. Попов, Л. Я. Пудан, С. Н. Колупаева и др.– Томск: Издательство Томского университета, 1990.– 185 с.

11. *Вержбицкий В.М.* Численные методы (математический анализ и обыкновенные дифференциальные уравнения): Учеб. пособие для вузов.– М.: Высшая школа, 2001.–382 с.

12. *Ракитский Ю.В., Устинов С.М., Чернорудский И.Г.* Численные методы решения жестких систем.– М.: Наука. Глав. ред. физ.-мат. лит., 1979.– 208 с.

13. *Shampine L.F., Gear C.W.* A User's View of Solving Stiff Ordinary Differential Equations // SIAM Review.– 1979.–Jan.– Vol. 21, no. 1.– Pp. 1–17.

14. *Каханер Д., Моулер К., Нэш С.* Численные методы и программное обеспечение.– М.: Мир, 1998.– 320 с.

15. *Семенов М.Е.* Математическая модель и комплекс программ для исследования пластической деформации скольжения в материалах с гранецентрированной кубической структурой: дис. … канд. физ. –мат. наук. – Том. гос. архит.-строит. ун-т, Томск, 2005.

16. Математическое моделирование от междислокационных взаимодействий до макроскопической деформации [Текст] : монография / под ред. В.А. Старенченко. – Томск : Изд-во Том. гос. архит.-строит. ун-та, 2015. – 540 с.

17. *Галанин М.П., Ходжаева С.Р.* Методы решения жестких обыкновенных дифференциальных уравнений. Результаты тестовых расчетов // Препринты ИПМ им. М.В.Келдыша. 2013. № 98. 29 с.
18. *Кирьянов Д.В., Кирьянова Е.Н.* Вычислительная физика – М.: Полибук Мультимедиа, 2006. – 352 с

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ОДНОВРЕМЕННОЙ (ПАРАЛЛЕЛЬНОЙ) ДОСТАВКИ ВИДЕО-КОНТЕНТА В НЕСКОЛЬКО СЕТЕЙ С ПОДДЕРЖКОЙ АДАПТИВНОГО БИТРЕЙТА

Хамуев В.В., Буторина Н.Б.
Томский государственный университет
manyaioan@mail.ru, nnatta07@mail.ru

Введение

Видеоконтент уже давно доминирует в общем объеме передаваемых данных в сети Интернет. По прогнозам компании Cisco, к 2021 г. общий объём интернет-трафика составит 3,3 зеттабайта, из которых 82% будет составлять передача видео (по сравнению с 73% сегодня).

Производителей живого видеоконтента можно разделить на следующие группы:

1. Пользователи, для которых создание и распространение видеоконтента – профильный род деятельности (телеканалы, интернет-порталы, видеоблогеры и т.п).

2. Пользователи, для которых создание и распространение видеоконтента – непрофильный род деятельности (организаторы спортивных и образовательных мероприятий, вебинаров и т.п).

Общей проблемой для этих групп является фрагментированность аудитории между разными социальными сетями. Инструментов для решения такой проблемы немного, в то же время они не адаптированы для "мобильного" использования.

1. Проведение аналитики

Программные комплексы, позволяющие вещать видеоконтент в социальные сети, и другие сервисы доставки видео контента по протоколу RTMP можно разделить на три основных сегмента. Ниже будут рассмотрены каждый сегмент с изучением его самого популярного представителя.

1. Бесплатное ПО с открытым исходным кодом.

Пример: Open Broadcast Project. Программный продукт для организации онлайн-вещания, поддерживающее передачу видео по протоколу RTMP. Большая часть трансляций с персональных компьютеров сегодня осуществляется через это приложение. **Плюсы:** бесплатность, наличие базового функционала, открытый исходный код. **Минусы:** отсутствие поддержки, долгий процесс внедрения нового функционала.

2. Коммерческое “коробочное ПО”.

2.1 Без явной целевой аудитории (**пример** – Telestream Wirecast). ПО для организации онлайн-вещания, поддерживающее передачу видео по протоколу RTMP. Лидер рынка, предоставляющий максимальную функциональность для решения разных задач вещания. Стоимость: от $495 до $2269. **Плюсы:** большое число поддерживаемых сервисов, наличие большого числа функций. **Минусы:** высокая стоимость, отсутствие гибкого ценообразования, перегруженный и неинтуитивный интерфейс.

2.2 ПО с четко выраженной целевой аудиторией (**пример** – XSplit). Это приложения, ориентированные на одну категорию создателей контента. Свои приложения есть для игровых трансляций, видеоподкастов. Стоимость: от 2,5 до 8 долларов в месяц. **Плюсы:** функционал, наиболее востребованный для той аудитории, для которой создано ПО, гибкая ценовая политика. **Минусы:** не подходят для универсального применения.

3. Сервисы для обработки уже готового видеосигнала и его ретрансляции в несколько сетей. **Примеры:** DaCast, Restream. Облачные сервисы, позволяющие ретранслировать уже закодированный сигнал в несколько сетей доставки контента. Используются в связке с ПО, описанным выше для решения проблемы фрагментации зрителей по разным платформам. Стоимость: от 0 до $8 в месяц. **Плюсы:** гибкие тарифы, облачное транскодирование. **Минусы:** необходимо покупать или устанавливать ПО для первоначального кодирования видео, работают только в связке с ним.

2. Формирование технического задания для реализации программного комплекса

В ходе исследования уже существующих ПО был сформирован следующий ряд требований к разрабатываемому программному комплексу.

Требования по обработке входящего видеопотока:

1. Поддержка захвата входного аудиосигнала с микрофона и выбора источника если их несколько

2. Поддержка входного видеосигнала из следующих источников: Web-камеры, экрана, файла.

3. Поддержка выбора изображения для последующего наложения поверх видео в качестве логотипа или водяного знака

4. При захвате экрана пользовательского устройства, на котором установлен программный продукт, необходимо обеспечить возможность разных режимов захвата экрана, а именно: захват всего экрана, с возможностью выбора экрана, если у пользователя их несколько; выбор разрешения, в котором видеоизображение будет транслироваться.

5. Для трансляции видеосигнала из медиафайла необходимо реализовать поддержку следующих форматов медиафайлов: .ts, .mp4, .mov, .avi, .flv, .mkv, .aac, .ogg, .mp3.

Требования к исходящему видеопотоку со стороны социальных сетей

В ходе технологической аналитики были уточнены требования к исходящему видеосигналу со стороны целевых социальных сетей (Facebook, Vk.com, Twitch, Youtube).

Разрешение видео	1280*720, 1920*1080, 3840*2160, 2560*1440, 854*480, 640*360, 426*240
Частота кадров	25 – 60
Минимальное количество ключевых кадров (I-frame) в секунду	0.5 – 1
Максимальный битрейт для видео	4000–9000 Кбит/с
Максимальный битрейт для аудио	128 Кбит/с
Поддерживаемый кодек для видео	H.264
Поддерживаемый кодек для аудио	AAC
Максимальная продолжительность	4 – 48 часов

Требования к программному комплексу при передаче выходного видеопотока

Программный комплекс должен использовать для вещания протокол RTMP (Real Time Messaging Protocol). Также иметь возможность вести вещание одновременно (параллельно) в несколько сервисов доставки видео контента. Выбор протокола передачи обусловлен тем, что большинство социальных сетей и других сервисов доставки видео контента используют именно его. Также для отказоустойчивости и бесперебойности вещания необходима поддержка адаптивного битрейта.

Дополнительные требования

Программный комплекс должен: иметь возможность сохранять трансляцию на компьютере пользователя; иметь возможность выводить транслируемый видеопоток в программный интерфейс; уметь анализировать технические характеристики компьютера, на котором он запущен, для установки рекомендуемых настроек качества транслируемого видео-контента.

3. Стек технологий используемых в программном комплексе

Выше был описан ряд требований к программному комплексу. Для их успешного выполнения необходимо было разработать или использовать уже готовое мультимедийное “ядро”.

Выбор мультимедийного ядра

На сегодняшний момент существует два основных мультимедийных фреймворка: FFmpeg и GStreamer. После анализа характеристик FFmpeg и GStreamer в качестве ядра было решено использовать GStreamer в силу его кроссплатформенности, наличия богатого выбора модулей для решения самых разнообразных задач, более низкого порога вхождения. К тому же, используя FFmpeg придется искать дополнительное решение для обеспечения аудио/видео ввода и вывода.

Выбор технологии для реализации GUI

Не смотря на то, что для полноценной работы в связке GStreamer и Qt придется решить ряд проблем, преимущества Qt стали решающими при его выборе для реализации GUI (Graphical User Interface).

Qt – кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++. Основные критерии выбора Qt: кроссплатформенность, общая кодовая база C/C++, легкость разработки программного интерфейса за счет хорошей документации и огромного выбора готовых компонент

4. Архитектура Pipeline

Pipeline – коллекция соединённых между собой элементов.

В GStreamer элементы связываются друг с другом, тем самым образуя цепочки, по которым идет видео или аудио поток. Это обусловлено тем, что в Gstreamer используется конвейерная архитектура для организации приложения. Для синхронизации у каждого элемента есть состояния, и оно распространяется от родителя к дочернему элементу, тем самым синхронизируя весь Pipeline [1].

Рассмотрим общую схему всего Pipeline, используемого в нашем программном комплексе.

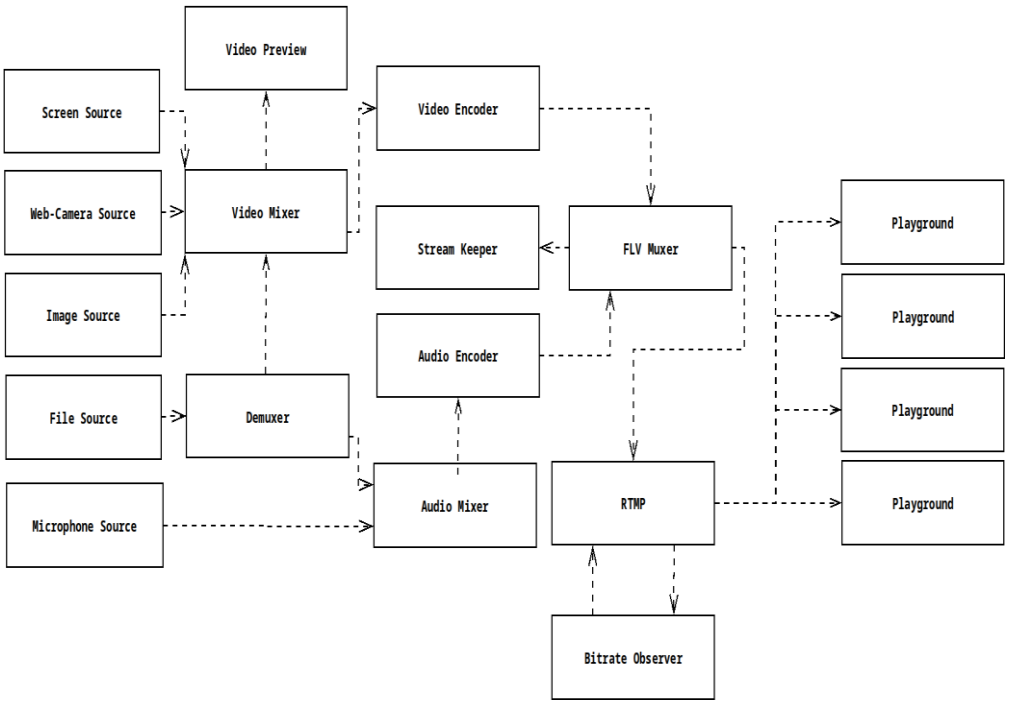


Рис. 1. Общая схема Pipeline

Каждый элемент на рис. 1 представляет собой Bin-контейнер, содержащий в себе несколько элементов. Это необходимо, так как для каждой отдельной операции, которую исполняет тот или иной модуль, требуется предварительная подготовка или преобразование данных, создание очередей для синхронизации и/или стабилизации.

Теперь перейдем к более детальному рассмотрению механизмов в нашем Pipeline и его отдельных модулей. Из источников (Screen, File, и т.д) поток данных выходит в “сыром виде”(video/x-raw или audio/x-raw). Такой формат данных еще не готов для передачи на в сети доставки контента, но уже может использоваться для дальнейшей обработки.

Структура контейнера Screen Source

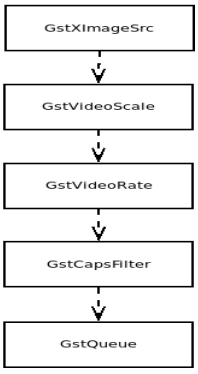


Рис. 2. Детальная схема Screen Source модуля

GstXImageSrc – это модуль, реализующий захват экрана в операционной системе (далее ОС) Linux.

GstVideoScale – это модуль, позволяющий наложить на видеопоток CapsFilter и впоследствии производить манипуляции с разрешением.

GstVideoRate – это модуль, позволяющий стабилизировать частоту кадров до установленной границы

GstQueue – это модуль, реализующий очередь, который может накапливать внутри себя видеопоток, в данной цепочке он необходим для стабилизации и синхронизации видеопотока

Примерно таким же образом устроены все остальные Source контейнеры, кроме FileSource, внутри него расположен Demuxer [2] – модуль, позволяющий получить отдельно видео и аудио дорожку из медиаконтейнера, а также декодер для этих дорожек, чтобы впоследствии без проблем преобразовать в используемый нами формат.

Структура контейнера Video/Audio Encoder

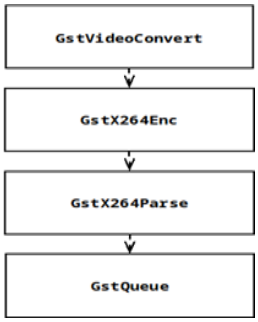


Рис.3. Video Encoder

GstVideoConvert – этот модуль используется для преобразования видеокадров из одного формата в другой. Может выполнять преобразование: видео формата, видео цветового пространства, цветность–размещение, размер видео.

GstX264Enc – этот модуль кодирует необработанное видео в сжатые данные H264, также известные как MPEG–4 AVC (Advanced Video Codec). Является хорошим выбором, так как при высокой степени сжатия сохраняет высокое качество.

GstX264Parser – Модуль реализующий синтаксический анализ потока битов как в формате AVC (с префиксом длины), так и в приложении B (префикс 0x000001 начального кода).

Video Encoder будет кодировать сырой видео поток с помощью стандарта H.264, а также выставлять битрейт и количество I–кадров и другие настройки необходимые для процесса кодирования [3].

Схожим образом устроен и Audio Encoder, его основа это модуль GstVoAacEnc который кодирует аудио с помощью кодека AAC(Advanced Audio Coding) отличается высокой эффективностью кодирования для стационарных и переходных сигналов, простой банк фильтров и лучшая управляемость частотами выше 16 кГц, поддерживает качество почти неотличимое от оригинальных аудио источников [4].

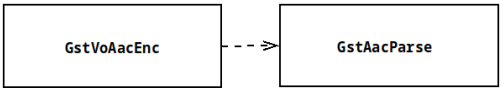


Рис. 4. Audio Encoder

GstAacParse - ищет блоки AAC в формате ADTS(Audio Data Transport Stream). Заголовок ADTS присутствует в AAC перед каждым необработанным участком данных или блоком от 2 до 4 таких участков данных во фрейме, что позволяет получить хорошую стабильность в таких средах, как мобильная связь или Интернет. Из-за этих дополнительных данных битрейт возрастает примерно на 2–3 kbps.

Решение задачи о параллельной доставке видео-контента в несколько сетей

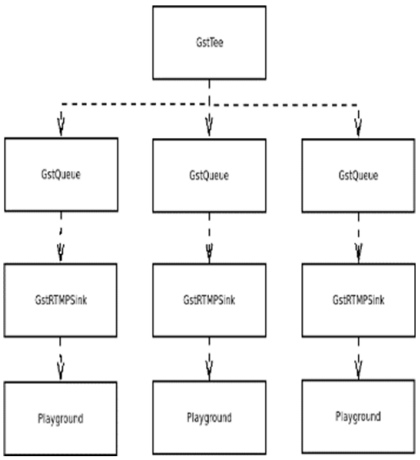


Рис. 5. Детальная схема RTMP модуля

Ключевым элементом в решении задачи о одновременном (параллельном) вещании потока на несколько сервисов является GstTee – это модуль который позволяет распараллеливать входящий в него поток данных. Тем самым модуль дает возможность вещать на любое количество сервисов, ограничиваясь лишь пропускной способностью соединения на устройстве. Модуль, реализующий очередь, в данной цепочке необходим для стабилизации и синхронизации данных в случае, если GstRTMPSink не будет успевать, так как encoder и muxer отдает данные неравномерно, в отличие от GstRTMPSink.

Адаптивный битрейт

Битрейт – количество бит, используемых для передачи/обработки данных в единицу времени. При передаче видеопотока может возникнуть ситуация, когда пропускная способность канала может упасть, что может повлечь за собой обрыв трансляции или возникновение задержек. Для решения данной проблемы был разработан модуль Bitrate Observer, который необходим для под-

держки адаптивного битрейта. Данный модуль анализирует состояния очередей в цепочке Pipeline и принимает решение о понижении или повышении битрейта.

Другие модули используемые в Pipeline

Audio/Video Mixer – назначение этих модулей в объединении нескольких аудио или видео потоков в один путем накладывания друг на друга.

Demuxer – модуль для разделения данных в медиа контейнере, необходим для разделение видео и аудио дорожки.

FLV Muxer – модуль для упаковки видео и фудио дорожки в медиаконтейнер в формате flv, который является стандартом для большинства сервисов распространения live-контента.

Stream Keeper – реализует сохранение трансляций на компьютере пользователя.

Video Preview – отображает видео поток, который получается после трансформации и наложения.

Модуль анализа технических характеристик персонального компьютера

В технических характеристиках, предъявляемых к программному комплексу, было указана возможность выставления рекомендованных значения для качества передаваемого видео-контента. Для реализации этой задачи был разработан модуль для анализа технических характеристик ПК, на котором происходит запуск программного комплекса. Важными характеристиками для анализа являются: объем оперативной памяти, модель процессора, количество ядер процессора, частота процессора. Для их получения решено было парсить ответ консольных команд, которые входят в комплект операционной системы. На основе количества и частоты ядер, а также объема оперативной вычисляется рекомендуемое значение качества кодирования и битрейт.

Разработка графического интерфейса пользователя

Основной идеей при проектировании интерфейса была простота работы с программой от этапа настройки до запуска вещания. Для ее достижения была применена концепция User Interface Wizard – пошаговая работа с приложением и его настройками в отдельных окнах.

Интерфейс был разбит на три блока:

1. Блок настройки сервисов и параметров исходящего видеопотока
2. Блок с параметрами устройств-источников
3. Блок с информацией о трансляции и ее управлением

Примечательным в реализации блока 2 и 3 является отображение передаваемого видеопотока в режиме реального времени, что достигается с помощью модуля Video Preview, передающего “сырой” видеопоток на QWidget , который занимается его отрисовкой.

Заключение

В данной работе был проведен анализ существующего программного обеспечения для вещания видеопотока через протокол RTMP. По итогам исследования был составлен ряд технических критериев, на основании которых был разработан программный комплекс для вещания видео-контента в несколько сетей доставки контента одновременно (параллельно). Программный комплекс обладает простым и интуитивно понятным интерфейсом с широким спектром возможностей, среди которых выделяется поддержка адаптивного битрейт, обеспечивающая высокую стабильность при передаче видео-контента.

ЛИТЕРАТУРА

1. *Wim Taymans, Steve Baker, Andy Wingo, Ronald S. Bultje, Stefan Kost.* GStreamer 1.10 Application Development Manual, 2017. - 160c.

2. *Edward J. Giorgianni, Thomas E. Madden.* Digital Color Management: Encoding Solutions, 2009. - 432c.

3. *Ричардсон Я.* Видеокодирование. H.264 и MPEG-4 - стандарты нового поколения. - М.: ТЕХНОСФЕРА, 2005. - 368с.
4. *Сэломон Д.* Сжатие данных, изображений и звука: Пер. с англ. - М.: Техносфера, 2004. - 368с.

ПАРСЕР ДЛЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ RHINESTONE

Чалых Е.П., Самохина С.И.
Томский государственный университет
egor.chalyh.98@gmail.com, sv.sam.tsk@gmail.com

Введение

На данный момент в мире существует большое число языков программирования, каждый из них создан с определённой целью. Изначальной целью проекта было создание мощного и быстрого языка программирования с поддержкой математических вычислений, таких как дифференцирование, интегрирование, работа с матрицами и другие. Но первая проблема нового языка возникла при выборе его типа: интерпретируемый или компилируемый. Очевидный минус первого типа заключается в том, что процесс выполнения кода будет довольно долгим, особенно на очень больших файлах исходного кода. Но главное достоинство – интерпретатор языка можно сделать кроссплатформенным. Компилируемый язык программирования будет работать гораздо быстрей, т.к. из исходного кода мы получим сразу исполняемый файл. А минус такого языка, естественно, платформозависимость.

1. Постановка задачи

Первый этап создания языка программирования – это создание работоспособной, желательно кроссплатформенной, виртуальной машины [1]. Второй – создание компилятора [2], который будет транслировать файлы с текстом исходного кода в код понятный для виртуальной машины.

Скорость работы компилятора можно повысить, если использовать язык C++ [3]. Кроссплатформенности можно добиться, используя систему сборки CMake [4], она сама выберет компилятор в зависимости от операционной системы, а также подготовит проект к сборке. Инструментом создания языка является мощная интегрированная среда разработки CLion [5] от компании JetBrains [6].

2. Реализация языка программирования

Язык реализован на языке C++, с использованием стандарта STDLIB 20. Исходный код реализации распространяется в рамках проекта OpenRSP (Open RhineStone Project) [7]. Это проект с открытым исходным кодом под лицензией Apache License 2.0 [8]. В нем есть 3 подпроекта: libs, vm, compiler. Первый – набор дополнительных библиотек, используемых в vm и compiler. Второй – реализация виртуальной машины RhineStone VM. Третий – реализация компилятора.

RhineStone VM – портированная на язык C++ виртуальная машина EmeraVM [9].

Компилятор разделен на 3 части. Первая – преобразование текста в структуру данных, которую понимает компьютер, в данном случае абстрактное синтаксическое дерево (далее АСД). Вторая – анализ АСД и добавление дополнительной информации (атрибутов). Третья – сборка дерева в байт-код, который будет выполнять виртуальная машина. Рассмотрим первый компонент компилятора – парсер.

2.1. Реализация парсера для языка программирования

Парсер – инструмент, который позволяет преобразовать текстовые данные в АСД. Парсинг – процесс такого преобразования. Он разделен на лексический и синтаксический анализы.

2.2. Лексический анализ

Лексический анализ – процесс выделения из текста определенных лексем (или токенов). Эти лексемы содержат информацию о типе лексемы (ключевое слово, оператор и т.п.) и значение лексемы (например, если тип – число, то значение – последовательность цифр). Также в лексеме можно хранить данные о местоположении в исходном коде. Лексический анализатор можно сгенерировать с помощью специальных утилит, например, ANTLR [10], GNU Bison [11] и других. Исходными данными для таких утилит служит описание грамматики в форме Бэкуса-Наура (БНФ) [12], иногда используется расширенная БНФ. Результатом выполнения утилиты является лексический анализатор на основе детерминированного конечного автомата [13]. Очевидным плюсом такого подхода является скорость работы автомата. А минус – очень сложно отлаживать программу, а также для работы такого анализатора требуется дополнительная библиотека из утилиты. Второй способ создать лексический анализатор – это самостоятельно описать грамматику языка программным способом, т.е. определить правила на языке C++ для каждого входного символа так, чтобы лексемы выделялись однозначно.

Принцип такого подхода следующий. Читаем символ из входного файла. Определяем его тип, например, пробел – разделитель, его мы пропускаем, цифра – является частью числа, значит применяем правило чтения числа, прочли букву – часть идентификатора или ключевого слова, и т.д.

Рассмотрим правило чтения идентификатора. У нас есть его начало, теперь продолжаем читать символы пока они удовлетворяют следующим требованиям: буква, цифра, знак нижнего подчеркивания, знак доллара. После прочтения у нас есть последовательность символов для идентификатора. Если эта последовательность принадлежит таблице зарезервированных слов, то лексема – ключевое слово, иначе идентификатор.

Для операторов принцип такой же, но для однозначности определения выбирается последовательность максимальной длины. Пример: «a >>> b». Символы a и b определятся как идентификаторы, пробелы пропустятся. Язык в своем наборе имеет операторы «>>» (сдвиг вправо) и «>» (больше чем), оператора «>>>» нет. Поэтому список лексем для этого выражения будет следующим: идентификатор «a», оператор «>>», оператор «>», идентификатор «b».

2.3. Абстрактное синтаксическое дерево

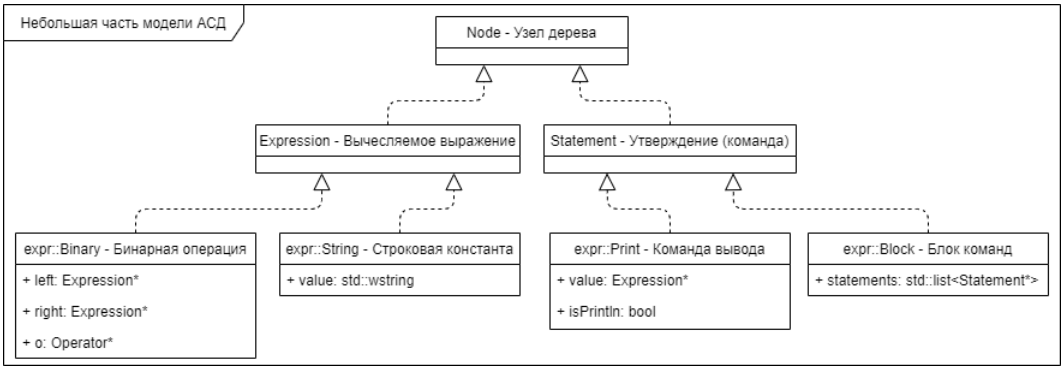
Представить исходный код программы с помощью дерева – логичный и очевидный шаг. Например, определение цикла While: условие и тело цикла. Узел будет состоять из узла для условия, и узла для тела. Подобно While можно представить условный оператор, определение функции и т.д.

Анализ дерева происходит путем его прямого обхода с помощью специального интерфейса «Посетитель» (IVisitor). С помощью этого интерфейса можно также перестраивать дерево (например, для оптимизации), преобразовывать в другие структуры данных (например, XML дерево, или string).

Это дерево является синтаксическим, т.к. было построено согласно определенным синтаксическим правилам. Пример правила для цикла While: ключевое слово «while», оператор «(«, правило «условное выражение», оператор «)», правило «последовательность команд». Правила «условное выражение» и «последовательность команд» также определены. Соответственно полученное дерево исключает в себе синтаксические ошибки.

Абстрактность понимается как отсутствие дополнительной информации в узле. Эта информация образуется после всех анализов дерева и используется в дальнейшем компилятором. Например, использование переменной до её объявления, можно выявить на этапе анализа дерева, и вывести соответствующую ошибку. Или определить общий тип

операции над числами типа integer и double. На этапе создания АСД, это выявить сложно. После всех возможных анализов дерево становится атрибутным деревом разбора.



2.4. Синтаксический анализ

На вход для анализа подается список лексем, полученных после лексического анализа, а на выходе получается АСД. Такое преобразование достигается описанием синтаксических правил методом рекурсивного спуска. Каждое правило должно определяться однозначно и порождаться одной или несколькими лексемами. Например, правило для While порождается лексемой «ключевое слово «while»» и определяется однозначно как While.

Принцип метода рекурсивного спуска следующий. Обозначим лексемы как терминальные (конечные) символы, а правила – нетерминальные символы. Берем один терминальный символ, определяем его принадлежность правилу, и спускаемся к этому правилу. В правиле берется следующий нетерминальный символ, и также идет спуск, пока правило полностью не определится как элемент АСД. Пример правила «блок или команда»: (терминал «{», нетерминал «команда» [0; N], терминал «}») или (нетерминал «команда»). При спуске в это правило мы проверим лексему, если встретилась фигурная открывающаяся скобка, то будем применять правило «команда» до тех пор, пока не встретится фигурная закрывающаяся скобка. Соответственно если не встретилась фигурная открывающаяся скобка, то выберется альтернатива – правило «команда», в которую спускаемся.

На этом этапе мы можем выделить синтаксические ошибки такие как неожиданный конец файла (когда правилу требуется больше лексем, чем есть), неожиданная лексема (например, после слова if должна стоять кругла скобка, а стоит точка с запятой) и другие.

3. Особенности парсера

1. По мере преобразования лексем актуальность прочитанных лексем теряется, поэтому парсер их удаляет, освобождая при этом память.

2. Есть возможность парсить несколько файлов асинхронно, но это нужно явно указывать в настройках компилятора.

3. Язык поддерживает арифметические выражения внутри строк, соответственно есть специальный подпарсер для этой цели. Пример "a + b = ${a + b}". Это выражение является константной строкой, но часть «${a + b}» будет посчитана при выполнении программы, и подставится в строку.

4. Внедрение новых правил требует небольших изменений в других правилах, как минимум в одном.

4. Примеры

В следующих примерах показана малая часть возможностей парсера, которых нет в других языках программирования.

4.1. Пример объявления функции

Возможности парсера позволяют объявить функцию несколькими способами:

1) func NAME(): (RESULT_VAR_NAME) {
   RESULT_VAR_NAME = SOMETHING
   }
   NAME – имя функции.
   RESULT_VAR_NAME – имя переменной для возвращаемого значения (может быть не одна).
   SOMETHING – какое-то определенное значение.

2) func NAME() = SOMETHING
   Если функция делает какое-либо простое арифметическое действие, то можно её таким образом сократить.

3) func NAME = SOMETHING
   Если у функции нет параметров, то скобки можно опустить.

4.2. Пример объявления функции с несколькими возвращаемыми значениями

```
func `111 my awesome function`(a, b): (quotient, modulo) {
  quotient = a \ b // частное ( \ - целочисленное деление)
  module = a % b // остаток от деления
}
```

Последовательность символов, заключенная в такие кавычки `` (косой апостроф), является идентификатором.

Вызвать такую функцию можно двумя способами:

1) a = 10
   q, m = `111 my awesome function`(a, 2)
   Этот способ поместит результат выполнения в переменные q и m
2) a, q, m = (1, 0, 0)
   `111 my awesome function`(a, q, m)
   В этом способе явно передаются переменные, в которые нужно сохранить результат

Заключение

Целью парсера является построение АСД на основе входных текстовых данных, выявление лексических и синтаксических ошибок и не более. Дальнейшая работа с АСД лежит уже на других компонентах компилятора.

В настоящей работе рассмотрен один из способов создания парсеров для языков программирования. Метод рекурсивного спуска очень прост в написании, но программист должен уверенно знать все аспекты языка для созданий правил.

ЛИТЕРАТУРА

1. Виртуальная машина [Электронный ресурс]: сайт / Википедия – URL: https://ru.wikipedia.org/wiki/Виртуальная_машина (дата обращения: 25.05.2020).

2. Компилятор [Электронный ресурс]: сайт / Википедия – URL: https://ru.wikipedia.org/wiki/Компилятор (дата обращения: 25.05.2020).

3. Язык программирования C++ [Электронный ресурс]: сайт / Википедия – URL: https://ru.wikipedia.org/wiki/C%2B%2B (дата обращения: 25.05.2020).

4. CMake [Электронный ресурс]: сайт / Википедия – URL: https://ru.wikipedia.org/wiki/CMake (дата обращения: 25.05.2020).

5. CLion [Электронный ресурс]: сайт / JetBrains – URL: https://www.jetbrains.com/clion (дата обращения: 25.05.2020).
6. JetBrains [Электронный ресурс]: сайт / JetBrains – URL: https://www.jetbrains.com (дата обращения: 25.05.2020).
7. OpenRSP [Электронный ресурс]: сайт / GitLab – URL: https://gitlab.com/rhinestone-project/openrsp (дата обращения: 25.05.2020).
8. Apache License 2.0 [Электронный ресурс]: сайт / Apache – URL: https://www.apache.org/licenses/LICENSE-2.0 (дата обращения: 25.05.2020).
9. *Чалых Е.П., Самохина С.И.* Виртуальная машина EmeraVM //Труды Томского государственного университета. – Т. 304. Серия физико-математическая: Математическое и программное обеспечение информационных, технических и экономических систем : материалы VII Междунар. молодежной науч. конф. Томск, 23-25 мая 2019 г. Томск: Издательский Дом Томского государственного университета, 2019. С. 210-215.
10. ANTLR [Электронный ресурс]: сайт / ANTLR – URL: https://www.antlr.org (дата обращения: 25.05.2020).
11. GNU Bison [Электронный ресурс]: сайт / GNU – URL: https://www.gnu.org/software/bison (дата обращения: 25.05.2020).
12. БНФ [Электронный ресурс]: сайт / Википедия – URL: https://ru.wikipedia.org/wiki/Форма_Бэкуса_—_Наура (дата обращения: 25.05.2020).
13. ДКА [Электронный ресурс]: сайт / Википедия – URL: https://ru.wikipedia.org/wiki/Детерминированный_конечный_автомат (дата обращения: 25.05.2020).

RESEARCH OF THE CONVERGENCE OF THE FLEXIBLE TOLERANCE METHOD DEPENDING ON THE PARAMETERS VALUES

Alimbaeva E.A., Balashova O.M., Keba A.V.
Tomsk State University
alimb97@mail.ru, balashovajkz@mail.ru, mir.na.mig7@mail.ru

Introduction

There are a considerable number of algorithms for solving the general nonlinear programming problem, which includes the optimization of an objective function subject, in the most general case, to both equality and inequality constraints.

Formally, the nonlinear programming problem can be formulated as follows [2]:

$$\text{minimize } f(\mathbf{x}),\ \mathbf{x} \in E^n, \tag{1}$$

with m linear and (or) nonlinear constraints in the form of equalities

$$h_i(\mathbf{x}) = 0,\ i = \overline{1,m}, \tag{2}$$

and $(p-m)$ linear and (or) nonlinear constraints in the form of inequalities

$$g_i(\mathbf{x}) \geq 0,\ i = \overline{m+1,p}. \tag{3}$$

Nonlinear programming techniques can be roughly divided into two broad categories: 1) direct search methods that depend upon a direct comparison of the values of the objective function; 2) gradient methods that seek the extremum by using first- and perhaps second-order derivatives [1]. Direct-search algorithms are more time consuming in their execution, but the net cost, including preparation time, for the solving the problem may be less than for gradient methods. Thus, the algorithm presented in the article was developed in accordance with the direct-search logic.

This work is focused in constrained nonlinear optimization using the Flexible Tolerance Method (FTM) proposed by Paviani and Himmelblau [5] for minimization of a functional subject to nonlinear equality and inequality constraints.

In this article, the convergence of FTM is investigated depending on the selected combination of parameter values. The article is organized as follows. In section 1, we provide statement of the problem. Section 2 presents a description of FTM and the Nelder-Mead Method (or the Flexible Polyhedron Method or FPM) and a flowchart of FTM. Section 3 illustrates the numerical results, where we provide various figures for different values of the model parameters, and we also numerically compare our results with exist in the literature.

Acceptable parameter values are presented for the two-dimensional cases of FTM based on the results obtained during the experiment.

1. Statement of the problem

FTM is a direct method of optimization, it improves the value of the objective function by using information provided by feasible points, as well as certain non-feasible points termed near-feasible points [2]. On the other hand, FTM employs a tolerance method for the constraint violation throughout the whole search, and thereby causes near-feasibility limits to be less as the search proceeds toward the optimum solution, until at the limit only feasible **x** vectors in the model are accepted [6].

Because of this strategy, the original problem (1), (2), (3) can be replaced by:

$$f(\mathbf{x}) \to \min,\ \mathbf{x} \in E^n,$$

$$\Phi^{(k)} - T(\mathbf{x}) \geq 0,$$

where $\Phi^{(k)}$ is flexible tolerance criterion at the *k*-th stage search, $T(\mathbf{x})$ is a positive function for all equality and/or inequality constraints of the problem, used as a degree measurement of the restriction violation extension. The tolerance criterion at the *k*-th iteration is given by equation (5) and the functional $T(\mathbf{x})$ is described by equation (4), which will be presented below.

FTM uses two searches to satisfy feasibility constraint. The external search is a variation of FPM [4]. This one seeks to minimize the objective function. The internal search minimizes the value of the positive function for all equality constraints and / or problem inequalities. This internal search can be performed by any unconstrained nonlinear optimization method.

In this article, the convergence of FTM depending on the parameters α, β, γ values is researched and the operating time of the algorithm for different parameters values is compared.

2. The Flexible Tolerance Method

FTM is a direct method of optimization that provides some advantages, such as simplicity, the ability to lead with equality and inequality constraints without employs derivative calculus. Consider this method in more detail.

2.1. Algorithm of the Flexible Tolerance Method

The algorithm of flexible tolerance increases knowledge about the objective function by using the information provided by feasible points, as well as some non-feasible points, called near-feasible points. FTM exploits the promising neighborhood individual by a search mechanism and minimizes a constraint violation of an objective function by a flexible tolerance criterion for near-feasible points. The algorithm for finding the optimal value by FTM is shown in Figure 1.

Consider the functional $T(\mathbf{x})$ over the set of constraints of the problem in the form

$$T(\mathbf{x}) = \sqrt{\sum_{i=1}^{m} h_i^2(\mathbf{x}) + \sum_{i=m+1}^{p} \chi_i g_i^2(\mathbf{x})}, \tag{4}$$

where χ_i – Heaviside operator: $\chi_i = \begin{cases} 0, & g_i(\mathbf{x}) \geq 0 \\ 1, & g_i(\mathbf{x}) < 0 \end{cases}$.

Consider the function Φ in the following form:

$$\Phi^0 = 2(m+1)t,$$

$$\Phi^{(k)} = \min\left\{\Phi^{(k-1)}, \frac{m+1}{r+1}\sum_{i=1}^{r+1}\left\|\mathbf{x}_i^{(k)} - \mathbf{x}_{r+2}^{(k)}\right\|\right\},\ k \geq 1, \tag{5}$$

where m is the number of equality constraints, t is the size of the original polyhedron, $\Phi^{(k-1)}$ is the value of the tolerance criterion in the $(k-1)$-th stage of the search, r is the number of degrees of freedom of the objective function $f(\mathbf{x})$ in the problem (1), (2), (3), and $r = n - m$, where n is the number of variables. $\mathbf{x}_i^{(k)}$ is the i-th vertex of the polyhedron in E^n, $\mathbf{x}_{r+2}^{(k)}$ is the vertex corresponding to the centroid of the polyhedron in E^n.

The tolerance criterion $\Phi^{(k)}$ behaves as a positive decreasing function of $\mathbf{x}$, and the vector $\mathbf{x}^{(k)}$ is classified as follows:

1) Feasible, if $T(\mathbf{x}^{(k)}) = 0$

2) Near-feasible, if $0 < T\left(\mathbf{x}^{(k)}\right) \le \Phi^{(k)}$

3) Non-feasible, if $T\left(\mathbf{x}^{(k)}\right) > \Phi^{(k)}$

The tolerance for near-feasible solutions is decreased until the limit when only feasible solutions are allowed: $\lim_{\mathbf{x} \to \mathbf{x}^*} \Phi^{(k)} = 0$, where $\mathbf{x}^*$ is the solution vector, within tolerance ε.

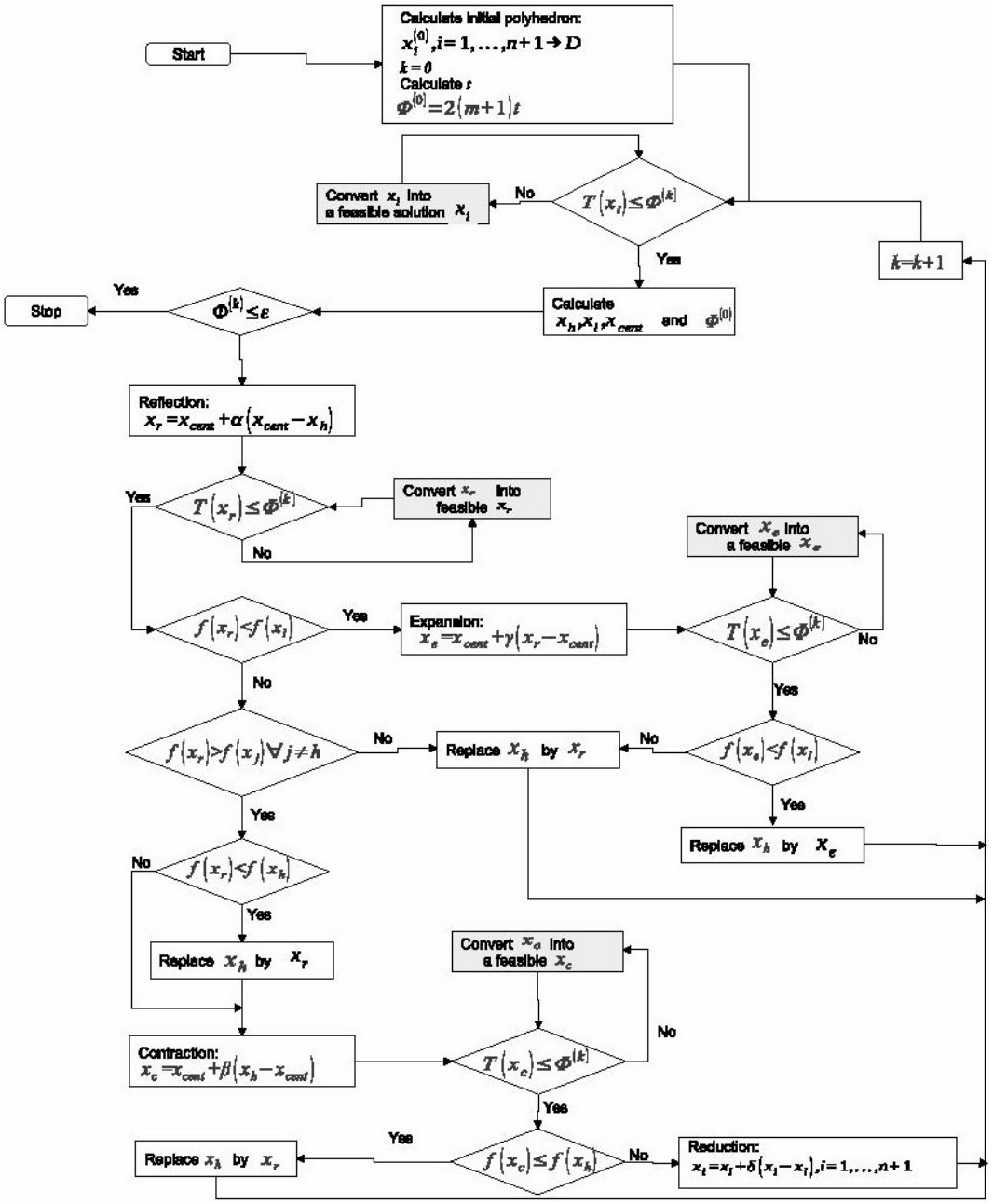


Fig. 1. The flowchart of FTM

2.2. The Flexible Polyhedron Method

Algorithm of FTM based on FPM. The FPM minimizes of n independent variables using $(n+1)$ vertices of a flexible polyhedron. The FPM starts with a set of $(n+1)$ vectors that represents the vertices of a regular simplex, described by matrix **D**, in which the columns represent the components of the vertices and the rows represent the coordinates:

$$\mathbf{D} = \begin{bmatrix} 0 & d_1 & d_2 & \dots & d_2 \\ 0 & d_2 & d_1 & \dots & d_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & d_2 & d_2 & \dots & d_1 \end{bmatrix}_{n\times(n+1)},$$

where $d_1 = \frac{t}{n\sqrt{2}}\left(\sqrt{n+1}+n-1\right)$ and $d_2 = \frac{t}{n\sqrt{2}}\left(\sqrt{n+1}-1\right)$, t is the distance between two vertices.

The procedure of finding a vertex in which the objective function has a better value involves four operations.

1) *Reflection.* Reflect $\mathbf{x}_h^{(k)}$: **x** that gives the maximum value of $f(\mathbf{x})$ through the centroid by computing:

$$\mathbf{x}_{n+3}^{(k)} = \mathbf{x}_{n+2}^{(k)} + \alpha\left(\mathbf{x}_{n+2}^{(k)} - \mathbf{x}_h^{(k)}\right),$$

where, $\alpha > 0$ is the reflection coefficient, and $\mathbf{x}_{n+2}^{(k)}$ is the centroid computed by $x_{n+2,j}^{(k)} = \frac{1}{n}\left(\sum_{i=1}^{n+1} x_{ij}^{(k)} - x_{hj}^{(k)}\right)$, in which index *j* designates each coordinate direction.

2) *Expansion.* If $f\left(\mathbf{x}_{n+3}^{(k)}\right) \le f\left(\mathbf{x}_h^{(k)}\right)$, expand the vector $\left(\mathbf{x}_{n+3}^{(k)} - \mathbf{x}_{n+2}^{(k)}\right)$ by computing:

$$\mathbf{x}_{n+4}^{(k)} = \mathbf{x}_{n+2}^{(k)} + \gamma\left(\mathbf{x}_{n+3}^{(k)} - \mathbf{x}_{n+2}^{(k)}\right),$$

where, $\gamma > 1$ is the expansion coefficient.

3) *Contractions.* If $f\left(\mathbf{x}_{n+3}^{(k)}\right) > f\left(\mathbf{x}_h^{(k)}\right)$ for all $i \ne h$, contract the vector $\left(\mathbf{x}_h^{(k)} - \mathbf{x}_{n+2}^{(k)}\right)$ by computing:

$$\mathbf{x}_{n+5}^{(k)} = \mathbf{x}_{n+2}^{(k)} + \beta\left(\mathbf{x}_h^{(k)} - \mathbf{x}_{n+2}^{(k)}\right),$$

where, $0 < \beta < 1$ is the contraction coefficient.

4) *Reduction.* If $f\left(\mathbf{x}_{n+3}^{(k)}\right) > f\left(\mathbf{x}_h^{(k)}\right)$, reduce all the vectors $\left(\mathbf{x}_i^{(k)} - \mathbf{x}_l^{(k)}\right)$, where $\mathbf{x}_l^{(k)}$ is the **x** that gives the minimum value of $f(\mathbf{x})$, and δ is the reduction coefficient (usually $\delta = 0.5$):

$$\mathbf{x}_{n+4}^{(k)} = \mathbf{x}_l^{(k)} + \delta\left(\mathbf{x}_i^{(k)} - \mathbf{x}_l^{(k)}\right).$$

All the above operations can be seen in Fig. 1. The grey boxes (Fig. 1) indicated the inner search that can be performed using FPM.

3. Numerical results

Within the framework of this work, an algorithm of FTM (Figure 1) had been implemented in the MathCad14 package. Let's examine the operating time of the FTM depending on the parameters α, β, γ values. To do this, we calculate the program runtime for two problems. We will minimize two functions in two-dimensional space. And further we will tell "problem 1" and "problem 2" about it.

“Problem 1” is minimization of function

$$F(\mathbf{x}) = 4x_0 - x_1^2 - 12 \tag{6}$$

with one constraint in the form of equality

$$h_1(\mathbf{x}) = -x_0^2 - x_1^2 + 25 = 0 \tag{7}$$

and three constraints in the form of inequalities

$$g_1(\mathbf{x}) = -x_0^2 - x_1^2 + 10x_0 + 10x_1 - 34 \geq 0,$$
$$g_2(\mathbf{x}) = x_0 \geq 0, \tag{8}$$
$$g_3(\mathbf{x}) = x_1 \geq 0.$$

"Problem 2" is minimization of function

$$F(\mathbf{x}) = x_0^2 + x_1^2 \tag{9}$$

with one constraint in the form of equality

$$h_1(\mathbf{x}) = x_0^2 + x_1^2 - 9x_1 + 4.25 = 0. \tag{10}$$

These problems are taken from [2]. We will compare the operating time of the program when changing one parameter of the method. The other two parameters remain fixed from recommended in the literature [2] $\alpha = 1$, $\beta = 0.5$, $\gamma = 2$.

3.1. Research of γ parameter value

First consider the coefficient of expansion γ at fixed parameters $\alpha = 1$, $\beta = 0.5$.

Fig. 2 shows the dependence of the optimum value of the function found by the algorithm and the operating time of the algorithm for "problem 1" and "problem 2" on various parameter γ.

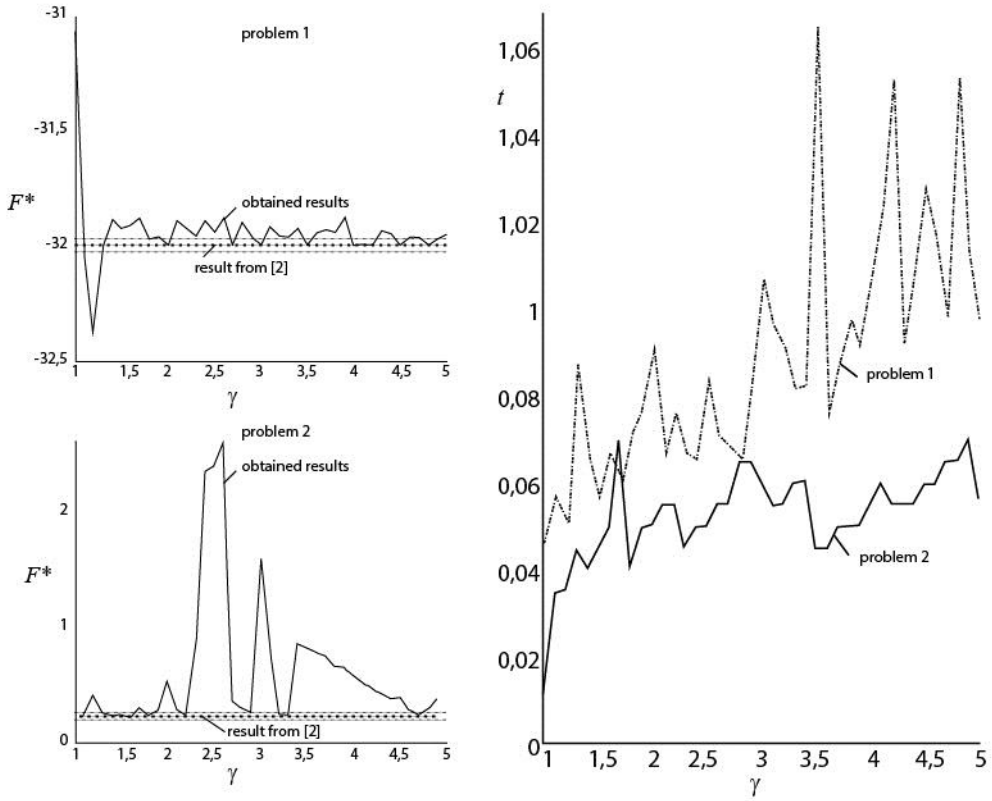


Fig. 2. The dependence of the optimum value of the function found by the algorithm and the operating time of the algorithm for "problem 1" and "problem 2" on various parameter γ

Curve «results from [2]» shows the optimum value of the function, which was obtained in [2] with recommended in literature values of parameters. Horizontal lines around the «results from [2]» curve are given as tolerances at a distance of 0.03 from this curve for better understanding of the scale. This figure shows that γ parameter values from 1.3 to 1.9 are acceptable for these problems.

3.2. Research of β parameter value

Then we select the optimal values of the coefficient of contraction β for "problem 1" and "problem 2" with fixed parameters $\alpha = 1$, $\gamma = 2$. Fig. 3 illustrates the dependence of the optimum value of the function found by the algorithm and the operating time of the algorithm for our two problems on various parameter β. All curves have the same meaning as in the previous section, but with respect to β parameter changes. Fig. 3 shows that β parameter values from 0.2 to 0.6 are acceptable for these problems.

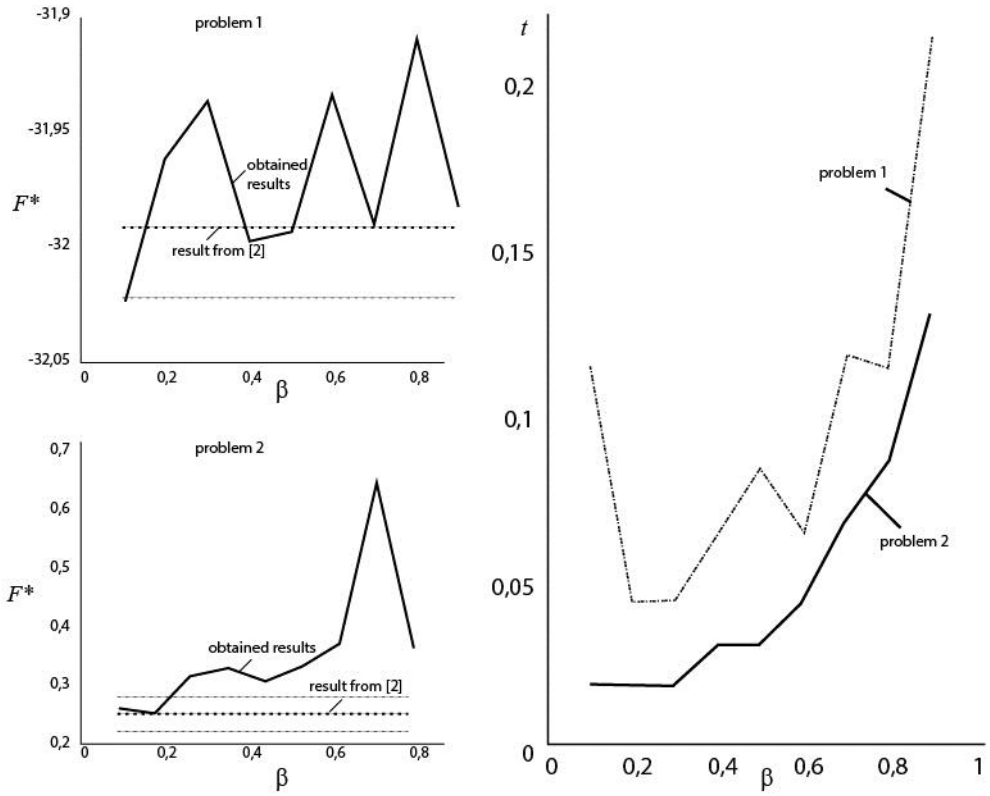


Fig. 3. The dependence of the optimum value of the function found by the algorithm and the operating time of the algorithm for "problem 1" and "problem 2" on various parameter β

3.3. Research of α parameter value

In the end, we will select the best coefficient of reflection α values in "problem1" and "problem 2" for fixed parameters $\beta = 0.5$, $\gamma = 2$.

Fig. 4 demonstrates the dependence of the optimum value of the function found by the algorithm and the operating time of the algorithm for problems (6), (7), (8) and (9), (10) on various parameter α. All curves have the same meaning as in the previous sections, but with respect to α parameter changes. This figure shows that α parameter values higher 0.6 are acceptable for these problems.

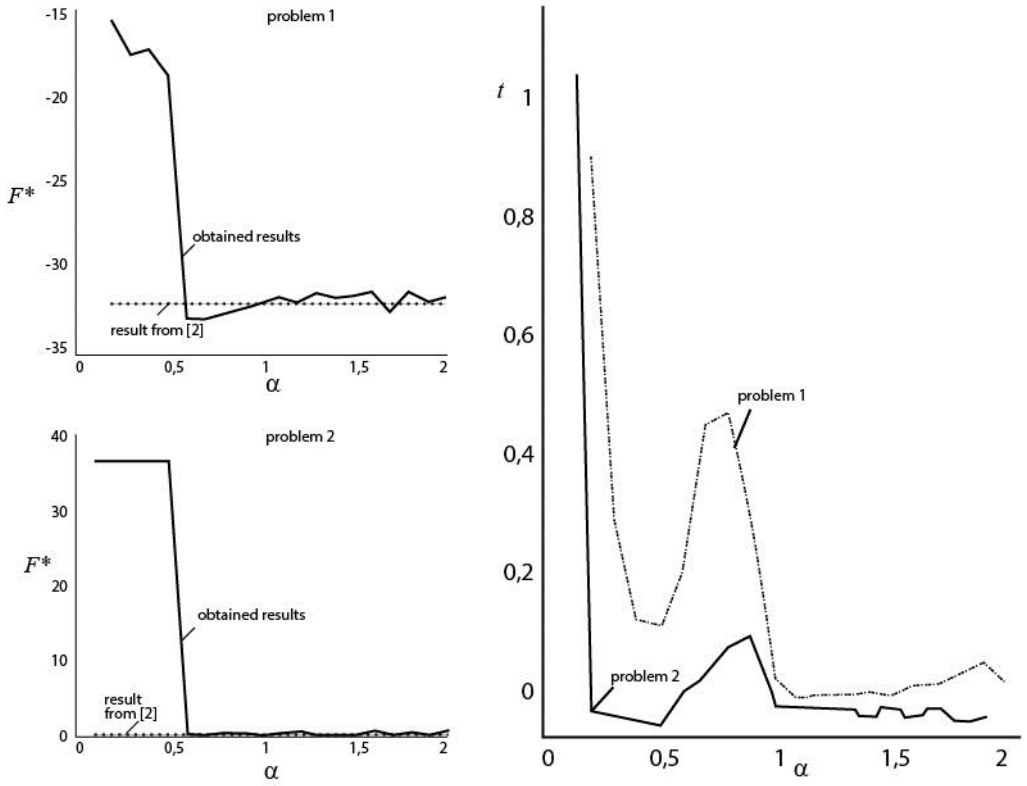


Fig. 4. The dependence of the optimum value of the function found by the algorithm and the operating time of the algorithm for "problem 1" and "problem 2" on various parameter α

Conclusion

We had minimized two functions (6) and (9) in two-dimensional space. We had identified the best parameter α, β, γ values for these problems. It is worth noting that recommended in literature parameters values are acceptable not for all problems. This can be seen from Table 1.

Table 1

Recommended parameters α, β, γ values

	γ	β	α
recommended values from [1]	2.0	0.5	1.0
recommended values from our experiments	1.3, 1.4, ..., 1.9	0.2, 0.3, ..., 0.6	0.6, 0.7, ...

For example, 2.0 is recommended parameter value for γ in literature, but 2.0 is not acceptable value for problems (6), (7), (8) and (9), (10) in paragraph 3.1 this article.

REFERENCES

1. *Colville A.R.* 1968. A Comparative Study on Nonlinear Programming Codes. IBM Technical Report No. 320–2949.

2. *Himmelblau D.M.* 1972. Applied Nonlinear Programming. McGraw-Hill. 498 p.

3. *Lima A.M.* 2015. Nonlinear Constrained Optimization with Flexible Tolerance Method Improvement and Application in System Synthesis of Mass Integration. 213 p.

4. *Nelder J.A., Mead R.* 1964. A Simplex Method for Function Minimization. Computer Journal. Vol. 7, P. 308–313.

5. *Paviani G., Himmelblau D.M.* 1969. Constrained Nonlinear Optimization by Heuristic Programming. Operations Research. Vol. 17, No. 5, P. 872–882.

6. *Shang W., Zhao S., Shen Y.* 2009. A Flexible Tolerance Genetic Algorithm for Optimal Problems with Nonlinear Equality Constraints. Adv. Eng. Inform. Vol. 23, P. 253–264.

III. ТЕСТИРОВАНИЕ И КОНТРОЛЕПРИГОДНОЕ ПРОЕКТИРОВАНИЕ ЛОГИЧЕСКИХ СХЕМ ВЫСОКОЙ ПРОИЗВОДИТЕЛЬНОСТИ И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

ОБНАРУЖЕНИЕ УТЕЧЕК РЕСУРСОВ В СПИСОЧНЫХ СТРУКТУРАХ ПРОИЗВОЛЬНОЙ ВЛОЖЕННОСТИ И СВЯЗНОСТИ

Бизяев Д.К.
Томский государственный университет
dicobi@rambler.ru

Введение

В настоящее время, статический анализ кода очень распространенный метод, который используется множеством компаний для обнаружения ошибок в исходных кодах своих проектов. Основная причина его применения заключается в соотношении таких показателей как цена/качество. Несмотря на то, что статические анализаторы не могут заменить полноценный обзор кода, проводимый группой разработчиков, они способны обратить внимание программиста на подозрительные места в коде. В том числе, эти инструменты, способны обнаруживать утечки памяти в программах. На сегодняшний день, поиску утечек памяти посвящено множество работ [1,2], в том числе использующих метод статического анализа [3].

Одним из критериев оценки качества статических анализаторов является их способность указать программисту на место в коде, которое содержит ошибку. Чем конкретнее и точнее будет сообщение, показанное анализатором, тем проще будет найти ошибку и исправить. Под точностью в данном случае понимается действительное наличие ошибки, в указанном месте, а не сообщение, о возможном наличии ошибки в подозрительном месте. Предложенный в данной работе алгоритм, реализован в виде консольного приложения, которое выводит программисту сообщение с указанием строчки кода, после которой произошла утечка памяти. И выводит последовательность операторов, в которых велась работа с потерянным элементом, для того чтобы было возможно определить, что привело к утечке памяти.

1. Списочные структуры

Списочная структура – это динамическая, рекурсивно-определенная структура данных, объявленная пользователем. Особенность этих структур заключается в том, что это структуры, которые внутри себя содержат ссылки на структуры такого же типа. Структура называется динамической, потому что ее форма и размер могут меняться в процессе выполнения программы. Элементы таких структур размещаются в динамической памяти программы или, другими словами, в куче. Доступ к элементам данных, размещенным в куче, может быть получен только через указатели. Сам указатель находится в статической области памяти программы и хранит адрес ячейки памяти, в которую размещается объект, в процессе работы программы (рис. 1).

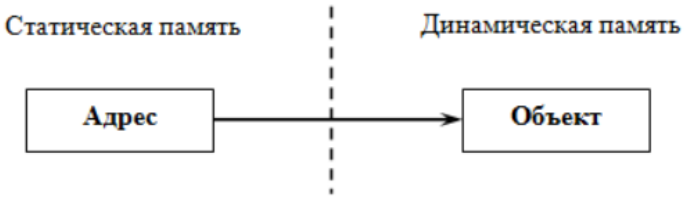


Рис. 1. Механизм доступа к элементам в куче через указатель

Основными разновидностями списочных структур являются односвязные и двусвязные списки, в которых каждый элемент данных хранит внутри себя указатель или ссылку на соседние элементы (рис. 2, 3) и какие то данные. Тем самым образуя цепочки элементов в виде списков, с элементами связанными между собой указателями. Как правило, для таких структур определяется набор характерных для них операций, таких как вставка, удаление элемента, обращение к элементу списка, подсчет количества элементов и т.д.

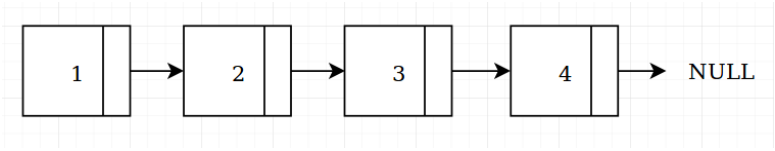


Рис. 2. Односвязный список

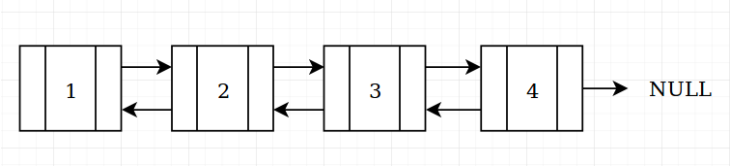


Рис. 3. Двусвязный список

Кроме перечисленных типов списочных структур, существуют списки произвольной вложенности, например, списки списков, когда элементы списка хранят внутри себя другие списки (рис. 4).

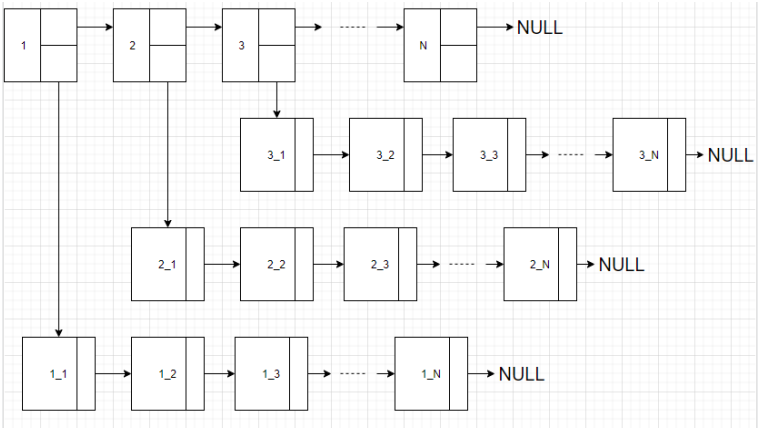


Рис. 4. Список списков

Т.к. работа с элементами, размещенными в динамической памяти, ведется с помощью указателей, то важно выделить свойство присущее всем элементам динамической памяти, в особенности элементам списочных структур. Это свойство связности. Связность определяет доступность элемента, размещенного в динамической памяти из программы через указатель.

В данной работе, поиск утечек памяти построен на проверке связей между элементами динамических структур и их доступности из программы через указатели.

2. Операции, приводящие к утечкам памяти

Как было сказано в предыдущем разделе, поиск утечек будет построен на проверке связей между элементами и наличии доступа к ним из программы через указатели. Наличие утечки памяти будем считать в случае отсутствия, через указатель, доступа к элементу, размещенному в куче. Для того чтобы выполнять поиск утечек памяти в про-

граммах использующих списочные структуры необходимо определить набор операций, которые оказывают влияние на кучу и могут привести к возникновению утечки памяти. К операциям такого типа, можно отнести:

● Сдвиг указателя

Операция относится к указателю, через который осуществляется доступ к структуре контейнерного типа. В отдельно взятой точке программы, такой указатель ссылается только на один элемент контейнера. Для динамических структур это может означать, что если сохранить в такой указатель адрес другого элемента контейнера, то может оказаться так, что возможности получить доступ к предыдущему элементу, нет. На рис. 5, 6 продемонстрирован сдвиг указателя на примере односвязного списка.

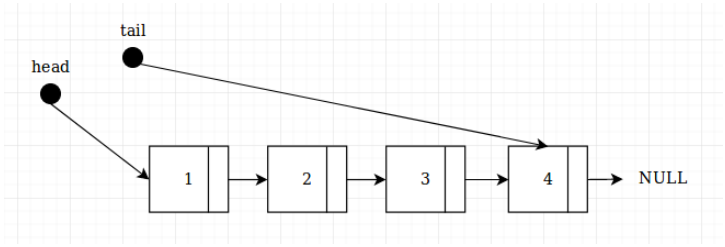


Рис. 5. Указатель *head* ссылается на первый элемент структуры

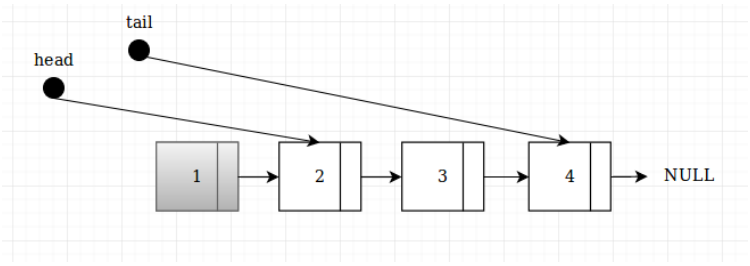


Рис. 6. Указатель *head* сдвигается на один элемент вперед, и первый элемент становится недоступным

● Переопределение указателя

Присвоение указателю, который содержит адрес элемента размещенного в куче, другого адреса без предварительного освобождения памяти или сохранения исходного адреса. Элементом может быть начало списка, дерева, массива или же просто объект. В любом случае доступ ко всем данным, так или иначе связанным с элементом будет утрачен. На рис. 7, 8 показан этот случай.

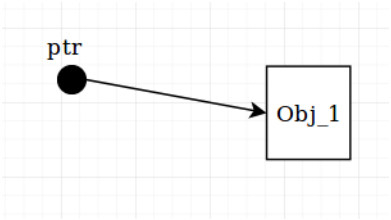


Рис. 7. Указатель на объект данных в куче

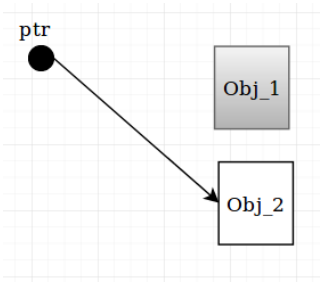


Рис. 8. Присвоение указателю нового адреса

● Операции над элементами динамических структур данных

Работа с самими элементами динамических структур может привести к утечкам в случае, если при работе с одним из элементов структуры будет изменено поле структуры, которое хранит адрес следующего элемента, в результате чего часть структуры будет утеряна. На рис. 9, 10 показана эта ситуация.

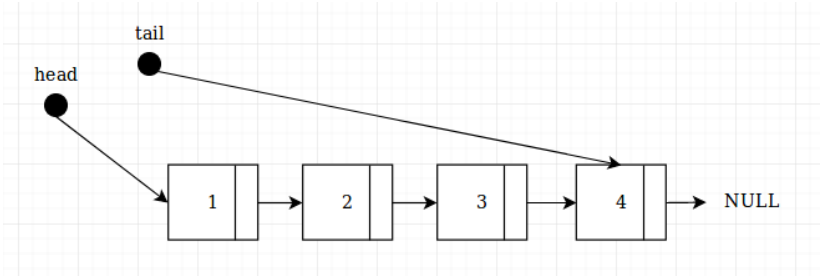


Рис. 9. Изначальный вид структуры

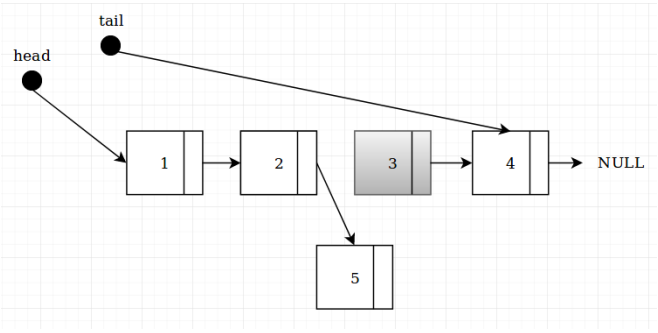


Рис. 10. Изменение поля элемента структуры ссылающегося на другой элемент

3. Символьное выполнение

Символьное выполнение – это метод статического анализа, в процессе которого строится модель анализируемой программы и на основе полученной модели формулируются правила для оценки программы. В случае поиска утечек памяти, правилами будем считать определенные в предыдущем разделе операции программы, которые могут привести к утечкам памяти. Таким образом, выполнение правила будет считаться успешным в том случае, когда в результате выполнения соответствующей операции не возникла утечка памяти. Оценкой, в этом случае, будет рассмотрение результата выполнения данных операций для получения ответа на вопрос, произошла утечка памяти или нет. Присвоим операциям, оказывающим влияние на кучу, следующие имена, которые будем использовать для обозначения выполнения или не выполнения правила:

Таблица 1

Соответствие между правилами и операциями оказывающими влияние на кучу

Правило 1	Сдвиг указателя
Правило 2	Переопределение указателя
Правило 3	Операция над элементом структуры

Процесс оценки программы заключается в исследовании поведения операторов программы. Это достигается путем имитации выполнения программы, при котором имитируется выполнение только относящихся к задаче операторов. Т.к. результат выполнения операторов, определенных в предыдущем разделе, может привести к изменению состояния кучи, например, изменив связи между элементами, добавив новые или удалив старые элементы, то помимо модели программы, необходимо также построить модель кучи.

Таким образом, имея в наличии модели объектов, за поведением которых нужно наблюдать и применяя символьное выполнение к операторам программы можно собрать требуемую, для решаемой задачи, информацию и на ее основе оценить программу на предмет наличия утечек памяти.

Рассмотрим, что собой представляет каждая из этих моделей, и какие компоненты нам в них потребуются для поиска утечек памяти.

3.1. Модель кучи

Куча – это динамически распределяемая память, в которую программа с помощью специальных функций может размещать элементы, получая от операционной системы адреса сегментов памяти.

В рамках нашей задачи будем представлять кучу как множество элементов. Каждый элемент имеет свой адрес, который может быть присвоен переменной указателю. Элементы кучи могут быть связаны друг с другом, если они являются членами динамической структуры данных.

В процессе символьного выполнения в кучу будут размещаться, и удаляться элементы. Между элементами кучи и указателями программы будут устанавливаться связи, и сами элементы будут также связываться друг с другом. Чтобы смоделировать такое поведение программы и отражать его на состоянии кучи, представим элемент кучи с помощью структуры предложенной в [4].

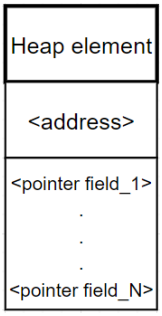


Рис. 11. Элемент кучи

На рис. 11 изображена структура элемента кучи, с помощью которой будут представлены все элементы, размещаемые в куче. У этой структуры есть два поля:

<address> – это адрес элемента в куче, который присваивается указателю и по которому можно обратиться к ячейке памяти для извлечения оттуда данных;

<pointer field> – если элемент в куче, является объектом динамической структуры, то это поле содержит столько указателей, сколько их объявлено в этой структуре.

Используя эту структуру, можно размещать элементы в куче и, присваивая их адреса переменным указателям и полям указателям других элементов, устанавливать свя-

зи, по которым в дальнейшем будет определяться наличие или отсутствие утечки памяти.

Определим набор операций для работы с моделью кучи. Набор операций будет следующим:

- добавить элемент;
- удалить элемент;
- проверить, есть ли к элементу доступ из программы через указатель;
- получить элемент по адресу.

Перейдем к рассмотрению модели программы.

3.2. Модель программы

Прежде чем анализировать поведение программы, определим те ее элементы, которые влияют на кучу и отражают ее собственное состояние. Доступ к элементам кучи возможен только через указатели, поэтому все операторы, функции и методы, в которых не ведется работа с указателями, исключаются из модели.

Любая программа состоит из функций. Чаще всего, говоря о функциях, мы представляем ее как сущность, у которой есть возвращаемый тип, значение, список аргументов и тело (последовательность операторов). В описываемой модели функция будет рассматриваться при дальнейшем анализе, только если в ней будет вестись работа с указателями. Т.е. если в списке аргументов функции, возвращаемом типе или теле функции не будет хотя бы одного указателя, то такая функция исключается из анализа.

Объекты пользовательских типов, например, классы или структуры, могут быть размещены в куче. Произвольные типы могут состоять из множества полей и методов. Знать о них всех нам не обязательно, достаточно в представлении типа хранить только те поля, которые являются указателями. То же касается и методов, достаточно сохранить информацию только о методах, работающих с указателями. Несмотря на то, что среди полей и методов пользовательского типа может не быть ничего, что могло бы как то повлиять на кучу, то такое «пустое», в нашем случае, представление о таком типе все равно нужно сохранить, т.к. экземпляр этого типа может быть размещен в куче.

Кроме информации о функциях и пользовательских типах и их определениях, важно хранить объявляемые в процессе работы программы указатели, переменные и контекст (область видимости) их объявления.

Работа программы заключается в исполнении операторов, в нашем случае происходит символьное выполнение. Чтобы начать символьное выполнение программы нужно сформировать возможные пути. Путь программы – это конечная последовательность операторов. Путей тем больше, чем больше и сложнее программа. Таким образом, множество путей является одним из важнейших компонентов модели программы. Т.к. куча будет строиться в процессе выполнения некоторого пути.

В результате, модель программы будет состоять из следующих элементов:

- множество функций;
- множество объявлений пользовательских типов данных;
- множество переменных указателей;
- множество всевозможных путей выполнения программы.

3.3. Множество путей выполнения программы

Рассмотрим, что собой представляет путь программы, и каким образом формируется множество всех путей более подробно.

Как уже было сказано выше, путь – это конечная последовательность операторов программы, которая будет выполнена после запуска программы. Программа – это отдельный процесс, содержащий внутри от одного до нескольких потоков выполнения, которые представляют собой последовательности команд, обрабатываемых процессором. Рассматривать работу многопоточного приложения мы не будем, т.к. нас сейчас интересует то, каким образом выполняются операторы программы в общем случае.

Точка входа в программу (обычно это функция main), – это место, откуда поток выполнения программы начинает последовательно исполнять команды программы до тех пор, пока они не закончатся либо не произойдет аварийное завершение программы, либо не будет выполнен преждевременный выход из программы, предполагаемый логикой ее работы в случае, например, обработки особой ситуации. При реальном выполнении программы, путь потока выполнения определяется входными данными и вычислениями, производимыми ей в процессе работы. В случае символьного выполнения, знать заранее какими будут входные данные программы невозможно, т.к. единственным источником информации о программе является лишь текст исходного кода, обрабатываемый инструментом статического анализа. Рассмотрим пример кода и множество возможных путей его выполнения:

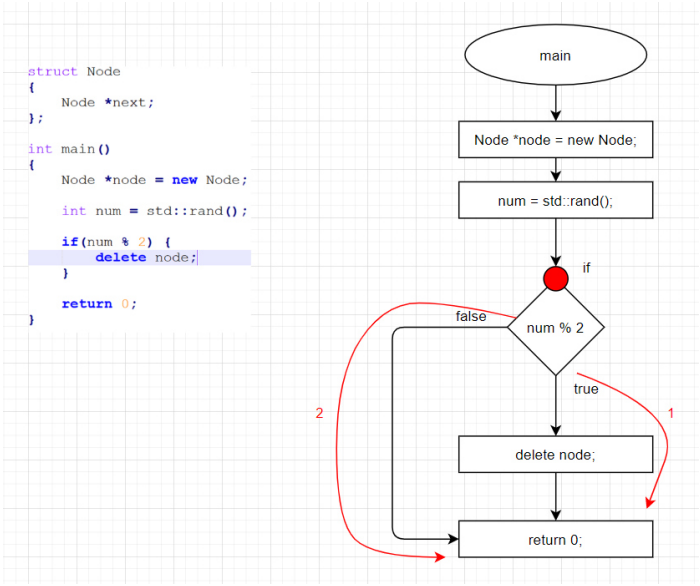


Рис. 12 Ветвление программы

В левой части рис. 12 показана небольшая программа, в правой части ее блок схема. В начале этой программы в куче выделяется память, в которую размещается объект типа *Node*, адрес этого сегмента памяти сохраняется в переменную указатель *node*. Далее происходит вызов метода генерирующего случайное число, значение которого будет определять, выполнение оператора *delete*, с целью освобождения выделенного участка памяти в куче. Анализируя программу статически, т.е. извлекая о ней информацию из абстрактного синтаксического дерева, невозможно определить последовательность выполняемых программой операторов, следовательно, чтобы дать ответ произошла утечка памяти или нет, необходимо рассмотреть оба варианта выполнения программы. Это позволит сказать, что при выполнении пути обозначенного на рисунке цифрой 1 в программе не произойдет утечки, но при выполнении пути под цифрой 2 элемент, размещенный в куче, не будет освобожден после завершения программы.

Исходя из вышесказанного, следует, что для определения возникновения в программе утечки памяти и для того, чтобы попытаться ответить на вопрос что послужило этому причиной, необходимо на этапе статического анализа исходного кода сформировать множество всех возможных путей ее выполнения.

4. Алгоритм поиска утечек памяти

В данной работе предлагается алгоритм поиска утечек памяти, который на текущем этапе разработки упрощает поиск утечек памяти, благодаря следующим возможностям:

- Указание переменной указателя, уничтожение которого в стеке привело к возникновению утечки памяти.
- Вывод последовательности операторов, которые были выполнены над элементом кучи, перед тем, как элемент стал недоступен.

Прежде чем программа приступит к поиску утечек памяти с использованием алгоритма, ей необходимо выполнить некоторые подготовительные действия для того, чтобы собрать нужную, для работы алгоритма, информацию. Эти действия состоят из двух этапов:

- Перевод во внутренний формат пользовательских типов данных и функций.
- Формирование всевозможных путей выполнения программы.

Перевод во внутренний формат пользовательских типов данных и функций нужен для того, чтобы выделить из исходного кода программы только ту информацию, анализ которой может помочь получить ответ на вопрос есть в программу утечка памяти или нет. Эта информация была обозначена в разделе с описанием модели программы. Таким образом, выполнив перевод программы во внутренний формат и сформировав множество всевозможных путей, будет получена модель программы, готовая для исследования на предмет наличия утечек памяти.

Полученное множество всевозможных путей передается на вход алгоритму для анализа и обнаружения на этих путях утечек памяти. Обозначим множество всевозможных путей программы Paths. В случае, если в результате работы алгоритма были обнаружены утечки памяти, то на выходе алгоритма будет получено множество, в котором с каждым потерянным элементом будет ассоциирована последовательность операторов, которая привела к этому. Обозначим это множество <LostElement, Operators>. Представим алгоритм поиска утечек памяти в виде следующих шагов:

Вход: Paths

1. Выбор пути; если все пути рассмотрены завершить алгоритм.
2. Символьное выполнение.
3. Если куча пустая, то перейти на шаг 1, иначе на шаг 4.
4. Сохранить все элементы, которые остались в куче после символьного выполнения программы.
5. Выполнить поиск утечек памяти.
6. Сохранить информацию о найденных утечках и перейти на шаг 1.

Выход: <LostElement, Operators>

Полученное множество последовательностей операторов, которые привели к утечкам, соответствующих данных форматируется и выводится пользователю. На пятом шаге алгоритма происходит повторное символьное выполнение того же самого пути, в ходе выполнения которого произошли утечки памяти. Для всех потерянных элементов, которые были сохранены на шаге 4 после выполнения оператора, оказывающего влияние на кучу, который обрабатывает один из этих элементов, производится проверка связности. По результату этой проверки определяется, привел ли оператор к возникновению утечки памяти. Алгоритм проверки связности можно представить в виде следующих шагов:

Вход: множество переменных указателей и множество элементов кучи

1. Выбирается элемент кучи, отмеченный как потерянный на предыдущем этапе алгоритма, если все элементы уже рассмотрены, завершить алгоритм;

2. Найти все элементы, указывающие на этот элемент; если среди них есть указатель программы, отметить элемент как НЕ потерянный и перейти на шаг 3, иначе на шаг 4;
3. Если из данного элемента можно получить доступ элементу, выбранному на шаге 1, путем обхода элементов, с помощью указателей, то отмечаем все элементы в цепочке как НЕ потерянные и уже рассмотренные и считаем, что утечки для исходного элемента нет, и переходим на шаг 1; иначе, просто отмечаем все элементы в цепочке как НЕ потерянные и уже рассмотренные, возвращаемся на шаг 2 и рассматриваем остальные элементы, которые указывают на исходный.
4. Если на текущий элемент ссылаются другие, НЕ рассмотренные элементы кучи, то отметить текущий элемент как неопределенный и для всех этих элементов начать выполнять проверку, начиная с шага 2. Продолжать выполнять проверку до тех пор, пока либо не будет установлено, что к исходному элементу есть доступ, либо что он потерян и произошла утечка.

Выход: ответ, потерян элемент или нет.

5. Экспериментальные результаты

Описанные в предыдущем разделе алгоритмы были проверены на нескольких тестовых примерах, в которых есть утечки памяти при использовании списочных структур. Сравнение проводилось с утилитой Predator [5], которая также, используя метод статического анализа, выявляет утечки памяти в программах на языке Си. Т.к. алгоритмы, предложенные в данной работе, обрабатывают код, написанный на языке С++, то все операции по размещению и удалению памяти, написаны с использованием, как операторов языка Си, так и языка С++. Ниже приведена таблица, в которой сравнивается вывод предложенного анализатора и Predator'а на точность указания причины утечки памяти относительны сформулированных правил. Сравнение будет проводится, по следующим критериям, а именно сообщил ли инструмент программисту следующую информацию, при условии что в программе использовались обозначенные правила:

1. Строка кода, после которой объект недоступен;
2. Сообщение с указанием причины утечки;
3. Какая переменная последняя имела доступ к потерянному объекту;
4. Последовательность операторов, которая привела к утечке.

В последние столбцы таблицы будет записано относительно какого правила в каждом коде какая информация была выведена инструментами. Например, если в коде использовалось правило 1 и для него оба инструмента сообщили всю указанную информацию то, запись в оба столбца будет выглядеть следующим образом: {1, {+, +, +, +}}, где, число означает номер правила, а заключенные в фигурные скобки знаки '+' (или '–') указывают, была ли выведена соответствующая информация данным инструментом.

Таблица 2

Сравнение вывода предложенного анализатора и инструмента Predator

Примеры	Правило 1	Правило 2	Правило 3	Предложенный анализатор	Predator
Код 1	+	–	–	{1, {+, +, +, +}}	{1, {+, +, –, –}}
Код 2	+	–	–	{1, {–, +, +, –}}	{1, {+, +, –, –}}
Код 3	+	–	–	{1, {+, +, +, –}	{1, {+,+, –, –}}
Код 4	+	+	–	{1, {+, +, +, +}}, {2, {+, +, +, +}}	{1, {+, +, –, –}}, {2, {+, +, –, –}}
Код 5	+	+	–	{1, {–, +, +, –}}, {2, {–, –, +, +}}	{1, {+, +, –, –}}, {2, {+, +, –, –}}
Код 6	+	+	–	{1, {+, +, +, –}}, {2, {–, +, +, –}}	{1, {+, +, –, –}}, {2, {+, +, –, –}}
Код 7	–	+	+	{2, {+, +, +, +}}, {3, {+, +, +, +}}	{2, {+, +, –, –}}, {3, {+, +, –, –}}

Примеры	Правило 1	Правило 2	Правило 3	Предложенный анализатор	Predator
Код 8	–	+	+	{2, {–, –, +, +}}, {3, {+, +, –, –}}	{2, {+, +, –, –}}, {3, {+, +, –, –}}
Код 9	–	+	+	{2, {–, +, +, –}}, {3, {+, +, –, +}}	{2, {+, +, –, –}}, {3, {+, +, –, –}}
Код 10	+	+	+	{1, {+, +, +, +}}, {2, {–, +, +, –}}, {3, {+, +, –, +}}	{1, {+, +, –, –}}, {2, {+, +, –, –}}, {3, {+, +, –, –}}

В примерах 1, 4 и 7 в кодах программы использовался односвязный список, каждый узел которого содержал какие-то данные и указатель на следующий элемент.

В примерах под номерами 2, 5 и 8 использовался массив указателей, каждый из которых ссылался на односвязный список, в которых структура узла тоже представляла собой объект, содержащий какие-то данные и указатель на следующий элемент.

В примерах 3, 6, 9 и 10 был использован двусвязный список и массив указателей на отдельные элементы этого списка. Для нарушения связей был определен метод переопределяющий указатели в списках, с последующим сохранением доступа к ним через массив указателей.

По результатам, представленным в табл. 2, видно, что предложенный анализатор дает более подробную информацию по возникшим утечкам памяти. Предложенный анализатор указывает на причины возникновения утечки памяти, сообщая программисту не только строки кода, после которых объект динамической памяти перестал быть доступен, но и имя переменной через которую в последний раз был возможен доступ к объекту.

Заключение

В данной работе основной акцент сделан на поиск утечек памяти в динамических структурах списочного типа и основное внимание уделено получению возможности обнаруживать нарушение связей между элементами списочных структур. Проведено исследование динамических структур списочного типа, метода статического анализа и определены основные операции над элементами динамических структур, которые могут привести к утечкам памяти. Предложен алгоритм, обнаруживающий утечки памяти в динамических структурах списочного типа произвольной вложенности и связности. Система предупреждений позволяет минимизировать время, требуемое для нахождения утечки памяти, сообщая последовательность операторов, которые привели к утечке.

ЛИТЕРАТУРА

1. Structured Specifications for Better Verification of Heap-Manipulating Programs / Cristian Gherghina [et al.] // FM 2011: Formal Methods - 17th International Symposium on Formal Methods. – 2011. – Limerick, Ireland. – P.386-401.

2. Template-Based Verification of Heap-Manipulating Programs / Viktor Malik [et al.]. – Formal Methods in Computer Aided Design (FMCAD). – 2018. – P.99-108.

3. *Scholz B.* Symbolic Pointer Analysis for Detecting Memory Leaks / Bernard Scholz, Johann Blieberger and Thomas Fahringer // ACM SIGPLAN Notices. – 1999. – P.34-45.

4. *Sai-ngern Sittisak* An address mapping approach for test data generation of dynamic linked structures. / Sittisak Sai-ngern, Chidchanok Lursinsap, Peraphon Sophatsathit. // Information and Software Technology – 2005. – P.199-214.

5. Predator: A Verification Tool for Programs with Dynamic Linked Data Structures / Kamil Dudka [et al.] – Tools and algorithms for the construction and analysis of systems. – 2012. – P.545-548.

ОБЕСПЕЧЕНИЕ КОНФЕДИНЦИАЛЬНОСТИ «ОБЛАЧНЫХ» ДАННЫХ В ЗАЩИЩЕННЫХ СУБД

Генрих В.В., Тренькаев В.Н.
Томский государственный университет
gvv856@mail.ru, tvnik@sibmail.com

Введение

В настоящее время широко распространены облачные сервисы, когда клиенту предоставляется третьей стороной (провайдером) сетевой доступ к распределенным и совместно используемым вычислительным ресурсам. Один из наиболее востребованных облачных сервисов – база данных как сервис (DBaaS – Database as a service). Потребителю предоставляются услуги хранения и обработки данных на «облаке». При этом возможна ситуация, когда провайдер корректно выполняет все действия необходимые для предоставления услуги, но еще ведет наблюдения за данными (так называемый честный, но любопытный провайдер). Таким образом, имеется риск раскрытия конфиденциальной информации потребителя облачной услуги.

Как известно, задача обеспечения конфиденциальности информации решается с помощью шифрования – такого преобразования, которое делает прочтение информации невозможным лицам, которые не обладают секретным ключом. Так называемое «прозрачное» шифрование баз данных (БД), когда данные шифруются перед записью на диск и расшифровываются во время чтения в память, – это распространенная опция современных систем управления БД (СУБД). Однако, такой подход небезопасен для случая «облачной» БД, т.к. ключ при этом становится доступен провайдеру. Существует несколько вариантов решения проблемы обеспечения конфиденциальности «облачных» данных.

Одно из решений – шифровать данные перед передачей в облачный сервис, но при помощи специализированных шифров (гомоморфные шифры, сохраняющие порядок шифры), которые позволяют производить операции над зашифрованными данными. На основе такого подхода реализованы СУБД CryptDB [1] и Monomi [2], а также исследовательские проекты [3,4]. Отметим, что достаточно высокая ресурсоемкость гомоморфного шифрования приводит к «проседанию» производительности такого рода СУБД.

В СУБД ZeroDB [5] выборка также осуществляется над зашифрованными данными, но на базе другого принципа. На сервере БД в зашифрованном виде хранятся индексы в формате B-дерева. В ходе выполнения запроса клиент взаимодействует с сервером через серию обращений, обходя узлы индекса и в итоге получая требуемые для выполнения запроса зашифрованные записи. Индексы состоят из сегментов, которые расшифровываются только на стороне клиента.

В СУБД Cipherbase [6], которая является расширением (дополнением) Microsoft SQL Server, данные шифруются перед отправкой в «облако». Проблема безопасности ключа и безопасной обработки запросов на «облаке» решается с помощью существования доверенного модуля. К представителям подхода на базе доверенного оборудования также относится TrustedDB [7].

Если требуется защитить только часть атрибутов (столбцов) таблицы (отношения) БД, то возможно применение техники, когда с зашифрованными столбцами связывают индексы [8]. Индексы представляют собой метаданные, сохраняющие свойства неконфиденциальных атрибутов, и могут вычисляться разными способами, например на базе хэш-функции. Столбцы, задающие индексы, используются при выполнении запросов на стороне провайдера.

В некоторых случаях важно обеспечить конфиденциальность не самих данных, а взаимосвязи между ними. Связь между элементами данных можно скрыть путем со-

хранения этих элементов в разных фрагментах БД [8]. Неотлеживаемость связей между данными гарантируется специальным разбиением отношения на части и двух-шаговым выполнением запроса (часть запроса выполняется провайдером, а часть – клиентом).

Для практического применения в качестве безопасных DDaaS требуются СУБД, которые предоставляют высокий уровень безопасности при производительности и наборе операций над данными, сравнимыми с обычными незащищенными СУБД. Далее подробно рассмотрены претенденты на такое индустриальное использование: CryptDB и Cipherbase. CryptDB выбрана как представитель решений по обеспечению конфиденциальности «облачных» данных на основе выполнения запросов над зашифрованными данными, а Cipherbase – на базе доверенного оборудования. Представлены ключевые идеи и базовые характеристики этих проектов, а также используемые модели угроз.

1. Ключевые особенности

Характерными особенностями CryptDB являются:

1. *Выполнение SQL-запросов над зашифрованными данными.* Идея основана на том, что SQL-запросы состоят из определенного набора примитивных операторов, которые требуется реализовать, но только над зашифрованными данными. Для этого используются шифрующие преобразования, позволяющие в дальнейшем обрабатывать зашифрованные данные. Эффективность подхода обеспечивается за счет отказа от полностью гомоморфного шифрования, которое на данный момент неприемлемо для практического применения.

2. *Настраиваемое и вложенное шифрование.* При шифровании применяется комбинация из различных видов шифров: шифр с сохранением порядка, частично гомоморфный шифр, традиционный шифр (детерминированный и вероятностный варианты), в виде компактного хранения нескольких шифртекстов друг в друге. Внешний слой в такой «луковице» самый безопасный, но не позволяет выполнять никакие операции над зашифрованными данными. Переход к внутренним слоям дает более слабые гарантии безопасности, но поддерживает больше операций над зашифрованными данными (сравнение, арифметические операции). Изначально все значения в защищенной БД шифруются до максимального уровня секретности. Далее в процессе обработки запросов удаляются «лишние» слои и достигается состояние, в котором все данные зашифрованы схемами, позволяющими выполнять наиболее часто встречающиеся для этих данных операций. Происходит динамическое регулирование уровня секретности.

3. *Связывание ключей шифрования с паролями пользователей.* Каждый элемент данных в БД можно расшифровать только через связку ключей, которая основана на пароле авторизованного пользователя. В результате, если злоумышленник не знает пароля, то не сможет расшифровать БД, даже если клиентское приложение и сервер БД полностью скомпрометированы.

Характерными особенностями Cipherbase являются:

1. *Политики шифрования и индексация.* Имеется возможность применять различные схемы шифрования для разных столбцов, а также выбрать политику шифрования, которая определяет баланс между уровнем защиты и уровнем производительности. Существует несколько таких политик. В зависимости от политики различные столбцы таблицы шифруются либо детерминированным (AES-ECB), либо вероятностным шифрами (AES-CTR). Также поддерживается алгоритм индексации с разделением индексов на два типа, каждый из которых предоставляет различные гарантии конфиденциальности.

2. *Использование доверенного модуля.* В целях обеспечения защищенного места для выполнения операций над зашифрованными данными в Cipherbase используется доверенный модуль на базе безопасного оборудования. Данный модуль поддерживает операции сравнения, а также арифметические и криптографические операции. Целевое

назначение модуля – выполнение операций, которые не могут быть совершены непосредственно над зашифрованными данными. Доверенный модуль является недоступным для злоумышленника, что позволяет в рамках модуля расшифровывать данные и проводить операции с ними.

3. *Применение программируемых логических интегральных схем (ПЛИС).* Хотя большая часть дизайна Cipherbase не зависит от выбора оборудования, описанный в [6] прототип реализован на основе ПЛИС. Новаторство реализации доверенного модуля на базе ПЛИС – включение модуля во внутренний цикл обработки запросов. При этом аппаратное и программное обеспечение спроектировано так, что управление циклами обработки запросов с использованием ПЛИС не сопровождается серьезным снижением производительности.

2. Обзор архитектуры

Архитектура CryptDB содержит следующие компоненты:

- пользовательская сторона;
- сервер приложений и прокси-сервер;
- сервер БД.

В прокси-сервере перехватываются SQL-запросы и переписываются для выполнения над зашифрованными данными. Также прокси-сервер шифрует и расшифровывает все данные и изменяет некоторые операторы запроса с сохранением семантики. В прокси-сервере хранится секретный ключ, схема БД и текущие уровни шифрования (слои луковиц шифрования) всех столбцов. Сервер БД не получает ключ, и таким образом, не имеет доступа к конфиденциальным данным.

На стороне сервера БД создаются определяемые пользователем функции (UDF), которые позволяют серверу БД выполнять определенные операции над шифрованными данными. CryptDB позволяет разработчику формировать так называемые аннотационные политики, указывая соответствие между принципалами (пользователями, группами пользователей) и элементами данных, к которым они имеют доступ.

Архитектура Cipherbase подразумевает разделение на две стороны: клиентский модуль (клиентская библиотека) и облачный сервер. Приложение клиента взаимодействует с сервером Cipherbase через клиентский модуль.

В клиентском модуле выполняется шифрование данных перед их отправкой на сервер и расшифрование возвращаемых результатов перед их отправкой в приложение.

Сервер Cipherbase состоит из доверенного модуля и базовой СУБД, модифицированной для взаимодействия с указанным модулем.

3. Обзор безопасности

Модель угроз CryptDB. Предполагается злоумышленник с полным доступом к облачной СУБД (*угроза 1*) или злоумышленник, получивший доступ к серверу приложений, прокси-серверу и/или серверу БД (*угроза 2*).

Угроза 1. Чтобы противостоять угрозе 1 на сервере БД выполняется обработка запросов только над зашифрованными данными. Тем не менее, возможна утечка метаданных. Например, если требуется выполнить операцию GROUP BY для некоторого столбца, то необходимо иметь возможность определять, какие элементы в этом столбце равны друг другу, хотя и не фактическое содержимое каждого элемента. Конфиденциальность таких метаданных как структура таблицы, количество строк, типы столбцов при этом не обеспечивается. Уровень безопасности также понижается (повышается) вследствие применения настраиваемого шифрования на основе запросов. Например, если приложение не формирует запросы со сравнением (больше, меньше, равно) элементов в столбце, то такой столбец шифруется с помощью вероятностного шифра, что дает максимальный уровень безопасности.

Угроза 2. Для противостояния угрозе 2 требуется формирование аннотационных политик, указывающих соответствие между принципалами и элементами данных, к которым они имеют доступ. Второе решение для устранения угрозы 2 – уменьшение количества информации, которое злоумышленник может получить, взломав систему. В CryptDB каждый элемент данных шифруется разными ключами, связанными с паролем пользователя, который имеет доступ к этим данным. Когда пользователь выходит из системы, то прокси-сервер удаляет его ключ. Таким образом, утечка информации неактивных пользователей становится невозможной.

Модель угроз Cipherbase. Предполагается злоумышленник, имеющий физический доступ к облачной БД и обладающий неограниченными возможностями наблюдения за облачным сервером БД. Вычисления внутри доверенного модуля злоумышленнику не доступны. Злоумышленник пассивен (честный, но любопытный провайдер): не изменяет содержимое БД и не влияет на процесс обработки запросов. Существует два типа злоумышленника (слабый и сильный). Слабый может только получить образ данных в состоянии покоя.

Для предотвращения данных угроз разработчиками Cipherbase проведена адаптация применяемого в CryptDB алгоритма настраиваемого шифрования. Существенное отличие от CryptDB заключается в отказе от использования частично гомоморфного шифрования и шифрования с сохранением порядка. Как результат, Cipherbase обеспечивает меньшую утечку информации, связанную с выполнением запросов. Тем не менее, полная конфиденциальность метаданных в Cipherbase не обеспечивается.

4. Результаты сравнительного анализа

Основные характеристики рассмотренных защищенных СУБД представлены в табл. 1.

Таблица 1

Сравнительный анализ CryptDB и Cipherbase

Характеристики	CryptDB	Cipherbase
Особенности архитектуры	На базе прокси-сервера	На базе доверенного модуля (ПЛИС)
Шифры		
Блочный шифр	AES-SIV	AES-CTR / AES-ECB
Шифр с сохранением порядка	mOPE	не используется
Частично гомоморфный шифр	Шифр Пэйе	не используется
Операции		
Набор операций	Ограниченный	Полный
Утечка информации, связанная с выполняемыми операциями	Имеется	Имеется, но в меньшей степени, чем в CryptDB
Модель угроз		
Возможности нарушителя (не изменяет содержимое БД и не влияет на процесс обработки запросов)	Имеется доступ к облачному серверу БД	Имеется доступ к облачному серверу БД
	Имеется доступ к прокси-серверу	Нет доступа к доверенному модулю
Оценка производительности (для набора запросов TPC-C) и затрат на реализацию		
Пропускная способность системы, нормализованная к пропускной способности базовой СУБД	от 46 до 95 % (в зависимости от типа запроса)	от 40 до 92 % (в зависимости от политики шифрования)
Объем зашифрованной БД (по сравнению с объемом открытого текста)	Увеличение в 3,76 раза	Увеличение в 1,29-4,34 раза (в зависимости от политики шифрования)
Дополнительное оборудование	Не требуется	Требуется
Изменение программного обеспечения	Практически не требуется	Практически не требуется

Заключение

В работе дан анализ защищенных СУБД CryptDB и Cipherbase, которые по сути являются надстройками над базовыми СУБД MySQL и SQL Server соответственно. Вы-

воды анализа: СУБД CryptDB и Cipherbase сравнимы по быстродействию и безопасности, но Cipherbase не имеет ограничений на выполняемые операции над данными, хотя при этом требует наличия на стороне провайдера специального доверенного модуля (дополнительное оборудование). В целом обеспечение конфиденциальности «облачных» данных приводит к снижению производительности базовых СУБД и росту объема БД. Выработка механизмов повышения быстродействия защищенных облачных СУБД без снижения уровня практической безопасности остается важных направлением для будущих исследований.

ЛИТЕРАТУРА

1. *Popa R., Redfield C., Zeldovich N., Balakrishnan H.* CryptDB: Protecting Confidentiality with Encrypted Query Processing // Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (SOSP'11), 2011. P. 85-100.

2. *Tu S., Kaashoek M.F., Madden S., Zeldovich N.* Processing analytical queries over encrypted data // Proceedings of the VLDB Endowment 6(5), 2013. P. 289–300.

3. *Shatilov K., Boiko V., Krendelev S., Anisutina D., Sumaneev A.* Solution for secure private data storage in a cloud // Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, 2014. P. 885-889.

4. *Глотов И.Н., Овсянников С.В., Тренькаев В.Н.* Защищённая СУБД с сохранением порядка // Прикладная дискретная математика. Приложение. 2014. № 7. С. 81-82.

5. *Egorov M., Wilkison M.* ZeroDB white paper, 2016 [Электронный ресурс]: ZeroDB white paper. URL: https://arxiv.org/abs/1602.07168

6. *Arasu A., Eguro K, Joglekar M, Kaushik R, Kossmann D, Ramamurthy R.* Transaction processing on confidential data using cipherbase // 2015 IEEE 31st International Conference on Data Engineering, Seoul, 2015, pp. 435-446.

7. *Bajaj S., Sion R.* TrustedDB: A Trusted Hardware based Database with Privacy and Data Confidentiality // IEEE Transactions on Knowledge and Data Engineering, 2011, Vol.26. P. 205-216,

8. *De Capitani Di Vimercati S., Foresti S., Livraga G., Paraboschi S., Samarati P.* Confidentiality Protection in Large Databases. In: S. Flesca et al. (eds.), A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years. Studies in Big Data, 2018, vol. 31. P. 457-472.

АЛГОРИТМЫ ПОСТРОЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ, ДОСТАВЛЯЮЩЕЙ ТЕСТОВЫЕ ПАРЫ ДЛЯ РОБАСТНО ТЕСТИРУЕМЫХ PDFS С ИСПОЛЬЗОВАНИЕМ ОПЕРАЦИЙ НАД ROBDD-ГРАФАМИ

Матросова А.Ю., Чернышов С.В.
Томский государственный университет
mau11@yandex.ru, semen.cher@mail.ru

Введение

Предлагается метод построения последовательности булевых векторов входных переменных, доставляющей тестовые пары (v_1, v_2) соседних векторов в пространстве входных и внутренних переменных для робастно тестируемых неисправностей задержек путей (робастных PDFs) в логических схемах с памятью. Булевы векторы называются соседними, если они отличаются значением только одной компоненты. Ранее в [1] был разработан метод нахождения всех тестовых пар соседних булевых векторов для робастно тестируемых неисправностей задержек пути в комбинационной схеме. На основе этих результатов в [2] были предложены методы построения тестовой последовательности для подмножества путей, гарантирующие нахождение тестовой пары для каждого из рассматриваемых путей комбинационной схемы, если такая пара существует. При тестировании [2] схем с памятью векторы полученной тестовой последовательности подаются на комбинационную составляющую в специальном режиме сканирования (Random Access Scan (RAS) режим). Эта технология требует больших дополнительных аппаратурных затрат. Целью данной работы является выяснение возможности построения тестовой последовательности для заданного подмножества путей без использования технологий сканирования, т.е. без дополнительных аппаратурных затрат в рамках ограничения на длину последовательности для отдельного пути. Имеется мно-

жество тестовых пар (v_1, v_2) соседних булевых векторов (всех или некоторых), обнаруживающих робастно тестируемую неисправность задержки пути в комбинационной составляющей схемы с памятью. Строится входная последовательность, доставляющую одну (любую) тестовую пару заданного множества из начального состояния q_0 схемы с памятью в рамках ограничения на длину последовательности. Предлагаются алгоритмы построения таких последовательностей в условиях, когда 1) начало пути отмечено входной переменной и 2) когда начало пути отмечено переменной состояний (внутренней переменной) схемы. Если искомые последовательности удается найти для каждого из путей заданного множества, то, совместив их, мы обнаруживаем задержки путей без дополнительных аппаратурных затрат. Проведенные эксперименты показывают, что тестовые последовательности удается построить не для всех путей (иногда ни для одного), для которых существуют тестовые пары в комбинационной составляющей схемы с памятью.

1. Некоторые свойства тестовых пар наборов для робастно тестируемых неисправностей задержек путей

Имеем одновыходную схему *C* и соответствующую эквивалентную нормальную форму (ЭНФ). В [3] рассматривается задача построения пар тестовых наборов для робастно и не робастно тестируемых неисправностей задержек путей на основе анализа ЭНФ. Наборы v_2 тестовых пар являются тестовыми наборами для константных неисправностей литер ЭНФ.

В случае rising transition тестируется 0-константная неисправность литеры ЭНФ, соответствующей пути α (a_p неисправность). В присутствии неисправности все вхождения литеры заменяются в ЭНФ константой 0. Тестовый набор, обнаруживающий эту неисправность, является набором v_2 тестовой пары, который порождает смену сигнала на выходе схемы с нулевого на единичное значение.

В случае falling transition тестируется 1-константная неисправность литеры ЭНФ, соответствующей пути α (b_p неисправность). В присутствии неисправности все вхождения литеры заменяются в ЭНФ константой 1. Тестовый набор, обнаруживающий эту неисправность, является набором v_2 тестовой пары, который порождает смену сигнала на выходе схемы с единичного на нулевое значение.

В [3] показано, что для тестовых пар, обнаруживающих робастно тестируемые неисправности задержек путей, выполняются следующие условия.

1. Если v_2 есть a_p тестовый набор, то v_1 есть b_p тестовый набор тестовой пары и наоборот;
2. На наборах тестовой пары, функция, реализуемая схемой *C*, принимает противоположные значения;
3. Минимально покрывающий интервал *и* векторов v_1, v_2 ортогонален всем конъюнкциям ЭНФ, не содержащим литеру, сопоставляемую пути α;
4. Наборы (v_1, v_2) тестовой пары порождаются одной и той же непустой конъюнкцией.

При замене набора v_1 набором v_2 схема может попасть в промежуточное состояние, являющееся тестовым набором v_1 для другого пути схемы. В результате мы можем определить задержку другого пути, а не рассматриваемого. Такая ситуация исключается при выполнении условия 3. Для проверки условия 3 в [3] предлагается воспользоваться ЭНФ схемы. Однако ЭНФ, как правило, оказывается громоздкой даже для небольших схем. Для соседних наборов тестовой пары (v_1, v_2) при переходе от v_1 к v_2 промежуточных состояний не возникает. Использование этого факта избавляет от необходимости анализировать ЭНФ.

Теорема. Соседние наборы тестовой пары (v_1, v_2), в которой v_2 есть a_p тестовый набор, а v_1 есть b_p тестовый набор и наоборот, обнаруживают робастно тестируемую неисправность задержки пути для rising transition и falling transition, соответственно.

Доказательство. Доказательство приведено в [1].

Итак, тестовая пара (b_p, a_p) обнаруживает задержку пути α, сопоставляемую rising transition, а тестовая пара (a_p, b_p) обнаруживает задержку пути α, сопоставляемую falling transition.

Множество всех тестовых пар соседних булевых векторов для робастно тестируемых неисправностей рассматриваемого пути в [1] предлагается представлять ROBDD-графом R_{rob}.

Из процедуры построения R_{rob} следует, что граф не содержит переменную x_i, отмечающую начало пути. Путь от корня графа R_{rob} до его 1-концевой вершины представляет конъюнкцию, такую, что булев вектор в пространстве переменных $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n$ задает тестовую пару в пространстве n переменных для рассматриваемого пути. Один из векторов пары получается приписыванием переменной x_i значения 0, а другой – значения 1.

Если на векторе v_2 и выходе одно выходной схемы, (для много выходной схемы-выходе, сопоставляемом пути α), достигается значение 1, то v_2 – вектор для rising transition, если 0, то v_2 – вектор для falling transition.

В последовательностной схеме пары соседних булевых векторов, заданных ROBDD-графом R_{rob}, сопоставляются полным состояниям, т.е. в векторах пары выделяются входные составляющие и составляющие по переменным состояния.

Из тестовой пары формируются тройки векторов, обнаруживающие задержки инверсных перепадов сигналов рассматриваемого пути, либо $v_2 - v_1 - v_2$, либо $v_1 - v_2 - v_1$. Это значит, что при использовании тестовых пар из соседних булевых векторов мы имеем возможность обнаруживать противоположные перепады значений сигналов рассматриваемого пути в комбинационной составляющей тремя векторами вместо четырех.

Однако если переменные состояния в комбинационной составляющей не доступны, предлагается из начального состояния q_0 последовательностной схемы доставлять по отдельности пары для каждой из последовательностей перепадов рассматриваемого пути. Дело здесь в том, что нам необходимо не только попасть в состояние, сопоставляемое началу последовательности из трех векторов, но далее оказаться в специальной цепочке переходов, порождаемой тройкой тестовых векторов, чем длиннее цепочка, тем ниже вероятность попадания в нее.

2. Алгоритмы построения последовательности, обнаруживающей робастно тестируемую неисправность задержки пути в схеме с памятью

Приведем алгоритмы построения последовательности векторов входных переменных последовательностной схемы, доставляющей тестовую пару на входы комбинационной составляющей схемы из начального состояния q_0. Напомним, что речь идет о соседних булевых векторах тестовой пары по переменной, отмечающей начало исследуемого пути. Напомним, что речь идет о соседних булевых векторах тестовой пары по переменной, отмечающей начало исследуемого пути. Эти алгоритмы формируют, вместе с соответствующими тестовыми парами, последовательности, обнаруживающие робастно тестируемые неисправности задержек путей, по одной входной последовательности для каждой последовательности перепадов значений сигналов рассматриваемого пути из заданного множества путей.

Пусть последовательностная схема получена из State Transition Graph (STG) описания, в котором выполнено кодирование состояний. Использовались либо коды минимальной длины, либо другие коды. Важно иметь в виду следующие факты.

1. Если в таблице переходов конечного автомата, сопоставляемой STG-описанию, отсутствуют петли, то невозможно построить последовательность, доставляющую тестовую пару для путей, начала которых отмечены входной переменной в последовательностной схеме.

2. Если в результате кодирования состояний в полученном множестве кодов (в рабочей области, задаваемой STG-описанием) отсутствуют соседние векторы, то невозможно построить последовательность, доставляющую тестовую пару для путей, начала которых отмечены переменной состояния (внутренней переменной) в последовательностной схеме.

Итак, множество всех пар соседних булевых векторов, задающих полные состояния схемы и обнаруживающих робастно тестируемые неисправности задержек рассматриваемого пути в комбинационной части последовательностной схемы, представлено ROBDD-графом R_{rob}. В ROBDD R_{rob} сначала выполняется разложение Шеннона по внутренним, а затем по входным переменным. Это следует из метода его построения [1].

Имея в виду, что тестовые пары (v_1, v_2), задаваемые R_{rob}, состоят из входной и внутренней составляющих, представим их в виде: $v_1 = v_1^{in}, v_1^s$; $v_2 = v_2^{in}, v_2^s$, т.е. тестовую пару далее будем задавать следующим образом: $(v_1^{in}, v_1^s; v_2^{in}, v_2^s)$.

Рассматриваются следующие ситуации.

a. Соседние булевы векторы, извлекаемые из R_{rob}, отличаются по входной переменной.
b. Соседние булевы векторы, извлекаемые из R_{rob}, отличаются по внутренней переменной.

Случай (а).

Рассмотрим тестовую пару $(v_1^{in}, v_1^s; v_2^{in}, v_1^s)$ векторов, которую необходимо подать на комбинационную составляющую последовательностной схемы для обнаружения задержки заданного пути α в условиях отличая векторов тестовой пары (составляющих v_1^{in}, v_2^{in}) по входной переменной x_i и при совпадении составляющих по переменным состояния. Для простоты положим, что переменная, отмечающая заданный путь, не имеет инверсии. Опишем процедуру доставки тестовой пары в соответствующие моменты времени.

1. Предварительно с помощью подходящей входной последовательности, построение которой будет описано далее, устанавливаем схему (в момент t_1) в состояние v_1^s. Нас не интересует ни входной сигнал, ни состояние, из которого мы попадаем в состояние v_1^s.

2. Из состояния v_1^s под действием входного сигнала v_1^{in}, поступающего в момент t_2 (именно в этот момент мы обеспечиваем подачу на входы комбинационной составляющей последовательностной схемы первого вектора v_1^{in}, v_1^s тестовой пары) переходим в состояние v_1^s, т.е. реализуем петлю в автоматной диаграмме переходов с целью достижения состояния v_1^s в момент t_3.

3. В состоянии v_1^s в момент t_3 подаем на входы схемы вектор v_2^{in} ,т.е. обеспечиваем поступление второго вектора v_2^{in}, v_1^s тестовой пары на входы комбинационной состав-

ляющей. Неважно, каким будет следующее состояние схемы, важно только, что должно произойти изменение сигнала на выходе, сопоставляемом рассматриваемому пути.

В момент t_4 наблюдаем задержку на соответствующем пути выходе комбинационной составляющей, если задержка имеет место.

Заметим, что соседние векторы тестовой пары поступают в следующие друг за другом моменты времени.

Переходы в моменты времени t_1, t_2, t_3 иллюстрируются рис. 1.

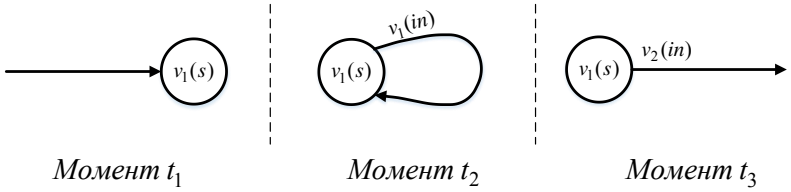


Рис. 1. Обеспечение цепочки переходов для соседних векторов, отличающихся по входной переменной

Итак, для доставки тестовой пары необходимо из начального состояния q_0 попасть в состояние v_1^s. Это можно сделать, построив последовательность, устанавливающую схему из начального состояния в некоторое состояние из множества состояний типа v_1^s. Причем, нас интересуют не все состояния, имеющие петлю, а только те, в которых петля реализуется по входной составляющей v_1^{in}, содержащейся среди полных состояний, представленных R_{rob}. Обеспечив поступление в этом состоянии входной составляющей, далее формируем вектор v_2^{in}, v_1^s, заменив в v_1^{in} значение переменной x_i инверсным.

Для построения установочной последовательности из начального состояния q_0 в состояние из заданного множества воспользуемся алгоритмом, предложенным в [4]. В нем множество состояний представлено ROBDD-графом R^{s_0}. Построение установочной последовательности из начального состояния является общей частью для различных типов перепадов сигналов и различных типов переменных, отмечающих начало пути. Для каждой из этих ситуаций приходится строить свой граф R^{s_0}.

Итак, строим множество R^{s_0}, для тестовой пары, в которой соседние векторы отличаются по входной переменной x_i. Из STG описания выделяем фрагмент, представляющий петли в таблице переходов соответствующего конечного автомата. Фрагмент состоит из строк вида:

$$k_1 q_i \to q_i$$
$$k_2 q_i \to q_i$$
$$\ldots$$
$$\ldots$$
$$k_m q_i \to q_i$$
$$k_{m+2} q_j \to q_j$$
$$\ldots$$
$$\ldots$$
$$k_{m+s} q_j \to q_j$$
$$\ldots$$

Здесь k_i – конъюнкции (троичные векторы) на множестве входных переменных схемы, q_i, q_j, ... – состояния STG-описания (состояния конечного автомата), представленные булевыми векторами при кодировании. Левую часть рассматриваемого фрагмента, т.е. конъюнкции от входных переменных вместе с приписанными им состояниями представляем в виде ROBDD R_n. Этот граф пригодится нам при рассмотрении всех

путей из предъявленного множества, начала которых отмечены входными переменными последовательностной схемы. Перейдем к рассмотрению заданного пути α.

Последовательность шагов для rising transition пути α

1. Из R_{rob} формируем a_p-тестовые наборы (наборы v_1^{in}, v_1^s) тестовой пары, сопоставляемые входной литере x_i, представляя их соответствующим ROBDD-графом: $R_{rob/x_i} = R_{rob} \,\&\, x_i$.
2. Среди a_p –тестовых наборов выбираем такие, которые порождают петли: $R_{a/n} = R_{rob/x_i} \,\&\, R_n$, т.е. получаем множество всех векторов вида v_1^{in}, v_1^s, сопоставляемых моменту времени t_2 при тестировании задержки пути.
3. Выделяем в $R_{a/n}$ множество внутренних состояний и представляем его графом R^{s_0}.
4. Получив вектор v_1^{in}, v_1^s, далее формируем вектор v_2^{in}, v_1^s, который порождается графом R_{rob}, и подаем его в момент t_3. Вектор v_2^{in} отличается от вектора v_1^{in} по переменной x_i.

При нахождении тестовой пары для falling transition выполняем следующие шаги алгоритма для b_p-тестовых наборов.

Последовательность шагов для falling transition пути α

1. Из R_{rob} формируем b_p-тестовые наборы (наборы v_1^{in}, v_1^s) тестовой пары, сопоставляемые входной литере $\overline{x}_i$, представляя их соответствующим ROBDD-графом: $R_{rob/\overline{x}_i} = R_{rob} \,\&\, \overline{x}_i$.
2. Среди b_p-тестовых наборов выбираем такие, которые порождают петли: $R_{b/n} = R_{rob/\overline{x}_i} \,\&\, R_n$, т.е. получаем множество всех векторов вида v_1^{in}, v_1^s, сопоставляемых моменту времени t_2, при тестировании задержки пути.
3. Выделяем в $R_{b/n}$ множество внутренних состояний и представляем его графом R^{s_0}.
4. Получив вектор v_1^{in}, v_1^s, далее формируем вектор v_2^{in}, v_1^s, который порождается графом R_{rob}, и подаем его в момент t_3. Вектор v_2^{in} отличается от вектора v_1^{in} по переменной x_i.

Случай (б).

Рассмотрим тестовую пару $\left(v_1^{in}, v_1^s; v_1^{in}, v_2^s\right)$ векторов, которую необходимо подать на комбинационную составляющую последовательностной схемы для обнаружения задержки заданного пути α в условиях отличая векторов тестовой пары (составляющих v_1^s, v_2^s) по переменной состояния z_i и при совпадении составляющих по входным переменным. Для простоты положим, что переменная, отмечающая заданный путь, не имеет инверсии. Опишем процедуру доставки тестовой пары в соответствующие моменты времени.

1. Предварительно с помощью подходящего входного воздействия устанавливаем схему (в момент t_1) в состояние v_1^s. Нас не интересует ни входной сигнал, ни состояние, из которого мы попадаем в состояние v_1^s.
2. Из состояния v_1^s под действием входного сигнала v_1^{in}, поступающего в момент t_2 (именно в этот момент мы обеспечиваем поступление на входы комбинационной составляющей последовательностной схемы первого вектора v_1^{in}, v_1^s тестовой

пары) переходим в состояние v_2^s, т.е. переходим в соседнее состояние, отличающееся от v_1^s по переменной z_i.

3. В состоянии v_2^s в момент t_3 подаем на входы схемы вектор v_1^{in}, т.е. обеспечиваем поступление второго вектора v_1^{in}, v_2^s тестовой пары на входы комбинационной составляющей. Неважно, каким будет следующее состояние схемы, важно только, что должно произойти изменение сигнала на выходе, сопоставляемом рассматриваемому пути.

В момент t_4 наблюдаем задержку на соответствующем пути выходе комбинационной составляющей, если задержка имеет место.

Заметим, что соседние векторы тестовой пары поступают в следующие друг за другом моменты времени.

Переходы в моменты времени t_1, t_2, t_3 иллюстрируются рис. 2.

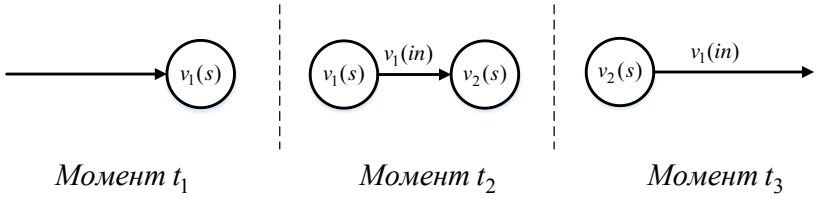


Рис. 2. Обеспечение цепочки переходов для соседних векторов, отличающихся по переменной состояния

Итак, для доставки тестовой пары во всех отмеченных выше ситуациях необходимо из начального состояния q_0 попасть в состояние v_1^s. Причем, нас интересуют не все состояния, имеющие переходы в соседние состояния, а только те, в которых переход реализуется по входной составляющей v_1^{in}, содержащейся среди полных состояний, представленных R_{rob}. Обеспечив поступление в этом состоянии входной составляющей, далее формируем вектор v_1^{in}, v_2^s, заменив в v_1^s значение переменной z_i на инверсное.

Из STG описания выделяем фрагмент, представляющий переходы в соседние состояния. Фрагмент состоит из строк вида:

$$k_1 q_{i_1} \to q_{j_1}$$
$$k_2 q_{i_1} \to q_{j_1}$$
$$\ldots$$
$$\ldots$$
$$k_m q_{i_1} \to q_{j_1}$$
$$k_{m+2} q_{i_2} \to q_{j_2}$$
$$\ldots$$
$$\ldots$$
$$k_{m+s} q_{i_2} \to q_{j_2}$$
$$\ldots$$

Здесь k_i – конъюнкции (троичные векторы) на множестве входных переменных схемы, q_{i_1}, q_{j_1}, q_{i_2}, q_{j_2}, ... – состояния STG-описания (состояния конечного автомата), представленные булевыми векторами при кодировании. Левую часть рассматриваемого фрагмента, т.е. конъюнкции от входных переменных вместе с приписанными им состояниями представляем в виде ROBDD R_s. Этот граф пригодится нам при рассмотрении всех путей из предъявленного множества, начала которых отмечены переменными состояний последовательностной схемы. Перейдем к рассмотрению заданного пути α.

Последовательность шагов для rising transition пути α

1. Из R_{rob} формируем a_p-тестовые наборы (наборы v_1^{in}, v_1^s) тестовой пары, сопоставляемые переменной состояний z_i, представляя их соответствующим ROBDD-графом: $R_{rob/z_i} = R_{rob} \,\&\, z_i$.
2. Среди a_p-тестовых наборов выбираем такие, которые порождают переходы в соседние состояния по переменной z_i: $R_{a/s} = R_{rob/z_i} \,\&\, R_s$, т.е. получаем множество всех векторов вида v_1^{in}, v_1^s, сопоставляемых моменту времени t_2 при тестировании задержки пути.
3. Выделяем в $R_{a/s}$ множество внутренних состояний и представляем их графом R^{s_0}.
4. Получив вектор v_1^{in}, v_1^s, далее формируем вектор v_1^{in}, v_2^s, который порождается R_{rob}, и подаем его в момент t_3. Вектор v_2^s отличается от вектора v_1^s по переменной z_i.

При нахождении тестовой пары для falling transition выполняем следующие шаги алгоритма для b_p-тестовых наборов.

Последовательность шагов для falling transition пути α

1. Из R_{rob} выбираем b_p-тестовые наборы (наборы v_1^{in}, v_1^s) тестовой пары, сопоставляемые входной литере $\overline{z}_i$, представляя их соответствующим ROBDD-графом: $R_{rob/\overline{z}_i} = R_{rob} \,\&\, \overline{z}_i$.
2. Среди b_p-тестовых наборов выбираем такие, которые порождают переходы в соседние состояния по переменной z_i: $R_{b/s} = R_{rob/\overline{z}_i} \,\&\, R_s$, т.е. получаем множество всех векторов вида v_1^{in}, v_1^s, сопоставляемых моменту времени t_2 при тестировании задержки пути.
3. Выделяем в $R_{b/s}$ множество внутренних состояний и представляем их графом R^{s_0}.
4. Получив вектор v_1^{in}, v_1^s, далее формируем вектор v_1^{in}, v_2^s, который порождается R_{rob}, и подаем его в момент t_3. Вектор v_2^s отличается от вектора v_1^s по переменной z_i.

Итак, во всех выше перечисленных алгоритмах (п. 3) отыскания тестовых пар формируется соответствующее множеств R^{s_0}.

Находим последовательность [4], устанавливающую схему из начального состояния q_0 в одно из состояний соответствующего отыскиваемой паре множества R^{s_0}, т.е. в состояние v_1^s. Получив это состояние, используем для тестовой пары этого же типа ROBDD-граф из множества $\{R_{a/n}, R_{a/s}, R_{b/n}, R_{b/s}\}$ и по нему находим первый вектор тестовой пары, а затем второй вектор способом, указанными в п. 4 соответствующего отыскиваемой паре алгоритма.

Строим для каждого пути из заданного множества и типа перепадов значений сигналов соответствующие последовательности, обнаруживающие задержку пути, объединяем их и получаем тестовую последовательность для заданного множества путей. Отметим, что полученная тестовая последовательность состоит из фрагментов, начинающихся в состоянии q_0.

3. Результаты экспериментов

Описанные выше алгоритмы были реализованы в виде программы. Для операций над ROBDD-графами использовалась библиотека CUDD. Были проведены эксперимен-

ты на некоторых бенчмарках, результаты которых приведены в таблице. В схемах, представленных в бенчмарках, рассматривались только самые длинные пути. Длина установочной последовательности ограничивалась тысячью входными векторами, однако в процессе эксперимента установочные последовательности не достигали и сотни входных векторов. В последнем столбце таблицы показана доля (в процентах) путей, для которых из начального состояния последовательностной схемы удалось доставить тестовую пару, обнаруживающую робастно тестируемую неисправность задержки пути. Эта доля определяется по отношению ко всем путям, для которых существуют тестовые пары для тех же неисправностей в условиях доступности входов, сопоставляемых переменным состояний комбинационного эквивалента.

Таблица 1

Результаты экспериментов

name	I	s	xr	xf	zr	zf	xob	zob	ob	N	%
s27	4	3	8	8	18	2	8	2	10	58	17.2%
s208	11	8	0	0	0	3	0	0	0	643	0%
s298	3	14	40	0	0	0	0	0	0	1179	0%
s386	7	6	45	57	320	572	0	160	160	1351	11.8%
s510	19	6	0	0	906	724	0	575	575	1373	41.9%
s820	18	5	34	451	1860	1392	0	1313	1313	3055	42.87%
s832	18	5	34	450	1859	1391	0	1312	1312	3052	42.98%
s1488	8	6	0	58	2578	2209	0	1612	1612	5384	29.94%
s1494	8	6	0	58	2628	2247	0	1626	1626	5351	30.39%

Обозначения:

name – название бенчмарка;

I – количество входов;

s – количество переменных состояний;

xr – количество путей, начало которых помечено входной переменной и для которых можно доставить тестовую пару из начального состояния схемы. Тестовая пара предназначена для rising transition;

xf – количество путей, начало которых помечено входной переменной и для которых можно доставить тестовую пару из начального состояния схемы. Тестовая пара предназначена для falling transition;

zr – количество путей, начало которых помечено переменной состояния и для которых можно доставить тестовую пару из начального состояния схемы. Тестовая пара предназначена для rising transition;

zf – количество путей, начало которых помечено переменной состояния и для которых можно доставить тестовую пару из начального состояния схемы. Тестовая пара предназначена для falling transition;

xob – количество путей, начало которых помечено входной переменной и для которых можно доставить тестовые пары для обоих перепадов значений сигналов из начального состояния схемы;

zob – количество путей начало которых помечено переменной состояний и для которых можно доставить тестовые пары для обоих перепадов значений сигналов из начального состояния схемы;

ob – общее количество робастно тестируемых путей, для каждого из которых обнаружимы оба перепада значений сигналов;

N – количество путей, для которых существуют непустые графы R_{rob};

% – доля (в процентах) робастно тестируемых путей, для которых удалось доставить тестовую пару, среди путей, для которых существуют непустые графы R_{rob}.

Заключение

Ранее был предложен метод отыскания всех тестовых пар, состоящих из соседних наборов булевых векторов, обнаруживающих робастно тестируемые неисправности задержек пути в условиях их доставки на входы комбинационного эквивалента схемы с памятью. В данной работе предложен способ доставки таких тестовых пар из начального состояния q_0 схемы с памятью в условиях ограничения на длину последовательности, основанный на операциях над ROBDD-графами. Работа ориентирована на исследование возможности отказа от использования техник сканирования, требующих больших аппаратурных затрат. В работе выделены классы автоматов и способы кодирования состояний, для которых не возможна доставка тестовых пар, существующих в комбинационной составляющей. Проведенные эксперименты на контрольных примерах (benchmarks) показали, что далеко не всегда удается доставить хотя бы одну тестовую пару, не важно, какую именно, для пути, имеющего тестовые пары для комбинационной составляющей в условиях доступности ее переменных состояния. Это значит, что для некоторых схем с памятью обнаружение робастно тестируемых неисправностей задержек путей без существенных аппаратурных затрат может оказаться невозможным.

ЛИТЕРАТУРА

1. *Matrosova A.Yu., Andreeva V.V., Nikolaeva E.A.* Finding Test Pairs for PDFs in Logic Circuits Based on Using Operations on ROBDDs //Russian Physics Journal. 2018. Vol. 61, № 5. pp. 994-999.

2. *Matrosova A.Yu., Andreeva V.V., Tychinskiy V.Z., Goshin G.G.* Applying ROBDDs for delay testing of logical circuits. Russian. Physics Journal. 2019.v. 62, № 5.pp. 86-94.

3. *Матросова А.Ю., Липский В.Б.* Свойства пар тестовых наборов, обнаруживающих неисправности задержек путей в логических схемах VLSI высокой производительности //Автоматика и телемеханика. 2015. № 4. С. 135-148.

4. *Matrosova A., Andreeva V., Melnikov A.* ROBDDs Application for Finding the Shortest Transfer Sequence of Sequential Circuit or Only Revealing Existence of this Sequence without Deriving the Sequence itself //Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2016). Yerevan: IEEE Computer Society, 2016. P. 513-516.

СИНТЕЗ ВЕНТИЛЬНЫХ СХЕМ, МАСКИРУЮЩИХ НЕИСПРАВНОСТИ, С ИСПОЛЬЗОВАНИЕМ SAT-РЕШАТЕЛЕЙ

Провкин В.А.
Томский государственный университет
prowkan@mail.ru

Введение

В процессе производства интегральных схем возможно изменение спецификации изготавливаемой схемы, возможно также обнаружение логических неисправностей на поздних стадиях производства схемы. В таких случаях приходится либо возвращать схемы на более ранние стадии производства, либо выбрасывать их как непригодные к использованию. Подобные ситуации снижают выход годных схем, что весьма нежелательно. Чтобы увеличить количество исправных схем, применяется ECO (Engineering Change Order) технологии. Неисправность в изготавливаемой схеме можно маскировать с помощью корректирующей схемы, подключаемой к входам корректируемой схемы и либо к выходам, либо к внутренним полюсам этой же схемы.

Ранее нами был предложен способ построения корректирующих схем, основанный на операциях над ROBDD-графами. Однако существуют такие схемы, в которых размеры ROBDD-графов, представляющих поведение схемы, растут экспоненциально с увеличением количества переменных, что исключает возможность применения разработанного метода. В данной работе предлагается альтернативный подход, основанный на использовании задачи выполнимости булевой формулы, записанной в конъюнктивной нормальной форме (КНФ). КНФ получается путём применения специального преобразования, а размер КНФ линейно зависит от сложности исходной формулы.

1. Постановка задачи

Рассмотрим комбинационную схему *C* (комбинационную составляющую последовательностной схемы), состоящую из вентилей. Каждому внутреннему полюсу v соответствует частично определённая булева функция (частичная булева функция). Частичная функция полюса определяется множествами единичных ($M_1(f_v)$) и нулевых наборов ($M_0(f_v)$). Пусть на некотором полюсе имеет место константная неисправность. Это значит, что на всех наборах значений входных переменных этому полюсу схемы сопоставляется одно и то же значение 1(0), т.е. искажается частичная функция, сопоставляемая полюсу.

Рассмотрим частичную функцию f_1 и полностью определенную функцию f_2, представленные парами множеств единичных и нулевых наборов значений своих переменных: $M_1(f_1)$, $M_0(f_1)$; $M_1(f_2)$, $M_0(f_2)$. Будем говорить, что полностью определенная функция f_2 реализует частичную функцию f_1, если выполняются условия: $M_1(f_1)\cap M_0(f_2)=\varnothing$, $M_0(f_1)\cap M_1(f_2)=\varnothing$. Это значит, что $M_1(f_2)\supseteq M_1(f_1)$ и $M_0(f_2)\supseteq M_0(f_1)$.

Каждому внутреннему полюсу в общем случае соответствует частичная булева функция, поскольку на некоторых наборах значений входных переменных (входных наборах) схемы соответствующее значение внутреннего полюса может не влиять на значения выходов схемы. Частичная функция, сопоставляемая внутреннему полюсу, определяется только на тех входных наборах, на которых смена соответствующего значения в рассматриваемом полюсе меняет значение на выходах схемы. Проявление логической неисправности внутреннего полюса схемы означает искажение частичной булевой функции этого полюса в области её определения. Соответственно, при построении маскирующей схемы («схемы-заплатки») необходимо использовать наборы из области определения частичной функции, чтобы найти, по возможности, более простую логическую схему, поведение которой представляется функцией f_2.

2. Представление частичной функции внутреннего полюса, зависящей от внешних входов, в виде логических схем

Рассмотрим многовыходную комбинационную схему *C* (комбинационную часть последовательностной схемы), обозначим символом v внутренний полюс, для которого будем строить частично определённую функцию. Выделим подсхему, выходом которой является полюс v, а входами – входы схемы *C*. Эта схема реализует некоторую функцию, обозначим её как $\phi(x_1,\ldots,x_n)$. Так же выделим одновыходные подсхемы, в которых входами является полюс *v* и полюсы схемы *C*, а выходом является один из выходов схемы *C*. Всего таких подсхем m_v – число выходов схемы *C*, которые связаны с полюсом *v* (т.е. существует путь (возможно, не единственный), началом которого является полюс *v*, а концом – один из выходов схемы *C*). Обозначим функцию, реализуемую такой подсхемой, как $f_i(v,x_1,\ldots,x_n)$, $i=\overline{1,m_v}$.

Набор значений входных полюсов схемы является тестом для одиночной константной неисправности внутреннего полюса одно выходной схемы, если смена значения в этом полюсе приводит к изменению значения на выходе схемы. Из этого следует, что тестовые наборы для неисправности в полюсе *v* являются корнями уравнения $f_i(0,x_1,\ldots,x_n)\oplus f_i(1,x_1,\ldots,x_n)=1$. Набор значений входных полюсов для многовыходной схемы является тестовым, если смена значения внутреннего полюса приводит к изменению значения хотя бы на одном из выходов схемы (возможно, сразу на нескольких). Таким образом, тестовые наборы для одиночной константной неисправности

внутреннего полюса v многовыходной схемы являются решениями уравнения $\bigvee_{i=1}^{m_v}\left[f_i\left(0,x_1,\ldots,x_n\right)\oplus f_i\left(1,x_1,\ldots,x_n\right)\right]=1$.

Для разделения множества всех тестовых наборов на множества тестовых наборов неисправности «константа 0» и неисправности «константа 1» используем функцию $\phi\left(x_1,\ldots,x_n\right)$. Множества, представляющие область определения частичной функции полюса *v*, имеют вид:

$$M_1=\left\{\left(x_1,\ldots,x_n\right)\in B_2^n:\ \phi\left(x_1,\ldots,x_n\right)\wedge\left(\bigvee_{i=1}^{m_v}\left[f_i\left(0,x_1,\ldots,x_n\right)\oplus f_i\left(1,x_1,\ldots,x_n\right)\right]\right)=1\right\},$$

$$M_0=\left\{\left(x_1,\ldots,x_n\right)\in B_2^n:\ \overline{\phi\left(x_1,\ldots,x_n\right)}\wedge\left(\bigvee_{i=1}^{m_v}\left[f_i\left(0,x_1,\ldots,x_n\right)\oplus f_i\left(1,x_1,\ldots,x_n\right)\right]\right)=1\right\}.$$

Здесь $B_2=\{0,1\}$, $B_2^n=\underbrace{\{0,1\}\times\{0,1\}\times\ldots\times\{0,1\}}_{n\text{ раз}}$.

Эти множества можно представить в виде логических схем. На рис. 1 показано представление множества M_1 в виде схемы.

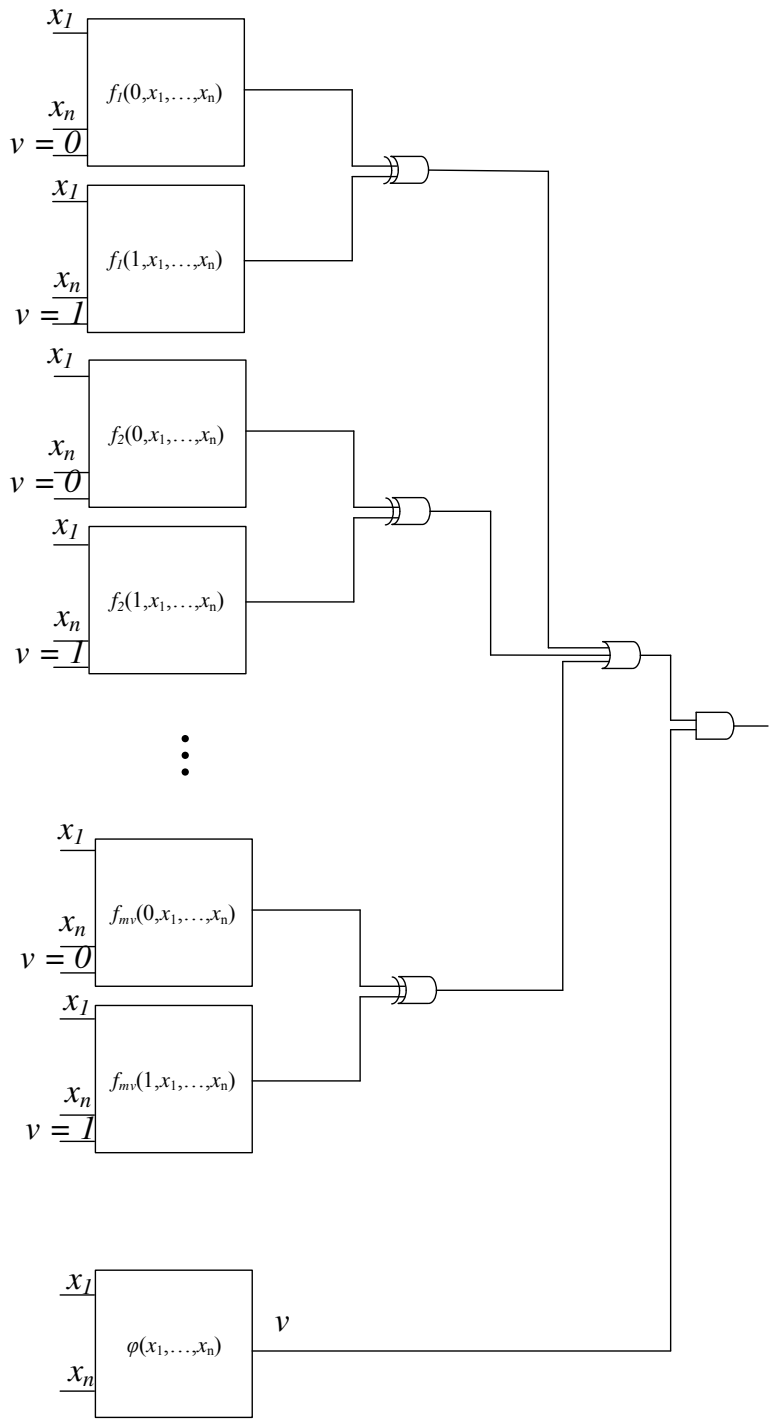


Рис. 1. Схемное представление множества M_1 частично определённой функции

Вообще говоря, по этим схемам уже можно строить формулу, по которой потом получается безызбыточная ДНФ. Однако, эти схемы можно упростить, что, как показывают экспериментальные результаты, значительно уменьшает время построения ДНФ.

3. Упрощение схем, представляющих множества частично определённой функции

Это упрощение основано на том, что в схемах, представленных в предыдущем разделе, содержатся элементы, такие, что подсхемы, выходом которых являются выходы этих элементов, реализуют одинаковые функции от одних и тех же внешних входов. Будем называть эти элементы изоморфными. Эти подсхемы не зависят от полюса v, и могут быть заменены одной общей подсхемой.

Алгоритм упрощения схемы:

1. Выбираем в схеме очередной элемент, такой, что подсхема, выходом которой является выход этого элемента, не зависит от переменной *v*. Если просмотрены все элементы, то работа алгоритма заканчивается.
2. Находим в схеме элемент, такой, что подсхема, выходом которой является выход этого элемента, реализует такую же функцию, что и в п. 1, от тех же входных переменных. Если такого элемента нет, то возвращаемся на шаг 1. Иначе переходим к шагу 3.
3. Удаляем изоморфный элемент из схемы, а те входные полюсы, которые были соединены с выходом этого элементом, соединяются с выходом элемента, выбранного на шаге 1. Далее переходим к шагу 1.

4. Построение безызбыточной ДНФ с использованием задачи выполнимости

Задача выполнимости булевой формулы (SAT-задача) формулируется следующим образом. Задана некоторая булева формула. Можно ли заменить переменные этой формулы соответствующими константами, так чтобы она обратилась в константу 1, т.е. выполнима ли эта формула? Если формула не выполнима, то она представляет константу 0. Вопрос о существование полиномиального алгоритма для определения выполнимость КНФ на данный момент является открытым: такого алгоритма не предложено, но при этом не доказано, что его не существует.

В 2018 г. была опубликована работа [1], в которой предложен алгоритм получения безызботочной ДНФ (БДНФ), основанный на определении выполнимости булевых формул. Этот алгоритм может применяться для построения БДНФ как для полностью определённых, так и для частичных булевых функций. При некоторых дополнительных условиях получаемая БДНФ является канонической (уникальной) для заданного порядка фиксирования константами переменных.

Для построения БДНФ по такому алгоритму необходимо получить такие формулы, которые бы принимали единичное значение на наборах, соответствующих единичным или нулевым наборам заданной функции и нулевое на всех остальных. Большинство SAT-решателей определяют выполнимость формул, записанных в конъюнктивной нормальной форме.

Традиционный способ построения КНФ из булевой формулы состоит в использовании тождественных преобразований (правил Де Моргана и дистрибутивного закона конъюнкции). Однако при таком подходе длина КНФ возрастает экспоненциально. Поэтому для построения КНФ используется преобразование Цейтина [2], при котором длина КНФ линейно зависит от размера формулы (числа элементов схемы, представляющей булеву функцию). Это преобразование заключается в следующем: каждой подформуле (каждому элементу схемы) сопоставляется новая внутренняя переменная. Затем для этой подформулы или элемента строится функция эквивалентности – функция вида $y \leftrightarrow \phi(x_1,\dots x_n)$, где $\phi(x_1,\dots x_n)$ – функция, реализуемая подформулой (элементом схемы). Для элемента схемы $x_1,\dots,x_n$ – переменные, сопоставляемые входам элемента, которые могут быть либо входными переменными схемы, либо ее внутренними переменными. Далее функция эквивалентности преобразуется в КНФ. Для эле-

ментарных функций эти КНФ могут быть построены заранее. Они имеют следующий вид:

$$y = \overline{x}: \ (x \vee y)(\overline{x} \vee \overline{y})$$

$$y = x_1 \wedge x_2 : (x_1 \vee \overline{y})(x_2 \vee \overline{y})(\overline{x_1} \vee \overline{x_2} \vee y)$$

$$y = x_1 \vee x_2 : (\overline{x_1} \vee y)(\overline{x_2} \vee y)(x_1 \vee x_2 \vee \overline{y})$$

$$y = x_1 \overline{\wedge} x_2 : (x_1 \vee y)(x_2 \vee y)(\overline{x_1} \vee \overline{x_2} \vee \overline{y})$$

$$y = x_1 \overline{\vee} x_2 : (\overline{x_1} \vee \overline{y})(\overline{x_2} \vee \overline{y})(x_1 \vee x_2 \vee y)$$

$$y = x_1 \oplus x_2 : \ (x_1 \vee x_2 \vee \overline{y})(\overline{x_1} \vee \overline{x_2} \vee \overline{y})(\overline{x_1} \vee x_2 \vee y)(x_1 \vee \overline{x_2} \vee y)$$

КНФ формулы (схемы) получается объединением знаком конъюнкции функций эквивалентности всех подформул (элементов схемы). В дальнейшем будем говорить о логических схемах и КНФ, им сопоставляемым.

Такая КНФ представляет собой все допустимые комбинации значений сигналов на входных, внутренних и выходных полюсах схемы. Для получения КНФ, представляющей единичные (нулевые) наборы функции, необходимо добавить к КНФ одно литеральный дизъюнкт, состоящей из переменной (её инверсии), соответствующей выходному полюсу схемы, получаем $КНФ_1$ ($КНФ_0$). Полученные КНФ не эквивалентны КНФ, полученным с помощью тождественных преобразований, но они являются *эквивыполнимыми*. Две КНФ называются эквивыполнимыми, если одна из них выполнима тогда и только тогда, когда выполнима другая. Между выполняющими наборами этих КНФ имеется взаимно однозначное соответствие.

Выполняющий набор для КНФ, полученной применением тождественных преобразований, получается из выполняющего набора КНФ, полученной преобразованием Цейтина, путём отбрасывания значений внутренних переменных, сопоставляемых выходам элементов схемы (внутренним полюсам схемы). Выполняющий набор КНФ, полученной преобразованием Цейтина, получается путём вычисления значения на внутренних полюсах схемы (двоичного моделирования) на заданном выполняющем наборе для КНФ, полученной применением тождественных преобразований.

Алгоритм построения безызбыточной ДНФ с использованием задачи выполнимости:

1. *Построение нового допустимого интервала.* Анализируется выполнимость КНФ, представляющей единичные значения функции ($КНФ_1$). Если КНФ не выполнима, то работа алгоритма завершена. Иначе, из полученного выполняющего набора выделяется набор входных переменных. Он образует новый допустимый интервал.
2. *Расширение допустимого интервала до максимального.* При подстановке значений входных переменных, полученных на предыдущем шаге, в КНФ, представляющую нулевые наборы ($КНФ_0$) функции, представляемой схемой, она становится невыполнимой. При расширении интервала по некоторой переменной x_i, в $КНФ_0$ подставляются значения всех заданных переменных, кроме x_i. Если $КНФ_0$ по-прежнему остаётся невыполнимой, то расширенный интервал не пересекается с областью нулевых значений функции, и переменная x_i может быть удалена из конъюнкции. Если же $КНФ_0$ становится выполнимой, то интервал не может быть расширен по этой переменной, т.к. такой интервал пересекается с областью нулевых значений функции.
3. После расширения интервала до максимального его предлагается исключить из $КНФ_1$. С этой целью в $КНФ_1$ добавляется *запрещающий дизъюнкт*, полученный

инверсией конъюнкции, представляющий максимальный интервал. Возвращаемся к п. 1, используя новую КНФ$_1$, при этом КНФ$_0$ должна сохраняться.

В результате получаем ДНФ, состоящую из максимальных интервалов. Из построения следует, что каждый интервал является безызбыточным по отношению к предыдущим интервалам, однако он может быть избыточным по отношению к последующим. Поэтому для построения безызбыточной ДНФ необходимо проверить каждый интервал на безызбыточность. Поступаем следующим образом.

Пусть мы хотим проверить на безызбыточность конъюнкцию K_i. Для этого мы строим КНФ, состоящую из инверсий всех остальных конъюнкций: $C = \overline{K_1} \wedge \ldots \wedge \overline{K_{i-1}} \wedge \overline{K_{i+1}} \ldots \wedge \overline{K_m}$. Далее, подставляем значения переменных, соответствующих этой конъюнкции (иначе говоря, получаем КНФ $K_i \wedge \overline{K_1} \wedge \ldots \wedge \overline{K_{i-1}} \wedge \overline{K_{i+1}} \ldots \wedge \overline{K_m}$). Полученную КНФ анализируем на выполнимость. Если КНФ невыполнима, то конъюнкция K_i является избыточной и может быть удалена из ДНФ. Если КНФ выполнима, то K_i нельзя удалить из ДНФ, т.к. это приведёт к потере некоторых единичных значений функции, представляемой схемой.

5. Маскирование неисправности внутреннего полюса схемой, соединяемой с внешними входами схемы

После получения множеств M_1, M_0 частично определённой функции для синтеза корректирующей схемы необходимо найти полностью определённую функцию, которая будет реализацией частичной функции полюса. В данной работе для этого применяется система ESPRESSO [3], которая реализует одноимённый алгоритм минимизации частично определённых булевых функций. Полученная реализация функции представляется в виде безызбыточной ДНФ. Эта ДНФ используется в качестве задания на синтез схемы. Для синтеза используется система ABC [4], разработанная в Калифорнийском университете Беркли.

Маскирование константной неисправности на внутреннем полюсе схемы выполняется следующим образом. Пусть корректируемая схема C имеет n входов и m выходов. Обозначим v – внутренний полюс схемы, на котором в процессе тестирования была обнаружена неисправность (рис. 2).

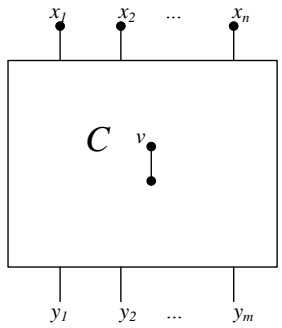


Рис. 2. Комбинационная схема с выделенным неисправным внутренним полюсом

Построив для этого полюса частичную функцию и её реализацию ранее описанным способом, мы получим некоторую вентильную схему, зависящую от входов схемы. С её помощью мы можем замаскировать неисправность полюса v при условии доступности входов схемы и внутреннего полюса. Для этого разрываются линии связи, идущие из полюса v, и выход маскирующей схемы C_p соединяется с разорванными входами элементов. Входы маскирующей схемы подключаются к соответствующим входам исходной схемы C. Результат показан на рис. 3, 4.

Если неисправный полюс являлся точкой ветвления (рис. 3), то в этом случае разрываются все линии связи, исходящие из этого полюса, а выход маскирующей схемы разветвляется, и каждая ветвь направляется на соответствующий внутренний полюс маскируемой схемы (рис. 4).

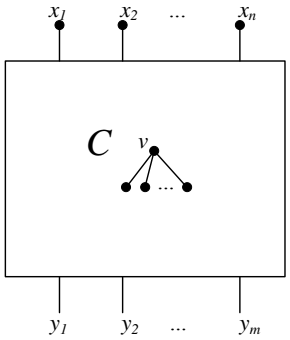


Рис. 3. Пример неисправного полюса, являющегося точкой ветвления

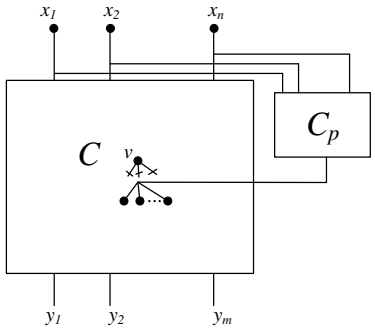


Рис. 4. Маскирование полюса, являющегося точкой ветвления

6. Построение частичной функции от внутренних полюсов методом отображения

Если технология изготовления схемы позволяет вводить дополнительные подсхемы, входы которых могут быть соединены с внутренними полюсами исходной схемы, то размер маскирующей схемы может быть существенно уменьшен. Для этого синтезируется схема, реализующая частичную функцию внутреннего полюса, зависящую от внутренних полюсов, предшествующих неисправному. Входы такой схемы соединяются с выходами внутренних полюсов, а выход соединяется с входами тех полюсов, входом которых является неисправный полюс.

Опишем метод построения частичной функции, зависящей от внутренних полюсов, основанный на использовании частичной функции, зависящей от внешних полюсов. Отметим, что множества M_0 и M_1 в этом случае должны быть представлены ортогональными ДНФ. Поэтому в алгоритм построения ДНФ, описанный в разделе 4, вносятся два дополнения:

1. При расширении интервала дополнительно проверяется, чтобы расширенный интервал не пересекался ни с одним из ранее полученных интервалов.
2. Т.к. в ортогональной ДНФ каждый единичный набор содержится только в одном интервале, то проверка интервалов на безызботочность не выполняется.

Пусть имеется логическая схема, в которой выделен неисправный полюс, а также некоторое подмножество предшествующих ему внутренних полюсов u1, …, uk. Обозначим частичную функцию внутреннего полюса v, зависящую от множества внутрен-

них полюсов $u_1,...,u_k$, как $f_v(u_1,...,u_k)$. Приведем алгоритм построения частичной функции от внутренних полюсов.

Алгоритм построения частичной функции для внутренних полюсов:

1. Для каждого троичного вектора из множества M_1 (M_0) в пространстве внешних переменных $x_1,...,x_n$ (входных переменных корректируемой схемы) находится его троичный образ в пространстве внутренних переменных $u_1,...,u_k$. Этот шаг выполняется с помощью троичного моделирования.
2. Для каждого двоичного набора, полученного из полученного троичного вектора, выполняется двоичное моделирование для вычисления значения в полюсе *v* с целью определения, может ли принадлежать этот набор множеству M_1 (M_0)
3. Для каждого двоичного набора, который может принадлежать множеству M_1 (M_0) частичной функции, проверяется его достижимость при заданных входных сигналах.

Процедура вычисления троичных векторов на внутренних полюсах при подаче троичных векторов на внешние полюсы выполняется с использованием SAT решателя.

Для каждого внутреннего полюса u_j выделяется подсхема с выходом в этом полюсе, входы подсхемы являются входами исходной схемы. Затем к этой подсхеме применяется преобразование Цейтина для получения КНФ. Далее вычисляется значение в каждом полюсе u_j на троичном векторе в пространстве переменных $x_1,...,x_n$ по следующему алгоритму:

1. Определяется выполнимость КНФ при условии, что переменная u_j равна 1. Для каждой переменной x_i, которая в троичном векторе имеет определённое значение (т.е. 0 или 1), значение этой переменной подставляется в КНФ. Если КНФ невыполнима, то значение полюса u_j на заданном троичном векторе равно 0.
2. Определяется выполнимость КНФ при условии, что переменная u_j равна 0, также с подстановкой в КНФ всех определённых компонент троичного вектора. Если КНФ не выполнима, то значение полюса u_j на заданном троичном векторе равно 1.
3. Если КНФ выполнима как при $u_j = 1$, так и при $u_j = 0$, то значение полюса на заданном троичном векторе равно «–».

Вычислив значения каждой переменной u_j на троичном векторе, мы получим троичный образ этого вектора, содержащий все наборы, которые могут входить в множество M_1 (M_0) частичной функции от переменных $u_1,...,u_k$. Однако, т.к. вычисление значений для каждого внутреннего полюса выполняется независимо от других, необходимо уточнить результаты троичного моделирования. Для этого выполняется двоичное моделирование каждого набора из интервала, представленного полученным троичным вектором. Двоичное моделирование выполняется по структуре схемы. Если троичный вектор в пространстве переменных $x_1,...,x_n$ принадлежит множеству $M_1(f_v(x_1,...,x_n))$ ($M_0(f_v(x_1,...,x_n))$), и результат двоичного моделирования набора, порождённого образом этого троичного вектора, равен 1 (0), то этот набор может принадлежать множеству $M_1(f_v(u_1,...,u_k))$ ($M_0(f_v(u_1,...,u_k))$), иначе набор гарантированно не принадлежит этому множеству.

После двоичного моделирования набора он проверяется на достижимость (т.е. проверяется, может ли быть получена некоторая комбинация сигналов на множестве внутренних полюсов при подаче на входы схемы троичного вектора). Достижимость проверяется следующим образом:

1. Выделяется подсхема, входами которой являются входы корректируемой схемы, а выходами являются полюсы $u_1,...,u_k$.

2. К выделенной подсхеме применяется преобразование Цейтина для получения КНФ.
3. Переменные $x_1,...,x_n$ фиксируются значениями определённых компонент троичного вектора. Переменные $u_1,...,u_k$ фиксируются значениями компонент двоичного вектора, проверяемого на достижимость.
4. Полученная КНФ проверяется на выполнимость. Если КНФ выполнима, то двоичный вектор является достижимым, и может принадлежать множеству M_1 (M_0). Если КНФ не выполнима, то двоичный вектор не является достижимым и гарантированно не принадлежит множеству M_1 (M_0).

Если вектор является достижимым, то он включается в соответствующее множество M_1 (M_0).

7. Маскирование неисправности внутреннего полюса схемой, соединяемой с внешними входами схемы

После получения множеств M_1 и M_0 частично определённой функции, зависящей от переменных $u_1,...,u_k$, строится полностью определённая функция, которая является реализацией частичной функции. Полученная реализация используется в качестве задания на синтез маскирующей схемы.

Маскирование константной неисправности на внутреннем полюсе схемы с помощью подсхемы, реализующей функцию от внутренних переменных, выполняется следующим образом. Снова рассмотрим корректируемую схему *C*, которая имеет *n* входов и *m* выходов. Обозначим *v* – внутренний полюс схемы, на котором в процессе тестирования была обнаружена неисправность, $u_1,...,u_k$ – предшествующие ему исправные полюсы (рис. 5).

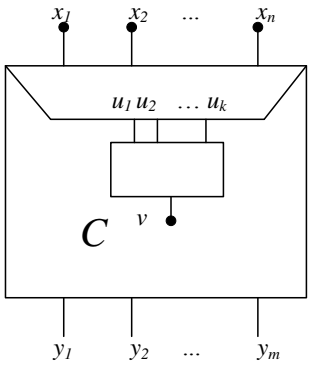


Рис. 5. Комбинационная схема с выделенной подсхемой, выход которой является неисправным

Построив для этого полюса частичную функцию и её реализацию ранее описанным способом, мы получим некоторую вентильную схему, зависящую от входов схемы. С её помощью мы можем замаскировать неисправность полюса *v* при условии доступности входов схемы и внутреннего полюса. Для этого разрываются линии связи, идущие из полюса *v*, и выход маскирующей схемы C_p соединяется с входом того полюса, входом которого являлся полюс *v*. Входы маскирующей схемы подключаются к выходам полюсов $u_1,...,u_k$. Результат показан на рис. 6.

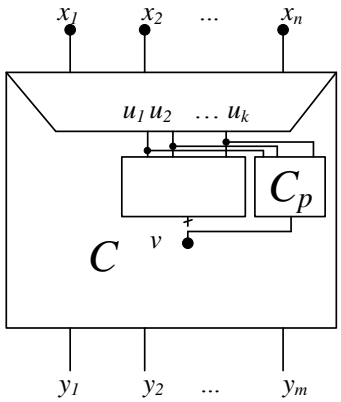


Рис. 6. Маскирование неисправности внутреннего полюса схемой
в условиях доступности предшествующих полюсов

Заключение

Рассматриваются комбинационные логические схемы из вентилей. Для таких схем исследуется возможность маскирования логической неисправности на внутреннем полюсе схемы с помощью внешней вентильной подсхемы в условиях доступности входов схемы и внутреннего неисправного полюса схемы, а также в условиях доступности внутреннего неисправного полюса схемы и предшествующих ему исправных внутренних полюсов. Представлены алгоритмы построения частичных функций внутреннего полюса схемы, зависящих как от внешних, так и от внутренних полюсов схемы, с использованием SAT-решателей. На основе полученных функций с использованием систем ESPRESSO и ABC строится маскирующая схема из вентилей.

ЛИТЕРАТУРА

1. *Petkovska A., Mishchenko A., Novo D., Owaida M.* Progressive Generation of Canonical Irredundant Sums of Products Using a SAT Solver. // Reis A., Drechsler R. Advanced Logic Synthesis. – 2018. – С. 169-188.

2. *Цейтин Г.С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. – 1968. – Т. 8. – С. 234-259.

3. ESPRESSO: Logic Minimization Software [Электронный ресурс] – Режим доступа: http://ramos.elo.utfsm.cl/~lsb/elo211/aplicaciones/aplicaciones/espresso/ESPRESSO Logic Minimization Software.htm – (Дата обращения: 27.04.2020).

4. ABC: A System for Sequential Synthesis and Verification [Электронный ресурс] – Режим доступа: https://people.eecs.berkeley.edu/~alanmi/abc/ – (Дата обращения: 27.04.2020).

ПОСТРОЕНИЕ МИНИМИЗИРОВАННОГО ПРОВЕРЯЮЩЕГО ТЕСТА ДЛЯ СИСТЕМЫ БЕЗЫБЫТОЧНЫХ ДНФ ,ОРИЕНТИРОВАННОЕ НА СОКРАЩЕНИЕ РАССТОЯНИЯ ПО ХЕММИНГУ МЕЖДУ СОСЕДНИМИ ТЕСТОВЫМИ НАБОРАМИ

Сампилов А.А., Андреева В.В.
Томский государственный университет
alesha1999@gmail.com, avv.21@mail.ru

Введение

Тестирование интегральных схем – необходимая процедура для корректной работы устройства. Сложность и скорость работы устройств постоянно растет, а их размер, в свою очередь, уменьшается, что приводит к увеличению требований к тестированию, в частности, к снижению потребляемой при тестировании мощности. Обеспечение этого свойства достигается сокращением длины тестовой последовательности при сохранении полноты покрытия относительно рассматриваемого класса неисправностей и со-

кращением числа перепадов сигналов (расстояния по Хеммингу)между соседними тестовыми наборами последовательности. Сокращение длины тестовой последовательности приводит также к сокращению времени тестирования и позволяет экономить память при хранении этой последовательности. Проблема снижения потребляемой мощности связана с увеличением скорости работы логических схем. В современных интегральных схемах высокой производительности, характеризующихся высокой скоростью функционирования и нано размерами транзисторов, в процессе их работы возникают непредусмотренные емкости, сопротивления, индуктивности. Их не удается рассчитать заранее существующими физическими методами. Они приводят к дополнительным задержкам, снижая тем самым расчетное быстродействие схем. Тесты, обнаруживающие такие задержки, должны поступать на схему с той же скоростью, с какой она функционирует в рабочем режиме. Тестирование задержек выполняется обычно на этапе производства каждой схемы. Именно тестирование задержек связано с большим потреблением мощности по сравнению с работой схемы в режиме функционирования. При таком тестировании вследствие перегрева схемы возможно ее разрушение. Повышение температуры во время тестирования вызвано частыми перепадами напряжения, обусловленными сменами значений сигналов в соседних наборах тестовой последовательности [1]. Проблема снижения потребляемой при тестировании мощности с ростом уровня интеграции схем и увеличением их быстродействия становится все более актуальной [2]. В современных интегральных схемах наряду с тестированием задержек желательно тестировать кратные константные неисправности на полюсах элементов схемы. Тестирование таких неисправностей можно заменить тестированием одиночных неисправностей литер системы безызбыточных ДНФ (БДНФ) в условиях применения к этой системе многоуровнего метода синтеза, основанного на делении ДНФ [3]. Снижение потребляемой мощности при использовании таких последовательностей также актуально. В данной работе рассматриваются одиночные неисправности литер именно таких ДНФ. Учитываются два критерия снижения потребляемой мощности: сокращение числа перепадов сигналов между соседними тестовыми наборами и сокращение длины тестовой последовательности. Разработанный ранее алгоритм, ориентированный на сокращение длины тестовой последовательности, модифицируется на основе дополнительного критерия – сокращения числа перепадов сигналов в строящейся последовательности.

В данной работе задача сокращения тестовой последовательности с одновременным сокращением числа перепадов сигналов рассматривается относительно одиночных *a*,*b*-неисправностей литер системы безызбыточных ДНФ (системы БДНФ). Предлагается алгоритм построения такой тестовой последовательности ДНФ.

1. Постановка задачи

Рассмотрим систему БДНФ, задающую множество функций $F = \{f_1, f_2, ..., f_k\}$, а также два типа неисправности *a*-неисправности и *b*-неисправности. Исчезновение некоторой конъюнкции из БДНФ называется *a*-неисправностью, исчезновение некоторой буквы из некоторой конъюнкции БДНФ называется *b*-неисправностью. Наборы, обнаруживающие такие неисправности, будем называть *a*,*b*-тестовыми наборами, соответственно.

Обозначим символом *A* множество *a*-тестовых наборов, обнаруживающих все *a*-неисправности БДНФ. Обозначим символом *B* множество *b*-тестовых наборов, обнаруживающих все *b*-неисправности БДНФ. В дальнейшем множество *A*(*B*) будем называть *A*(*B*) тестом БДНФ.

Назовем *T проверяющим тестом* системы БДНФ, который получается путём объединения множеств *A*(*B*), построенных для каждой БДНФ системы. Заметим, что при построении проверяющего теста для системы в целом возможно совмещение *a*(*b*)-

тестовых наборов для различных функций системы, а также совмещение *a* и *b*-тестовых наборов, относящихся к различным функциям системы. Причем, для одной и той же функции системы совмещение *a*-тестовых наборов невозможно.

Целью данной работы является, разработка алгоритма сокращения мощности проверяющего теста *T* с сохранением полноты относительно рассматриваемых класса неисправностей, причем, сокращение мощности сопровождается сокращением числа перепадов в последовательности, построенной из элементов этого множества.

2. Построение *A*, *B* тестов

Рассмотрим БДНФ D, состоящую из простых импликант, $D = K_1 \vee K_2 \vee ... \vee K_s$. Конъюнкцию $K(x_i)$, полученную из *K* путем замены знака переменной x_i на инверсный, будем называть дополнением конъюнкции *K* по переменной x_i.

ДНФ $D^*(K_i)$ строится из БДНФ удалением из нее конъюнкции K_i и конъюнкций ортогональных K_i, а также удалением всех переменных присутствующих в K_i из оставшихся конъюнкций.

Построение *a*-тестовых наборов для рассматриваемой литеры сводится к нахождению корня уравнения $D^*(K_i) = 0$. Пусть r_i – корень уравнения $D^*(K_i) = 0$, тогда булев вектор, обращающий в единицу логическое произведение $K_i^a = r_i \wedge K_i$, есть *a*-тест, где конъюнкция K_i^a представляет в общем случае множество *a*-тестовых наборов, обнаруживающих исчезновения конъюнкции K_i из *D*. Вектор (в общем случае троичный), представляющий конъюнкцию K_i^a, добавляется в множество *A* тестовых наборов.

Построение *b*-тестовых наборов для рассматриваемой литеры сводится к нахождению корня уравнения $D(K_i(x_j)) = 0$, где ДНФ $D(K_i(x_j))$ строится из *D* путем вычеркивания из нее конъюнкции ортогональных $K_i(x_j)$, и вычеркиванием всех букв, присутствующие в $K_i(x_j)$, из оставшихся конъюнкций.

Пусть r_i – корень уравнения $D(K_i(x_j)) = 0$, тогда булев вектор, обращающий в единицу логическое произведение $K_i^b = r_i \wedge K_i(x_j)$, есть *b*-тест, где конъюнкция K_i^b представляет множество *b*-тестовых наборов, обнаруживающих исчезновения некоторой переменной x_j конъюнкции K_i из *D*. Вектор (в общем случае троичный), представляющий конъюнкцию K_i^b, добавляется в множество *B* тестовых наборов.

Решение логических уравнений в общем случае связано с реализацией вычислений, объем которых экспоненциально зависит от числа переменных уравнения. Однако, учитывая свойства некоторых уравнений, а также требования к искомым корням вычисления можно сократить. В общем случае, для логического уравнения вида $F(x_1,...,x_n) = \sigma$, $\sigma \in \{0,1\}$ процесс поиска можно свести к обходу некоторой части ROBDD или Free BDD графа [4] до получения первой концевой вершины, помеченной σ. Если такая вершина не найдена, то необходимо вернуться в ближайшую к корню вершину и продолжить процесс обхода графа. Если, в результате обхода концевая вершина со значением σ не найдена, следовательно, корня не существует. В противном случае, конъюнкция, соответствующая простой цепи, соединяющая эту вершину с корнем графа, задает корень уравнения.

Напомним, что в ROBDD порядок разложения фиксирован, в то время как во Free BDD графе порядок разложения может быть выбран в каждой внутренней вершине, что позволяет проводить дополнительный анализ при выборе переменной обхода. В нашем случае, необходимо в каждой вершине выбирать именно те переменные обхода, кото-

рые бы способствовали сокращению ранга конъюнкции, представляющей корень. Для нахождения корней уравнения $D = 0$ воспользуемся методом, предложенным в [5]. Данный метод позволяет находить корни уравнения, представляемые конъюнкцией, как можно меньшего ранга, что дает возможность при формировании тестовой последовательности сокращать число перепадов между соседними векторами последовательности. Предлагается модификация этого метода, ориентированная на усиление этой возможности.

Метод поиска корня логического уравнения $D = 0$, $D = K_1 \vee K_2 \vee ... \vee K_s$, основывается на анализе двух булевых матриц L, R, где i-е строки матриц L_i, R_i представляют конъюнкцию K_i, следующим образом: в булевом векторе L_i единичными значениями отмечены компоненты встречающиеся в K_i без знака инверсии, а в булевом векторе R_i единичными значениями отмечены компоненты встречающиеся в K_i со знаком инверсии.

Например, для ДНФ $D = x_1\bar{x}_2\bar{x}_3 \vee x_2\bar{x}_3 \vee x_1\bar{x}_2x_3$ матрицы L, R будут иметь вид:

$$L = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}, \; R = \begin{matrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{matrix}.$$

В [5] сформулированы условия, позволяющие выяснить существование корня. В частности, если в матрицах L, R нет нулевых строк, то корень уравнения существует, а нахождение корня осуществляется либо по матрице L, либо по матрице R путем построения покрытия строк матрицы столбцами. Такая ситуация порождает ряд следствий, например, если существуют нулевые строки только в одной из матриц, то конъюнкция, представляющая корень уравнения, находится покрытием столбцами строк матрицы без нулевых строк [5].

Построим корень уравнения для примера, представленного выше. Корень можно найти путем получения покрытия либо по матрице L, либо, по матрице R. Построим покрытие для матрицы L. Для данного примера в покрытие войдут столбцы 1, 2 и корень будет иметь вид $r_L = 00-$. Для матрицы R в покрытие войдут столбцы 2, 3 и корень будет иметь вид $r_R = -11$.

Опишем основные шаги метода из [5] с предлагаемой модификацией.

П. 1. В текущей вершине Free BDD-графа проверяется соответствующая ей ДНФ на отсутствие в матрицах L, R нулевых строк. Если нулевые строки есть в обеих матрицах, то переходим к П. 4.

П. 2. Если в матрицах L, R отсутствуют нулевые строки, то строим покрытия r_L и r_R соответственно. Конъюнкции r_L, r_R умножаем на конъюнкцию, порожденную простой цепью, соединяющей рассматриваемую вершину с корнем Free BDD-графа. Полученные в результате конъюнкции представляют корни рассматриваемого уравнения. Полученную пару корней будем называть связанной парой корней.

П. 3. Если в одной из матриц L, R нет нулевых строк, то строим ее покрытие r. Аналогично процедуре, изложенной в П. 2, находим конъюнкцию, порожденную простой цепью, которая соединяет рассматриваемую вершину с корнем Free BDD-графа, и умножаем ее на r. Полученная в результате конъюнкция является корнем уравнения.

П. 4. Выбираем следующую переменную разложения, согласно правилам описанным в [5].

Рассматриваем множество $A(B)$ тестовых наборов, представленное в общем случае троичными векторами, обнаруживающих a,b-неисправности всех БДНФ системы. Формируем проверяющий тест T путем объединения $A(B)$ множеств. Далее применим к

множеству T метод минимизации проверяющего теста на основе дерева разложения с учетом перепадов сигналов.

3. Минимизация проверяющего теста на основе дерева разложения с учетом перепадов сигналов

Минимизация проверяющего теста T выполняется с помощью дерева разложения. Дерево разложения – это структура, которая позволяет более эффективно находить пересечения векторов, определять заведомо ортогональные множества векторов, а также сортирует исходное множество относительно компонент разбиения [6].

Обозначим:

а) P_0 – множество векторов, у которых компонента $x_i = 0$;

б) P_1 – множество векторов, у которых компонента $x_i = 1$;

в) DC – множество векторов, у которых компонента $x_i = '-'$.

Дерево разложения обладает следующими свойствами:

а) внутренние вершины сопоставляются множеству векторов;

б) у внутренних вершин и корня может быть до трёх потомков P_0, P_1, DC;

в) лист содержит единственный вектор или же является множеством DC, для рассматриваемой на данном уровне компоненты;

г) количество перепадов между соседними листьями, содержащими единственный вектор, минимально.

Обозначим символом C результат минимизации множества T.

Идея минимизации проверяющего теста состоит в разложении исходного тестового множества по максимально определенной немонотонной компоненте на множества P_0, P_1, DC.

Сопоставим тестовое множество T корню дерева разложения. Найдем в T максимально определенную немонотонную компоненту x_i. Разложим T на три множества P_0, P_1, DC в соответствии со значением x_i, в каждом конкретном векторе, полученные множества будут являться потомками нашего корня. Повторяя вышеописанные действия с множествами P_0, P_1, до тех пор, пока не будет возможным определить максимально определенную немонотонную компоненту, это будет означать, что все векторы в данном множестве имеют общее пересечение.

После разложения получим дерево, в котором листы, не являющиеся DC, сопоставляются соответствующему единственному вектору. Далее, выполняя обход, поднимаясь по дереву от листьев, пытаемся пересекать полученные векторы с векторами из множеств DC на текущем уровне, и в каждой внутренней вершине объединяем результаты, пришедшие от потомков P_1 и P_0, сокращая количество перепадов. Поднявшись в корень от каждого листа, получим C. Для того чтобы дополнить множество C, применим описанный метод для векторов, из всех векторов DC, которые не попали в C. В результате получим множество C_{CD}, после чего добавим C_{CD} к C , сокращая количество перепадов.

Сформулируем алгоритм сокращения количества перепадов для тестовых множества V и W.

Сокращение количества перепадов между множествами векторов V и W при объединении будем выполнять следующим образом:

1. Выбираются крайние векторы обоих множеств, v_1, v_n из первого и w_1, w_m из второго множества.
2. Подсчитывается количество перепадов между v_1, w_m и между v_n, w_1. Обозначим количество перепадов в первом случае символом t_1, а во втором – t_2.

3. Выбирается минимальное значение из t_1 и t_2. Если $t_1 < t_2$, то в результирующем множестве сначала располагаются векторы из множества *W*, а затем векторы из множества *V*, в противном случае, иначе.

После получения множества *C* из него необходимо удалить, для каждой связанной пары векторов, вектор, который порождает большее количество перепадов. Далее, при доопределении компонент, выполняется сокращение перепадов значением ближайшего вектора, у которого данная компонента определена, если таких нет, то данная компонента определяется любым значением из $\{1,0\}$ [7].

4. Экспериментальные результаты

Были проведены эксперименты предложенного алгоритма на системах БДНФ для бенчмарков. Экспериментальные результаты предложенного алгоритма приведены в табл. 1.

Таблица 1

Экспериментальные результаты

Название контрольного примера	*S*	*N*	*P*	n_I	n_C	*drops*	*peak*	*a.n.d*	%
TBK	13	14	2224	20696	4109	7889	10	1.9	80%
S8	5	8	43	220	71	126	6	1.8	68%
BBSSE	16	13	120	682	91	297	7	3.3	87%
CSE	16	13	193	1402	196	573	7	2.9	86%
DK14	13	9	149	822	153	271	6	1.8	81%
DVRAM	26	16	233	1308	87	240	12	2.8	93%
EX1	34	15	375	1966	207	563	10	2.7	89%
LION	6	6	21	81	19	30	3	1.6	77%
MODULO12	6	7	31	114	22	58	4	2.7	81%
PLANET1	36	15	584	3148	284	932	12	3.2	91%
RIE	41	17	246	1146	144	322	6	2.2	87%
TRAIN4	6	6	19	75	18	38	4	2.2	76%
TRAIN11	8	8	42	161	34	82	5	2.5	79%
SAND	22	19	680	4404	826	1916	10	2.3	81%

S – количество БДНФ;
N – количество переменных в системе;
P – количество конъюнкций в системе;
n_I – мощность сгенерированного тестового множества;
n_C – мощность сокращенного тестового множества;
drops – количество перепадов в сокращенном тестовом множестве;
peak – пиковое значение перепадов между соседними тестами;
a.n.d. – среднее количество перепадов в сокращенном тестовом множестве, рассчитывается по формуле $a.n.d = drops/(n_C - 1)$, где $n_C - 1$ – количество пар между которыми подсчитывается количество перепадов;
% – процент сокращения мощности тестового множества, вычисляемый по формуле $(s-r)/s$, здесь s – мощность сгенерированного тестового множества, r – мощность минимизированного тестового множества.

В табл. 1 демонстрируется результат проведенных испытаний, в частности, длина полученного проверяющего теста и длина сокращенного теста для рассматриваемых контрольных примеров. Сокращение проверяющего теста выполнялось с помощью предложенного подхода, который направлен не только на сокращение длины, но и, по возможности, на сокращение перепадов сигналов в этой тестовой последовательности. В последнем столбце таблицы, показан процент сокращения тестовой последовательности относительно исходной. Итак, предложенный подход позволяет значительно со-

кратить как длину тестовой последовательности, так и количество перепадов относительно несокращенной тестовой последовательности.

Заключение

Разработан и реализован эвристический алгоритм построения минимизированного проверяющего теста для системы безызбыточных ДНФ, ориентированный на сокращение числа перепадов сигналов в тестовой последовательности и сокращение ее длины. Предложенный подход сводится к построению конъюнкций, по возможности минимальных рангов, представляющих тестовые наборы, что дает возможность при формировании тестовой последовательности сокращать число перепадов между соседними векторами последовательности. В дальнейшем планируется продолжить исследования предложенной модификации с целью оценок качества сокращения на различных примерах.

ЛИТЕРАТУРА

1. ScanTest [Электронный ресурс] // URL: https://semiengineering.com/knowledge_centers/test/scan-test-2 (дата обращения: 31.08.2019).

2. *Toshinori H., Atsushi H., Yukari Y., Masayuki A.* A Low Capture Power Test Generation Method Based on Capture Safe Test Vector Manipulation // IEICE T INF SYST - 2017. - P. 2118-2125.

3. *Matrosova A., Loukovnikova E., Ostanin S., Zinchuck A., Nikolaeva E.* Test Generation for Single and Multiple Stuck-at Faults of a Combinational Circuit Designed by Covering Shared ROBDD with CLBs // 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007). doi:10.1109/dft.2007.42

4. *Матросова А.Ю., Жидкова Е.В.* Решение логических уравнений и анализ BDD-графов // 5-я Междунар. конф. “Проектирование дискретных систем”. – Минск, 2004. - С. 1 – 55.

5. *Андреева В.В., Матросова А.Ю.* Построение минимизированного проверяющего теста, обнаруживающий неисправности безызбыточной ДНФ // Вестник ТГУ. Приложение. – 2006. – № 18. – С. 34–39.

6. *Андреева В.В., Сорудейкин К.А.* Сокращение длины проверяющего теста на основе дерева декомпозиции // Известия вузов. Физика. - 2013. - Т. 56, № 9/2. С. 187-190.

7. *Матросова А.Ю., Андреева В.В., Тычинский В.З., Гошин Г.Г.* Использование ROBDD графов для тестирования задержек логических схем // Известия вузов. Физика. 2019. – Т. 62, № 5. – С. 86-94.

ПОЛУЧЕНИЕ ТЕСТОВЫХ ПАР ДЛЯ РОБАСТНО ТЕСТИРУЕМЫХ НЕИСПРАВНОСТЕЙ ЗАДЕРЖЕК ПУТЕЙ С ИСПОЛЬЗОВАНИЕМ SAT-РЕШАТЕЛЕЙ

Тычинский В.З., Андреева В.В.
Томский государственный университет
tvz.041@gmail.com, avv.21@mail.ru

Введение

Изобретение интегральных схем на рубеже 50–60-х гг. 20-го века позволило существенно уменьшить размеры электронно-вычислительных машин. В настоящее время интегральные схемы всё ещё являются основной элементной базой: они используются как в бытовых приборах, так и в узкоспециализированном оборудовании для космических аппаратов. Однако одновременное увеличение скорости работы схем вместе с усложнением техпроцесса приводит к возрастанию числа неисправностей в процессе создания схем, в том числе на заключительных этапах их производства. Кроме того, при усложнении логики и сокращении размеров схем всё чаще начинают проявляться новые типы неисправностей, ранее не влиявшие на их работоспособность. Одним из таких типов неисправностей являются неисправности задержек путей (Path Delay Faults (PDFs)).

Различают робастно тестируемые и не робастно тестируемые неисправности задержек путей. Если неисправность является робастно тестируемой, её обнаружение позволяет однозначно определить путь, на котором она возникла. В случае обнаружения не робастно тестируемых неисправностей однозначно определить путь не представляется возможным. После обнаружения в логической схеме робастно тестируемых

неисправностей, она передаётся разработчикам. В свою очередь, они могут либо понизить быстродействие схемы, тем самым исключив влияние задержек, либо же корректировать саму схему, чтобы сохранить требуемое быстродействие.

Как правило, целесообразно тестировать не все пути, а лишь те, время прохождения сигналов по которым которых близко к максимальному. Для обнаружения задержки пути необходимы два вектора – тестовая пара (v_1, v_2). Векторы пары отличаются инверсным значением переменной, отмечающей начало пути (в то же время они могут различаться в значениях и по другим переменным). В общем случае для каждого пути требуется получить две пары, поскольку задержки противоположных (rising transitions и falling transitions) перепадов значений сигналов могут отличаться.

В то же время очень важно получать тестовые последовательности, ориентированные на потребление как можно меньшей мощности. Потребляемая мощность тем меньше, чем меньше смен значений переменных при переходе от одного вектора тестовой последовательности к другому вектору. Минимизация числа таких смен крайне важна, поскольку потребление мощности во время тестирования (как правило) выше, чем в штатном режиме работы схемы. Не учитывая этот факт, во время тестирования можно разрушить схему, которая могла бы корректно функционировать в рабочем режиме.

В работе предлагается алгоритм построения тестовых пар для обнаружения робастно тестируемых неисправностей задержек путей с использованием SAT-решателя. Обсуждаются возможности разработанного алгоритма на основе анализа результатов экспериментов с бенчмарками ISCAS'89. Анализ включает в себя как общую оценку затрат ресурсов на поиск тестовых пар, так и оценку возможностей алгоритма по снижению количества перепадов в построенной тестовой последовательности, обнаруживающей робастно тестируемые неисправности задержек путей в схемах с памятью.

1. Обнаружение робастно тестируемых неисправностей задержек путей

На практике, как известно, обнаружение неисправностей задержек путей осуществляется с помощью методов сканирования [1]. При этом вектор v_2 получается из вектора v_1 – либо сдвигом v_1, либо как реакция комбинационной составляющей схемы с памятью на вектор v_1. Очевидно, что при подобный подход в ряде случаев не позволяет сформировать тестовые пары для робастно тестируемых неисправностей задержек путей. Поэтому, наряду с традиционными методами тестирования развиваются альтернативные методы тестирования, например, Random Access Scan (RAS) методы, позволяющие существенно сократить время тестирования, потребление мощности и множество тестовых наборов [2,3]. В рамках этих методов могут быть использованы тестовые пары, гарантированно обнаруживающие неисправности задержек путей, в частности, робастно тестируемые неисправности задержек путей. В свою очередь, последовательности тестовых пар, ориентированные как на снижение потребляемой мощности при тестировании, так и на сокращение длины теста в целом, могут быть получены с помощью подмножеств тестовых пар для робастно тестируемых неисправностей задержек путей, построенных для каждого из выбранных путей.

В рассмотренном в данной работе алгоритме для получения тестовых пар используется булева разность рассматриваемого пути [4].

2. Вычисление булевой разности для пути α

Путём в комбинационной схеме называют последовательность элементов, в которой выход предыдущего элемента является входом последующего элемента. Один из входов первого элемента последовательности является входом схемы. Выход последнего элемента последовательности является выходом схемы.

Представим путь α одновыходной комбинационной схемы *C*, входы которой обозначены переменными $x_1,...,x_n$, в виде последовательности символов: $x_i, u_1,...,u_k$. Здесь k – длина пути α, x_i – переменная, отмечающая начало пути (вход схемы *C*), переменные $u_1,...,u_k$ отмечают выходы элементов пути. Переменная u_k отмечает выход схемы *C*. Заметим, что схема *C* может быть комбинационной частью последовательностной схемы.

Пусть переменные u_i, u_{i-1} отмечают выходы соседних элементов пути α. Рассмотрим подсхему C_{u_i} схемы *C*. Выход этой подсхемы отмечается переменной u_i, а входы – переменными $x_1,...,x_n,u_{i-1}$. Здесь переменная u_{i-1} является входной переменной подсхемы C_{u_i} наряду с переменными $x_1,...,x_n$ и одновременно входной переменной элемента с выходом, отмеченным переменной u_i.

Обозначим символом $D_{u_i}/D_{u_{i-1}}$ булеву разность, вычисляемую для функции, реализуемой подсхемой C_{u_i}, по переменной u_{i-1}. Тогда булева разность для пути α принимает следующий вид:

$$D_\alpha = \left(D_{u_r}/D_{u_{r-1}}\right)\wedge\left(D_{u_{r-1}}/D_{u_{r-2}}\right)\wedge\ldots\wedge\left(D_{u_2}/D_{u_1}\right)\wedge\left(D_{u_1}/D_{x_i}\right).$$

3. Получение тестовых наборов для робастно тестируемых неисправностей с использованием КНФ логической схемы

Рассмотрим элемент *u* комбинационной схемы *C* с входами $x_1,...,x_n$. Пусть требуется найти булеву разность D_u/D_{x_i}, при этом возможны следующие ситуации:

1. Тривиальный случай: $D_u/D_{x_i}=1$, если элемент *u* является одновходовым (инвертор) или двухвходовым элементом вида XOR / NXOR (исключающее ИЛИ / эквивалентность):

$$D_u/D_{x_i} = \left(x_k\oplus\left(x_i=0\right)\right)\oplus\left(x_k\oplus\left(x_i=1\right)\right) = x_k\oplus\overline{x}_k = 1,$$

$$D_{u_i}/D_{u_j} = \left(x_k\sim\left(x_i=0\right)\right)\oplus\left(x_k\sim\left(x_i=1\right)\right) = \overline{x}_k\oplus x_k = 1.$$

2. $D_u/D_{x_i}=\left(x_1\wedge\ldots\wedge x_{i-1}\wedge x_{i+1}\wedge\ldots\wedge x_n\right)$, если *u* является элементом вида AND / NAND (И / НЕ И):

$$D_u/D_{x_i} = \left(x_1\wedge\ldots\wedge\left(x_i=0\right)\wedge\ldots\wedge x_n\right)\oplus\left(x_1\wedge\ldots\wedge\left(x_i=1\right)\wedge\ldots\wedge x_n\right) =$$

$$= 0\oplus\left(x_1\wedge\ldots\wedge x_{i-1}\wedge x_{i+1}\wedge\ldots\wedge x_n\right) = \left(x_1\wedge\ldots\wedge x_{i-1}\wedge x_{i+1}\wedge\ldots\wedge x_n\right),$$

$$D_u/D_{x_i} = \left(\overline{x_1\wedge\ldots\wedge\left(x_i=0\right)\wedge\ldots\wedge x_n}\right)\oplus\left(\overline{x_1\wedge\ldots\wedge\left(x_i=1\right)\wedge\ldots\wedge x_n}\right) =$$

$$= 1\oplus\left(\overline{x_1\wedge\ldots\wedge x_{i-1}\wedge x_{i+1}\wedge\ldots\wedge x_n}\right) = \left(x_1\wedge\ldots\wedge x_{i-1}\wedge x_{i+1}\wedge\ldots\wedge x_n\right).$$

3. $D_u/D_{x_i}=\left(\overline{x_1\vee\ldots\vee x_{i-1}\vee x_{i+1}\vee\ldots\vee x_n}\right)$, если *u* является элементом вида OR / NOR (ИЛИ / НЕ ИЛИ):

$$D_u/D_{x_i} = \left(x_1\vee\ldots\vee\left(x_i=0\right)\vee\ldots\vee x_n\right)\oplus\left(x_1\vee\ldots\vee\left(x_i=1\right)\vee\ldots\vee x_n\right) =$$

$$= \left(x_1\vee\ldots\vee x_{i-1}\vee x_{i+1}\vee\ldots\vee x_n\right)\oplus 1 = \left(\overline{x_1\vee\ldots\vee x_{i-1}\vee x_{i+1}\vee\ldots\vee x_n}\right),$$

$$D_u/D_{x_i} = \left(\overline{x_1\vee\ldots\vee\left(x_i=0\right)\vee\ldots\vee x_n}\right)\oplus\left(\overline{x_1\vee\ldots\vee\left(x_i=1\right)\vee\ldots\vee x_n}\right) =$$

$$= \left(\overline{x_1\vee\ldots\vee x_{i-1}\vee x_{i+1}\vee\ldots\vee x_n}\right)\oplus 0 = \left(\overline{x_1\vee\ldots\vee x_{i-1}\vee x_{i+1}\vee\ldots\vee x_n}\right).$$

В случаях 2 и 3, если элемент *u* является двухвходовым (с входами x_i и x_j), выражение D_u / D_{x_i} упрощается до x_j и $\overline{x}_j$ соответственно.

Таким образом, используя выражения для булевых разностей элементов пути, функцию D_α булевой разности пути α можно представить в виде надстройки над исходной схемой *C* (рис. 1, 2).

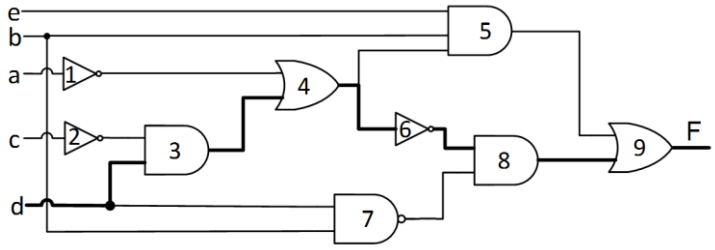


Рис. 1. Комбинационная схема *C*

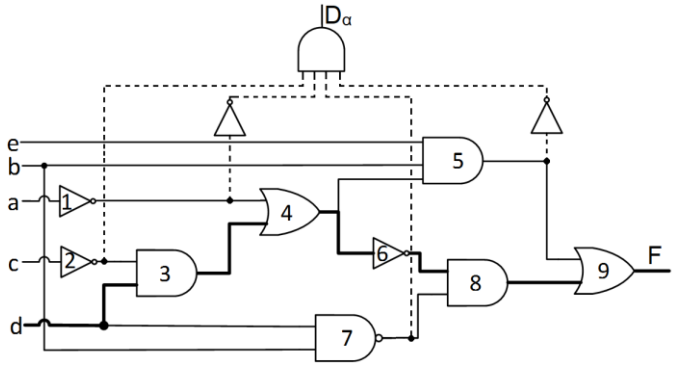


Рис. 2. Реализация булевой разности D_α в виде логической схемы C^*

КНФ элемента у комбинационной схемы C^* строится в виде т.н. «КНФ разрешения» [5] по следующим правилам:

$$y = \overline{x} : (x \vee y) \wedge (\overline{x} \vee \overline{y});$$

$$y = x_1 \oplus x_2 : (\overline{x}_1 \vee x_2 \vee y) \wedge (x_1 \vee \overline{x}_2 \vee y) \wedge (x_1 \vee x_2 \vee \overline{y}) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{y});$$

$$y = \overline{x_1 \oplus x_2} : (x_1 \vee x_2 \vee y) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee y) \wedge (\overline{x}_1 \vee x_2 \vee \overline{y}) \wedge (x_1 \vee \overline{x}_2 \vee \overline{y});$$

$$y = x_1 \wedge x_2 \wedge \ldots \wedge x_n : (x_1 \vee \overline{y}) \wedge (x_2 \vee \overline{y}) \wedge \ldots \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \ldots \vee y);$$

$$y = \overline{x_1 \wedge x_2 \wedge \ldots \wedge x_n} : (x_1 \vee y) \wedge (x_2 \vee y) \wedge \ldots \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \ldots \vee \overline{y});$$

$$y = x_1 \vee x_2 \vee \ldots \vee x_n : (\overline{x}_1 \vee y) \wedge (\overline{x}_2 \vee y) \wedge \ldots \wedge (x_1 \vee x_2 \vee \ldots \vee \overline{y});$$

$$y = \overline{x_1 \vee x_2 \vee \ldots \vee x_n} : (\overline{x}_1 \vee \overline{y}) \wedge (\overline{x}_2 \vee \overline{y}) \wedge \ldots \wedge (x_1 \vee x_2 \vee \ldots \vee y).$$

«КНФ разрешения» булевой разности пути α схемы *C* представляет собой произведение «КНФ разрешений» элементов подсхемы C^* с выходом D_α (рис. 2). Полученная КНФ K_α является функцией не более чем от $n+m$ переменных, где n – число входов схемы *C*, m – число элементов подсхемы C^* с выходом D_α.

Полученная КНФ K_α содержит в себе как область единичных, так и область нулевых наборов функции D_α. Для извлечения конъюнкций ДНФ D_α необходимо положить переменную, соответствующую выходу схемы C^*, в единицу. Затем полученная КНФ $K_\alpha^{D_\alpha=1}$ подаётся на вход SAT-решателя. Полученное решение в виде интервала возможно дополнительно расширить. Для этого необходимо заменить одну из его компонент неопределённым значением и выполнить подстановку значений входных переменных в

КНФ области нулевых наборов K_α ($K_\alpha^{D_\alpha=0}$). Если полученная КНФ является невыполнимой, расширение интервала допустимо.

ДНФ D_α представляет все тестовые наборы v_2 для rising и falling transitions, которые не отделены друг от друга. Необходимо их разделить. ДНФ для rising transition обозначим в виде $D_{rise} = D_\alpha \wedge x_i$, ДНФ для falling transition обозначим через $D_{fall} = D_\alpha \wedge \overline{x}_i$.

Обозначим символом D'_{rise} ДНФ, полученную из D_{rise} удалением переменной x_i, а символом D'_{fall} ДНФ, полученную D_{fall} удалением переменной x_i. В обоих случаях наличие инверсии над переменной x_i не имеет значения.

Обозначим символом D_{rob} ДНФ, представляющую тестовые пары соседних по переменной x_i наборов для робастно тестируемых неисправностей задержек пути α: $D_{rob} = D'_{rise} \wedge D'_{fall}$.

Заметим, что ДНФ D_{rob} не содержит переменной x_i. Конъюнкции ДНФ D_{rob} таковы, что соответствующие им булевы векторы в пространстве переменных $x_1,...,x_{i-1},x_{i+1},...,x_n$ задают тестовые пары в пространстве n переменных. Один из векторов пары получается приписыванием переменной x_i значения 0, а другой – значения 1. Из тестовой пары формируются тройки векторов, обнаруживающие задержки инверсных перепадов сигналов рассматриваемого пути.

4. Используемые технологии

4.1. Язык C++

Язык программирования C++ представляет собой высокоуровневый компилируемый язык программирования общего назначения со статической типизацией, который подходит для создания самых различных приложений. На сегодняшний день C++ является одним из самых популярных и распространенных языков. C++ является мощным языком, унаследовав от Си богатые возможности по работе с памятью, поэтому он нередко находит свое применение в системном программировании, в частности, при создании операционных систем, драйверов, различных утилит, антивирусов и т.д. Однако применение данного языка не ограничивается только системным программированием. C++ можно использовать в программах любого уровня, где важны скорость работы и производительность. В отличие от Си язык C++ позволяет писать приложения в объектно-ориентированном стиле, представляя программу как совокупность взаимодействующих между собой классов и объектов, что упрощает создание сложных приложений.

4.2. SAT-решатель MiniSat

В дискретной математике проблема булевой выполнимости (The Boolean Satisfiability Problem, сокращенно SATISFIABILITY или SAT-задача) – это проблема определения, существует ли такой набор значений переменных, который обращает данную булеву формулу в 1. Если такой набор существует, формула называется выполнимой. В противном случае (когда на любом наборе значений входных переменных формула обращается в 0), формула называется невыполнимой. Например, формула $a \wedge \overline{b}$ является выполнимой, потому что можно найти значения $a = 1$ и $b = 0$, при которых $a \wedge \overline{b} = 1$. Напротив, формула $a \wedge \overline{a}$ – невыполнима.

Основной задачей *SAT-решателей* (SAT-solvers) и является проверка булевой формулы на выполнимость. При этом исходной задачей может быть как простая проверка на выполнимость (т.е., обнаружение одного набора входных переменных, обращающих формулу в 1), так и обнаружение всех подобных наборов.

В данной работе был использован SAT-решатель MiniSat [6]. Данный решатель является проектом с открытым исходным кодом и распространяется под лицензией MIT.

В то же время целью данной работы является скорее сравнение времени работы алгоритма, получающего тестовые последовательности для неисправностей задержек путей с использованием КНФ логической схемы, при различных требованиях к получаемой тестовой последовательности (т.е., к *количеству* решений, получаемых с помощью SAT-решателя). Поскольку сами решатели постоянно совершенствуются, время выполнения предложенного алгоритма может существенно меняться, однако соотношение между его вариациями с разными параметрами будет сохраняться.

5. Результаты бенчмарков

Для анализа алгоритма использовались бенчмарки ISCAS'89. Для каждого выхода каждой схемы было выбрано не менее 10 самых длинных путей. Общая информация о бенчмарках, использованных при тестировании, представлена в табл. 1.

Таблица 1

Информация об использованных бенчмарках

№ п/п	Схема	Число входов	Число выходов	Число вентилей	Выбрано путей		Доля робастно тестируемых путей	Общее количество перепадов	Пиковые значения перепадов
					Всего	Робастно тестируемых			
1	s298	17	20	119	146	95	65%	267	5
2	s344	24	26	160	159	111	70%	338	10
3	s400	24	27	162	258	213	83%	611	7
4	s444	24	27	181	237	142	60%	348	6
5	s641	54	42	379	309	137	44%	382	5
6	s820	23	24	289	232	230	99%	628	9
7	s953	45	52	395	338	313	93%	1101	9
8	s1196	32	32	529	334	162	49%	500	12
9	s1488	14	25	653	312	291	93%	1004	8
10	s1494	14	25	647	336	306	91%	1067	8

В табл. 2 представлены результаты работы алгоритма, выполняющего построение тестовых последовательностей с использованием КНФ логической схемы и SAT-решателя:

Таблица 2

Экспериментальные результаты алгоритма, использующего КНФ логической схемы для построения тестовой последовательности

№ п/п	Общее количество перепадов	Пиковые значения перепадов
1	267	5
2	338	10
3	611	7
4	348	6
5	382	5
6	628	9
7	1101	9
8	500	12
9	1004	8
10	1067	8

Приведённый выше алгоритм извлекает все тестовые наборы, в то время как его особенностью является возможность получения лишь части тестовых наборов. Это скажется на качестве тестовой последовательности и на числе обнаруживаемых неисправностей (поскольку если при построении ДНФ D_α исключить часть конъюнкций, т.е. вероятность, что одна из ДНФ D_{rise}, D_{fall} будет пустой), но в то же время увеличит скорость работы алгоритма. Сравнительные данные алгоритмов, в которых размер D_α ограничивался k конъюнкциями, приведены в табл. 3 (за эталон принимался алгоритм, обнаруживающий все тестовые наборы):

Таблица 3

Сравнительные результаты алгоритма, использующего КНФ логической схемы, при разных требованиях к количеству извлекаемых конъюнкций

№ п/п	Уменьшение времени работы			Покрыто робастно тестируемых путей			Общее количество перепадов			Пиковые значения перепадов		
1	0%	13%	26%	100%	100%	99%	98%	99%	96%	100%	100%	100%
2	33%	55%	76%	98%	95%	92%	104%	88%	81%	125%	88%	88%
3	23%	46%	69%	96%	87%	62%	95%	81%	57%	100%	100%	100%
4	8%	23%	52%	100%	96%	77%	87%	83%	68%	100%	100%	100%
5	2%	5%	33%	100%	100%	100%	109%	109%	109%	83%	83%	83%
6	1%	6%	25%	100%	100%	97%	105%	104%	102%	100%	100%	100%
7	7%	15%	40%	100%	100%	97%	123%	122%	116%	82%	73%	73%
8	32%	51%	68%	98%	96%	95%	99%	97%	95%	100%	100%	100%
9	1%	1%	21%	100%	99%	96%	122%	121%	116%	80%	80%	80%
10	0%	3%	23%	100%	100%	95%	122%	121%	115%	89%	89%	89%
k	10	5	2	10	5	2	10	5	2	10	5	2

Экспериментальные результаты показывают, что даже в тех случаях, когда с помощью SAT-решателя извлекаются только две конъюнкции ДНФ D_α, получаемая на их основе тестовая последовательность обнаруживает свыше 90% робастно тестируемых неисправностей задержек путей (напомним, что, согласно табл. 1, таких путей может быть от 40% до 90% относительно общего числа путей в схеме). При этом время, затраченное на построение тестовой последовательности, снижается пропорционально количеству получаемых решений (в среднем ускорение составляет 11%, 22% и 43% по сравнению с методом, извлекающим все решения).

Заключение

Анализ результатов показывает, что реализация предложенного алгоритма с использованием SAT-решателей имеет значительное преимущество в количестве обнаруженных неисправностей по сравнению с используемыми в настоящее время эвристическими методами. В то же время он может быть применим и к достаточно большим схемам, позволяя за счёт снижения требований к получаемой тестовой последовательности оставаться на требуемом уровне быстродействия.

Следует отметить, что в сфере тестирования неисправностей задержек путей существуют и другие проблемы, в частности, проблема генерации соседних пар тестовых наборов, что в свою очередь позволяет снизить разогрев схем при их тестировании. Использование алгоритмов, сокращающих число перепадов тестовой последовательности [7], в совокупности с возможностью задания количества получаемых тестовых наборов в рассмотренном алгоритме, позволяет получать тестовые последовательности, состоящие из фрагментов с минимальным потреблением мощности; при совмещении фрагментов потребляемая мощность, по возможности, также минимизируется.

ЛИТЕРАТУРА

1. *Agrawal V.D., Cheng R.T., Johnson D.D., Lin T.S.* “Designing Circuits with Partial Scan ”, IEEE Design & Test of Computers. Vol.5, N 2, pp. 8-15, 1988.

2. *Fu Y.Hu., Fan X., Fujiwara H.* “Localized Random Access Scan: Towards Low Area and Routing Overhead,” Proceedings of the 2008 Asia and South Pacific Design Automation Conference, p.p. 565-570, IEEE Computer Society Press, 2008.

3. *Adiga R., Arpit G., Singh V., Satuja K.K., Fujivara H., Singh A.D.* “On Minimization of Test Application Time for RAS” in 2010 23 International Conference on VLSI design, pp. 393-398, IEEE, 2010.

4. *Matrosova A.Yu., Andreeva V.V., Nikolaeva E.A.* Finding Test Pairs for PDFs in Logic Circuits Based on Using Operations on ROBDDs //Russian Physics Journal. 2018. Vol. 61, № 5. P. 994-999.

5. *Черемисинова Л.Д.* Поиск кратчайшей установочной последовательности схемы с памятью на Dтриггерах – Весці Нацыянальнай акадэміі навук Беларусі. Серыя фізіка-матэматычных навук. - 2015. - С. 119-128.

6. MiniSat: a minimalistic, open-source SAT solver, developed to help researchers and developers alike to get started on SAT [Электронный ресурс] — Режим доступа: http://minisat.se/ — (Дата обращения: 23.03.2020).

7. *Тычинский В.З., Андреева В.В.* О свойствах ROBDD-графов, представляющих тестовые пары для робастно тестируемых неисправностей задержек путей – Математическое и программное обеспечение информационных, технических и экономических систем. Материалы VII Международной молодежной научной конференции. Сер. "Физико-математическая" / под общей ред. И.С. Шмырина. – Томск, 2019. – С. 190-196.

IV. СИСТЕМНЫЙ АНАЛИЗ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

АЛГОРИТМЫ ИДЕНТИФИКАЦИИ И ПРОГНОЗИРОВАНИЯ ДЛЯ КОМБИНИРОВАННЫХ МОДЕЛЕЙ

Вилкина И.Ю.[1], Дмитриев Ю.Г.[2], Кошкин Г.М.[2]
[1]*Томский государственный архитектурно-строительный университет,*
[2]*Томский государственный университет*
win32_86@mail.ru, dmit@mail.tsu.ru, kgm@mail.ru

Введение

Построение математической модели зависимости выходных переменных от входных переменных стохастического объекта является важной задачей. Для решения этой проблемы используются как параметрические, так и непараметрические подходы. Представляет интерес улучшение качества восстановления зависимости с привлечением дополнительной информации, которой может располагать исследователь. Имеется обширная литература по оцениванию вероятностных характеристик с использованием дополнительной информации (априорной догадки). В данной работе предлагаются комбинированные алгоритмы идентификации и прогнозирования стохастических объектов с использованием линейной комбинации непараметрических и параметрических оценок регрессии.

Комбинированные статистические адаптивные оценки с априорной догадкой и их свойства рассматривались в ряде работ, например, в [1–6]. Мы рассмотрим случай, когда имеется предположение о виде оцениваемой функции, интерпретируемое как априорная догадка.

В качестве математической модели стохастического объекта часто используют функцию регрессии

$$r(\vec{x}) = E\left(Y \mid \vec{X} = \vec{x}\right) = \int y \cdot p\left(y \mid \vec{x}\right) dy = \frac{\int y \cdot p(\vec{x}, y) dy}{p(\vec{x})},$$

где $\left(\vec{X}, Y\right) = \left(X^{\langle 1\rangle}, ..., X^{\langle p\rangle}, Y\right)$ – $(p+1)$-мерный вектор p входов и выхода, $p(\vec{x}, y)$ – их общая плотность распределения, $p(\vec{x})$ – плотность распределения входов, $p\left(y \mid \vec{x}\right)$ является условной плотностью распределения.

В условиях непараметрической априорной неопределенности обычно применяется оценка Надарая – Ватсона [7,8]

$$\hat{r}(\vec{x}) = \hat{r}\left(\vec{x}; X_1, ..., X_n\right) = \frac{\sum_{i=1}^{n} Y_i \cdot K\left(\frac{\vec{x} - \vec{X}_i}{\vec{h}_n}\right)}{\sum_{i=1}^{n} K\left(\frac{\vec{x} - \vec{X}_i}{\vec{h}_n}\right)}, \qquad (1)$$

где $K\left(\frac{\vec{x} - \vec{X}_i}{\vec{h}_n}\right) = K\left(\frac{x_1 - X_i^{\langle 1\rangle}}{h_n^{\langle 1\rangle}}\right) ... K\left(\frac{x_p - X_i^{\langle p\rangle}}{h_n^{\langle p\rangle}}\right)$ – p-мерное ядро (произведение p одномерных ядер), $\vec{h}_n = \left(h_n^{\langle 1\rangle}, ..., h_n^{\langle p\rangle}\right)$ – p-мерный вектор параметров размытости, $\left(\vec{X}_i, Y_i\right) = \left(X_i^{\langle 1\rangle}, ..., X_i^{\langle p\rangle}, Y_i\right)$, $i = \overline{1, n}$ – независимые наблюдения случайного вектора $\left(\vec{X}, Y\right)$.

Обычно исследователь имеет некоторую информацию о характере зависимости выхода объекта от входов, например, в виде предполагаемой функции $\varphi\left(\vec{x};\vec{\theta}\right)$, где $\vec{\theta}=\left(\theta^{(1)},...,\theta^{(s)}\right)$ – вектор известных параметров. Этот тип информации мы назовем априорной догадкой. Возникает вопрос об использовании такой дополнительной информации при оценивании регрессии (1). Подход с использованием комбинаций различных оценок изучался, например, в [9–13].

1. Комбинированные оценки для статической модели

В случае статических моделей в качестве комбинированной оценки регрессии возьмем [9–11]

$$\hat{R}_\lambda\left(\vec{X}_i\right)=(1-\lambda)\cdot\hat{r}\left(\vec{X}_i\right)+\lambda\varphi\left(\vec{X}_i;\vec{\theta}\right), \tag{2}$$

где λ – весовой коэффициент, определяемый из минимума критерия

$$\sum_{i=1}^{n}\left\{\hat{R}_\lambda(\vec{X}_i)-Y_i\right\}^2\xrightarrow[\lambda]{}\min. \tag{3}$$

Минимизируя (3), получаем оптимальное $\lambda=\lambda_{\text{опт}}$:

$$\lambda_{\text{опт}}=\frac{\sum_{i=1}^{n}\left\{\left(\hat{r}\left(\vec{X}_i\right)-Y_i\right)\left(\hat{r}\left(\vec{X}_i\right)-\varphi\left(\vec{X}_i;\vec{\theta}\right)\right)\right\}}{\sum_{i=1}^{n}\left\{\hat{r}\left(\vec{X}_i\right)-\varphi\left(\vec{X}_i;\vec{\theta}\right)\right\}^2}. \tag{4}$$

Заметим, что весовой коэффициент $\lambda_{\text{опт}}$ в (4) не зависит от выбора *x*. Далее, подставляя (4) в (3), после преобразований, получаем:

$$\sum_{i=1}^{n}\left\{\hat{R}_\lambda(\vec{X}_i)-Y_i\right\}^2=\sum_{i=1}^{n}\left\{\hat{r}\left(\vec{X}_i\right)-Y_i\right\}^2-\frac{\left[\sum_{i=1}^{n}\left\{\left(\hat{r}\left(\vec{X}_i\right)-Y_i\right)\left(\hat{r}\left(\vec{X}_i\right)-\varphi\left(\vec{X}_i;\vec{\theta}\right)\right)\right\}\right]^2}{\sum_{i=1}^{n}\left\{\hat{r}\left(\vec{X}_i\right)-\varphi\left(\vec{X}_i;\vec{\theta}\right)\right\}^2}. \tag{5}$$

Второй член в (5) показывает, насколько СКО комбинированной оценки $\hat{R}_\lambda\left(\vec{X}_i\right)$, с учетом априорной догадки, уменьшается по сравнению с $\hat{r}\left(\vec{X}_i\right)$. Таким образом, с учетом (4), оценка (2) принимает вид

$$\hat{R}_{\lambda_{опт}}\left(\vec{X}_i\right)=(1-\lambda_{\text{опт}})\cdot\hat{r}\left(\vec{X}_i\right)+\lambda_{\text{опт}}\cdot\varphi\left(\vec{X}_i;\vec{\theta}\right). \tag{6}$$

Очевидно, что оценка (6) является оптимальной по критерию (3).

2. Комбинированные оценки для динамической модели

Рассмотрим динамическую модель (см. [14–21])

$$Y_t=f\left(Y_{t-1},...,Y_{t-s}\right)+\xi_t, \tag{7}$$

где Y_t – выход объекта в момент времени *t*, *f* является неизвестной функцией, $s\geq 1$, ξ_t является случайной величиной с нулевым средним и конечной дисперсией.

Предположим, что *f* ограничена и ее вид не меняется в изучаемом интервале времени. В качестве априорной догадки о виде *f* возьмем функцию $\varphi\left(\vec{y};\vec{\theta}\right)$ и рассмотрим следующую комбинированную оценку модели (7):

$$\hat{R}\left(Y_{i-1}\right)=\left(1-\overline{\lambda}_{\text{опт}}\right)\hat{r}\left(\vec{Y}_{i-1}\right)+\overline{\lambda}_{\text{опт}}\,\varphi\left(\vec{Y}_{i-1};\vec{\theta}\right), \tag{8}$$

где

$$\vec{Y}_{i-1}=\left(Y_{i-1},...,Y_{i-s}\right), \tag{9}$$

$$\overline{\lambda}_{\text{опт}} = \frac{\sum_{i=1}^{n}\left\{\left(\hat{r}\left(\vec{Y}_{i-1}\right)-Y_i\right)\left(\hat{r}\left(\vec{Y}_{i-1}\right)-\varphi\left(\vec{Y}_{i-1};\vec{\theta}\right)\right)\right\}}{\sum_{i=1}^{n}\left\{\hat{r}\left(\vec{Y}_{i-1}\right)-\varphi\left(\vec{Y}_{i-1};\vec{\theta}\right)\right\}^2}. \quad (10)$$

Комбинированная оценка (8) позволяет делать прогнозы на один шаг с учетом априорной догадки.

3. Применение алгоритмов идентификации и прогнозирования на поступлениях НДФЛ в бюджет города Томска

Работоспособность рассматриваемых алгоритмов проверялась на данных ежемесячных поступлений налога на доходы физических лиц (НДФЛ) в бюджет муниципального образования «Город Томск» (далее бюджет города Томска).

Для построения модели прогноза поступлений в бюджет города Томска НДФЛ были взяты данные ежемесячных поступлений налогов с 2004 г. (выборка объема 141).

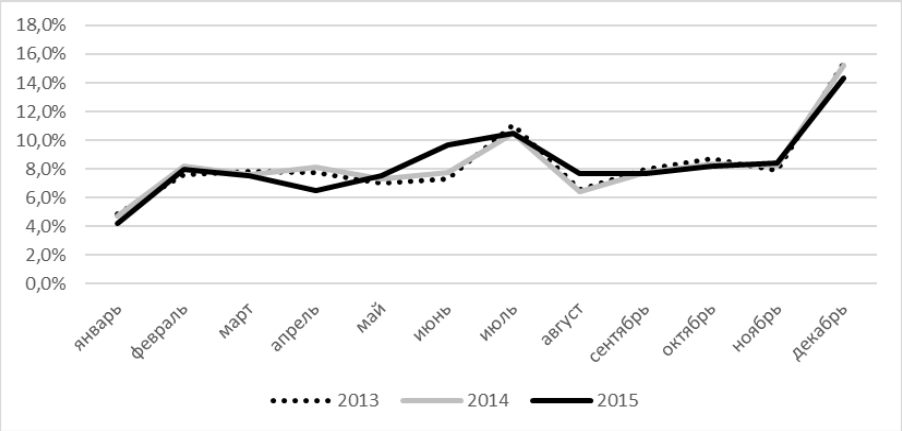


Рис. 1. Графики месячных поступлений НДФЛ в процентах к годовому поступлению за 3 года

Ввиду того, что НДФЛ, согласно Бюджетному кодексу РФ в бюджеты муниципальных образований поступают не в полном объеме, а согласно нормативу, который определяет Бюджетный кодекс и субъект РФ, в рассматриваемый период норматив менялся от 40% в 2004 г. до 25% в 2016 г. для бюджета города Томска. Таким образом, рассматриваемая выборка наблюдений анализировалась не в разрезе фактических поступлений в городской бюджет, а в разрезе собираемости налога с территории по нормативу 100%, при этом наблюдалась сезонность по поступлениям данного налога. В частности, за период с 2004 г. поступления в июле составляют 10–12% годовых, а поступления в декабре 14–15% годовых. Данный факт объясняется уходом сотрудников в ежегодные оплачиваемые отпуска на летний период (в основном это профессорско-преподавательский состав учебных заведений города Томска), а также выплатами 13-й заработной платы и годовых премий в конце года.

Для решения задачи идентификации и прогнозирования использовалась оценка (8) с $s = 12$, согласно (9) и (10). Динамическая модель (7) в данном случае задается формулой $Y_t = f\left(Y_{t-1},...,Y_{t-12}\right)+\xi_t$, где $t = \overline{13,n}$, Y_t – поступление НДФЛ в момент времени *t*. В качестве априорной догадки возьмем следующую параметрическую функцию: $\varphi\left(Y_{t-12};\theta^{(1)},\theta^{(2)}\right)=\theta^{(1)}+\theta^{(2)}Y_{t-12}$, где $\theta^{(1)}=0$, $\theta^{(2)}=2$. Тогда оценка (1) принимает вид

$$\hat{Y}_t = \hat{r}\left(Y_{t-1}, Y_{t-2}, Y_{t-3}\right) = \frac{\sum\limits_{i\geq 4, i\neq t}^{n} Y_t K\left(\frac{Y_{t-1}-Y_{i-1}}{h_n^{Y_{t-1}}}\right) K\left(\frac{Y_{t-2}-Y_{i-2}}{h_n^{Y_{t-2}}}\right) K\left(\frac{Y_{t-3}-Y_{i-3}}{h_n^{Y_{t-3}}}\right)}{\sum\limits_{i\geq 4, i\neq t}^{n} K\left(\frac{Y_{t-1}-Y_{i-1}}{h_n^{Y_{t-1}}}\right) K\left(\frac{Y_{t-2}-Y_{i-2}}{h_n^{Y_{t-2}}}\right) K\left(\frac{Y_{t-3}-Y_{i-3}}{h_n^{Y_{t-3}}}\right)},$$

а комбинированная оценка определяется формулой

$$\overline{Y}_t = \hat{R}_{\lambda_{\text{опт}}}\left(Y_{t-1},...,Y_{t-12}\right) = \left(1-\overline{\lambda}_{\text{опт}}\right)\hat{r}\left(Y_{t-1}, Y_{t-2}, Y_{t-3}\right) + \overline{\lambda}_{\text{опт}}\, 2Y_{t-12}\,.$$

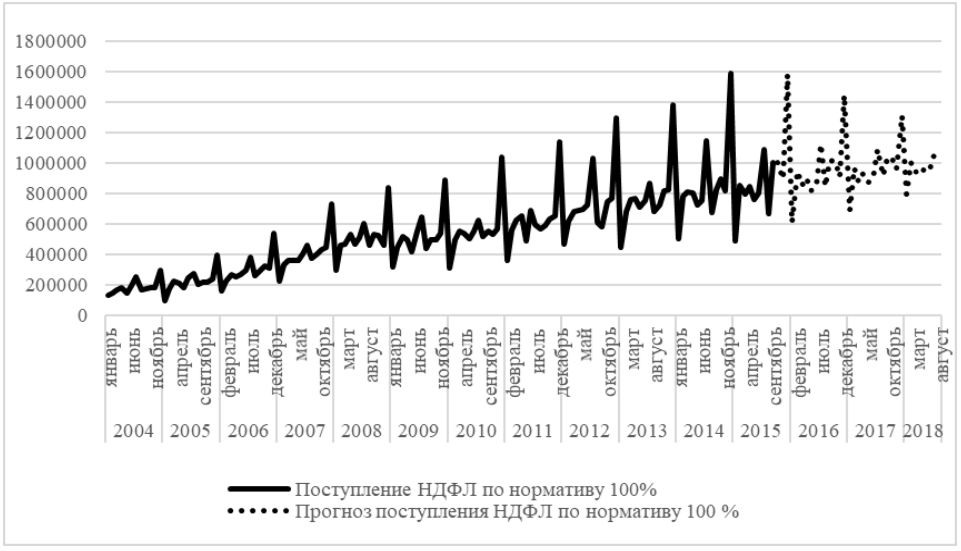


Рис. 2. Временной ряд НДФЛ с 2004 года по октябрь 2015 года и прогноз НДФЛ до 2018 г.

Данная комбинированная оценка была применена для прогнозирования поступления НДФЛ за октябрь–декабрь 2015 г. и 2016 г. и плановый период 2017–2018 гг. в рамках утверждения Думой Города Томска бюджета на 2016 г. и плановый период 2017–2018 гг.. Как правило, наибольший интерес представляет прогнозирование очередного года. На плановый период прогнозируют минимальные поступления для того, чтобы бюджет не принял обязательств сверх возможных поступлений. Результаты прогнозирования представлены на рисунке 2 в виде пунктирной линии.

Качество идентификации и прогнозирования характеризовалось средними относительными ошибками $\delta_{\text{real}}\left(\hat{r}\right)$ и $\eta_{\text{real}}\left(\hat{r}\right)$, соответственно:

$$\delta_{\text{real}}(\hat{r}) = \frac{1}{129}\sum_{i=13}^{141}\frac{\left|Y_i - \hat{r}\left(Y_{t-1},...,Y_{t-12}\right)\right|}{Y_i}\cdot 100\%\,,$$

$$\eta_{\text{real}}(\hat{r}) = \frac{1}{15}\sum_{i=13}^{141}\frac{\left|Y_i - \hat{r}\left(Y_{t-1},...,Y_{t-12}\right)\right|}{Y_i}\cdot 100\%\,.$$

При неизвестном параметре θ он оценивается по методу наименьших квадратов. В этом случае оценку (6) будем обозначать следующим образом:

$$\tilde{R}_{\hat{\bar{\lambda}}_{\text{опт}}}\left(\vec{Y}_{i-1}\right) = \left(1-\hat{\bar{\lambda}}_{\text{опт}}\right)\cdot\hat{r}\left(\vec{Y}_{i-1}\right) + \hat{\bar{\lambda}}_{\text{опт}}\cdot\varphi\left(\vec{Y}_{i-1};\hat{\tilde{\theta}}\right).$$

Таблица 1

Средние относительные ошибки идентификации и прогнозирования

	$\delta_{\text{real}}(\hat{r})$	$\delta_{\text{real}}\left(\hat{R}\right)$	$\delta_{\text{real}}\left(\tilde{R}\right)$
Идентификация	1,98%	1,52%	1,43%

	$\eta_{\text{real}}(\hat{r})$	$\eta_{\text{real}}(\hat{R})$	$\eta_{\text{real}}(\tilde{R})$
Прогнозирование	2,5%	2,1%	1,9%

Заключение

Результаты расчетов, приведенных в табл. 1, показывают преимущество (в точности оценивания) адаптивных комбинированных оценок по сравнению с оценками непараметрической регрессии. Целесообразность применения предлагаемого подхода на практике иллюстрируется при анализе данных ежемесячных поступлений НДФЛ в бюджет муниципального образования «Город Томск».

ЛИТЕРАТУРА

1. *Dmitriev Yu.G., Tarasenko P.F.* The use of a priori information in the statistical processing of experimental data // Russian Physics Journal. 1992. Vol. 35. P. 888–893.

2. *Dmitriev Yu.G., Tarima S.S.* Statistical estimation with possibly incorrect model assumptions // The Bulletin of Tomsk State University: control, computing, informatics. 2009. Vol. 3. No. 8. P. 87–99.

3. *Dmitriev Yu., Tarassenko P., Ustinov Yu.* On estimation of linear functional by utilizing a prior guess // A. Dudin et al. (Eds.): ITMM 2014. 2014. Vol. 487. P. 82–90.

4. *Dmitriev Yu., Tarassenko P.* On adaptive estimation using a prior guess // Proceedings The International Workshop, Applied Methods of Statistical Analysis. Nonparametric Approach. Novosibirsk, Russia. Novosibirsk, 14-15 September, 2015: NSTU publisher. 2015. P. 49–55.

5. *Tarima S.* Statistical estimation in the presence of possibly incorrect model assumptions // Journal of Statistical Theory and Practic. – 2017. No. 1(3**). P. 449–467.

6. *Dmitriev Yu,G., Koshkin G.M., Lukov V.Yu.* Combined identification algorithms // Applied Methods of Statistical Analysis. Nonparametric Methods in Cybernetics and System Analysis - AMSA'2017, Krasnoyarsk, Russia, 18-22 September, 2017: Proceedings of the International Workshop. – Novosibirsk: NSTU publisher, 2017. - P. 19-27.

7. *Кошкин Г.М.* Об одном подходе к исследованию функционалов от условных распределений при статистической неопределенности // Автоматика и телемеханика. 1978. № 8. С. 53-65.

8. *Кошкин Г.М.* Асимптотические свойства функций от статистик и их применения к непараметрическому оцениванию // Автоматика и телемеханика. 1990. № 3. С. 82-97.

7. *Dmitriev Yu.G., Koshkin G.M.* On the use of a priori information in nonparametric regression estimation // IFAC Proceedings Series. 1987. Vol. 2. P. 223–228.

8. *Dmitriev Yu.G., Koshkin G.M.* Using additional information in nonparametric estimation of density functionals // Automat. and Remote Control. 1987. Vol. 48. No. 10. P. 1307–1316.

9. *Скрипин С.В.* Свойства комбинированной оценки регрессии при конечных объемах выборок // Известия Томского политехнического университета. 2008. Т. 313. № 5. С. 10–14.

10. *Dmitriev Yu.G., Koshkin G.M., Lukov V.Yu.* Combined identification algorithms // Applied Methods of Statistical Analysis. Nonparametric Methods in Cybernetics and System Analysis - AMSA'2017, Krasnoyarsk, Russia, 18-22 Sept., 2017: Proceedings of the International Workshop. – Novosibirsk: NSTU publisher. 2017. P. 19-27.

11. *Dmitriev Yu.G., Koshkin G.M., Lukov V.Yu.* Combined identification and prediction algorithms \\ Advances in Computer Science Research (ACSR), volume 72, IV International Research Conference "Information Technologies in Science, Management, Social Sphere and Medicine" (ITSMSSM 2017), Atlantis Press, 2017, pp. 244-247.

12. *Dmitriev Yu.G., Koshkin G.M.* On distribution functionals estimation with auxiliary information // Applied Methods of Statistical Analysis. Nonparametric Methods in Cybernetics and System Analysis - AMSA'2017, Krasnoyarsk, Russia, 18-22 Sept., 2017: Proceedings of the International Workshop. – Novosibirsk: NSTU publisher, 2017. P. 9-18.

13. *Dmitriev Yu.G., Koshkin G.M.* Nonparametric estimators of probability characteristics using unbiased prior conditions // Statistical Papers. 2018. Vol. 59. No. 4. P. 1559-1575.

14. *Vasiliev V.A., Koshkin G.M.* Nonparametric identification of autoregressions // Theory Probab. 1998. Vol. 43. No. 3. P. 507–517.

15. *Kitayeva A.V., Koshkin G.M.* Nonparametric identification of the production functions // Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2011, WCE 2011, (6-8 July, 2011, Imperial College London, London,U.K.). – Hong Kong: Newswood Limited. 2011. Vol.1. P. 276–280.

16. *Kitayeva A.V., Koshkin G.M.* Nonparametric semirecursive identification in a wide sense of strong mixing processes // Problems Inform. Trans. 2010. Vol. 46. No.1. P. 22-37.

17. *Китаева А.В., Кошкин Г.М.* Полурекуррентная непараметрическая идентификация в широком смысле нелинейной гетероскедастической авторегрессии // Автоматика и телемеханика. 2010. № 2. С. 92-111.

18. *Kitayeva A.V., Koshkin G.M.* Nonparametric identification of static and dynamic production functions // IAENG International Journal of Applied Mathematics. 2011. Vol. 41. Issue 3. P. 228–234.

19. *Кошкин Г.М., Глухова И.Ю.* Непараметрическая идентификация нелинейных ARX-процессов // Вестник ТГУ. Управление, вычислительная техника и информатика. 2012. № 3(20). С. 55-61.

20. *Кошкин Г.М., Глухова И.Ю.* Сходимость в среднеквадратическом непараметрического алгоритма идентификации ARX-процесса // Вестник ТГУ. Управление, вычислительная техника и информатика. 2012. № 4(21). С. 71-78.

21. *Koshkin G.M., Lukov V.Yu., Piven I.G.* Nonparametric algorithms of identification and prediction in the ARX-models // Proceedings. The Second International Symposium on Stochastic Models, in Reliability Engineering, Life Science, and Operations Management / Ilia Frenkel and Anatoly Lisnianski (Eds.). SMRLO 2016, February 15-18, 2016, Beer Sheva, Israel. Conference Publishing Services The Institute of Electrical and Electronics Engineers, Inc. 2016. P. 620-623.

ДЕТЕРМИНАЦИОННЫЙ АНАЛИЗ ОПРОСОВ ПО ТЕСТАМ РЕДДИНА

Дмитриев Ю.Г., Ерёмина Н.Л., Тарасенко В.Ф.
Томский государственный университет
dmit70@mail.ru, 26051971@mail.ru, vtara54@ mail.ru

Введение

При переходе к цифровой экономике прослеживается тенденция применения математических методов в управленческом процессе. В данной работе описано исследование результатов опросов по тестам Реддина, отражающим стиль менеджмента респондентов. Изложена методика обработки данных, проанализированы найденные закономерности.

1. Практическая постановка задачи

Основным из постулатов 3D-теории организационной и управленческой эффективности [1] является утверждение о том, что на успех в бизнесе влияют три равнозначных фактора: конкретная ситуация, личность менеджера и эффективность его стиля руководства. В свете такого подхода важное значение приобретает изучение личностных установок и сложившегося управленческого стиля конкретного менеджера и оценка их релевантности внешним и внутренним условиям, в которых осуществляется управление. С этой целью д-ром Реддиным (Reddin W.J.) [1] были разработаны тесты, представляющие собой утверждения, с которыми респондент может соглашаться или не соглашаться. Ответы респондента оцениваются в баллах, отражающих степень совпадения точки зрения респондента с позицией современных концепций менеджмента. Тесты Реддина применяются в системе профессионального образования скандинавских и других стран [2, 3].

В исследовании использованы данные, полученные в результате тестирования более 4000 респондентов в течение 15 лет. Тестирование проводилось в курсе занятий по менеджменту в рамках учебных дисциплин программ высшего образования будущих менеджеров, а также курсов повышения квалификации для действующих управленцев. Респондентам предлагалось пять последовательных тестов по темам: 1) «Коммуникации в организации»; 2) «Информированность работников»; 3) «Человеческие отношения»; 4) «Изменения, реформы, преобразования»; 5) «Индивидуальное аттестационное собеседование». Каждый из которых включал в себя по 80 суждений. Непосредственной задачей тестирования была индивидуализация образовательной траектории респондента. В опросе участвовали сотрудники Администрация Томской области, студенты ТГУ, ТПУ, ТУСУР, СурГУ, КемГУ и др., что дало возможность собрать значительный массив эмпирических данных для исследования.

Были рассмотрены следующие признаки: пол, возраст респондента; для действующих управленцев – место работы; для обучающихся – вуз, факультет (косвенно свидетельствующий о направлении обучения) и отношение к проблеме. Исследовалась структура связей между этими признаками и личностными предпочтениями.

Все исследуемые признаки измеряются в номинальных шкалах, что является типичным для социально-экономической информации. Это обстоятельство существенно ограничивает применение методов, работающих с качественными данными. Другая особенность состоит в том, что связи прослеживаются не между признаками в целом, а между их отдельными значениями. Поэтому в качестве метода исследования был вы-

бран детерминационный анализ, ориентированный на работу с данными, измеряемыми в номинальных шкалах, связанными между собой на уровне значений признаков [4].

2. Математическая постановка задачи

Рассмотрим совокупность результатов тестирования, сопоставив каждому признаку (пол, возраст, отношение к проблеме и т.д.) переменную x_i, принимающую значения из множества X_i. Тогда массив эмпирических данных представляет собой веер отображений вида $E \to X_i$, $i = \overline{1,n}$, где E –множество тестов, X_i – множество значений переменной x_i, n – количество компонент веера.

Детерминация – логическая импликация $a \to b$, $a \in X_i$, $b \in X_j$, представляющая собой высказывание вида «если a, то b». Иначе говоря, детерминация определяет зависимость между истинностью высказывания «признак x_i принимает значение a» (наступлением события «признак x_j принимает значение a») и истинностью высказывания «признак x_j принимает значение b» (наступлением события «признак x_j принимает значение b»). При этом a носит название объясняющего свойства данной детерминации, а b – ее объясняемого свойства.

Детерминация обладает следующими характеристиками:

– интенсивность, понимаемая как условная частота b при условии a и отражающая истинность утверждения «если a, то b»: $I(a \to b) = P(b \mid a) = \dfrac{|E(ab)|}{|E(a)|}$,

– емкость, понимаемая как условная частота a при условии b и отражающая полноту рассматриваемого утверждения: $C(a \to b) = P(a \mid b) = \dfrac{|E(ab)|}{|E(b)|}$.

Анализ интенсивности и емкости и их приращений при изменении состава свойств позволяет делать выводы о характере связей между значениями свойств [4].

Детерминация может быть рассмотрена не на всем множестве наблюдений, а на некотором его подмножестве, определяемом некоторым общим для всех наблюдений значением k признака X_k. В этом случае говорят, что детерминация $ka \to kb = k(a \to b)$ существует в контексте k.

В ходе исследования были рассмотрены две задачи.

Задача 1 состоит в получении объяснений, т.е. установлении зависимости между объясняемым свойством b и объясняющей переменной x в контексте k. Для этого необходимо найти все решения основного уравнения детерминационного анализа

$$\begin{cases} I(k(x \to b)) \geq p, \\ C(k(x \to b)) \geq q, \end{cases}$$

где p – минимально допустимая точность объяснения, q – минимально допустимая полнота объяснения, заданные исследователем.

Задача 2 заключается в установлении существенности объясняющих свойств переменной.

Пусть объясняющее свойство a, являющееся решением основного уравнения детерминационного анализа $k(a \to b)$, может быть представлено в виде разложения $a = a_1 a_2$, $a_1 \in X_{a_1}$, $a_2 \in X_{a_2}$, т.е. значение a соответствует тому, что некоторый признак X_{a_1} принимает значение a_1 и одновременно с этим другой признак X_{a_2} принимает значение a_2.

Под существенностью объясняющего свойства a_1 понимается изменение интенсивности детерминации $a_2 \to b$ в контексте k при учете влияния свойства a_1. Суще-

ственность может быть вычислена по формуле $S\left(k\left(a_1a_2 \to b\right)\right) = I\left(k\left(a_1a_2 \to b\right)\right) - I\left(k\left(a_2 \to b\right)\right)$, где $k\left(a \to b\right)$ – решение основного уравнения детерминационного анализа, $a = a_1a_2$ – разложение рассматриваемой переменной *a*.

3. Исследование эмпирических данных методом детерминационного анализа

Метод детерминационного анализа позволяет исследовать связи сразу между всеми значениями всех признаков. Очевидно, что для *n*-мерного веера отображений количество рассматриваемых детерминаций можно вычислить по формуле $|D| = \prod_{i=1}^{n} |X_i|$, где *D* –множество всех возможных детерминаций, X_i – множество значений переменной x_i, *n* – количество компонент веера. Столь большие объёмы обрабатываемых данных неизбежно требуют автоматизации описанного процесса.

В качестве инструментария автоматизированной обработки предлагается использование возможностей электронных таблиц GoogleTabs с применением алгоритмов, реализованных на GoogleAppsScript.

Выбор инструментария обусловлен, с одной стороны, простотой использования его конечными пользователями, которыми являются специалисты сфер менеджмента, управления персоналом, социологии и пр. С другой стороны, данные средства позволяют использовать современные облачные технологии хранения информации, что существенно облегчает вопрос не только непосредственного хранения данных, но и обмена информацией в процессе обработке данных, позволяя реализовать работу в удаленном режиме коллективов исследователей, включающих в себя как конечных пользователей, так и специалистов по обработке данных.

Метод детерминационного анализа был реализован в виде скрипта гугл-таблицы, с помощью которого были исследованы данные опросов по тестам Реддина. Были рассмотрены детерминации вида $a_1a_2 \to b$, где a_1 и a_2 – значения признаков x_1 и x_2, отражающих пол, возраст, место работы/учебы, факультет/направление деятельности респондента, а *b* –реакция респондента (согласие либо несогласие) на один из вопросов теста. Для каждой детерминации были получены значения ее основных характеристик – емкости и интенсивности, а также проанализирована существенность каждого из объясняющих свойств a_1 и a_2.

Выявленные закономерности отражают влияние исследуемых признаков на стиль менеджмента респондента, и могут послужить основой для рекомендаций, способствующих индивидуализации обучения по дисциплине «Менеджмент» в рамках основных образовательных программ вуза, а также программ повышения квалификации. Отметим, что подобные рекомендации могут быть применены в педагогическом дизайне как традиционного, так и дистанционного курса менеджмента, в том числе с применением методов адаптивного обучения.

4. Выявление закономерностей на основе детерминации

В качестве примера, иллюстрирующего возможности метода, покажем, какие выводы можно получить из анализа детерминаций, связанных с реакцией респондента всего лишь на одно из четырехсот утверждений тестов Реддина.

Рассмотрим вопрос 4 из теста №1 «Коммуникации в организации». Он сформулирован в следующем виде: «Company policy need not be communicated below the management level» – «Политику предприятия следует сообщать только его руководителям».

Ответ на этот вопрос (согласие или несогласие с утверждением) характеризует не только отношение респондента к прозрачности корпоративного управления, но и косвенно свидетельствует о склонности респондента к авторитарному стилю управления.

Было изучено влияние пола и места работы \ направления подготовки (для действующих и будущих управленцев) на характер ответа на вопрос. Все объясняющие свойства были рассмотрены как по отдельности, так и в комбинации попарно друг с другом. Для каждой из детерминаций были рассчитаны интенсивность и емкость, а для детерминаций, отражающих совместное влияние двух объясняющих свойств, была вычислена существенность каждого из свойств.

В результате анализа данных были получены следующие выводы.

1. На основании значений интенсивности детерминаций установлено, что мужчины проявили большую, нежели женщины, склонность к закрытости и авторитарному стилю управления, что подтверждается следующими результатами.

Фрагмент 1 таблиц детерминаций для b = «получен отрицательный ответ на вопрос 4» (табл. 1).

Таблица 1

Таблица детерминаций для b = «получен отрицательный ответ на вопрос 4» (фрагмент 1)

X_1	I
Пол	**Интенсивность**
1	0,3017
2	0,6983

Значение 1 соответствует мужскому полу, 2 – женскому полу.

2. Дальнейший анализ проводился в контексте места работы / направления учебы, раздельно для студентов и действующих управленцев в государственной сфере.

Были изучены результаты тестов сотрудников Администрации Томской области (значение 1 признака «Вуз /факультет/ место работы».

Фрагмент 2 таблиц детерминаций для b = «получен отрицательный ответ на вопрос 4» (табл. 2).

Таблица 2

Таблица детерминаций для b = «получен отрицательный ответ на вопрос 4» (фрагмент 2)

X_1	X_2	I	C	S_1	S_2
Пол	**Вуз/ факультет/ место работы**	**Интенсивность**	**Ёмкость**	**Существенность значения X_1**	**Существенность значения X_2**
1	1	0,2	0,0009	– 0,1017	0,1986
2	1	0,8	0,0016	0,1017	0,7986

Среди сотрудников областной администрации больше женщин, чем мужчин, и это объясняет значения интенсивности. Однако анализ существенности обоих признаков показывает, что в данном случае место работы оказывает большее влияние на наличие зависимости, чем пол сотрудника. Можно говорить о том, что жизненный и профессиональный опыт действующих управленцев не дает им согласиться с весьма категоричным утверждением, а корпоративный стиль управления, принятый в данном исполнительном органе, поддерживает открытость и публичность управленческих процессов.

3. Среди студентов вузов г. Томска, значительную часть которых составляют студенты ведущих вузов России, выраженных связей между значениями признаков не обнаружено.

4. Однако связи между направлением подготовки и характером ответа на вопрос ярко выражены у студентов других вузов. Пол в этом случае также оказывает меньшее

влияние на результат, чем внешняя образовательная среда, т.е. место учебы и направление подготовки студента.

Фрагмент 3 таблиц детерминаций для b = «получен отрицательный ответ на вопрос 4» (табл. 3).

Таблица 3

Таблица детерминаций для b = «получен отрицательный ответ на вопрос 4» (фрагмент 3)

X_1	X_2	I	C	S_1	S_2
Пол	**Вуз/ факультет**	**Интенсивность**	**Ёмкость**	**Существенность значения X_1**	**Существенность значения X_2**
1	9	0,3731	0,0884	0,0714	0,3015
1	11	0,0833	0,0018	– 0,2184	0,0767
2	9	0,6269	0,0642	– 0,0714	0,5554
2	11	0,9166	0,0086	0,2184	0,9100
-	9	0,0715	1	0	– 0,9285
-	11	0,0066	1	0	– 0,9934

Прочерк в графе «пол» соответствует детерминациям, где в качестве объясняющего свойства учитывалось только направление обучения.

Заключение

В настоящей работе рассмотрено применение детерминационного анализа результатов опросов по тестам Редина респондентов, изучающих управленческие науки.

Другие возможные подходы к обработке и анализу подобных данных изложены в [5–10].

ЛИТЕРАТУРА

1. *Reddin W.J.* Tests for the ounput oriented manager/ - UK: Kogan-Page. 1990 – 301p.

2. Reddin Trainingen [Электронный ресурс]: сайт «Learning programma verhoogt de Reddin Methode de management effectiviteit». – Режим доступа: http://www.reddin.nl (дата обращения 30.08.2020).

3. Reddin International [Электронный ресурс]: сайт «Management Education and Management Training». – Режим доступа: http://www. wjreddin.co.uk (дата обращения 30.08.2020)

4. *Чесноков С.В.* Детерминационный анализ социально-экономических данных/М.: Наука, 1982. – 168 с.

5. *Тарасенко В.Ф.* Моделирование систем менеджмента: моногр. / В.Ф. Тарасенко. – Томск: Изд-во Томск. гос. ун-та систем управления и радиоэлектроники, 2018. – 172 с.

6. *Берестнева О.Г.* Конструирование диагностических решений в слабоструктурированных проблемных областях / О.Г.Берестнева, Е.А.Муратова., А.М.Уразаев, Н.Л.Ерёмина Вестник Томского государственного педагогического университета,. № 6 (43). – Томск: 2004 - С. 81-84.

7. *Еремина Н.Л.* Подходы к оценке и управлению конкурентоспособностью студентов университетов / Еремина Н.Л., Краковецкая И.В., Лопухин Я.Н. Обозрение прикладной и промышленной математики. 2019. Т. 26. № 2 – Томск: 2019 -. С. 160-161.

8. *Yury G. Dmitriev, Peter F. Tarassenko, Yuri K. Ustinov.* Determinacy Analysis of Weights as Mathematical Basis of the Future Sociology //ACSR. 2017. Vol. 72. P. 238-243

9. *Мартынова С.Э., Дмитриев Ю.Г., Устинов Ю.К.* Интерпретация количественной оценки удовлетворённости граждан муниципальными публичными услугами на основе детерминационного анализа //Инновационная наука. 2016. Т. 6, № 3. С. 252-257.

10. *Dmitriev Yu.G., Martynova S.E., Tarasenko P.F., Ustinov Iu.K.* Application of Determinacy Analysis to the Study of Citizen Satisfaction with the Service-Based Public Management //Mediterranean Journal of Social Sciences. 2015. Vol. 6, № 6 S2. P. 444-452.

ПРИМЕНЕНИЕ СТАТИСТИЧЕСКИХ ОЦЕНОК В УПРАВЛЕНИИ ПРОЕКТАМИ ПО РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Змеев Д.О., Дмитриев Ю.Г.
Томский государственный университет
denis.zmeev@accounts.tsu.ru, dmit70@mail.ru

Введение

Управление проектами по разработке программного обеспечения является достаточно противоречивой и сложной сферой IT деятельности. С одной стороны, управление проектами считается частью области знаний программной инженерии и, следовательно, к управлению проектами должны быть применимы стандартные инженерные подходы. Однако, несмотря на многие сходства с инженерными дисциплинами (формальное проектирование, разработка эксплуатируемых продуктов, сложная технологическая природа разработанных систем), программная инженерия в целом и управление проектами в частности не сильно поддаются стандартным приёмам, применимых в других инженерных отраслях. Причины этого подробно разобраны в [1], одна из них – специфика разработки программного обеспечения. Программные продукты несмотря на то, что их можно классифицировать, сравнивать и даже формировать классы и типы (ERP системы, сайты-визитки, форумы, социальные сети и др.), по своей природе являются уникальными изделиями. Основные инженерные методики направленны на процессы оптимизации производственной цепочки одного и того же изделия. По мнению авторов, именно это принципиальное различие приводит к тому, что попытки применять хорошо зарекомендовавшие себя в других отраслях статистические методы приводят к противоречивым результатам и проблемам при практическом применении, что подтверждается результатами в [2].

С другой стороны, несмотря на инженерную природу проектов по разработке программных продуктов, они не могут быть с лёгкостью оценены и декомпозированы на полную последовательность связанных задач и ресурсов (например, для построения диаграммы Ганта). Необходимо отметить, что теоретическая возможность это сделать существует, однако при выполнении большинства проектов по разработке программного обеспечения, построение такой последовательности работ до начала самих работ приводит к большим временны затратам, сравнимых с затратами на всю разработку программного продукта. Еще одной проблемой является высокая вероятность того, что даже построенная последовательность работ теряет смысл, если стейкхолдеры поменяют требования к системе. В реальных проектах это часто встречающаяся ситуация. В результате даже современные методы [3–5], основанные на статических оценках в управлении проектами, мало применимы к реальным проектам по разработке программного обеспечения.

В данной работе будет рассмотрен альтернативный подход к использованию статистических оценок. Вместо анализа и прогнозирования общего плана проекта на основе глобальных высокоуровневых оценок, авторы сосредоточились на приёмах, которые могут помочь менеджменту проекта принимать более взвешенные решения, и таким образом в частных вопросах положительно влиять на ход всего проекта в целом.

1. Описание ситуации

Обычно в теории управления проектами, проект рассматривают как направленную последовательность работ, нацеленных на получение фиксированного результата определённого качества, при условии ограниченных ресурсов и времени. Проект можно описать как комбинацию (Q,R,T), где Q – качество или требования к результату, R – ресурсы для достижения результата, а T – время, за которое должен быть достигнут результат. В целом считается, что эти переменные зависимы, и что качество должно

определять и время, и ресурсы. Однако зачастую для проектов по разработке программного обеспечения цепочка зависимости немного другая: $J(Q)=T$, $V(T)=R$.

В большинстве случаев до выполнения проекта проводится оценка сложности и реализации проекта в человеко/часах (или аналогичных величин). На основе этих оценок определяются стоимость и финансовые ресурсы на проект. В результате оценка задач является одной из наиболее критичных деятельностей, осуществляемых менеджментом проекта, поскольку оценка задач влияет на все остальные критичные параметры (время и стоимость). И именно оценка задач, сильно подвержена ряду проблем и сложностей:

1. Субъективность. Несмотря на то, что могут использоваться экспертная оценка, общее голосование, оценка будущим исполнителем, оценка по методу Story Point Poker, и другие, все они субъективны и их эффективность определяется тем, кто оценивает задачи.
2. Более формальные инженерные подходы для оценивания задач, которые позволяют повышать точность оценок зачастую несут накладные расходы на их поддержку, и фактически уместны только при разработке систем очень высокой сложности и качества.
3. Предыдущий опыт влияет на текущую оценку проекта иногда очень противоречиво. После проекта, который был сопряжён с трудностями и проблемами, следующий проект будут склонны оценивать с большим запасом (пессимистично), даже если следующий проект более простой.
4. Помимо оценивания самой задачи также критично, кто именно будет исполнять эту задачу. Даже не учитывая набор необходимых навыков (предполагается, что проектные менеджеры не будут в обычных условиях назначать исполнителей на задачи, которые те выполнить не в состоянии), разные исполнители обладают разными рабочими характеристиками, которые тоже обычно учитываются на интуитивном уровне.

Как результат, большая часть будущих проблем возникает до начала работ, и природа этих проблем заключена в природе экспертной оценки задач командой исполнителей. При этом затраты на более строгие инженерные подходы могут превысить потенциальные потери от ошибок планирования. Однако, предыдущий опыт, который имеется у проектных команд, может быть использован не только на уровне интуитивной пессимистической оценки, но и как источник данных для поиска дополнительной информации при оценивании нового проекта.

Большинство команд по разработке программного обеспечения используют системы управления проектами или их аналоги (Redmine, Trello, Jira, Kanban и пр.) и т.к. оценка задач является одной из основных работ любого проекта, то часть команд также практикуют подход, при котором задача, получаемая исполнителем, имеет оценку по времени исполнения до её начала, и при этом после выполнения фиксируется сколько времени было потрачено. Фактически существуют команды по разработке программного обеспечения, у которых есть накопленная информация о своих оценках о предыдущих проектах, и в какой-то момент времени есть информация о предыдущих выполненных задачах, что можно описать следующим образом.

Пусть у нас имеются N задач одного проекта, состоящих из: x_i – оценки времени исполнения задачи i исполнителем до выполнения, и y_i – затраченное время для исполнения задачи i, $i=\overline{1,N}$. Представляют интерес используемые и интерпретируемые характеристики исполнителя, которые можно получить на основе статистического анализа этих выборок. Будем считать, что $y_i,...,y_N$ – выборка, полученная из распределения $G(x)$, а $x_1,...,x_N$ – из $F(x)$. Согласно [6] точность оценок вероятностных характери-

стик исполнителя после выполнения задачи можно улучшить, используя информацию, содержащуюся в выборке $x_1,...,x_N$.

2. Статистические методы

Представляет интерес проверка гипотезы о том, что выборки $x_1,...,x_N$ и выборки $y_i,...,y_N$ однородны, т.е. $H_0: F(x) \equiv G(x)$. Проверим гипотезу однородности с помощью критерия Вилкоксона, который можно вычислить на основе следующей статистики

$$U = \sum_i \sum_j c\left(x_i - y_j\right), \; c(x) = \begin{cases} 0, & x \le 0, \\ 1, & x > 0. \end{cases}$$

Статистика *W* Вилкоксона и статистика *U* связаны линейным соотношением

$$W + U = N^2 + \frac{N(N+1)}{2}.$$

Предположим, что $G(x) = F(x-\theta)$, где θ – величина сдвига. Если принято решение о неоднородности выборки, то можно оценить параметр сдвига θ по формуле

$$\hat{\theta} = \text{median}\left\{\left(x_i - y_j\right)\right\}, \; i, j = \overline{1, N}.$$

Критерий Вилкоксона был применён на нескольких выборках данных, полученными в результате проведённых экспериментов со студентами высшей ИТ школы ТГУ. Эксперимент заключался в следующем, Все студенты получали одно и то же задание, которое они должны были оценить по времени выполнения, далее его индивидуально выполнить и зафиксировать время, которое потребовалось на его выполнение. Поскольку однородность выборок позволяет проводить оценку разработчиков, то достаточно критично иметь дополнительную информацию об исполнителях (в данном случае уровень успеваемости и достижений студентов), чтобы соотносить результаты, полученные на основе данного критерия с общей информацией об исполнителях (предполагается, что менеджер обладает дополнительной неформализованной информацией об уровне навыков своих исполнителей). В большинстве случаев гипотеза однородности отвергалась, что в целом согласуется с практическими результатами (очень часто даже примерный порядок “сложности” или ранга задачи не точны). Однако, отвергая гипотезу однородности, критерий Вилкоксона позволяет оценить сдвиг θ, который сложно использовать для оптимизации оценок, но можно применять для оценивания исполнителей-разработчиков. Продемонстрируем это на результатах, полученных во время экспериментов со студентами (табл. 1)

Таблица 1

Значения оценки сдвига в минутах, подсчитанные на основе экспериментов со студентами

Студент	Величина сдвига
1	49,5
2	13,5
3	25
4	0
5	–2,5
6	5
7	5,25

Подсчёт сдвига приводит к интересным результатам: чем больше величина сдвига, тем больше разработчик по тем или иным причинам себя «недооценивает» (иными словами либо его навыки лучше, чем он о себе думает, либо он склонен считать, что задачи более сложные). Фактически разработчик планирует свои оценки с запасом, больше времени, чем необходимо; чем меньше величина сдвига, тем больше себя «переоценивает», планируя меньше времени, чем того потребуется. Достаточно интересный прак-

тический кейс использования данной величины – чем более задачу можно охарактеризовать как сложную (следовательно, связанную с рисками), тем большую величину сдвига должен иметь разработчик, который будет её оценивать, т.к. это позволяет создать запас времени для проектного плана (разработчик с большим сдвигом оценит задачи с запасом) и не нарушить договорённости с заказчиком.

Однако эксперименты, проводимые со студентами высшей ИТ школы, достаточно специфичны. Исполняемые задачи не являлись частью одного проекта, они были обособленными и между задачами проходило достаточно много времени. Другой интерес вызывает возможность посмотреть, что можно получить, если анализировать выборку, полученную именно по одному связанному проекту с последовательным и регулярным выполнением задач исполнителем. Компания DevGuild предоставила подобные данные для статистических экспериментов. Авторам был передан анонимизированный набор данных по задачам одного проекта с двумя различными исполнителями (чьи имена скрыты), где по задачам было известно кто из двух исполнителей оценивал и выполнял эту задачу, оценку времени исполнения задачи в человека/часах и сколько потребовалось времени на исполнения задачи в человека/часах. Одним из интересных вопросов при управлении проектами является анализ зависимости между оценкой времени исполнения и временем выполнения задачи. Определить же степень связи между оценками задач x_i и временем исполнения y_i можно с помощью рангового коэффициента корреляции Спирмена $p = 1 - \frac{6}{N(N-1)(N+1)} \sum_{i=1}^{N} (R_i - S_i)^2$, где R_i – ранг x_i, и S_i – ранг y_i.

Данный коэффициент позволяет оценить наличие и степень связи между величинами. Подсчёт данного коэффициента применялся к выборке разработчиков из DevGuild. Для разработчика с наибольшим объёмом задач в выборке (размера 79) данный коэффициент оказался равен 0,9218. Высокий коэффициент корреляции позволяет использовать регрессию для прогноза время выполнения задачи по времени исполнения задачи. Однако, поскольку законы распределения $F(x)$ и $G(x)$ неизвестны, то было принято решение применить непараметрическую регрессию с ядром Гаусса [7]:

$$r(x) = \frac{\sum_{i=1}^{N} y_i e^{-\frac{\left(\frac{x-x_i}{h}\right)^2}{2}}}{\sum_{i=1}^{N} e^{-\frac{\left(\frac{x-x_i}{h}\right)^2}{2}}},$$

где h – ширина окна. Данная непараметрическая регрессия была применена к выборке из компании DevGuild на основе данных разработчика с наибольшим количеством задач. В общем случае, при применении ко всей выборе регрессия ожидаемо улучшает общий результат (совокупная разница $x_i - y_i$ меньше, чем $x_i - \hat{y}_i$). Однако возникает вопрос: можно ли использовать непараметрическую регрессию для улучшения прогноза? (т.е. на основании оценок экспертов прогнозировать новую величину с использованием непараметрической регрессии, которую использовать вместо изначальных оценок). Для этого вся выборка была случайным образом поделена на обучающую (на основе которой регрессия определяла свои параметры и в ней оказалось 42 задачи), и тестовую (на которой проверяли результаты обученной регрессии, в ней оказалось 37 задач). Изначально предполагалось, что регрессия позволит улучшить качество оценок задач. Результат оказался достаточно интересным (рис. 1)

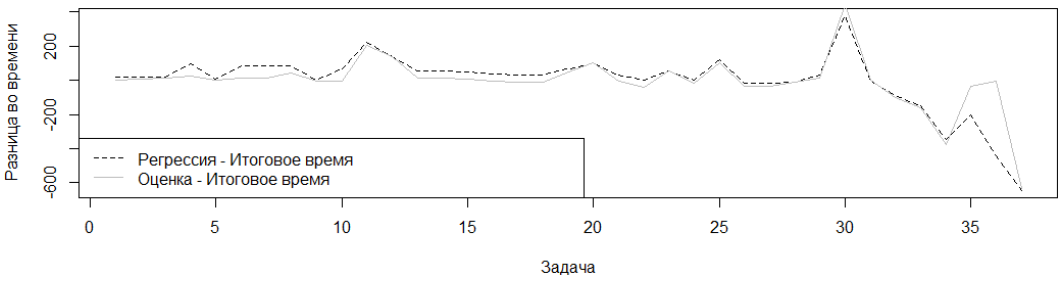


Рис. 1. Разница между регрессией и итоговым временем

На графика представлена разница (в минутах) между изначальной оценкой и итоговым затраченным временем (прямая линия серого цвета) по каждой задаче и разница между регрессионным временем и итоговым временем (пунктирная линия чёрного цвета). Использование регрессии сделало оценки времени исполнения по отдельным задачам «хуже» (т.е. дальше от итогового времени). Однако интересным результатом является то, что, если посчитать сумму «ошибок» (т.е. то насколько далеко оценка или регрессия от итогового времени), то получим, что для изначальной (экспертной) оценки времени исполнения задачи разница составит минус 270 минут, а для регрессионной оценки 11 минут. В результате вместо проигрыша в оценке почти в 4,5 человека часа (что в реальной разработке можно округлить до одного рабочего дня), получили «выигрыш» в 11 минут. Общее время запланированных задач было 5892 минуты (или же 98,2 часа, или же 2,5 рабочих недели, т. е. фактически 12–13 рабочих дней), в данном случае получилось, что применённая регрессия дала выигрыш в пределах 7,8–8,3% если говорить про ошибку планирования рабочего времени.

Заключение

В результате проделанной работы можно сделать следующие выводы: не создавая специализированных «лабораторных» условий при разработке программного обеспечения, фактически только на основе базовой информации, достаточно сложно получить эффективные и практически значимые выводы на основе стандартных статистических оценок. Однако, используя статистические оценки, можно получить вспомогательную информацию, которая позволяет дополнительно оценивать разработчиков. Результаты показывают, что непараметрическая регрессия имеет потенциал для корректировки «планов» разработчика, однако для полноценного вывода потребуется дополнительные исследования (больше выборок из действующих проектных команд). Необходимо отметить, что в данной работе разбирались только те ситуации, в которых исполнитель одновременно и оценивал задачу и выполнял её. Однако могут встречаться иные ситуации, при которых оценку задач осуществляет не будущий исполнитель, а другое лицо (например, самый опытный член команды разработчиков, отдельно менеджер или коллективное голосование за сложность задач). Рассмотрение такой ситуации – в последующей работе авторов.

Авторы благодарят компания DevGuild в лице Неверова Михаила Павловича за предоставленные данные, а также студентов высшей ИТ школы, которые активно приняли участие в экспериментах.

ЛИТЕРАТУРА

1. *Boehm B., Rombach H.D., Zelkowitz M.* (2005). Foundations of Empirical Software Engineering: The Legacy of Victor R. Basili. Fraunhofer IESE.

2. National Research Council and Natl Research Coun. 1996. Statistical Software Engineering. National Academy Press, USA.

3. *Miguel A., Madria W., Polancos R.* (2019). Project Management Model: Integrating Earned Schedule, Quality, and Risk in Earned Value Management. 622-628. 10.1109/IEA.2019.8714979.

4. *Sackey S., Lee D.E., Kim B.S.* (2020). Duration Estimate at Completion: Improving Earned Value Management Forecasting Accuracy. KSCE Journal of Civil Engineering. 10.1007/s12205-020-0407-5.
5. *Baqerin M.H., Shafahi Y., Kashani H.* (2016). Application of Weibull analysis to evaluate and forecast schedule performance in repetitive projects. Journal of Construction Engineering and Management, 142(2), 04015058.
6. *Dmitriev Y.G., Zmeev D.O.* On a Combined Estimator of Probabilistic Characteristics //Eighth International Conference on Risk Analysis and Design of Experiments, Vienna, 23-26 april 2019 : book of abstracts. Vienna, 2019. P. 147-148.
7. *Хардле В.* Прикладная непараметрическая регрессия. — 1989.

КЛАССИФИКАЦИЯ СОВРЕМЕННЫХ ПОРТФЕЛЬНЫХ ТЕОРИЙ

Иштуганов Р.А.
Томский политехнический университет
r.ishtuganov@gmail.com

Введение

Для создания любого продукта в целом и программного приложения в частности, требуется фундаментальная основа, которая будет описывать механизм его работы, регулировать взаимодействия компонентов внутри продукта на формальном языке. В качестве формального языка выступает язык математики, во всем многообразии ее разделов, а основой, фундаментом продукта – та или иная математическая теория или модель, реализующая результатом своего применения к набору данных знание, которое в свою очередь продукт предоставляет своим пользователям.

В рамках настоящей работы нами предпринята попытка выполнить классификацию современных портфельных теорий, ставящих своей задачей подбор оптимального портфеля ценных бумаг, с максимальной доходностью при минимальном риске, основываясь на математическом аппарате, подходах и методах, которые легли в основу той или иной портфельной теории.

1. Классификация

Если мы выполним классификацию всех известных портфельных теории по времени их возникновения, то можно прийти к следующему:

1. Традиционные теории:
   1.1. Теория Доу-Джонса;
   1.2. Теория случайного блуждания;
   1.3. Формульная теория.
2. Современные теории:
   2.1. Модели минимизации дисперсии среднего;
   2.2. Модели паритета риска;
   2.3. Байесовские модели.

2. Традиционные теории

Традиционные портфельные теории не имели в своей основе математического аппарата, позволяющего формально выразить в первую очередь риск, а также связь риска с доходностью.

Теория Доу-Джонса в первую очередь описывает тенденции движения цен на активы и их цикличность и в плане управления портфелем только выдает сигналы на открытие или закрытие позиций.

Теория случайного блуждания и близкая ей теория эффективного рынка описывает движения цен на активы как совершенно случайные события и в рамках настоящего дискурса не предоставляет никаких аналитических инструментов[*], а в плане управления портфелем рекомендует только в пассивные индексные фонды рынка.

[*] Хотя теория случайного блуждания и упоминает риск, говоря о том, что невозможно постоянно получить доходность выше рыночной с учетом риска, но само понятие риска аналитически не определено.

Формальная теория в целом устанавливает правила (формулы) распределение активов между двумя видами портфелей, а также содержит набор правил по открытию и закрытию позиций. В плане управления риском, данная теория оперирует только на уровне минимизации убытков портфеля (то, что мы называем «наивным» пониманием риска и это именно то понятие, которое интуитивно ассоциируется с риском у большинства).

3. Современные теории

Только в 1952 г. Г. Марковицем был предложен формальный подход к определению риска, показана функциональная зависимость доходности от риска и предложена формальная методика выбора портфеля по соотношению риск/доходность. Эта теория в настоящее время известна как «Современная портфельная теория», которая дала начало применению аналитических методов при оценке портфелей финансовых активов.

4. Модели минимизации дисперсии среднего

Как уже отмечалась, годом зарождения портфельного анализа как области научного знания является 1952 г., когда Г. Марковиц опубликовал статью «Portfolio Selection» [1]. Теория утверждает, что каждый инвестор стремится реализовать две цели: максимизировать доходность и минимизировать неопределённость, связанную с фактом получения такой доходности. Таким образом, в момент принятия решения о приобретении той или иной ценной бумаги или их набора, необходимо сбалансировать эти две цели, что и помогает сделать предложенная теория.

Основные положения теории заключаются в следующем:

В момент t_0 инвестор не знает каким будет уровень доходности потенциальных портфелей[*] ценных бумаг, поэтому ожидаемые доходности стоит считать случайной величиной. Случайная величина среди прочего имеет две характеристики – среднее значение и стандартное отклонение. Поэтому, инвестору необходимо оценить ожидаемые доходности потенциальных портфелей, которые в подавляющем большинстве случаев выражаются через средние исторические доходности и риски, которые также в подавляющем большинстве случаев выражаются как стандартное отклонение исторических доходностей. После этого необходимо выбрать лучший портфель с точки зрения минимального риска (стандартного отклонения) при максимально возможной доходности (среднего значения). Поэтому методы, применяющие данный подход для выбора ценных бумаг и получили название методов минимизации дисперсии среднего значения.

Стоит отметить, что данная теория неявно делает два предположения:

1. О ненасыщаемости – т.е. при одинаковом уровне риска инвестор всегда предпочтет большую доходность

2. Об избегании риска – т.е. при одинаковой доходности инвестор всегда предпочтет минимальный риск.

Расчет доходности портфеля представлен в формуле (1):

$$r_p = \sum_{i=1}^{N} X_i r_i \ , \qquad (1)$$

где r_p – ожидаемая доходность портфеля, X_i – процентная доля i-й ценной бумаги в портфеле, r_i – ожидаемая доходность i-й ценной бумаги, N – количество ценных бумаг в портфеле.

Расчет риска портфеля представлен в формуле (2):

[*] За единственным исключением приобретения Treasury Inflation-Protected Securities.

$$\sigma_p = \sqrt{\sum_{i=1}^{N}\sum_{J=1}^{N} X_i X_j \sigma_{ij}}\ , \qquad (2)$$

где X_i – процентная доля i-й ценной бумаги в портфеле, X_j – процентная доля j-й ценной бумаги в портфеле, $\overline{\sigma}_{ij}$ – ковариация доходностей i-й и j-й ценных бумаг.

На основании данных формул мы можем рассчитать т.н. достижимое множество, т.е. комбинацию всех возможных портфелей на данном рынке (рис. 1).

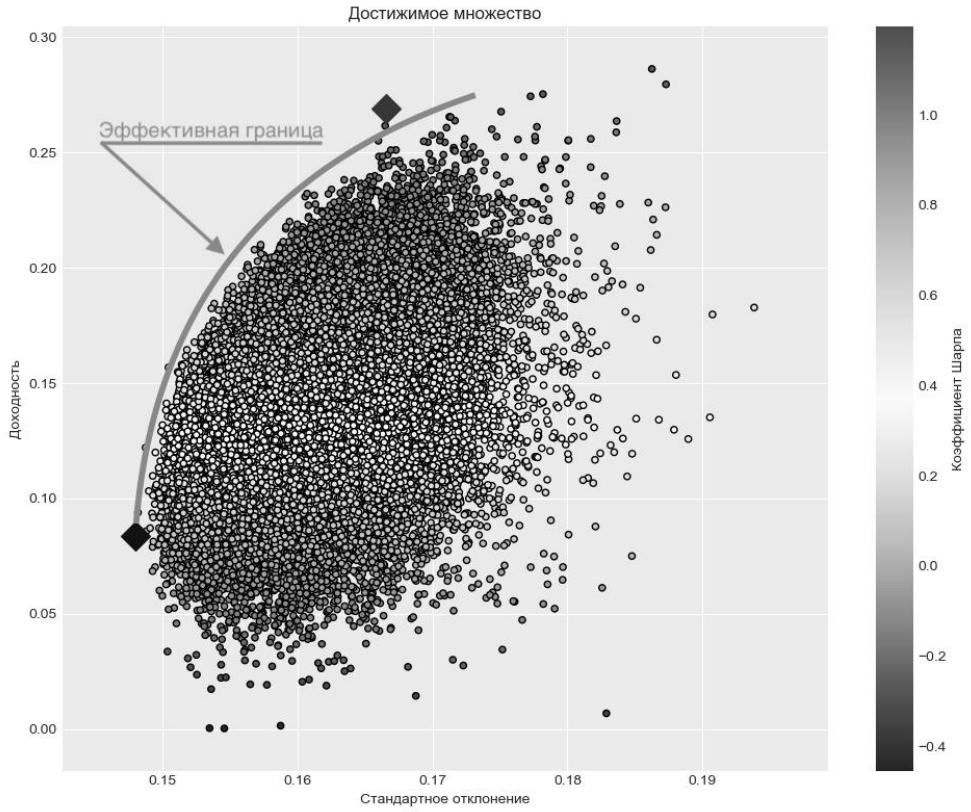


Рис. 1. Достижимое и эффективное множества

Однако инвестора должно интересовать только определенное подмножество портфелей из данного достижимого множества. А именно, множество портфелей, лежащих на левой границе достижимого множества портфелей, и известное как «эффективное множество». Это те портфели, которые обеспечивают максимальный уровень доходности при минимальном риске. Для того, чтобы подобрать оптимальный для себя портфель, инвестору необходимо построить кривые безразличия, характеризующие его/ее отношение к риску и выбрать портфель, находящийся в точке касания кривой безразличия эффективного множества.

Однако процесс построения кривых безразличия не тривиален, требует определения функции полезности для каждого индивидуального инвестора и затем определения степени принятия риска.

Поэтому более простой способ выбора портфеля был предложен У. Шарпом в 1966 г. [2] и заключается в ранжировании портфелей по коэффициенту отношения доходности к риску, известному как коэффициент Шарпа. Расчет данного коэффициента представлен в формуле (3):

$$SR = \frac{R_p - R_f}{\sigma_p}, \tag{3}$$

где R_p – доходность портфеля, R_f – безрисковая ставка, σ_p – стандартное отклонение доходности портфеля.

На интуитивном уровне данная формула понимается следующим образом – вычисляемый коэффициент показывает сколько единиц рискованной доходности приходится на 1 единицу риска. Соответственно, инвестору следует выбирать портфель с наивысшим коэффициентом Шарпа. Другой выбор, который может сделать инвестор – это портфель с минимальным риском, т.е. с минимальным стандартным отклонением. На рисунке данные портфели помечены ромбами.

Стоит отметить, что, исходя из рыночной модели*, риск портфеля состоит из двух компонентов, так же, как и риск отдельной ценной бумаги – это рыночный риск и собственный риск и может быть выражен формулой (4).

$$\sigma_p = \sqrt{\beta_{pI}^2 \sigma_I^2 + \sigma_{\varepsilon p}^2}, \tag{4}$$

где β_{pI} – коэффициент «бета», характеризующий степень зависимости доходности портфеля от доходности всего рынка, который в свою очередь является средневзвешенным коэффициентом бета всех ценных бумаг, находящихся в портфеле, σ_I^2 – дисперсия доходности рынка, $\sigma_{\varepsilon p}^2$ – дисперсия случайной погрешности доходности портфеля, которая в свою очередь является средневзвешенной случайной погрешностью доходностей ценных бумаг, входящих в портфель. Данный член уравнения «компенсирует» разницу между расчётной доходностью в соответствии с рыночной моделью и фактической.

Коэффициент «бета» портфеля в свою очередь рассчитывается согласно формуле (5):

$$\beta_{pI} = \sum_{i=1}^{N} X_i \frac{\sigma_{iI}}{\sigma_I^2}, \tag{5}$$

где X_i – доля i-й ценной бумаги в портфеле, σ_{iI} – ковариация доходности i-й ценной бумаги в портфеле, σ_I – дисперсия доходности рынка.

Включение в портфель все большего количества ценных бумаг (или диверсификация) не оказывает значительного влияния на рыночный риск и ведет к его усреднению. На интуитивном уровне это объясняется тем, что изменения на рынке в связи с теми или иными политическими или экономическими событиями приведут к соответствующему изменению цен всех ценных бумаг на рынке в той или иной мере.

Случайная погрешность портфеля рассчитывается согласно формуле (6):

$$\sigma_{\varepsilon p}^2 = \frac{1}{N}\left(\frac{\sigma_{\varepsilon 1}^2 + \sigma_{\varepsilon 2}^2 + \sigma_{\varepsilon N}^2}{N}\right), \tag{6}$$

где N – количество ценных бумаг в портфеле, $\sigma_{\varepsilon i}^2$ – дисперсия доходности i-й ценной бумаги в портфеле.

Как следует из формулы (6), включение в портфель все большего количества ценных бумаг (или диверсификация) оказывает значительное влияния на собственный риск портфеля и ведет к его снижению. На интуитивном уровне это объясняется тем, что высокий риск одной ценной бумаги компенсируется низким риском другой ценной бумаги.

Влияние диверсификации портфеля на его риск обобщено на рис. 2.

* Данная модель предполагает, что доходность ценной бумаги неким образом связана с доходностью всего рынка, выраженного через какой-либо индекс.

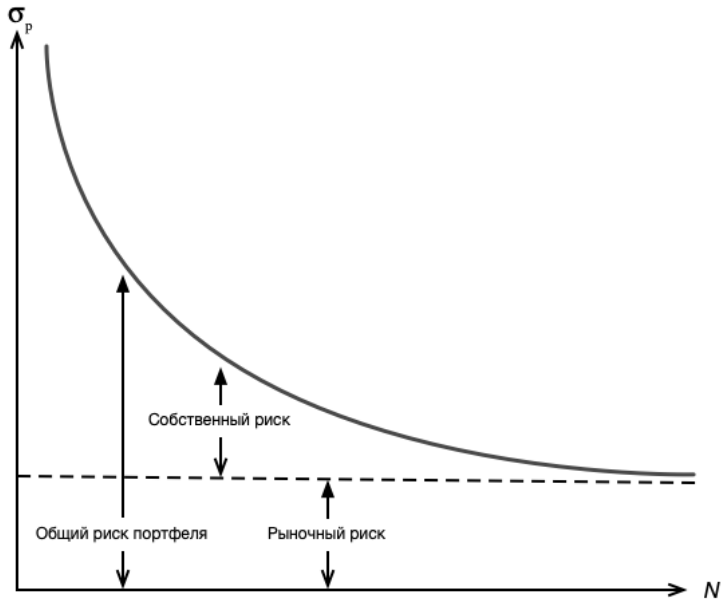


Рис. 2. Влияние диверсификации на риск

5. Пост-современная портфельная теория

Данная теория, впервые упомянутая в прессе в 1993 г. [3] и основанная на работах Х. Форси и Ф. Сортино [4] скорее является дополнением теории Марковица, а не совершенно новой теорией, призванной решить некоторые противоречия современной портфельной теории.

Основные ограничения портфельной теории Марковица состоят в том, что она подразумевает следующее:

1. Стандартное отклонение доходностей ценных бумаг портфеля является истиной мерой риска;
2. Доходности ценных бумаг портфеля могут быть адекватно представлены нормальным распределением.

Т.е. портфельная теория Марковица использует такие меры риска и доходности, которые не всегда точно отражают объективную реальность. Так, использование нормального распределения подразумевает, что риск получения доходности выше ожидаемого (или среднего, при использовании исторических доходностей) равен риску получения доходности ниже ожидаемого. Также, портфельная теория Марковица оценивает ценные бумаги с доходностями, в которых количество положительных отклонений от среднего превышает количество отрицательных, как более рискованные чем они есть на самом деле. Обратное искажение характерно и для доходностей, где преобладают отрицательные доходности.

Интересно, что сам Г. Марковиц отмечал [5], что в некоторых случаях его теория не точно отражает поведение и ожидания инвесторов и предполагал, что модель с использованием полудисперсии будет более предпочтительней. Но из-за сложности расчетов полудисперсии он все же взял за основу среднее и стандартное отклонение.

Поэтому для снятия данных противоречий была предложена т.н. «Пост-современная портфельная теория», которая использует полудисперсию, а также вводит понятие «целевой доходности» для сравнения с фактической доходностью, что более точно отражает объективную реальность. Именно отрицательная разница между целевой доходностью и фактической доходностью и является истиной мера риска в данной теории.

Пример расчет данной меры риска представлен на формуле (7):

$$\sigma_{DR} = \sqrt{\int_{-\infty}^{t} (t-r)^2 \, f(r) dr}, \tag{7}$$

где t – целевая требуемая доходность ценной бумаги или портфеля, r – средняя годовая доходность ценной бумаги и портфеля, $f(r)$ – функция распределения доходности.

Данную формулу интуитивно можно понимать так, что мы учитываем вероятностно взвешенные квадраты доходностей ниже целевой. Возведение в квадрат экспоненциально «наказывает» «плохие» доходности, что более точно соответствует интуитивному пониманию риска [6].

Для ранжирования портфелей, теория предлагает коэффициент, аналогичный коэффициенту Шарпа – коэффициент Сортино [7], расчет которого представлен на формуле (8). Как уже отмечалась выше, коэффициент Шарпа, как и вся теория Марковица, исходят из предположения нормального распределения доходности. Соответственно, при рассмотрении доходностей, которые не отвечают критерию нормального распределения, а, например, скошены вправо (присутствуют положительные сюрпризы, которые инвестор будет оценивать как желаемые) коэффициент Шарпа будет оценить как высокорискованные:

$$SrR = \frac{R_p - T_p}{\sigma_{DRp}} \tag{8}$$

где R_p – средняя годовая доходность ценной бумаги или портфеля, T_p – целевая требуемая доходность ценной бумаги и портфеля, σ_{DRp} – «Downside Risk» портфеля.

В остальном процедура выбора портфеля соответствует подхожу, предложенному в теории Марковица – поиск портфеля либо с минимальным значением риска или с максимальным значением коэффициента Сортино.

6. Бюджетирование риска

Кроме вышеперечисленных проблем, характерных для современной портфельной теории, другой проблемой, отмеченной в работах [8,9] является высокая чувствительность модели к входным параметрам и в первую очередь к параметру ожидаемой доходности. Так, было показано, что при изменении доходности и риска на одинаковую величину, влияние доходности на состав портфеля оказывается в 10 раз более сильным. А ведь именно оценки доходности являются наиболее сложным процессом инвестиционной деятельности и характеризуются максимальной неопределенностью. Параметр риска в этом смысле гораздо более стабилен.

Поэтому были предложены методы оптимизации портфеля, которые бы не учитывали показатель доходности, а оперировали только риском. Построенные в ходе применения этих методов портфели получили название «портфели паритета риска».

Базовая идея метода состоит в том, чтобы подобрать активы в портфель таким образом, чтобы величина вклада каждого актива в общий риск портфеля была одинакова.

Можно использовать различные меры риска, например, VaR, но «классический» подход – это использование стандартного отклонения исторических доходностей.

Определив стандартное отклонение портфеля по историческим данным, мы может рассчитать вклад каждого актива в общий риск портфеля в соответствии с формулой (9):

$$RC_i = w_i \frac{\sum_{j=1}^{N} w_i \operatorname{cov}(X_i X_j)}{\sigma_p}, \tag{9}$$

где w_i – вес *i*-й ценной бумаги в портфеле, N – количество ценных бумаг в портфеле, $\mathrm{cov}\left(X_i X_j\right)$ – коэффициент ковариации доходностей *i*-й и *j*-й ценной бумаги, σ_p – стандартное отклонение доходности портфеля.

На интуитивном уровне данная формула показывает взвешенный коэффициент чувствительности риска актива к риску всего портфеля.

Получив данный коэффициент чувствительности для всех активов портфеля, необходимо их уровнять так, чтобы $RC_i = \frac{1}{N}\sigma_p$, где N – количество ценных бумаг в портфеле, σ_p – стандартное отклонение доходности портфеля.

Аналитическое решение данной задачи существует только для случая диагональной матрицы дисперсии активов (т.е. при условии, что доходности активов не коррелированы между собой).

Для всех остальных случаев необходимо использовать численные методы. В настоящее время предложено большое количество алгоритмов на разных языках программирования (R, Python, Matlab) в которых применяются различные подходы к решению данной задачи – метод Ньютона, циклический покоординатный спуск и др.

В общем, можно сказать, что данный подход является частным случаем метода бюджетирования риска, в котором производится явное распределение риска в соответствии с профилем риска той или иной ценной бумаги. Другими словами, коэффициенты вклада риска каждой ценной бумаги в портфель не подразумеваются равными, а задаются явно и задача сводится к определению весов на основании уже имеющихся коэффициентов вклада риска.

7. Байесовские методы

Байесовская статистика – это теория в области статистики, основанная на байесовской интерпретации вероятности, когда вероятность отражает степень доверия событию. Степень доверия может основываться на априорных знаниях о событии, таких как результаты предыдущих экспериментов или личном доверии событию. Самым известной моделью, применяемой данный подход для решения проблемы оптимизации портфеля ценных бумаг, является модель «Блэка – Литтермана».

8. Модель Блэка-Литтермана

Данная модель была разработана Ф. Блэком и Р. Литтерманов во время их работы в банке «Голдман Сакс» и впервые опубликована в 1992 г. [10]. Данная модель совмещает априорные оценки доходности с апостериорными оценками ожидаемых доходностей. Но данная модель не только позволяет инвестору учесть свой прогноз относительно ожидаемых доходности той или иной ценной бумаги как в абсолютных значениях, так и относительных, но и степень уверенности в том или ином прогнозе. Расчет оптимального портфеля выполняется в несколько шагов.

Первым этапом необходимо рассчитать апостериорные значения. Оценка начинается с расчета рыночного риска согласно формуле (10):

$$\lambda = \frac{R - R_f}{\sigma^2}, \tag{10}$$

где R – рыночная, равновесная доходность, R_f – безрисковая ставка, σ^2 – дисперсия рынка.

Затем выполняется расчет рыночной или равновесной доходности портфеля согласно формуле (11):

$$\Pi = \lambda \Sigma w_{\mathrm{mkt}}, \tag{11}$$

где $\Pi_{N\times 1}$ – вектор равновесных доходностей, N – количество ценных бумаг в портфеле, λ – рыночный риск, $\Sigma_{N\times N}$ – ковариационная матрица доходностей активов в портфеле, w_{mkt} – удельный вас актива в портфеле на основании его капитализации.

Если у инвестора нет никакого мнения относительно будущей доходности активов в портфеле, то вычисления прекращаются на данном этапе и, в таком случае модель Блэка – Литтермана, по сути, рекомендует держать рыночно нейтральный портфель ценных бумаг.

Вторым этапом необходимо рассчитать апостериорные значения. Если у инвестора имеется мнение касательно будущих доходностей активов, то их необходимо выразить либо в абсолютных, либо в относительных значениях. Например, «Доходность акций компании А составит 10%» или «Акции компании В покажут доходность на 5% выше, чем акции компании С». Мнение инвестора формализуется с помощью трех матриц:

Матрица P отражает факт наличие или отсутствие мнения инвестора и тип мнения – абсолютный или относительный (рис. 3).

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix} \text{ или } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0{,}8 & -0{,}2 \end{bmatrix}$$

Рис. 3. Пример матрицы мнений

Матрица Q отражает мнение инвестора относительно ожидаемой доходности активов (рис. 4).

$$\begin{bmatrix} 10 \\ 5 \end{bmatrix}$$

Рис. 4. Пример матрицы предполагаемых доходностей

Матрица Ω – матрица уверенности инвестора во мнении относительно ожидаемой доходности. Причем в данной матрице на этапе расчётов используются не непосредственно значения уверенности, а дисперсия ошибок (отличие предполагаемой доходности от фактической) в виде ковариационной матрицы ошибок. Существует несколько методов расчета данной матрицы, классическим, предложенный создателями модели, является использование априорный дисперсий (на интуитивном уровне это можно объяснить так, что актив, который характеризовался высокой дисперсией имеет высокую степень неопределенности прогноза). Рассчитывается данная матрица согласно формуле (12):

$$\Omega = \text{diag}(P(\tau\Sigma)P^T), \tag{12}$$

где P – матрица мнений инвестора о доходности активов, τ – весовой коэффициент, отражающий степень неуверенности во мнении, $\Sigma_{N\times N}$ – ковариационная матрица доходностей активов в портфеле.

Такой подход учитывает и неуверенность мнения инвестора касательно доходности актива и априорную подразумеваемую неопределенность, выраженную дисперсией исторических доходностей.

При этом необходимо рассчитать коэффициент τ, который характеризует степень неуверенности инвестора во мнении. Одним из походов является использование формулы (13):

$$\tau = \frac{1}{T}, \tag{13}$$

где T – количество наблюдений в выборке, либо можно использовать заранее определенное значение, равное 0,05.

После этого можно рассчитать новый комбинированный вектор доходности, представленный формулой (14):

$$E_r = (\tau\Sigma)^{-1} + P^T\Omega^{-1}P^{-1}\left(\tau\Sigma\right)^{-1}\Pi + P^T\Omega^{-1}Q\,, \qquad (14)$$

где τ – весовой коэффициент, отражающий степень неуверенности во мнении, $\Sigma_{N\times N}$ – ковариационная матрица доходностей активов в портфеле, $P_{K\times N}$ – матрица, отражающая факт наличие или отсутствие мнения инвестора и тип мнения, K – количество прогнозов, N – количество ценных бумаг в портфеле, $Q_{K\times 1}$ – матрица, отражает мнение инвестора относительно ожидаемой доходности активов, $\Omega_{K\times K}$ – матрица, характеризующая уверенности инвестора во мнении относительно ожидаемой доходности актива, $\Pi_{N\times 1}$ – вектор равновесных доходностей.

На интуитивном уровне, данная формула рассчитывает средневзвешенную сумму вектора подразумеваемой доходности актива и вектора прогнозной доходности, где весами выступают подразумеваемая неопределённость доходностей, неопределённость мнений и степени неопределенности мнений.

Получив ожидаемый вектор доходностей, его необходимо подставить в формулу (11) и решить для каждого компонента вектора, получив тем самым веса активов в целевом портфеле.

Заключение

В ходе выполнения работы, был предложен подход к классификации портфельных теорий, применяющихся на практике в современном портфельном управлении, а именно на основе математического аппарата, методов и подходов, лежащих в основе той или иной портфельной теории. В рамках каждой математической модели была рассмотрена наиболее популярная теория ее представляющая.

ЛИТЕРАТУРА

1. *Markovitz H.* 1952. Portfolio Selection. The Journal of Finance, 7(1), P. 77-91.
2. *Sharpe W.* 1966. Mutual Fund Performance. The Journal of Business, 39(1), P. 119-138.
3. *Brian R.M., Ferguson K.W.* 1993. Post-Modern Portfolio Theory Comes of Age. The Journal of Investing, 2(4), P. 27-33.
4. *Sortino F., Satchell S.* Managing Downside Risk in Financial Markets // F. Sortino, S. Satchell. Oxford: Reed Educational and Professional Publishing Ltd., 2001.
5. *Sharpe W.F.* 1964. Capital Asset Prices: A Theory of Market Equilibrium under Consideration of Risk. Journal of Finance, 19(3), P. 425-442.
6. *Fishburn P.* 1997. Mean-Risk Analysis with Risk Associated with Below-Target Returns. The American Economic Review, 67(2), P. 116-126.
8. *Sortino F., Price L.* 1994. Performance Measurement in a Downside Risk Framework. The Journal of Investing, 3(3), P. 59-64.
9. *Merton R.* 1980. On estimating the expected return on the market: An exploratory investigation. Journal of Financial Economics, 8(4), P. 323-361.
10. *Chopra V., Ziemba W.* 1993. The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice. The Journal of Portfolio Management, 19(2), P. 6-11.

ТЕХНОЛОГИЯ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛИЗА РЕШЕНИЯ ПРОБЛЕМ НА ПРЕДПРИЯТИИ В УСЛОВИЯХ ОГРАНИЧИТЕЛЬНЫХ МЕР ДЛЯ НАСЕЛЕНИЯ

Кодочигов А.В., Тарасенко В.Ф.
Томский государственный университет
av.kodochigov@gmail.com, vtara54@mail.ru

Введение

Текущая ситуация в стране и в мире, с экономической точки зрения напоминающая мировой финансовый кризис 2008 г., диктует новые условия для нормального функци-

онирования как предприятий малого и среднего бизнеса, так и населения страны и мира в целом. Ограничительные меры, принимаемые государством для сохранения и поддержки здоровья граждан, так или иначе, приводят к значительному снижению спроса на практически все категории товаров и услуг. Многие предприятия вынуждены сокращать штат, либо переводить сотрудников на удаленный режим работы в связи с падением спроса и сокращением мощностей соответственно.

Для недопущения роста уровня безработицы принимаются меры поддержки, позволяющие предприятиям не идти на крупные сокращения или ликвидацию совсем. Полная остановка экономической деятельности некоторых организаций даже на относительно короткий период времени может привести к их полной ликвидации.

Главная задача руководства во время кризиса – оптимизировать экономические процессы внутри предприятия и разработать меры по поддержанию конкурентоспособности предлагаемой продукции. Для того, чтобы сформировать антикризисную модель поведения предприятия, управляющему персоналу необходимо понимать природу кризиса и какие изменения для хозяйственной деятельности он несет. Следствием развития кризиса является изменение доходности всех хозяйств: государств, компаний, частных домохозяйств. Чаще всего развитие кризиса сопровождается эффектом домино [1], который можно описать следующими друг за другом процессами:

1. Наблюдается падение реального дохода населения и его покупательской способности.
2. Сокращение покупок на рынке.
3. Падение продаж и доходов B2C предприятий.
4. Сокращение заказов от B2C компаний на сырье, оборудование, материалы и др.
5. Сокращение продаж и доходов B2B предприятий.

Такое изменение рыночной структуры выносит на первое место роль системного анализа, как инструмента для разработки упредительных решений. Переход рынка из фазы стабильного развития, в соответствии с его жизненным циклом, в фазу активной стагнации в период кризиса, заставляет топ-менеджмент менять принципы и систему управления, превентивно внедряя систему антикризисных мер. Управленческие решения и стратегические инициативы, разработанные на каждом уровне управления маркетингом предприятия, предусматривают формирование целей и целевых показателей, позволяющих контролировать уровень достижения запланированных результатов в области рынка и клиентской составляющей. Стратегический план развития состоит из системы целей, которые в зависимости от разрабатываемой стратегии могут быть направлены во внешнюю среду, или определять необходимые достижения внутри предприятия [1].

Для разработки моделей управления в данных условиях можно применить технологию прикладного системного анализа для комплексного изучения проблемы, в частности, 1) сформулировать проблему с точки зрения заказчика; 2) поставить диагноз проблемы; 3) составить список стейкхолдеров; 4) определить проблемное месиво; 5) построение конфигуратора; 6) целевыявление; 7) определение критериев; 8) экспериментальное исследование систем; 9) построение и совершенствование моделей; 10) генерирование альтернатив; 11) выбор или принятие решения; 12) реализация решения [2].

Разработка математических моделей с применением метода проб и ошибок позволяет снизить риски для предприятия в условиях экономической нестабильности, и увеличить прибыль в условиях отсутствия значительных внешних воздействий, способных повлиять на показатели спроса и цены на готовую продукцию.

Описание проблемы в данной статье осуществлено на примере крупного завода по производству кабельной продукции.

1. Объект исследования

Само производство кабельной продукции осуществляется в несколько этапов. Не углубляясь в подробности, можно сказать, что на первом этапе из медного или алюминиевого сплава изготавливаются проволоки, которые впоследствии скручиваются в токопроводящие жилы. Затем, этим жилы проходят через экструзионную линию, где кабель покрывается изоляцией и обретает конечный вид [3–5].

В качестве объекта исследования выступает крупный завод по производству кабельной продукции, в том числе силовых кабелей ВВГ-П. Такие кабели используются в системах передачи электроэнергии от электроснабжающих и жилищно-коммунальных предприятий к потребителю и разработаны они в основном для линий с напряжением до 1 кВ. Также существует модификации, которые способны выдержать напряжение до 10 кВ.

Руководство предприятия, прежде всего, должно организовать производство и поставку кабельной продукции по долгосрочным обязательствам, заключенным с компаниями заказчиками. Все прочие виды деятельности должны быть направлены на реализацию заключенных соглашений.

2. Постановка задачи и экономико-математическая модель оптимизации работы предприятия

Постановка задачи и экономико-математическая модель оптимизации работы завода по производству кабельной продукции в условиях ограничений выполнена на основе концептуальных положений прогнозно-адаптивного подхода к управлению компаниями [6].

Допустим, что завод по производству кабельной продукции имеет n заготовочных линий, каждую из которых обслуживает k производственных бригад. Линии могут заготавливать кабели различных марок f. К примеру, ВВГ-П 3х2,5, ВВГ-П 3х4, ВВГ-П 3х6. По имеющимся договорным обязательствам предприятие должно выпустить необходимое количество кабельной продукции конкретных марок. Цена на электроэнергию и материалы варьируется в зависимости от внешних условий, но в определенный момент времени она имеет постоянное значение.

Перед топ-менеджерами предприятия стоит задача максимизации прибыли в условиях, когда рынок стабилен, и производимая кабельная продукция востребована на внутренних и внешних рынках [7].

Естественно, что в период стабильного экономического развития повышенной спрос на кабельную продукцию стимулирует у производителя желание реализовать выпуск дополнительного количества готовой продукции, помимо того, которое необходимо выпустить по имеющимся договорным обязательствам.

В период нестабильности рынка, связанной с иными причинами, управляющий персонал таких предприятий уже не может рассчитывать на получение значительных прибылей от реализации готовой продукции (см. для примера Приложение). Более того, для обеспечения непрерывной работы производства, вынуждены нести дополнительные расходы. Эти расходы связаны с тем, что продолжающее работать предприятие в условиях пониженного, или даже отсутствующего спроса на готовую кабельную продукцию, несет издержки на материалы, электроэнергию, заработную плату и т.д. Поэтому в ограниченных или нестабильных условиях мирового рынка предприятие должно стремиться свести к минимуму издержки [1,8,9].

3. Целевая функция предприятия в условиях стабильного развития мировой экономики

В условиях стабильного развития рынка, целевая функция Φ_1 работы предприятия по производству кабельной продукции может быть представлена в следующем виде

$\Phi_1 = \Pi^T \to \max$, где Π^T – прибыль предприятия от реализации готовой продукции кабеля всех марок в период времени T.

Прибыль от реализации готовой продукции всех марок кабельной продукции в период T рассчитывается по формуле [6]:

$$\Pi^T = \sum_f \left(I_f^T C_f^T \right) - \sum_f \left(I_f^T \left(\sum_n S_{fn}^{ET} + \sum_n \sum_m S_{fnm}^{MT} + \sum_k \sum_n Z_{fkn}^T \right) \right) - S^{OT},$$

где C_f^T – стоимость единицы выпущенной продукции f-й марки кабеля в период T, тыс. руб; I_f^T – объем выпуска кабельной продукции f-й марки за период времени T, т; S_{fn}^{ET} – стоимость расходов на электроэнергию E на единицу выпущенной продукции для выпуска f-й марки кабеля на n-й заготовительной линии за период времени T, тыс. руб.; S_{fnm}^{MT} – стоимость расходов на материалы M по m-му материалу на единицу выпущенной продукции для производства f-й марки кабеля в n-й линии за период времени T, тыс. руб.; Z_{fkn}^T – суммарная заработная плата рабочих, осуществлявших производство кабельной продукции, на единицу выпущенной продукции f-й марки k-й бригады на n-й линии предприятия за период времени T, тыс. руб.; S^{OT} – общие издержки, которые предприятие несет в течение всего периода T.

Издержки S^{OT} компания несет даже тогда, когда не производит готовую продукцию совсем. В состав указанных издержек можно включить амортизацию, заработную плату, налоговые и страховые отчисления, плата за аренду, кредитные платежи и др.

Учитывая положения, описанные в п. 2 настоящей статьи, предприятие в условиях стабильного экономического развития стремится наладить выпуск увеличенного объема продукции, учитывая интересы и потребности рынка. Решение о выпуске дополнительных объемов продукции принимается руководством после тщательного и оперативного анализа рынка, иначе возможно изготовление такого количества продукции, которое в результате не только будет невозможно реализовать, но и могут возникнуть трудности с его хранением.

Поэтому выражение для Π^T можно записать в следующем виде:

$$\Pi^T = \sum_f \left(\left(I_{f\,dog}^T + I_{f\,dop}^T \right) C_f^T \right) - \sum_f \left(\left(I_{f\,dog}^T + I_{f\,dop}^T \right) \sum_n \left(S_{fn}^{ET} + \sum_m S_{fnm}^{MT} + \sum_k Z_{fkn}^T \right) \right) - S^{OT},$$

где $I_{f\,dog}^T$ – объем выпуска f-й марки кабеля, который предприятие должно произвести в соответствии со всеми своими договорными обязательствами за период времени T, т; $I_{f\,dop}^T$ – дополнительный объем выпуска f-й марки кабеля за период времени T, т.

Т.к. издержки S^{OT} являются, как правило, неизменными в период времени T, то S^{OT} может рассматриваться как постоянная $S^{OT} = \text{const}$, поэтому в исследуемой целевой функции это слагаемое можно опустить.

Использование целевой функции Φ_1 возможно при соблюдении следующих дополнительных условий:

1. Технологические ограничения:

Каждая n-я линия должна 30 дней в квартал непрерывно выпускать кабельную продукцию марки ВВГ-П 3х2,5. Все остальное время указанного периода может выпускаться кабельная продукция других марок;

2. Объем выпускаемого кабеля каждой марки f не должен быть меньше необходимых объемов производства, предусмотренных имеющимися договорными обязательствами.

3. Выпуск дополнительных объемов кабеля должен коррелировать с текущим спросом на рынке.

4. Объем производимого кабеля, которое предприятие не сможет реализовать покупателям, должен вписываться в возможности завода по складированию и хранению.

5. Общий объем произведенной кабельной продукции в целом за период времени T не может превышать суммарной потенциальной мощности всех производственных агрегатов предприятия: $\sum_f I_f^T \le \sum_f \sum_n O_{fn} t$, где O_{fn} – среднесуточная производительность n-й линии заготовки кабельной продукции марки f; t – количество суток в периоде T.

4. Целевая функция предприятия в условиях колебаний экономических показателей в связи введением ограничительных мер

Целевую функцию работы предприятия по производству кабельной продукции, в условиях ограничений или нестабильности рынка, можно представить как функцию минимизации потерь, которые предприятие вынуждено нести из-за трудностей с реализацией готовой продукции. Так же, из-за необходимости непрерывной работы предприятия за ним сохраняется практически все основные статьи расходов. Следовательно, представим функцию в виде $\Phi_2 = \min F\left(S^{ET}, S^{MT}, Z^T, P_d^T\right)$, где S^{ET} – затраты предприятия на электроэнергию на период времени T, тыс. руб.; S^{MT} – затраты предприятия на сырье и материалы за период времени T, тыс. руб.; Z^T – заработная плата рабочих предприятия за период времени T, тыс. руб.; P_d^T – дополнительные нерегламентированные потери промышленного предприятия за период времени T, вызванные внешними воздействиями кризисного характера, тыс. руб. [1].

В результате целевая функция Φ_2 будет иметь вид:

$$\Phi_2 = \sum_f \left(I_f^T \sum_n \left(S_{fn}^{ET} + \sum_m S_{fnm}^{MT} + \sum_k Z_{fkn}^T \right) \right) + P_d^T \to \min .$$

Представленная выше функция Φ_2 поможет свести к минимуму расходы предприятия на электроэнергию, материалы, заработную плату и дополнительные потери, связанные с колебаниями рыночных показателей.

Оценка деятельности предприятия за период больший T осуществляется по формуле:

$$\Phi_2^0 = \sum_T \left(S^{ET} + S^{MT} + Z^T + P_d^T \right) \to \min .$$

Уменьшить значение общих потерь можно, сократив объемы выпуска готовой продукции.

Заключение

Показатели спроса и цены на готовую продукцию могут сильно колебаться в периоды экономического спада, это может привести к затяжным проблемам, или даже попаданием в долговую яму. Поэтому руководство предприятия должно быть готово оперативно реагировать на различного рода кризисные проявления.

Сложность управления предприятием кроется еще и в сложности представления краткосрочных прогнозов динамики складывающихся ситуаций, поэтому такие параметры как список стейкхолдеров, проблемное месиво являются динамическими.

Это требует разработки АИС поддержки технологии системного анализа для непрерывного мониторинга проблемного месива, выявления и решения вновь выявляемых проблем [10].

Таким образом, для обеспечения работы предприятия в условиях нестабильности рынка, системные аналитики должны предоставлять управляющему персоналу компании варианты экономически обоснованных бизнес-прогнозов, с помощью которых

предприятие сможет адаптироваться к нестабильной ситуации. Управляющий персонал должен тщательно проанализировать предложенные варианты и выбрать наиболее приемлемые из них с помощью АИС поддержки технологии системного анализа [11–14].

Выбираемые решения должны позволять предприятию существенно уменьшить затраты на электроэнергию, сырье и материалы, необходимые для производства готовой продукции, а также на заработную плату рабочим, обеспечивая при этом выполнение долгосрочных договорных обязательств в полном объеме и сокращение дополнительных нерегламентированных потерь кризисного характера в соответствии с выбранной целевой функцией.

ЛИТЕРАТУРА

1. *Максимов А., Коренная К., Логиновский* Адаптивное управление промышленной корпорацией в условиях неопределенности (на примере ферросплавных производств). Проблемы теории и практики управления. –2012. – № 9-10. – С. 145-150.

2. *Тарасенко Ф.П.* Прикладной системный анализ: учебное пособие. 2-е изд., перераб. и доп. М.: КНОРУС, 2017. 322 с.

3. *Коваленко А.В., Уртенов М.Х., Трахова С.Ш.* Математическое моделирование финансово-экономического кризиса на предприятии с использованием канонических катастроф складки и сборки. Научный журнал КубГАУ, №63(09), 2010.

4. Силовые кабели - https://sts-kabel.ru/catalog/kabelno_provodnikovaya_produktsiya/kabeli_silovye/prop_mechanic_protection_net/?PAGEN_1=2

5. "Электрик Инфо" - онлайн журнал про электричество. Теория и практика. http://electrik.info/device/1367-kak-delayut-kabeli-i-provoda.html

6. *Коренная К.А., Логиновский О.В., Максимов А.А.* Математическая модель оптимизации работы экспортно-ориентированного предприятия в условиях мировой финансово-экономической нестабильности – Вестник ЮУрГУ, №23, 2012.

7. *Коренная К.А.* Информационно-ресурсное обеспечение управления промышленными предприятиями на основе прогнозно-адаптивного подхода Информационные ресурсы России. – 2012. - №2. – С. 16-20.

8. *Зайцева Т.Ю.* Кризис и маркетинговый аудит предприятия: проблемы управления предприятием в условиях экономического кризиса. Российское предпринимательство, 2010, №2 (1).

9. *Акаев А.А., Коротаев А.В.* Прогноз и моделирование кризисов и мировой динамики. М.: ЛКИ, 2010.

10. *Тарасенко В.Ф.* Моделирование систем менеджмента. Томск: Изд-во ТУСУР, 2018. 172с.

11. *Тарасенко В.Ф., Чернышова Ю.В., Жуковский О.И.* Прикладной системный анализ и технологии проектирования программного обеспечения // Информационные технологии в науке, управлении, социальной сфере и медицине : сб. науч. тр. IV Междунар. науч. конф., 5-8 дек. 2017 г. Томск: Изд-во ТПУ, 2017. С. 147-149.

12. *Степин В.С., Тарасенко В.Ф.* Автоматизированная информационная система поддержки технологии прикладного системного анализа // Новые информационные технологии в исследовании сложных структур. Материалы девятой Российской конференции с международным участием. Томск: Изд-во НТЛ, 2012. С. 120.

13. *Тарасенко В.Ф., Степин В.С.* Современный рынок АИС для подбора персонала при приеме на работу //Актуальные проблемы модернизации управления и экономики: российский и зарубежный опыт : материалы Всероссийской научно-практической конференции (с международным участием), Томск, 29-30 марта 2012 г. Томск: Изд-во Том. ун-та, 2012. С. 371-377.

14. *Кононенко Р.Ф., Тарасенко В.Ф.* Автоматизация процесса решения проблем с использованием технологии прикладного системного анализа // Инноватика-2014: сб. материалов X Всероссийской школы-конференции студентов, аспирантов и молодых ученых с международным участием. 23–25 апреля 2014 г. г. Томск, Россия. Томск: Изд-во Том. ун-та, 2015. С. 448-450.

ЧИСЛЕННОЕ СРАВНЕНИЕ ПРОЦЕДУР ОЦЕНИВАНИЯ ПАРАМЕТРА АВТОРЕГРЕССИИ С АДДИТИВНЫМ ШУМОМ

Пупков А.В.

Томский государственный университет

andrewpupkov@gmail..com

Введение

Случайные процессы, окружающие нас, могут быть аппроксимированы стохастическими моделями, которые представляют из себя совокупность детерминированной и случайной компонент. В качестве детерминированной части выступает некоторая функциональная зависимость между наблюдаемыми элементами процесса. В свою очередь случайная компонента является некоторой непредсказуемой величиной, которая,

например, позволяет не учитывать некоторые зависимости процесса, которые недоступны исследователю. В частности, большой популярностью пользуются процессы авторегрессионного типа (AR) за свою простоту в применении и интерпретируемость. Написано множество работ, связанных с оцениванием процессов авторегрессии. Некоторые из них основаны на использовании ядерных методов оценивания [1], другие на модификациях метода наименьших квадратов (МНК). В [2] рассмотрена ситуация, когда наблюдаемый процесс AR(1) обладает бесконечной дисперсией шума (тяжелые хвосты) и предложена модификация метода МНК, учитывающая высокие вероятности выброса. В [3] исследуется процесс AR-типа, который обладает случайной аддитивной составляющей при значениях оцениваемого параметра, и описывается последовательная модификация МНК.

Часто на практике истинные значения процесса не доступны исследователю. Вместо этого наблюдается сигнал с аддитивной помехой. В таком случае возникает множество проблем, одной из которых является смещение оценок МНК. Одним из способов решения данной проблемы может выступать фильтрация поступающего сигнала. В [4] рассматривается задача состоятельного оценивания авторегрессионного сигнала в случае дополнительно накладываемого шума, путем использования рекурсивного метода фильтрации, использующего два параллельно работающих фильтра. Ключевая идея алгоритма состоит в применении одного из фильтров для оценивания ненаблюдаемого значения процесса, при параллельном оценивании параметра другим. Результат работы первого фильтра улучшает выводы второго и наоборот. В [5] предложена процедура оценивания авторегрессионого сигнала, искаженного дополнительным шумом, через численное решение методом Ньютона специального уравнения, зависящего от параметра, имеющего смысл дисперсии аддитивной ошибки.

Целью этой работы ставится задача провести численное сравнение процедуры идентификации процесса AR(p), загрязненного аддитивной шумовой составляющей, предложенной В.В. Коневым, с другим алгоритмом, разработанным для аналогичной цели, в частности, с алгоритмом, представленным в [5].

1. Описание исследуемого процесса

Рассмотрим процесс с аддитивной ошибкой, накладываемой на наблюдения, вида

$$x_k = X_{k-1}^{'}\theta + \varepsilon_k,\ y_k = x_k + \eta_k,\ k = 1,2,\ldots, \qquad (1)$$

где $X_{k-1} = \left(x_{k-1},\ldots,x_{k-p}\right)'$, $\theta = \left(\theta_1,\ldots,\theta_p\right)'$, $\{\varepsilon_k\}$ и $\{\eta_k\}$ – последовательности шумом с нулевыми математическими ожиданиями $E\varepsilon = E\eta = 0$ и дисперсиями $E\varepsilon_k^2 = \sigma^2$, $E\eta_k^2 = \Delta^2$ соответственно. Предполагается, что начальное значения процесса не зависит от последовательностей $\{\varepsilon_k\}$ и $\{\eta_k\}$. Штрих (') обозначает транспонирование. Ставится задача оценить неизвестный параметр θ по значениям наблюдаемой последовательности $\{y_k\}$, представляющей из себя зашумленный процесс авторегрессии.

2. Последовательная модификация оценки Юла – Уокера

Сначала рассмотрим последовательную оценку параметра авторегрессии, искаженного аддитивным шумом, которую предложил В.В. Конев. Представим множество индексов в виде суммы множеств

$$T(n) = \{2p+1, 2p+2,\ldots,n\} = \bigcup_{i=1}^{p+1} T_i(n),\ ,\ T_i(n) = \{k = (p+1)j + p + i - 1 : j = 1,2,\ldots; k \le n\}.$$

Введем последовательность моментов остановки

$$\tau_l^{(i)}(h)=\inf\left\{n>\tau_{l-1}^{(i)}(h):\sum_{k=\tau_{l-1}^{(i)}(h)+1}^{n}\chi_{k\in T_i(n)}y_{k-p-l}^2\ge h\right\},\ l=1,...,p,$$
$$\tau(h)=\max_{1\le i\le p+1}\tau_p^{(i)}(h). \qquad (2)$$

Определим коэффициенты по следующим формулам

$$\sum_{k=\tau_{l-1}^{(i)}+1}^{\tau_l^{(i)}(h)-1}\chi_{\left\{k\in T_i\left(\tau_l^{(i)}(h)\right)\right\}}y_{k-p-l}^2+\alpha_l^{(i)}(h)y_{\tau_l^{(i)}(h)-p-l}^2=h\ ,\ \tau_0^{(i)}(h)=2p\ ,$$

$$\beta_{l,k}^{(i)}(h)=\begin{cases}1,\ \text{если}\ \tau_{l-1}^{(i)}(h)<k<\tau_l^{(i)}(h),\\ \alpha_l^{(i)}(h),\ \text{если}\ k=\tau_l^{(i)}(h),\end{cases}\ ,\ \gamma_{l,k}=\sum_{i=1}^{p+1}\mu_{k,l}^{(i)}\sqrt{\beta_{l,k}^{(i)}(h)}\ ,\ \mu_{k,l}^{(i)}=\chi_{\{k\in T_i(k)\}}\chi_{\left(\tau_{l-1}^{(i)}(h),\tau_l^{(i)}(h)\right]}(k)\ .$$

Определение 1. Для каждого $h>0$ последовательный план оценивания вектора параметров $\theta=\left(\theta_1,...,\theta_p\right)'$ в модели (1) задается парой $\left(\tau(h),\theta^*(h)\right)$, где $\tau(h)$ – длительность процедуры, определенная в (2), $\theta^*(h)=\left(\theta_1^*(h),...,\theta_p^*(h)\right)'$ – вектор оценок, задаваемый формулами $\theta^*(h)=G^{-1}(h)\vartheta(h)$. Здесь $\vartheta=\left(\vartheta_1(h),...,\vartheta_p(h)\right)'$ – p-мерный вектор с координатами $\vartheta_l(h)=\sum_{k=2p+1}^{\tau(h)}\gamma_{l,k}y_{k-p-l}y_k$, $1\le l\le p$, $G(h)$ – матрица, размера $p\times p$, с элементами $\left\langle G(h)\right\rangle_{l,s}=\sum_{k=2p+1}^{\tau(h)}\gamma_{l,k}y_{k-p-l}y_{k-s}$.

В следующем разделе рассмотрим метод, предложенный в [5].

3. Обобщенный МНК

Приведем алгоритм, описанный в [5].

Шаг 0. Рассчитать величины

$$\hat R=\frac{1}{N}\sum_{k=1}^{N}Y_{k+1}Y_{k+1}^{'}\ ,\ \hat r=\frac{1}{N-1}\sum_{k=1}^{N-1}Y_{k+1}y_{k+1}\ ,\ \beta^{(0)}=\kappa\min\lambda(\hat R)\ ,\ 0.5\le\kappa\le0.99\ ,$$

где $Y_{k+1}=\left(y_k,y_{k-1},...,y_{k-p+1}\right)'$, $\lambda(x)$ – собственные значения матрицы x.

Шаг 1. Найти $\hat\theta^{(k)}$ из уравнения $\left(\hat R-\beta^{(k)}I_p\right)\hat\theta^{(k)}=\hat r$, I_p – ед. матрица размерности $p\times p$.

Шаг 2. Вычислить $\sigma^2(k)=\left(\hat r[1]-\hat r[2]'\hat\theta^{(k)}\right)\Big/\hat\theta_1^{(k)}$, где $\hat r[1]=\frac{1}{N-1}\sum_{k=1}^{N-1}y_k y_{k+1}$, $\hat r[2]=\frac{1}{N-2}\sum_{k=1}^{N-2}Y_{k+1}y_{k+2}$.

Шаг 3. Рассчитать $f\left(\beta^{(k)}\right)=\beta^{(k)}+\sigma^2(k)+\hat r'\hat\theta^{(k)}-\frac{1}{N}\sum_{k=1}^{N}y_k^2$, $f'\left(\beta^{(k)}\right)=1+\left\|\hat\theta^{(k)}\right\|$, $\|a\|^2=a'a$.

Шаг 4. Найти $\beta^{(k+1)}=\beta^{(k)}-f\left(\beta^{(k)}\right)\Big/f'\left(\beta^{(k)}\right)$ и $\hat\theta^{(k+1)}=\theta_{LS}+\beta^{(k+1)}\hat R^{-1}\hat\theta^{(k)}$, где $\theta_{LS}=\hat R^{-1}\hat r$.

Шаг 5. Если выполняются условия

$$\left|\frac{\beta^{(k+1)}-\beta^{(k)}}{\beta^{(k)}}\right|<\delta_1\ \text{и}\ \frac{\left\|\hat\theta^{(k+1)}-\hat\theta^{(k)}\right\|}{\left\|\hat\theta^{(k)}\right\|}<\delta_2\ ,$$

то действие алгоритма заканчивается, иначе $k := k+1$ и на **Шаг 1.** Здесь $\delta_1, \delta_2 > 0$ – пороги точности.

Замечание 1. Смысл величины $\beta^{(k)}$ объясняется в теореме 1 [5].

4. Сравнение численных результатов

Приведем результаты численного моделирования. Рассмотрим усредненные оценки параметра процесса AR(1) с аддитивной ошибкой. Усреднение производилось по 1000 реализаций процесса. Объем выборки для последовательной модификации оценки Юла – Уокера рассчитывался по формуле (2) при значении $h = 1000$. Объем выборки $\bar{\tau}$ обобщенного МНК вычислялся путем усреднения 1000 реализаций момента остановки (2). В табл. 1 указаны усредненные оценки параметра AR(1) при различных значениях параметра θ, в случае гауссовских шумов при $\sigma^2 = 1$ и $\Delta^2 = 0.25$.

Таблица 1

Усредненные оценки параметра AR(1): h = 1000, $\varepsilon_k \sim N(0,1)$, $\eta_k \sim N(0,0.25)$, к = 0.7, $\delta_1 = \delta_2$ =0.01, 1000 повторений

θ	$\theta^*(h)$	$\hat{\theta}$	$\bar{\tau}$	θ	$\theta^*(h)$	$\hat{\theta}$	$\bar{\tau}$
–1	–0.998	–0.970	85.7	0.1	0.126	0.219	1313.1
–0.9	–0.896	–0.901	316.1	0.2	0.195	0.259	1277.7
–0.8	–0.799	–0.807	550.5	0.3	0.294	0.336	1225.1
–0.7	–0.697	–0.710	747.8	0.4	0.395	0.427	1146.9
–0.6	–0.598	–0.616	911.2	0.5	0.498	0.521	1043
–0.5	–0.500	–0.521	1042.3	0.6	0.600	0.618	912.2
–0.4	–0.402	–0.430	1148.1	0.7	0.700	0.709	750.1
–0.3	–0.298	–0.334	1227.9	0.8	0.798	0.807	552.2
–0.2	–0.197	–0.258	1278.3	0.9	0.895	0.901	316.7
–0.1	1.073	–0.198	1311	1	0.999	0.974	85.9
0	–3.321	–0.331	1322.9	–	–	–	–

В табл. 2 приведены результаты оценивания при экспоненциальном распределении шумов. В частности, шум процесса ε_k имеет экспоненциальное распределение с параметром $\lambda = 1$, центрированным на величину своего математического ожидания. Аддитивный шум имеет, центрированное на математическое ожидание, экспоненциальное распределение с параметром $\lambda = 2$.

Таблица 2

Усредненные оценки параметра AR(1): h = 1000, $\varepsilon_k \sim \exp(1) - 1$, $\eta_k \sim \exp(2) - 0.5$, к = 0.7, $\delta_1 = \delta_2$ =0.01, 1000 повторений

θ	$\theta^*(h)$	$\hat{\theta}$	$\bar{\tau}$	θ	$\theta^*(h)$	$\hat{\theta}$	$\bar{\tau}$
–1	–0.999	–0.974	87.1	0.1	0.060	0.215	1338.8
–0.9	–0.897	–0.904	316.2	0.2	0.190	0.268	1309.5
–0.8	–0.797	–0.807	561.7	0.3	0.295	0.335	1253.3
–0.7	–0.697	–0.713	757.4	0.4	0.395	0.433	1171.7
–0.6	–0.599	–0.615	921.3	0.5	0.501	0.520	1064.2
–0.5	–0.499	–0.518	1059.1	0.6	0.601	0.616	928.2
–0.4	–0.397	–0.425	1167.7	0.7	0.699	0.714	755.6
–0.3	–0.294	–0.336	1251	0.8	0.798	0.807	554.7
–0.2	–0.192	–0.270	1318.8	0.9	0.898	0.903	313.5
–0.1	–0.069	–0.217	1346.5	1	0.998	0.976	86.7
0	–58.322	–0.285	1354.7	–	–	–	–

Заключение

В настоящей работе представлены алгоритмы идентификации процесса авторегрессии AR(p) в случае зашумленных наблюдений. Проведено численное сравнение рассматриваемых алгоритмов для процесса AR(1) . Приведены результаты моделирования для случая симметричного и несимметричного распределения ошибок. Результаты

моделирования говорят о том, что последовательная модификация оценок Юла – Уокера, для процесса AR(1), дает в среднем более близкие к истинному значению результаты.

ЛИТЕРАТУРА

1. *Васильев В.А.* Непараметрическая идентификация авторегрессий / В. А. Васильев, Г. М. Кошкин // Теория вероятн. и ее примен. 1998. Том 43, № 43, С. 577—588.

2. *Марков А.С.* Оценивание параметра авторегрессии с бесконечной дисперсией шума // Автоматика и телемеханика. 2009. №1. С. 104 — 118.

3. *Кашковский Д.В.* Последовательная идентификация параметров авторегрессии со случайными коэффициентами // Вестник Томского государственного университета: Серия «Информатика. Кибернетика. Математика». 2006. №293. С. 105 — 109.

4. *Labarre D.* Consistent estimation of autoregressive parameters from noisy observations based on two interacting Kalman filters / D. Labarre, E. Grivel, Y. Berthoumieu, E. Todini, M. Najim // Signal Processing. 2006. V.86. P. 2863 – 2876.

5. *Xia Y.* Novel parameter estimation of autoregressive signals in the presence of noise / Y. Xia, W. X. Zheng // Automatica. 2015. V. 62. P. 98 – 105.

КОМБИНИРОВАННАЯ ОЦЕНКА В КЛАССИФИКАЦИИ КАРДИОГРАММ

Скрипин С.В., Дмитриев Ю.Г.
Томский государственный университет
skripinsv@mail.ru, dmit70@mail.ru

Введение

Проблема доврачебной экспресс-диагностики сердечных заболеваний актуальна как в наше время, так и в будущем: в условиях внезапных приступов, удалённости пунктов медобслуживания, снижения количества врачей-кардиологов, а также, их квалификации. Широкое распространение мобильных и компактных диагностических устройств, позволяющих измерять пульс, давление и короткие кардиограммы, открывает возможности упрощённого доврачебного экспресс-анализа кардиограмм. Это позволит сократить время получения предварительного диагноза, определить, вызван ли приступ нарушением сердечной деятельности или другими причинами, своевременно принять решение о приёме лекарств, обращения к врачу или вызова скорой помощи. В условиях обработки большого количества кардиограмм доврачебный экспресс-анализ позволит разгрузить врачей-кардиологов.

Особенностью указанных устройств является короткое время измерения кардиограмм, вплоть до нескольких секунд. В этих условиях важно надёжное распознавание заболеваний сердца, а так же, отдельных видов этих заболеваний. В математическом описании – это задача отнесения наблюдений к одному из известных классов заболеваний (классификации) при наличии обучающих (эталонных) выборок малого объема, которые решаются различными методами, в том числе методами математической статистики (задача дискриминантного анализа). Она решается с использованием различных математических моделей, как из класса параметрических, так и непараметрических. Повысить качество решений может применение комбинированных оценок классификации. В указанных условиях выгоднее использовать адаптивные комбинированные оценки [1], модифицируя их способом, предложенным в [2].

1. Оценки задачи классификации

Представим эталонную выборку, содержащую наблюдения из k классов, с количеством наблюдений в каждом классе $n_1,...,n_k$. Обозначим независимые наблюдения случайного вектора размерности m из класса $t=\overline{1,k}$ как $\mathbf{X}_t=\left\{\mathbf{X}_{1,t},...,\mathbf{X}_{nt,t}\right\}$, с неизвестной плотностью вероятности $f_t(\mathbf{x})$, $\mathbf{x}\in R^m$. В условиях малых объемов выборок n_t и отсут-

ствия информации о классах априорные вероятности отнесения новых наблюдений в каждый класс можно принять равными. Тогда в соответствии с оптимальными процедурами классификации (Байеса) параметрическую оценку отнесения наблюдения в точке **x** в класс *t* представим в виде

$$J\left(\mathbf{x};\hat{\boldsymbol{\theta}}_t\right)=\frac{f_t\left(\mathbf{x};\hat{\boldsymbol{\theta}}_t\right)}{\sum_{j=1}^{k} f_j\left(\mathbf{x};\hat{\boldsymbol{\theta}}_j\right)},\ t=\overline{1,k}\ .$$

Здесь $f_t\left(\mathbf{x};\hat{\boldsymbol{\theta}}_t\right)$ – параметрическая оценка плотности вероятности в классе *t*, $\hat{\boldsymbol{\theta}}_t$ – оценка вектора параметров θ_t некоторой размерности в классе *t*. Если каждый класс описывается *m*-мерной нормальной плотностью, то функция плотности вероятности класса *t* в точке **x** может быть представлена параметрической оценкой вида [3]

$$f_t\left(\mathbf{x};\hat{\boldsymbol{\theta}}_t\right)=(2\pi)^{-m/2}\left|\mathbf{V}_t\right|^{-1/2}\exp\left(-0.5d^2\left(\mathbf{x};\hat{\boldsymbol{\theta}}_t\right)\right),\ t=\overline{1,k}\ ,$$

где $\left|\mathbf{V}_t\right|$ – детерминант матрицы ковариаций из класса *t*, элементы которой получены по эталонной выборке методом максимального правдоподобия, $d^2\left(\mathbf{x};\hat{\boldsymbol{\theta}}_t\right)$ – обобщенный квадрат расстояния от точки **x** до центра группы наблюдений из класса *t*. Если используется объединенная матрица ковариаций, его можно записать

$$d^2\left(\mathbf{x};\hat{\boldsymbol{\theta}}_t\right)=\left(\mathbf{x}-\hat{\bar{\mathbf{n}}}_t\right)^{\mathrm{T}}\left(\mathbf{V}_t\right)^{-1}\left(\mathbf{x}-\hat{\bar{\mathbf{n}}}_t\right),\ t=\overline{1,k}\ ,$$

где $\hat{\bar{\mathbf{n}}}_t=\left(\hat{\bar{n}}_{1,t},...,\hat{\bar{n}}_{m,t}\right)$ – оценка вектора математических ожиданий компонент случайного вектора $\mathbf{X}_t$, $\mathbf{V}_t$ – объединенная матрица ковариаций, полученная методом максимального правдоподобия.

Непараметрическую оценку отнесения наблюдения в точке **x** в класс *t* представим

$$\hat{J}_t(\mathbf{x})=\frac{f_{n_t,t}(\mathbf{x})}{\sum_{j=1}^{k} f_{n_j,t}(\mathbf{x})},\ t=\overline{1,k}\ .$$

Здесь $f_{n_t,t}(\mathbf{x})$ – непараметрическая оценка плотности вероятности следующего вида:

$$f_{n_t,t}\left(x^{(1)},...,x^{(m)}\right)=\frac{1}{n_t H_t}\sum_{i=1}^{n_t}\prod_{j=1}^{m} K\left[\left(x^{(j)}-X_{i,t}^{(j)}\right)\Big/h_t^{(j)}\right],$$

где $K\left(u\right)$ – заданное ядро (некоторая функция плотности вероятности); $h_t^{(j)}$ – параметры масштаба в классе *t*, удовлетворяющие требованиям: при количестве наблюдений класса $n_t\to\infty$, $h_t^{(j)}\to 0$, $n_t H_t\to\infty$, $h_t^{(j)}=h_t^{(j)}\left(n_t\right)$.

2. Комбинированная оценка классификации

Рассмотрим комбинированную оценку классификации наблюдения в точке **x** в класс *t* следующего вида:

$$\hat{J}\left(\mathbf{x};\lambda_t\right)=\lambda_t J\left(\mathbf{x};\hat{\boldsymbol{\theta}}_t\right)+\left(1-\lambda_t\right)\hat{J}_t(\mathbf{x})\ ,\ t=\overline{1,k}\ ,\qquad(1)$$

где λ_t – весовой коэффициент в классе *t*.

Можно предложить различные способы выбора наилучших оценок λ_t в (1), задавая различные критерии оптимальности. Выбор наилучших оценок $\lambda_t(\mathbf{x})$, по критерию минимума ошибок классификации всех наблюдений из обучающей (эталонной) выборки объема $N=n_1+...+n_k$, рассматривался в [2,4]. Воспользуемся предложенной в них оценкой коэффициента $\lambda_t(\mathbf{x})$ следующего вида:

$$\hat{\lambda}_t(\mathbf{x}) =$$

$$= \frac{\sum_{i=1}^{N}\left(Y_{i,t} - \hat{J}_t(\mathbf{X}_i)\right)\left(\hat{J}_t(\mathbf{X}_i) - J\left(\mathbf{X}_i;\hat{\boldsymbol{\theta}}_t\right)\right)\left(1-1/\left(1+a_{1,t}(\mathbf{X}_i)b_{1,t}\right)\right)\left(1-1/\left(1+a_{2,t}(\mathbf{X}_i)b_{2,t}\right)\right)}{\sum_{i=1}^{N}\left(\left(\hat{J}_t(\mathbf{X}_i) - J\left(\mathbf{X}_i;\hat{\boldsymbol{\theta}}_t\right)\right)\left(1-1/\left(1+a_{1,t}(\mathbf{X}_i)b_{1,t}\right)\right)\left(1-1/\left(1+a_{2,t}(\mathbf{X}_i)b_{2,t}\right)\right)\right)^2} \times, \quad (2)$$

$$\times\left(1-1/\left(1+a_{1,t}(\mathbf{x})b_{1,t}\right)\right)\left(1-1/\left(1+a_{2,t}(\mathbf{x})b_{2,t}\right)\right),$$

где $Y_{i,t}$ – числовая величина априорной доли отнесения каждого наблюдения обучающей выборки к своему и чужим классам [4]

$$Y_{i,j}\left(t \mid \mathbf{X}_i\right) = \begin{cases} 1-\alpha > 0, & j \in t, \\ \alpha/(k-1) > 0, & j \notin t, \end{cases}, \; i = \overline{1,N}, \; j = \overline{1,k}, \; t = \overline{1,k}, \; \alpha < 1,$$

выражения $a_{1,t}(u)$, $a_{2,t}(u)$ имеют вид

$$a_{1,t}(u) = \frac{\left(\left(\hat{J}_t(u)\right)^2 - \hat{J}_t(u)\right)\frac{1}{NH_t}\sum_{i=1}^{N}Y_{i,t}\prod_{j=1}^{m}K\left[\left(u^{(j)} - X_i^{(j)}\right)/h_t^{(j)}\right]}{\frac{1}{NH_t}\sum_{i=1}^{N}\left(Y_{i,t}\right)^2\prod_{j=1}^{m}K\left[\left(u^{(j)} - X_i^{(j)}\right)/h_t^{(j)}\right] - \hat{J}_t(u)\frac{1}{NH_t}\sum_{i=1}^{N}Y_{i,t}\prod_{j=1}^{m}K\left[\left(u^{(j)} - X_i^{(j)}\right)/h_t^{(j)}\right]},$$

$$a_{2,t}(u) =$$

$$= \frac{\frac{1}{NH_t}\sum_{i=1}^{N}\left(Y_{i,t}\right)^2\prod_{j=1}^{m}K\left[\left(u^{(j)} - X_i^{(j)}\right)/h_t^{(j)}\right] - \hat{J}_t(u)\frac{1}{NH_t}\sum_{i=1}^{N}Y_{i,t}\prod_{j=1}^{m}K\left[\left(u^{(j)} - X_i^{(j)}\right)/h_t^{(j)}\right]}{NH_t\left(\Delta_{0,t}(\mathbf{x})\left(1-1/\left(1+(NH_t)^{3/4}\left(\Delta_{0,t}(\mathbf{x})\right)^2\right)\right)\right)^2} +$$

$$+ \frac{\left(\hat{J}_t(u)\right)^2 - \hat{J}_t(u)\frac{1}{NH_t}\sum_{i=1}^{N}Y_{i,t}\prod_{j=1}^{m}K\left[\left(u^{(j)} - X_i^{(j)}\right)/h_t^{(j)}\right]}{NH_t\left(\Delta_{0,t}(\mathbf{x})\left(1-1/\left(1+\left(NH_t\right)^{3/4}\left(\Delta_{0,t}(\mathbf{x})\right)^2\right)\right)\right)^2},$$

$$\Delta_{0,t}(u) = \frac{1}{NH_t}\sum_{i=1}^{N}Y_{i,t}\prod_{j=1}^{m}K\left[\left(u^{(j)} - X_i^{(j)}\right)/h_t^{(j)}\right] - J\left(u;\hat{\boldsymbol{\theta}}_t\right)\frac{1}{NH_t}\sum_{i=1}^{N}\prod_{j=1}^{m}K\left[\left(u^{(j)} - X_i^{(j)}\right)/h_t^{(j)}\right],$$

коэффициенты $b_{1,t}$, $b_{2,t}$ подбираются из выборочного критерия оптимальности

$$Q\left(b_{1,t}, b_{2,t}\right) = \sum_{i=1}^{N}\left[Y_{i,t} - J\left(\mathbf{X}_i;\hat{\lambda}_t(\mathbf{x})\right)\right]^2 \to \min_{b_{1,t},\,b_{2,t}}.$$

Используя оценку (2), получим адаптивную комбинированную оценку вида

$$J\left(\mathbf{x};\hat{\lambda}_t(\mathbf{x})\right) = \hat{\lambda}_t(\mathbf{x})J\left(\mathbf{x};\hat{\boldsymbol{\theta}}_t\right) + \left(1-\hat{\lambda}_t(\mathbf{x})\right)\hat{J}_t(\mathbf{x}), \; t = \overline{1,k}. \quad (3)$$

Наблюдение в точке (**x**) отнесём к классу, в котором оценка (3) получит максимальное значение. Проверим свойства оценки вида (3) с оценкой коэффициента $\lambda_t(\mathbf{x})$ вида (2) в имитационном эксперименте.

3. Условия имитационного эксперимента

Для простоты интерпретации результатов имитационный эксперимент проведен с одномерной величиной $x \in R^1$. Сравнение свойств оценок выполнено со случайным выбором точек X_i из двух одномерных случайных величин X_1 и X_2, представляющих два класса, $k = 2$. Для имитационного эксперимента выбраны следующие условия:

- для моделирования данных выборки генерировались с наблюдениями из двух одномерных случайных величин X_1 и X_2 с нормальным законом распределения;

- случайные величины X_1 и X_2 имели одинаковые значения среднеквадратического отклонения $\sigma_1 = \sigma_2 = 1,0$ и математические ожидания, равные $EX_1 = 0,0$, $EX_2 = 2,0$;
- для формирования области перемешивания наблюдений из разных классов расстояние между математическими ожиданиями случайных величин выбрано равным двум среднеквадратическим отклонениям, $EX_2 - EX_1 = 2,0$;
- отбор наблюдений из разных классов в исходные выборки выполнялся с вероятностью 0,5;
- сравнение качества оценок в условиях малых объемов выборки выполнено для объемов выборок в диапазоне $N = 5 - 95$ с интервалом $\delta = 5$;
- для каждой выборки соблюдалось условие: в выборке должно присутствовать не менее двух наблюдений из каждого класса;
- для каждого значения N числовые результаты были получены по серии выборок количеством $K = 90000$ (с одинаковым объемом наблюдений N в каждой выборке).

Сравнение качества оценок классификации у трех моделей $J_1 = J\left(x;\hat{\boldsymbol{\theta}}_t\right)$, $J_2 = \hat{J}_t\left(x\right)$, $J_3 = J\left(x;\hat{\lambda}_t\left(x\right)\right)$ проводилось по следующим критериям.

Критерии, вычисленные для каждой выборки.

1. Оценка вероятности безошибочной классификации наблюдений эталонной выборки по всем классам

$$S_{1,j} = \frac{1}{N}\sum_{t=1}^{k} u_t\,,\ \ j = \overline{1,3}\,, \tag{4}$$

где u_t – количество правильно классифицированных наблюдений эталонной выборки из класса t; j – номер модели классификации, указанной выше.

2. Оценка вероятности классификации наблюдений эталонной выборки в точке X_i в свой класс t – для случая равных значений величин $S_{1,j}$

$$S_{2,j} = \frac{1}{N}\sum_{i=1}^{N} \hat{p}_i\left(t \mid X_i \in t\right),\ \ j = \overline{1,3},\ t = \overline{1,k}\,. \tag{5}$$

Критерии, вычисленные по серии K=90000 выборок.

3. Среднее оценок вероятностей безошибочной классификации наблюдений эталонной выборки по всем классам

$$Q_{1,j} = \frac{1}{K}\sum_{i=1}^{K}\left(S_{1,j}\right)_i\,,\ \ j = \overline{1,3}\,. \tag{6}$$

4. Среднее оценок вероятностей классификации наблюдений эталонной выборки в свой класс

$$Q_{2,j} = \frac{1}{K}\sum_{i=1}^{K}\left(S_{2,j}\right)_i\,,\ \ j = \overline{1,3}\,. \tag{7}$$

Выражения (4–7) нормированы в диапазоне [0,1].

Оценки непараметрической модели классификации для каждого наблюдения выборки получены по методу скользящего эксперимента (с исключением из выборки наблюдения i в точке X_i, в которой оценивается классификация). При указанных условиях моделирования в имитационном эксперименте обеспечивается преимущество (в смысле качества) оценкам параметрической модели $J\left(x;\hat{\boldsymbol{\theta}}_t\right)$.

В общем случае параметры масштаба $h_t^{(j)}$ в непараметрической и комбинированной моделях могут выбираться с использованием разных критериев оптимизации и мо-

гут *различаться*. С целью упрощения условий имитационного эксперимента параметры масштаба в указанных моделях выбраны *одинаковыми*.

4. Результаты эксперимента

Результаты эксперимента по критериям (9) и (10) представлены на графиках рис. 1, 2. Кривые на графиках рис. 1, 2 статистически значимо различаются на уровне значимости меньше 0,05, за исключением областей пересечения кривых.

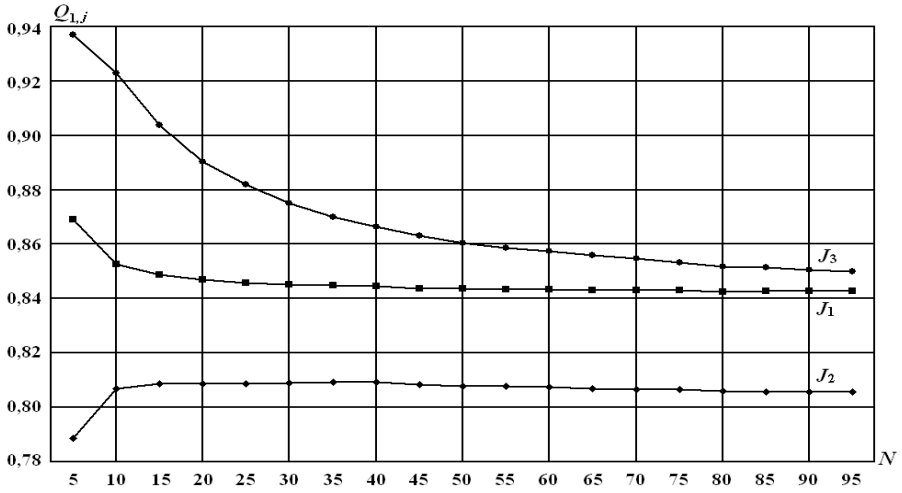


Рис. 1. Среднее оценок вероятностей безошибочной классификации $Q_{1,j}$

На рис. 1 представлены средние оценок классификации по критерию $Q_{1,j}$. По горизонтальной оси указан объем выборки N, по вертикальной – величина среднего оценок вероятностей безошибочной классификации $Q_{1,j}$.

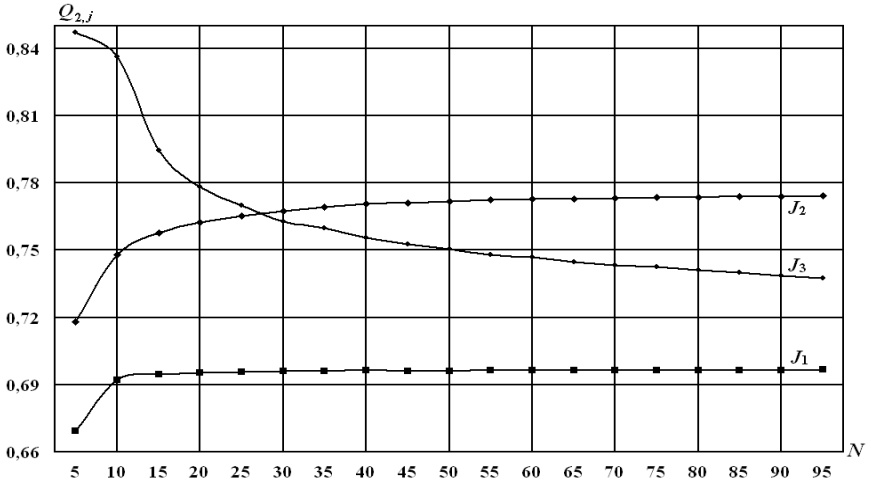


Рис. 2. Среднее оценок вероятностей классификации в свой класс $Q_{2,j}$

На рис. 2 представлены средние оценок классификации по критерию $Q_{2,j}$. По горизонтальной оси указан объем выборки N, по вертикальной – величина среднего оценок вероятностей классификации в свой класс $Q_{2,j}$.

5. Выводы по эксперименту

По результатам эксперимента можно сделать следующие выводы.

1. Оценки модели классификации:

- *параметрической* J_1 дают меньшие, чем у непараметрической J_2 ошибки классификации (рис. 1, кривая J_1). При этом значения среднего оценок вероятностей классификации в свой класс (рис. 2, кривая J_1) самые низкие среди рассматриваемых моделей. Следовательно, качество оценок параметрической модели J_1 в имитационном эксперименте высокое;
- *непараметрической* J_2 дают наибольшие ошибки классификации среди представленных моделей (рис. 1, кривая J_2). Значения среднего оценок вероятностей классификации в свой класс самые высокие среди рассматриваемых моделей (рис. 2, кривая J_2). Это свидетельствует о невысоком качестве оценок непараметрической модели J_2 в имитационном эксперименте;
- *комбинированной* J_3 дают наименьшие ошибки классификации (рис. 1, кривая J_3) среди рассматриваемых моделей. Значения среднего оценок вероятности классификации в свой класс (рис. 2, кривая J_3) при $N < 30$ наибольшие среди рассматриваемых моделей, а при $N > 30$ плавно приближаются к значениям параметрической модели J_1. Следовательно, оценки модели J_2 являются наилучшими в имитационном эксперименте. Это подтверждает заявленные свойства комбинированных оценок классификации (1).

2. Оценки комбинированной модели J_3 дают наибольший выигрыш в области малых значений объемов выборки, вплоть до $N < 10$ наблюдений, несмотря на неоптимальную оценку параметров масштаба $h_i^{(j)}$ для комбинированной модели. При увеличении объемов выборки преимущество комбинированных оценок (1) сохраняется.

3. В эксперименте подтверждены правильность выбора наилучшей оценки весового коэффициента $\lambda_t(\mathbf{x})$ при заданных критериях качества классификации.

6. Применение в анализе кардиограмм

6.1. Задачи распознавания кардиограмм

Для решения проблем распознавания кардиограмм применим комбинированные (непараметрические) оценки классификации, предложенные выше. Применение таких оценок требует представления исходных данных в виде таблиц типа объект (наблюдение)-признак. Кардиограммы представлены в виде двумерного графика типа время-амплитуда. Это требует введения понятия «наблюдение» для кардиограмм и создания производных переменных для них. При этом необходимо учесть, что основную информацию о заболеваниях сердечно-сосудистой системы содержит форма кривой кардиограммы и её привязка к шкале времени. Перечислим основные задачи:

– формирование наблюдений и производных переменных, описывающих элементы графиков кардиограмм для перехода от кардиограмм к таблицам типа объект-признак;
– определение эффективности применения производных переменных для использования в качестве входных данных в процедурах дискриминантного анализа;
– определение возможности применения комбинированных (непараметрических) оценок для решения задач распознавания кардиограмм;
– определение качества (разрешаюшей способности) решения указанной задачи выбранными моделями.

6.2. Формирование наблюдений

На рис. 3 представлена кардиограмма с диагностированным заболеванием ишемической болезни сердца (ИБС). Самые высокие пики кардиограммы обозначены как зубцы *R*. Выделим из графика кардиограммы все «полные» удары сердца в интервалах от зубца R_i до зубца R_{i+1}. На рис. 3 таких интервалов 11. Будем предварительно считать одним наблюдением интервал от зубца R_i, включая его, до зубца R_{i+1}, исключая его. В дальнейшем описании уточним понятие наблюдения.

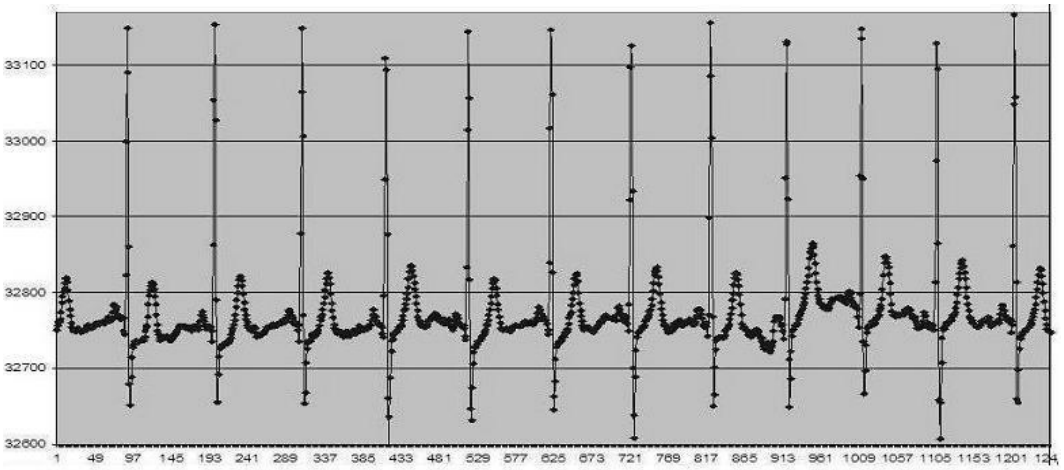


Рис. 3. Кардиограмма заболевания ИБС.

6.3. Формирование производных переменных

Внутри наблюдения данные представленны в виде графика типа время-амплитуда (T,V). Здесь: переменная T – горизонтальная ось времени измерений через равные интервалы времени, переменная V – вертикальная ось амплътуды сигнала, измеренного в точках между интервалами в условных значениях. В интервале от зубца R_i до зубца R_{i+1}, в точках времени $T_1,...,T_n$, будут измерены значения амплитуды $V_1,...,V_n$. Сформируем p производных переменных $C_1,...,C_p$ следующим способом. Каждая переменная C_i может использовать от 1 до r значений амплитуды $V_1,...,V_r$ из диапазона $1,...,n$, в позициях, не обязательно подряд расположенных, и связанных некоторой функциональной зависимостью: $C_i = f_i(V_1,...,V_r)$. Для C_i можно предложить множество способов выбора значений $V_1,...,V_r$ и видов функциональных зависимостей $f_i()$.

Зафиксируем выбранные значения $V_1,...,V_r$ и виды $f_i()$ для каждой переменной C_i. Переменные C_i описывают части графика одного удара сердца в количестве $p < n$. Будем считать одним наблюдением последовательный набор из p переменных C_i для одного удара сердца. При этом последовательность формирования переменных C_i в каждом интервале удара сердца должна быть одинаковой.

6.4. Формирование классов

На рис. 4 представлена кардиограмма с диагностированным заболеванием АВ блокада.

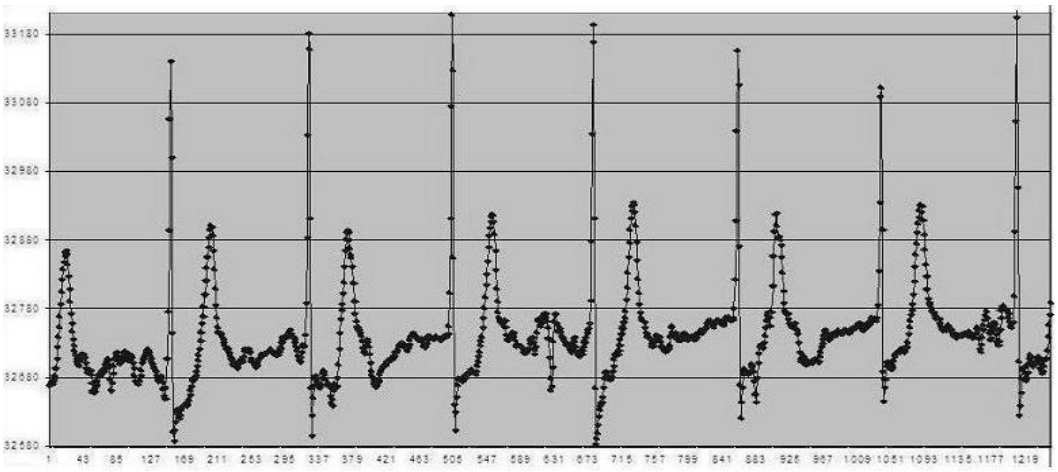


Рис. 4. Кардиограмма .заболевания АВ блокада

В связи с замедленным биением сердца на кардиограмме можно выделить 6 «полных» ударов сердца. Сформируем 6 наблюдений для этого заболевания аналогично выше описанному. Определим, что один класс наблюдений должен представлять один вид заболевания сердца. Получим 11 наблюдений из класса «заболевание ИБС» и 6 наблюдений из класса «заболевание АВ блокада». Добавим к каждому наблюдению переменную C_{p+1} с информацией о номере класса. В результате сформируем таблицу объект-признак (обучающую выборку) как входные данные для процедуры классификации. Она будет содержать 17 наблюдений из двух классов: $\mathbf{X}_{i,t} = \{C_{1,t},...,C_{p+1,t}\}$, $i = \overline{1,17}$, $t = 1,2$.

6.5. Условия и результаты экспериментов

1. Условия. Сформируем выборки с наблюдениями из $p = 5$ и $p = 10$ производных переменных для оценки влияния p на качество классификации. Критерием качества оценок выберем минимум суммы ошибок классификации по всем классам. При поиске наилучших оценок комбинированной модели применим процедуру полного перебора состава и сочетания переменных C_i в модели от 1 до p для параметрической и непараметрической оценок.

2. Результаты.

2.1. Эксперимент при $p = 5$. Наилучшие оценки представлены в табл. 1.

Таблица 1

№ набл.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Класс 1	0.971527	0.952801	0.981557	0.966093	0.999873	0.999882	0.971106	0.974481	0.150657
Класс 2	0.028473	0.047199	0.018443	0.033907	0.000127	0.000118	0.028894	0.025519	0.849343
№ кл.	1	1	1	1	1	1	1	1	2
№ набл.	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	
Класс 1	0.974003	0.968015	0.116492	0.744210	0.120610	0.376028	0.337415	0.084027	
Класс 2	0.025997	0.031985	0.883508	0.255790	0.879390	0.623972	0.662585	0.915973	
№ кл.	1	1	2	1	2	2	2	2	

В модели оставлены 4 переменные: 1, 3, 4, 5. Качество классификации по оценкам вероятности отнесения: в класс 1 $Q_1 = 0.882353$, в класс 2 $Q_2 = 0.831248$, по всем классам $Q_{12} = 0.811106$. В классе 1 – одна ошибка классификации из 11 наблюдений (набл. 9), в классе 2 – одна ошибка классификации из 6 наблюдений (набл. 13).

2.2. Эксперимент при $p = 10$. Наилучшие оценки представлены в табл. 2.

Таблица 2

№ набл.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Класс 1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
Класс 2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
№ кл.	1	1	1	1	1	1	1	1	1
№ набл.	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	
Класс 1	1.000000	1.000000	0.048945	0.132407	0.100668	0.002252	0.005056	0.013772	
Класс 2	0.000000	0.000000	0.951055	0.867593	0.899332	0.997748	0.994944	0.986228	
№ кл.	1	1	2	2	2	2	2	2	

В модели оставлены 9 переменных: 1, 2, 4, 5, 6, 7, 8, 9, 10. Качество классификации по оценкам вероятности отнесения: в класс 1 $Q_1 = 1.000000$, в класс 2 $Q_2 = 0.982171$, по всем классам $Q_{12} = 0.986115$. Ошибок классификации нет.

6.6. Выводы по экспериментам

При $p = 10$ достигнута безошибочная классификация наблюдений в классы заболеваний. Дальнейшее увеличение количества производных переменных C_i из-за их высокой «эффективности», может привести к ошибкам классификации.

Отбрасывание «неэффективных» C_i при $p = 5$ и при $p = 10$ указывает, что формирование производных переменных требует более глубокого анализа.

Заключение

1. В работе представлена адаптивная комбинированная оценка классификации вида (3) с оценкой весового коэффициента $\lambda_t(\mathbf{x})$ вида (2) и продемонстрирована ее работоспособность при конечных объемах выборок N, а также её свойства получать оценки лучше, чем у каждой из построенных моделей (параметрической или непараметрической). Результаты имитационного эксперимента демонстрируют преимущества ее оценок даже при $N < 10$.

2. Применение в анализе кардиограмм позволяет заключить следующее:

- описание графиков кардиограмм производными переменными является работоспособным в задаче классификации при диагностике кардиозаболеваний;
- использование схемы «один удар сердца в кардиограмме – одно наблюдение для задачи классификации» также является работоспособным. Это позволяет работать с короткими кардиограммами (даже в несколько ударов сердца);
- необходим поиск «наилучших» вариантов описания кардиограмм производными переменными, а также подбора их количества в моделях классификации;
- необходима углублённая работа по формированию «эталонных» выборок заболеваний, т.к. внутри общего названия «заболевание ...» могут быть выявлены подклассы, имеющие свои особенности. Это необходимо и в случае различий в кардиограммах у разных больных, разных возрастов, различий пола и т.д.. Необходимо особо обрабатывать ситуации смешанных заболеваний (более одного вида кардиозаболеваний у одного больного).

ЛИТЕРАТУРА

1. *Дмитриев Ю.Г., Скрипин С.В.* О комбинированной оценке вероятности безотказной работы по полной выборке // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. Томск: Изд-во Томского государственного университета, 2012. – № 4(21). – С. 32–38.

2. *Скрипин С.В.* Модификация комбинированной оценки в задаче дискриминиции // Новые информационные технологии в исследовании сложных структур: Тезисы доклада IX Российской конференции с международным участием. – Томск, 5–8 июня 2012. – Томск: Изд-во НТЛ, 2012. – С. 109.

3. *Андерсон Т.В.* Введение в многомерный статистический анализ. – М.: Физматгиз, 1963. – 500 с.

4. *Скрипин С.В.* Свойства комбинированной оценки в задаче классификации наблюдений // Известия Томского политехнического университета. – 2009. – Т. 314. – № 5. – С. 21–26.

АНАЛИЗ ПРОДАЖ ТОВАРА С УЧЕТОМ АНОМАЛЬНОГО СПРОСА

Тюменцева Л.С., Зенкова Ж.Н.
Томский государственный университет
tl442@mail.ru, thankoffjean@mail.ru

Введение

Для составления эффективной стратегии управления запасами, любой компании необходимо проводить адекватную классификацию номенклатуры [1]. В этом может помочь достаточно известный и доступный метод ABC-XYZ-анализа [2,3]. Он используется для группировки ассортимента предприятия в зависимости от выручки или непосредственно прибыли от продаж относительно ABC-метода. Касательно XYZ-части принято рассматривать объем продаж.

Однако, не стоит забывать, что данная классификация проводится на основе продаж, на которые влияет спрос потребителей. Зачастую на практике можно наблюдать случаи, когда единичное событие, например, период празднования Международного женского дня 8 марта) приводит к резкому кратковременному колебанию спроса, выбросу, которое быстро стабилизируется. При этом всплеск этот бывает настолько силён, что может оказывать весьма продолжительное влияние на результаты классификации ABC-XYZ анализа, вплоть до следующего подобного события. Это существенно искажает результаты классификации и приводит к потере оптимальности системы управления запасами предприятия.

Поэтому необходимо учитывать любой аномальный спрос (как в большую, так и в меньшую сторону потребления) товара для составления адекватной стратегии по управления запасами. Для этого были разработаны различные модификации методов классификации, в зависимости от выбора главного критерия

1. Постановка задачи

Предлагается рассмотреть ABC-XYZ-анализ в целом, а также его модификацию с использованием критерия Граббса [4], позволяющего идентифицировать наличие выбросов [5]. А далее предполагается применение изученного метода на реальных данных (табл. 1) до и после исключения выбросов.

Таблица 1

Исходные данные по продаже товаров за 7 месяцев

Наименование	Средняя цена за период, руб./ед.	Продажи, ед./мес.						
		Январь	Февраль	Март	Апрель	Май	Июнь	Июль
Один	98	1202	1003	1123	989	1075	1102	1098
Два	250	119	127	165	116	122	107	93
Три	194	322	360	345	312	325	339	328
Четырег	350	100	110	90	100	105	95	100
Пять	200	103	112	106	98	89	109	112
Шесть	100	98	95	93	114	108	120	119
Семь	30	280	295	287	279	293	300	288
Восемь	286	20	25	23	19	18	15	16
Девять	100	50	60	70	30	0	40	60
Десять	230	18	18	19	20	19	22	21

2. Классический ABC-XYZ-анализ

Алгоритм классического ABC-анализа [5] можно представить следующим образом:

1. Сортируем товары по доходности R_i, $i=\overline{1,N}$ за указанный период n в порядке убывания.
2. Рассчитываем удельный вес [3] для каждого товара, $i=\overline{1,N}$:

$$d_i = \frac{R_i}{\sum_{i=1}^{N} R_i} \cdot 100\% .$$

3. Рассчитываем нарастающий итог [3] для каждого товара, $i=\overline{1,N}$: $S_i = S_{i-1} + d_i$, $S_0 = 0$.
4. Группируем товары: $S_i \le 80\%$ – относим товар к группе A, $80\% < S_i \le 95\%$ – относим товар к группе B, $S_i > 95\%$ – относим товар к группе C.

Таблица 2

ABC-классификация на реальных данных

Наименование	**Доход R_i за период, тыс. руб.**	**d_i, %**	**S_i, %**	**ABC-классификация**
Один	744 016	36.5	36.5	A
Три	452 214	22.2	58.7	A
Четыре	245 000	12.0	70.7	A
Два	212 250	10.4	81.1	B
Пять	145 800	7.2	88.3	B
Шесть	74 700	3.7	92.0	B
Семь	60 660	3.0	95.0	C
Восемь	38 896	1.9	96.9	C
Десять	31 510	1.6	98.5	C
Девять	31 000	1.5	100.0	C
Общий доход	**2 036 046**	–	–	–

XYZ-анализ базируется на вычислении коэффициента вариации по следующей формуле: $v = \frac{\sigma}{\bar{x}} \cdot 100\%$, где $\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$ – среднее значение уровня спроса на запас или продаж за n рассматриваемых периодов, $x_i = (x_1, x_2, ..., x_n)$ является выборкой, состоящей из объем продаж продукта, n – размер выборки, в нашем случае – количество месяцев в заданном периоде, $\sigma = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} \left(x_i - \bar{x}\right)^2}$ – среднеквадратическое отклонение (СКО), оцененное по выборке.

Классификация проводится по значению коэффициента вариации, рассчитанного для каждого товара, по следующему правилу: $v \le 10\%$ – относим товар к группе X, $10\% < v \le 30\%$ – относим товар к группе Y, $v > 30\%$ – относим товар к группе Z.

Таблица 3

XYZ-классификация на реальных данных

Наименование	**$\bar{x}$**	**σ**	**v**	**XYZ-классификация**
Один	1 084.57	72.61	6.69%	X
Два	121.29	22.29	18.38%	Y
Три	333.00	16.12	4.84%	X

Наименование	$\bar{x}$	σ	v	XYZ-классификация
Четыре	100.00	6.45	6.45%	X
Пять	104.14	8.36	8.02%	X
Шесть	106.71	11.43	10.71%	Y
Семь	288.86	7.73	2.68%	X
Восемь	19.43	3.60	18.52%	Y
Девять	44.29	23.70	53.53%	Z
Десять	19.57	1.51	7.72%	X

Объединив результаты двух методов, составим матрицу ABC-XYZ-анализа, которой уже можно пользоваться для эффективного управления запасами представленных товаров.

Таблица 4

ABC-XYZ-анализ на реальных данных

Наименование	ABC-классификация	XYZ-классификация
Один	A	X
Два	B	Y
Три	A	X
Четыре	A	X
Пять	B	X
Шесть	B	Y
Семь	C	X
Восемь	C	Y
Девять	C	Z
Десять	C	X

3. Модифицированный ABC-XYZ-анализ

Внимательно изучив табл. 1, можно отметить слегка завышенную мартовскую продажу товара «Два» из группы BY и отсутствие майской продажи товара «Девять» из группы CZ. Данные отклонения могли существенно повлиять на общие результаты. Для большей ясности и точности вычислений рассмотрим товары «Два» и «Девять» как отдельные выборки $X_2 = (119,127,165,116,122,107,93)$ и $X_9 = (50,60,70,30,0,40,60)$ при $n = 7$.

Таким образом, сформулируем и рассмотрим статистическую гипотезу относительно товара «Два»:

$$H_0\text{: } X_{2,3} = 165 \text{ является выбросом,}$$

альтернативная же гипотеза

$$H_1\text{: } X_{2,3} = 165 \text{ не является выбросом.}$$

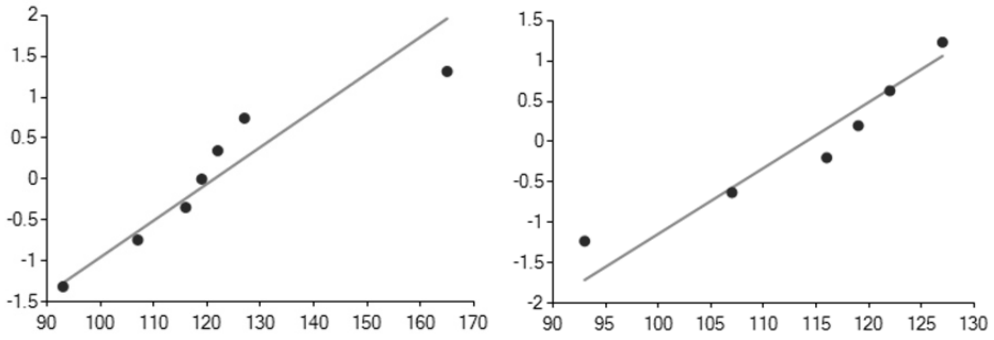
Для проверки данной гипотезы воспользуемся критерием Граббса [4,6] для определения наличия верхнего выброса, а именно: $G = \frac{1}{\sigma} \cdot \max\limits_{i=1,n} |x_i - \bar{x}|$.

В нашем случае максимальным значением является $X_{2,3} = 165$, $\bar{x} = 121.29$, $\sigma = 22.29$, а значит $G = 1.961$. Табулированные значения процентных точек критерия Граббса приведены в [6]. В нашем случае при $n = 7$ табличное значение $G^* = 1.828$. Получается, что $G = 1.961 > G^* = 1.828$, следовательно, мы принимаем гипотезу H_0 и считаем, что $X_{2,3} = 165$ является выбросом.

Теперь необходимо проверить выборку без выброса на нормальность. Это можно сделать с помощью критерия Шапиро – Уилка [4] по следующей формуле:

$$W = \frac{\left(\sum_{i=1}^{n/2} a_i \cdot \left(x_{n-i+1} - x_i\right)\right)^2}{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}.$$

В нашем случае $W = 0.922$, а значение $p\text{-}value = 0.522$, что при заданном уровне значимости $\alpha = 0.1$ позволяет утверждать о нормальности полученной выборки, т.к. $p\text{-}value = 0.522 > \alpha = 0.1$.



a) b)

Рис. 1 Нормальные вероятностные графики продаж товара «Два» a) с выбросом; b) без выброса

Далее необходимо рассчитать урезанное среднее. Для этого упорядочим исходный набор по возрастанию, а затем отбросим слева и справа по одному значению, что позволит исключить выброс: $\bar{x}^* = \frac{1}{n-2} \cdot \sum_{i=2}^{n-1} x_i$. Следовательно, урезанное среднее значение $\bar{x}^* = 118.2$, а 7-месячный доход от товара «Два» с учетом усечений будет составлять 206 850 тыс. руб., а не 212 250 тыс. руб., как было подсчитано ранее.

Данные преобразование влечет за собой изменение общего дохода, удельного веса и нарастающего итога, что требует пересмотр распределения по группам в ABC-XYZ-анализе. Для этого необходимо рассчитать новое среднеквадратическое отклонение (СКО), рассматривая исходные упорядоченные значения выборки и новое среднее, используя следующую формулу: $\sigma^* = \sqrt{\frac{1}{n-3} \cdot \sum_{i=2}^{n-1}\left(x_i - \bar{x}^*\right)^2}$. Тогда коэффициент вариации будет рассчитываться по формуле: $\nu^* = \frac{\sigma^*}{\bar{x}^*} \times 100\%$.

В нашем случае, для товара «Два» среднеквадратическое отклонение (СКО) $\sigma^* = 7.463$, а значит коэффициент вариации теперь не 18.38%, а 6.31%. В итоге получаем, что определение товара «Два» в группу BY было не совсем верным решением. Целесообразней и правильней определить его в группу BX.

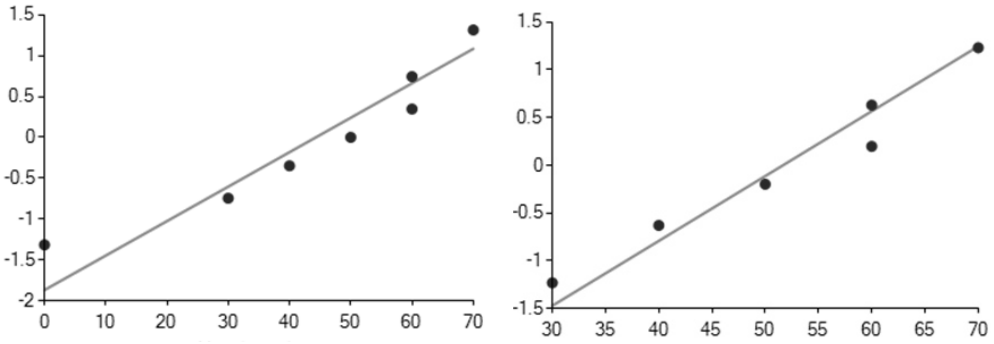
Теперь выдвинем гипотезу относительно товара «Девять», имеющее распределение $X_9 = (50, 60, 70, 30, 0, 40, 60)$:

H_0: $X_{9,5} = 0$ является выбросом,

альтернативная же гипотеза

H_1: $X_{9,5} = 0$ не является выбросом.

В это раз для проверки гипотезы воспользуемся критерием Граббса для определения наличия нижнего выброса: $G = \frac{1}{\sigma} \cdot \min_{i=1,n} |x_i - \bar{x}|$. Очевидно, что минимальным значением является $X_{9,5} = 0$, среднее значение выборки $\bar{x} = 44.29$, среднеквадратическое отклонение $\sigma = 23.70$, а значит $G = 1.868$. Табличный критерий Граббса остается тем же, т.к. $n = 7$: $G^* = 1.828$. Получается, что $G = 1.868 > G^* = 1.828$, следовательно, мы принимаем гипотезу H_0 и считаем, что $X_{9,5} = 0$ является выбросом.



c) d)

Рис. 2 Нормальные вероятностные графики продаж товара «Девять» c) с выбросом; d) без выброса

После проверки выборки на нормальность по критерию Шапиро – Уилка без выброса ($W = 0.958$, $p\text{-}value = 0.804 > \alpha = 0.1$), выполняем те же расчеты, что и для товара «Два».

Тогда усеченное среднее значение для товара «Девять» $\bar{x}^* = 48$, 7-месячный доход составит уже 33 600 тыс. руб., что на 2 600 больше дохода с учетом выброса. Это приведет к увеличению удельного веса и уменьшению нарастающего итога. Среднеквадратическое отклонение (СКО) уменьшится на 10.662 условные единицы: $\sigma^* = 13.038$, а коэффициент вариации почти вдвое уменьшится и составит 27.16%.

Таким образом, после исключения всех выбросов, повторно проводим ABC-XYZ-анализ, который будет в большей мере соответствовать действительности (табл. 5).

Отметим, что далеко не всегда выбросы являются очевидными, поэтому при небольшом количестве товаров удобно рассчитать коэффициент Граббса для всех товаров, чтобы выявить как нижние, так и верхние выбросы (табл. 6). По получившейся таблице можно легко выявить их наличие, сравнив полученный коэффициент с табличным значением (в данном случае для $n = 7$).

Таблица 5

ABC-XYZ-классификация на реальных данных без выбросов

Наименование	**Доход R_i за период, тыс. руб.**	**ABC-классификация**	**XYZ-классификация**
Один	744 016	A	X
Три	452 214	A	X
Четыре	245 000	A	X
Два	206 850	B	X
Пять	145 800	B	X
Шесть	74 700	B	Y
Семь	60 660	B	X
Восемь	38 896	C	Y
Девять	33 600	C	Y
Десять	31 510	C	X
Общий доход	**2 033 246**	–	–

Таким образом, сравнив каждое из полученных чисел с табличным значением, в нашем случае $G^* = 1.828$, можно выявить выбросы, искажающие результат ABC-XYZ-анализа, что влияет на эффективность метода. Можно отметить, что мы исключили все имеющиеся в данной выборке сомнительные значения, препятствующие составлению адекватной ABC-XYZ –модели.

Таблица 6

Расчетные значение критерия Граббса

Наименование	G_{max}	G_{min}
Один	1.617	1.316
Два	1.961	1.269
Три	1.674	1.302
Четыре	1.549	1.549
Пять	0.940	1.812
Шесть	1.163	1.200
Семь	1.441	1.275
Восемь	1.548	1.231
Девять	1.085	1.868
Десять	1.606	1.039

Можно сделать вывод, что при использовании ABC-XYZ-анализа необходимо обращать внимание не только на завышенный спрос потребителя, связанный с различными факторами, но и на заниженный спрос на товар (или полное его отсутствие), т.к. это влияет на эффективность деятельности предприятия. Изменение системы управления запасов даже одного товара может значительно снизить затраты компании и привести к увеличению дохода

Также данный метод позволяет избежать возникновения эффекта хлыста, представляющий собой ситуацию, когда незначительное изменение спроса конечного потребителя способно привести к значительным отклонениям в планах цепи управления запасами. Одним из преимуществ рассмотренного метода является простота. Его реализация не требует наличие специального программного обеспечения, что позволяет сэкономить некоторые денежные и трудовые ресурсы компании. Данный вопрос актуален для малого и среднего бизнеса.

Заключение

В работе были рассмотрены такие методы управления запасами, как ABC- и XYZ-анализ с различными модификациями, а также совмещение данных классификационных методов, называющееся ABC-XYZ-анализом. В частности, указанные методы были применены к реальным данным о продаже некоторых товаров, взяв за определяющий критерий доход от продаж каждого товара за указанный период.

Для статистической проверки гипотезы о наличии выбросов был использован критерий Граббса. С его помощью было выявлено два выброса, являющихся последствием аномального спроса потребителя. Чтобы получить более адекватные результаты анализа, был проведен перерасчет. Усеченное среднее значение выборки было необходимо для вычисления усеченного стандартного отклонения и расчета нового дохода и коэффициента вариации. После исключения из выборки имеющихся выбросов, результаты перерасчета классификации значительно отличаются от полученных ранее.

ЛИТЕРАТУРА

1. *Зенкова Ж.Н.* Логистический подход в управлении предприятием. Уч.-метод. комплекс. – Томск: Томский государственный университет, 2012.

2. *Бадокин О.В., Лукинский В.В., Малевич Ю.В., Степанова А.С., Шульженко Т.Г.*; под общ. и научн. ред. Лукинского В. С. Управление запасами в цепях поставок. Учеб. пособие. – СПб.: СПбГИЭУ, 2010. – 372 с.

3. *Стерлигова А.Н.* Управление запасами в цепях поставок. Учебник. / Высшее образование. – М.: ИНФРА-М, 2008. – 430 с.

4. *Козбарь А.И.* Прикладная математическая статистика для инженеров и научных работников. М.: ФИЗМАТЛИТ, 2006. 816 с.

5. *Zenkova Z.N., Kabanova T.V.* The ABC-XYZ Analysis Modified for Data with Outliers. Proceeding GOL'2018 The 4th IEEE International Conference on Logistics Operations Management. April 10-12, 2018, Le Havre, France. P. 63-68 DOI: 10.1109/GOL.2018.8378073.

6. *Заляжных В.В.* Статистическая обработка результатов испытаний (измерений) [Электронный ресурс]. – Электрон. дан. – URL: https://arhiuch.ru/lab4a.html.

V. ПРИКЛАДНАЯ ТЕОРИЯ ВЕРОЯТНОСТЕЙ, КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

ИМИТАЦИОННАЯ МОДЕЛЬ УПРАВЛЯЕМОГО ЗАНЯТИЯ РЕСУРСОВ СИСТЕМЫ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ ИЗ ДВУХ ГРУПП ВИРТУАЛЬНЫХ МАШИН

Бурцева С.А., Хакимов А.А., Григорьева Т.В., Власкина А.С., Кочеткова И.А.
Российский университет дружбы народов
sofiyaburtseva@gmail.com, khakimov-aa@rudn.ru, tgrigoryeva97@gmail.com, vlaskina-as@rudn.ru, igudkova@sci.pfu.edu.ru

Введение

Постоянный рост числа потребителей услуг сотовой связи ставит перед поставщиками услуг все новые и новые задачи. Необходимо не просто обслужить клиентов, но и сделать это оптимально, используя собственные ресурсы, не пожертвовав качеством оказываемых услуг и не превысив некоторые пределы допустимого времени ожидания услуг со стороны потребителей. В качестве одного из подходов к решению данной проблемы можно рассматривать деление потребителей на некоторые классы, выделяя потребителей по тому или иному признаку и резервируя ресурс под каждый тип клиентов. Однако, подобный подход выявляет новые проблемы. Необходимо понять, как распределять обслуживающие устройства между типами пользователей. В данной статье рассмотрена система массового обслуживания с двумя параллельными очередями, имеющая в своём распоряжении две разнородные группы обслуживающих приборов. Рассматриваются различные возникающие в процессе работы системы события, зависящие от степени загрузки очереди и приборов. Система анализируется в устойчивом состоянии. Для заданной структуры затрат задаются правила, устанавливающие политику оптимального распределения обслуживающих устройств между очередями. Кроме того, формулируются и рассматриваются основные показатели эффективности системы.

1. Системная модель

Рассмотрим два сервера, на каждом из которых развёрнуто N_1 и N_2 виртуальных машин соответственно. Предположим, что первый сервер более быстродействующий, чем второй, а время обслуживания запросов на виртуальных машинах данных серверов распределено по экспоненциальному закону с интенсивностями μ_1 и μ_2, $\mu_1 \geq \mu_2$.

В данную систему пользователи посылают запросы, которые с некоторой вероятностью разделяются на две группы, первого и второго типа соответственно ($j = \{1,2\}$). Будем считать, что запросы первого типа заданы пуассоновским потоком с частотой λ_1, а второго типа – потоком с частотой λ_2 соответственно. Если в момент поступления запроса все виртуальные машины заняты, запрос принимается в очередь. Причём запросы каждого типа образуют отдельную, выделенную под данный тип очередь [1]. Обе очереди имеют бесконечную ёмкость и характеризуются количеством находящихся в них в конкретный момент времени запросов. Схематическое изображение системы представлено на рис. 1.

Стоимость, рассматриваемая далее в работе – это стоимость затрат на эксплуатацию и простой оборудования. Тот факт, что запрос находится в ожидании обслуживания, негативно сказывается на прибыли провайдера. Следовательно, нам необходимо минимизировать данные потери.

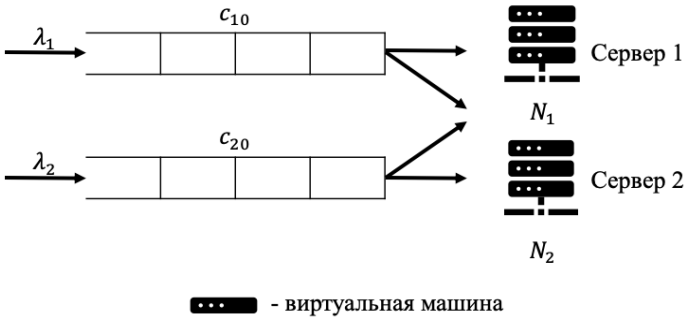


Рис. 1. Схема модели

Таблица 1

Характеристики системы

Параметры серверов	
Параметры	**Описание**
k	номер сервера, $k = \{1,2\}$
N_1	количество виртуальных машин на первом сервере
N_2	количество виртуальных машин на втором сервере
μ_1	интенсивность экспоненциального распределения времени вычисления блоков данных в виртуальных машинах первого сервера
μ_2	интенсивность экспоненциального распределения времени вычисления блоков данных в виртуальных машинах второго сервера
d_{11}	количество запросов первого типа на виртуальных машинах первого сервера
d_{12}	количество запросов второго типа на виртуальных машинах первого сервера
d_{21}	количество запросов первого типа на виртуальных машинах второго сервера
d_{22}	количество запросов второго типа на виртуальных машинах второго сервера
Параметры пользователей и очередей	
j	Тип поступившего запроса $j = \{1,2\}$
λ_1	Интенсивность поступление запросов первого типа
λ_2	Интенсивность поступление запросов второго типа
q_1	Количество запросов в очереди 1
q_2	Количество запросов в очереди 2
Стоимость	
c_{j0}, ($c_{j0} > 0$)	стоимость ожидания в очереди для заявки типа j
c_{j1}	стоимость обслуживания заявки типа j на приборе соответствующей группы
c_{j2}, ($c_{j2} > c_{j1}$)	стоимость обслуживания заявки типа j на приборе альтернативной группы

Когда запрос ожидает обслуживания, провайдер в среднем тратит c_{10} и c_{20} на содержание запроса в первой и второй очередях соответственно. Если запрос обслуживается на своем сервере и выделенной ему виртуальной машине, то стоимость обслуживания равна для заявок первого типа – c_{11}, для второго – c_{21}. Если же обслуживание запроса происходит не на своём сервере и выделенной ему виртуальной машине, затраты на эксплуатацию будут больше. Примем следующее обозначение для стоимости обслуживания запросов первого и второго типа не на своём сервере: c_{12} и c_{22} соответственно.

Считается, что, используя облачные вычисления, потребители стараются минимизировать свои расходы – на строительство собственных центров обработки данных, закупку серверного и сетевого оборудования, аппаратных и программных решений по обеспечению непрерывности и работоспособности – эти расходы "поглощаются" провайдером облачных услуг. Обслуживать запросы, требующие большой мощности и электроэнергии, на малопроизводительных серверах будет убыточно для провайдера. Однако из-за простоя оборудования или долгой обработки посланного запроса потери будет серьёзнее, поэтому такие случаи в системе допускаются. Средние затраты скла-

дываются из затрат на включение/отключение виртуальной машины; затрат, связанных с пребыванием запроса в очереди и затрат на эксплуатацию серверов.

Потребители определенного типа могут обслуживаться без каких-либо штрафных санкций на сервере соответствующей группы. Подключение клиентов к вычислительным машинам другой группы разрешено, но только в соответствии с политикой распределения обслуживающих устройств и за дополнительную плату. Прерывание обслуживания клиента недопустимо, все вычислительные машины должны быть заняты, если в очереди есть заявки.

Политика занятости приборов предполагает следующее.

Представим поступление заявки как событие **A**. Возможны следующие варианты работы системы при возникновении данного события:

A1. Если при поступлении заявки в систему, нет очереди и прибор, соответствующий данному типу заявки, свободен, то заявка поступает на данный прибор, который начинает обслуживание.

A2. Если этот прибор занят, то заявка переходит на обслуживание на приборе другого типа, если тот свободен.

A3. В противных случаях заявка встает в очередь своего типа.

После того, как некоторая заявка была обслужена, необходимо рассмотреть возникающее событие с точки зрения состояния системы в этот момент. Поскольку поступившая заявка покидает сервер, он может взять на обслуживание новую. В связи с разными затратами на содержание запросов различного типа на вычислительных машинах разнородных групп, необходимо рассмотреть данное событие для каждого возможного варианта.

Представим завершение обслуживания заявки на первом сервере как событие **B**.

B1. Если после обслуживания заявки обе очереди свободны, то виртуальная машина простаивает.

B2. В случае, когда оказалась занята очередь первого типа и при этом вторая пустует, то на освобождённую виртуальную машину первого сервера поступает заявка из первой очереди.

B3. Если же заявок не оказалось в первой очереди, но при этом занята вторая, то на свободную виртуальную машину будет взята заявка из второй очереди.

B4. В случае, когда заняты обе очереди, для группы вычислительных машин первого сервера заявки будут выбраны из первой очереди.

Аналогично и для обслуживания заявок на втором сервере. Представим завершение обслуживания заявки на втором сервере как событие **C**.

C1. Если после обслуживания заявки обе очереди свободны, то виртуальная машина простаивает.

C2. Если занята очередь второго типа и при этом первая пустует, то на освобождённую виртуальную машину второго сервера поступает заявка из второй очереди.

C3. Если же заявок не оказалось во второй очереди, но при этом занята первая, то на свободную виртуальную машину будет взята заявка из первой очереди.

C4. В случае, когда заняты обе очереди, для группы вычислительных машин второго сервера заявки будут выделяться из второй очереди.

Примеры описанных выше событий проиллюстрированы на рис. 2, 3

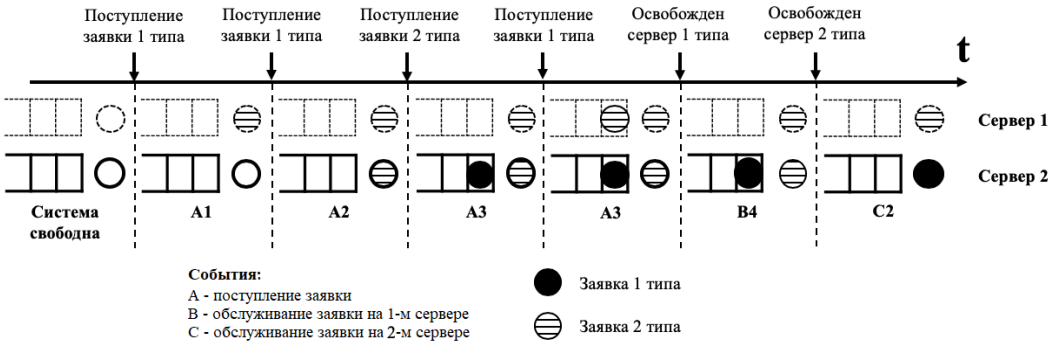


Рис. 2. Поведение системы в течение некоторого времени

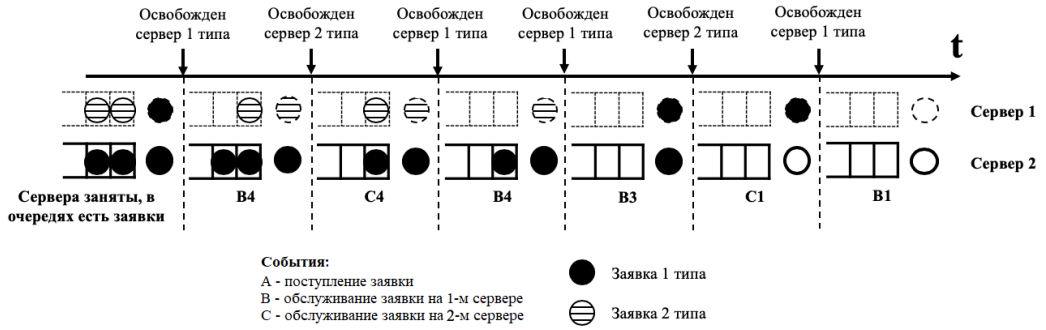


Рис. 3. Поведение системы в течение некоторого времени

2. Математическая модель

В предыдущем разделе был рассмотрен конкретный пример исследуемой в статье системы, перейдем к более общему описанию. Пусть анализируется система массового обслуживания с двумя параллельными очередями и двумя наборами приборов в стационарном режиме. Предполагается, что существуют потребители двух типов (классов), создающих две очереди. Схематическое изображение СМО представлено на рис. 4.

Состояния системы во времени описываются многомерной цепью Маркова с непрерывным временем: $\{X(t)\}_{t\geq 0} = \{Q_1(t), Q_2(t), D_{11}(t), D_{12}(t), D_{21}(t), D_{22}(t)\}$, где для каждого конкретного состояния x будет выявлен ряд параметров.

Пространство состояний системы описывается следующим образом:

$$E_x = \bigcup_{q_1=0}^{\infty} \bigcup_{q_2=0}^{\infty} E(q_1, q_2),$$

$$E(0,0) = \left\{ (0,0,d_{11},d_{12},d_{21},d_{22}) :\ d_{kj} \in E_{D_{kj}},\ d_{k1} + d_{k2} \leq N_k \right\},$$

$$E(q_1,q_2) = \left\{ (q_1,q_2 d_{11},d_{12},d_{21},d_{22}) :\ q_1 + q_2 \geq 1,\ d_{kj} \in E_{D_{kj}},\ d_{k1} + d_{k2} = N_k \right\}.$$

Определим основные характеристики системы. Пусть λ $(0 < \lambda < \infty)$ – параметр входящего пуассоновского потока заявок двух типов j, $j = \{1,2\}$. Заявки поступают в систему со средней скоростью потока λp_j для j-го потока заявок ($p_1 + p_2 = 1$). Число обслуживающих устройств (приборов), предназначенных для k-типа потребителей (их заявок) – N_j, тогда $N = N_1 + N_2$ – общее число обслуживающих устройств в системе. Время обслуживания устройств на приборе соответствующего типа: $0 < \mu_2 \leq \mu_1 < \infty$.

Для описанной системы мы пытаемся найти стационарную политику – векторную функцию $f=(f_{11},f_{12},f_{21},f_{22})\colon E_X\to(A_{11}(x),A_{12}(x),A_{21}(x),A_{22}(x))$, что предписывает выбор правильного действия $a\in A_{kj}(x)$ всякий раз непосредственно перед завершением обслуживания заявки из очереди j на сервере k, если $d_{kj}>0$ и $q_1(x)+q_2(x)\geq 1$. Любой другой переход не может подразумевать выбор какого-либо действия.

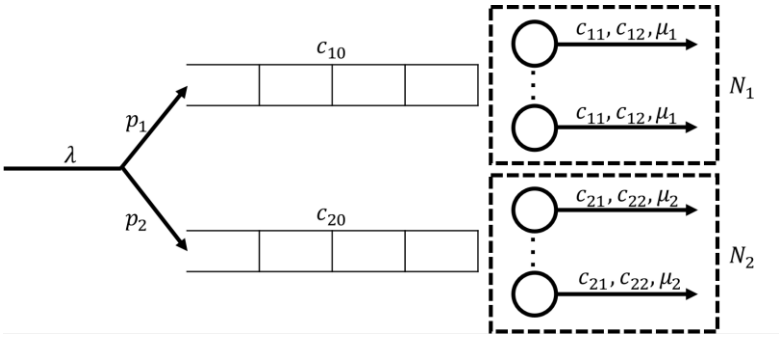


Рис. 4. Схема системы массового обслуживания

Запишем всё изложенное выше в виде функции выбора:

$$A_{kj}(q_1,q_2,d_{11},d_{12},d_{21},d_{22})=\begin{cases} A;\ q_1\geq 1;\ q_2\geq 1;\\ \{1\};\ q_1\geq 1;\ q_2=0;\\ \{2\};\ q_1=0;\ q_2\geq 1;\end{cases}$$

где q_1 и q_2 – количество заявок, находящихся в очередях первого и второго типа, $A=\{1,2\}$ – множество возможного выбора очереди из представленных значений. Получается, что $A_{kj}(x)$ – подмножество действий в состоянии x с заявкой, пришедшей из очереди j на прибор группы k.

Введём для удобства ещё несколько множеств: $Q_k(t)$ – множество числа клиентов в очереди $k\in\{1,2\}$ для каждого момента времени, $D_{kj}(t)$, $k,j\in\{1,2\}$ – множество числа занятых приборов для обслуживания заявок из очереди j на приборах группы k для каждого момента времени.

Алгоритм выбора оптимальной политики представлен в [2].

3. Характеристики системы

Чтобы оценить, насколько корректно функционирует система и насколько оптимально использует доступные ресурсы важно вычислять ее характеристики. Знание их позволит в том числе оптимально распределять обслуживающие устройства между двумя группами.

Как известно, для эргодических цепей Маркова с затратами долгосрочная средняя стоимость за единицу времени для политики f существует и совпадает с соответствующим средним [3]: $q^f=\sum_{y\in E_X}c(y)\pi_y^f$, где $c(y)$ – средняя стоимость в состоянии $y\in E_X$. Она считается по формуле:

$$c(x)=\sum_{k=1}^{2}\left(c_{k0}q_k(x)+c_{k1}d_{k1}(x)+c_{k2}d_{k2}(x)\right).$$

Для понимания того, сколько заявок каждого типа в среднем ожидает обслуживания, можно вычислить такой показатель как среднее число заявок j типа в очереди:

$$\overline{Q}_j=\sum_{x\in E_X}c(x)\pi_y^f=\sum_{q_1=0}^{\infty}\sum_{q_2=0}^{\infty}\sum_{d_{11}+d_{12}=N_1}\sum_{d_{21}+d_{22}=N_2}q_j\pi^f_{(q_1,q_2,d_{11},d_{12},d_{21},d_{22})},$$

где $c(x) = q_k(x)$, $x \in E_X$.

Кроме того, необходимо понимать, сколько серверов обычно занимают пользователи каждого типа. Среднее число обслуживающих устройств, занятых потребителями класса j:

$$\overline{C}_j = \sum_{x \in E_X} c(x)\pi_y^f = \sum_{q_1=0}^{\infty}\sum_{q_2=0}^{\infty} \sum_{d_{11}+d_{12}=N_1} \sum_{d_{21}+d_{22}=N_2} \left(d_{k1} + d_{k2}\right)\pi^f_{(q_1,q_2,d_{11},d_{12},d_{21},d_{22})},$$

где $c(x) = \sum_{j=1}^{2} d_{kj}(x)$.

Среднее число потребителей в системе вычисляется по формуле:

$$\overline{N} = \sum_{x \in E_X} c(x)\pi_y^f = \sum_{j=0}^{2}\left(\overline{Q}_j + \overline{C}_j\right),$$

где $c(x) = \sum_{k=1}^{2}\left[q_k(x) + \sum_{j=1}^{2} d_{kj}(x) \right]$.

Заключение

В настоящей работе рассмотрены системная и математическая модель системы с двумя типами услуг. Получение численного значения представленных в работе параметров является дальнейших исследований

ЛИТЕРАТУРА

1. 3GPP TS 23.501 V15.5.0: System architecture for 5G System (5GS). – 2019.

2. *Efrosinin D., Gudkova I., Stepanova N.* Algorithmic analysis of a two-class multi-server heterogeneous queueing system with a controllable cross-connectivity. – ASMTA, 2020.

3. *Башарин Г.П.* Лекции по математической теории телетрафика. – М.: РУДН, 2004.

ИССЛЕДОВАНИЕ SPLIT-MERGE СИСТЕМЫ С ДВУМЯ КЛАССАМИ ТРЕБОВАНИЙ И ПОТЕРЯМИ

Гуркова В.М., Осипов О.А.

Саратовский государственный университет

viktorija.gurkova@mail.ru, oleg.alex.osipov@gmail.com

Введение

Системы массового обслуживания (СМО) [1] с делением и слиянием требований (Fork-Join Queueing Systems) [2,3] представляют собой модели реальных систем с параллельной обработкой (многопроцессорные системы, GRID-системы, MapReduce и т.д.). В таких системах поступающие для обработки требования делятся на более простые для обслуживания фрагменты, которые распределяются по системе, занимая выделенные для них ресурсы. После завершения обслуживания фрагменты освобождают выделенные им ресурсы, но исходное требование будет считаться обслуженным и покинет систему только после завершения обслуживания всех его фрагментов. В последнее время анализ систем обслуживания с делением и слиянием требований является актуальным предметом исследований [4].

В настоящей работе рассматривается система массового обслуживания с двумя классами требований типа split-merge. В результате исследования было получено стационарное распределение вероятностей состояний системы, а также вычислены её основные характеристики. Для исследования и анализа данной системы обслуживания была разработана программа для численного расчёта её стационарных характеристик, а также построена имитационная модель [5], позволяющая рассмотреть систему в случае очередей неограниченной вместимости.

1. Описание модели

Рассматривается split-merge система массового обслуживания, состоящая из M обслуживающих приборов, с двумя очередями ограниченной вместимости c_1 и c_2 соответственно с дисциплиной обслуживания FCFS. В систему обслуживания поступают два класса требований, время между интервалами поступления которых распределено по экспоненциальному закону с параметрами λ_1 и λ_2 соответственно. Требования первого класса поступают в первую очередь, требования второго класса – во вторую очередь.

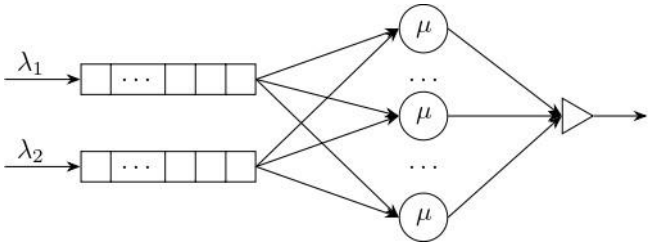


Рис. 1. Split-merge система массового обслуживания с двумя классами требований

Требование, поступающее на обслуживание, состоит из заданного числа фрагментов. Пусть d_1 – количество фрагментов в первом классе требований, а d_2 – количество фрагментов во втором классе требований. В том случае, когда свободных приборов достаточно для обслуживания всех фрагментов требования, фрагменты одновременно занимают свободные приборы и начинают обслуживаться. Если приборов недостаточно для обслуживания требований первого или второго класса, требования поступают на ожидание в очередь. В связи с ограниченной вместимостью очередей, требования, не заставшие свободных мест в первой или второй очереди, теряются, т.е. возвращаются обратно в источник. Предполагается следующая последовательность извлечения требований из очередей: если первая очередь не пуста и имеется достаточное количество свободных приборов, на обслуживание поступает требование первого класса, иначе, если вторая очередь не пуста, – требование второго класса. Таким образом, требования первого класса имеют приоритет по отношению к требованиям второго класса.

Длительность обслуживания фрагмента любого класса на приборе есть экспоненциально распределённая случайная величина с параметром μ.

Следует отметить, что фрагмент, обслуживание которого завершено, не освобождает обслуживающий прибор, а занимает (блокирует) его до момента, когда все родственные ему фрагменты, т.е. фрагменты одного и того же требования, не завершат своё обслуживание. Только после завершения обслуживания последнего фрагмента требование освобождает приборы и покидает систему.

2. Состояния системы

Состояние описанной выше системы обслуживания определяется как вектор

$$\mathrm{s} = (q_1, q_2, A, B),$$

где

- q_1 – число требований первого класса, находящихся в очереди;
- q_2 – число требований второго класса, находящихся в очереди;
- $A = \{a_1, \ldots, a_n\}$ – мультимножество, отображающее число обслуживаемых фрагментов каждого из требований первого класса; здесь $a_i > 0$, $i = \overline{1,n}$, обозначает число фрагментов некоторого требования первого класса, находящихся в состоянии обслуживания, n обозначает число требований первого класса на обслуживающих приборах системы;

- $B=\{b_1,\ldots,b_m\}$ – мультимножество, отображающее число обслуживаемых фрагментов каждого из требований второго класса; здесь $b_j>0$, $j=\overline{1,m}$, обозначает число фрагментов некоторого требования второго класса, находящихся в состоянии обслуживания, m обозначает число требований второго класса на обслуживающих приборах системы.

Процесс $\{\mathrm{s}(t),t\ge 0\}$ есть цепь Маркова с непрерывным временем, определенная на пространстве состояний S, которое определяется следующим образом:

$$\mathrm{S}=\Big\{\big(q_1,q_2,\{a_1,\ldots,a_n\},\{b_1,\ldots,b_m\}\big): q_1\le c_1, q_2\le c_2; nd_1+md_2\le M;$$
$$a_i\in\{1,\ldots,d_1\}, i=1,\ldots,n; b_j\in\{1,\ldots,d_2\}, j=1,\ldots,m; n,m\ge 0\Big\}.$$

Число приборов, занятых требованиями первого или второго класса, определим как

$$x(\mathrm{s})=|A|d_1+|B|d_2,$$

где $|A|$, $|B|$ – мощности соответствующих множеств A и B.

Обозначим через $\alpha(\mathrm{s},\mathrm{s}')$ интенсивность перехода системы из состояния s в состояние s'.

Рассмотрим все возможные события для системы обслуживания и определим соответствующие им интенсивности переходов между состояниями.

1. *Поступление требования в очередь*

а) если $M-x(s)<d_1$ и $q_1<c_1$: $\alpha\big((q_1,q_2,A,B),(q_1+1,q_2,A,B)\big)=\lambda_1$,

б) если $M-x(s)<d_2$ и $q_2<c_2$: $\alpha\big((q_1,q_2,A,B),(q_1,q_2+1,A,B)\big)=\lambda_2$.

2. *Поступление требования и немедленное начало его обслуживания*

а) если $M-x(s)\ge d_1$: $\alpha\big((0,q_2,A,B),(0,q_2,A+\{d_1\},B)\big)=\lambda_1$,

б) если $M-x(s)\ge d_2$: $\alpha\big((q_1,0,A,B),(q_1,0,A,B+\{d_2\})\big)=\lambda_2$.

3. *Завершение обслуживания одного фрагмента без ухода требования*

а) если $A\ne\varnothing$, $a_n>1$: $\alpha\big((q_1,q_2,\{a_1,\ldots,a_n\},B),(q_1,q_2,\{a_1,\ldots,a_n-1\},B)\big)=\mu a_n|A|_{a_n}$,

$|\cdot|_x$ определяет число элементов в мультимножестве, равных x,

б) если $B\ne\varnothing$, $b_m>1$: $\alpha\big((q_1,q_2,A,\{b_1,\ldots,b_m\}),(q_1,q_2,A,\{b_1,\ldots,b_m-1\})\big)=\mu b_m|B|_{b_m}$.

4. *Завершение обслуживания всего требования*

а) если $a_n=1$, $q_1\ge 1$: $\alpha\big((q_1,q_2,\{a_1,\ldots,a_{n-1},1\},B),(q_1-1,q_2,A-\{1\}+\{d_1\},B)\big)=\mu|A|_1$,

б) если $a_n=1$, $q_1=0$:

$\alpha\big((0,q_2,\{a_1,\ldots,a_{n-1},1\},B),(0,q_2-k,A-\{1\},B+\{d_2\}\times k)\big)=\mu|A|_1$,

где $k=\min\{(M-x(s)+d_1)/d_2,q_2\}$;

в) если $b_m=1$: $\alpha\big((q_1,q_2,A,\{b_1,\ldots,b_{m-1},1\}),(q_1-r_1,q_2-r_2,A',B')\big)=\mu|B|_1$,

где $A'=A+\{d_1\}\times r_1$, $B'=B-\{1\}+\{d_2\}\times r_2$, $r_1=\min\{(M-x(s)+d_2)/d_1,q_1\}$,

$r_2=\min\{(M-x(s)+d_2-r_1d_1)/d_2,q_2\}$.

Операции «+», «–» и «×» для множеств A, B в данном случае определяют добавление, удаление и повторение одного элемента в (из) мультимножества.

3. Стационарное распределение и характеристики системы обслуживания

Обозначим через **Q** инфинитезимальный оператор цепи Маркова. Тогда стационарное распределение $\boldsymbol{\pi}$ может быть найдено как решение уравнений глобального баланса вместе с условием нормировки: $\boldsymbol{\pi}\mathbf{Q}=\mathbf{0}$, $\boldsymbol{\pi}\mathbf{1}=\mathbf{1}$.

Используя найденное стационарное распределение, определим следующие стационарные характеристики системы:

1. Математическое ожидание Q_1, Q_2 числа требований в очередях для каждого класса: $Q_i=\sum_{s\in S} q_i\pi(s)$, $i=1,2$.
2. Вероятность отказа, т.е. вероятность того, что требование застанет очередь полностью заполненной, для требований первого и второго класса:
$p_{fi}=\sum_{(q_1,q_2,A,B)\in S,\,q_i=c_i}\pi\big((q_1,q_2,A,B)\big)$, $i=1,2$.
3. Математическое ожидание W_i длительности пребывания требования в очереди для каждого класса: $W_i=\dfrac{Q_i}{\lambda_i\left(1-p_{fi}\right)}$, $i=1,2$.
4. Математическое ожидание G числа свободных приборов в системе:
$G=\sum_{s\in S}\big(M-x(s)\big)\pi(s)$.
5. Математическое ожидание S_1, S_2 числа требований на приборах для каждого класса: $S_i=\sum_{s\in S}\dfrac{x_i(s)\pi(s)}{d_i}$, $i=1,2$, где $x_i(s)$, $i=1,2$ – число приборов, занятых требованиями первого или второго класса соответственно.
6. Математическое ожидание S числа требований на приборах: $S=S_1+S_2$.
7. Математическое ожидание длительности пребывания требований каждого класса в системе.

Представим данную характеристику как сумму математического ожидания длительности пребывания в очереди и математического ожидания длительности обслуживания на приборах. Очевидно, что длительность пребывания требования на приборах в силу функционирования системы является максимумом из независимых одинаково распределённых случайных величин.

Для максимума из k независимых одинаково распределённых случайных величин справедливо:

$$\xi=\max\{\xi_1,\ldots,\xi_k\},\ P\{\max\{\xi_1,\ldots,\xi_k\}<x\}=R(x),$$

$$\max\{\xi_1,\ldots,\xi_k\}<x\Leftrightarrow(\xi_1<x)\,\&\cdots\&\,(\xi_k<x)\Rightarrow P\{\xi_1<x\}\times\cdots\times P\{\xi_k<x\}=\left(1-e^{-\mu x}\right)^k.$$

Тогда для математического ожидания получим

$$E[\xi]=\int_0^{+\infty}xdR(x)=\left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)\frac{1}{\mu}=H_k\frac{1}{\mu},$$

где H_k есть k-я частная сумма гармонического ряда, $H_k=\sum_{i=1}^{k}\dfrac{1}{i}$.

Таким образом, математическое ожидание длительности пребывания требований каждого класса в системе имеет вид $T_i=W_i+H_{d_i}\dfrac{1}{\mu}$, $i=1,2$.

8. Математическое ожидание длительности пребывания требований в системе может быть вычислено с использованием формулы Литтла:

$$T = \frac{Q_1 + Q_2 + S_1 + S_2}{\lambda_1 \left(1 - p_{f1}\right) + \lambda_2 \left(1 - p_{f2}\right)}$$

4. Численный эксперимент

Рассматривается система массового обслуживания типа split-merge, состоящая из $M = 4$ обслуживающих приборов, с двумя очередями вместимости $c_1 = 5$ и $c_2 = 5$ с дисциплиной обслуживания FCFS. В систему обслуживания поступают два класса требований. Интенсивность поступления требований второго класса равна $\lambda_2 = 1$. Требования первого класса состоят из $d_1 = 3$ фрагментов, а требования второго класса из $d_2 = 2$ фрагментов. Интенсивность обслуживания требований приборами равна $\mu = 3$.

Выявим зависимость математического ожидания длительности пребывания требований в системе и вероятности отказа для первого и второго класса от параметра λ_1 Полученную зависимость можно увидеть на рис.2, 3, представленных ниже.

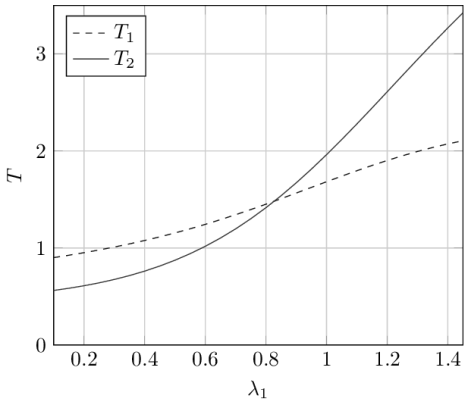


Рис. 2. Зависимость T_1, T_2 от λ_1

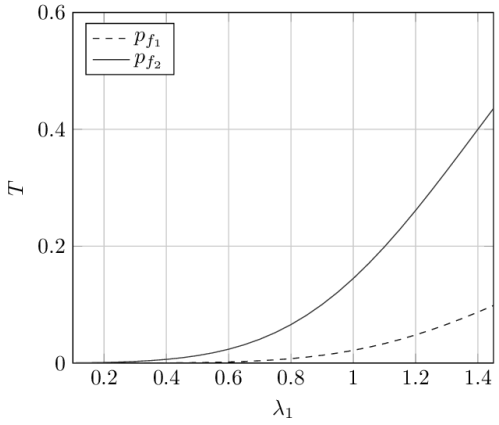


Рис. 3. Зависимость p_{f1}, p_{f2} от λ_1

Проанализировав первый график, можно сделать вывод о том, что с увеличением интенсивности поступления требований первого класса λ_1 стремительно увеличивается математическое ожидание длительности пребывания в системе требований второго класса, и менее стремительно – математическое ожидание длительности пребывания в системе требований первого класса. Такая зависимость может обуславливаться установленным приоритетом обслуживания – сначала обслуживать, насколько это возможно, требования первого класса, а затем требования второго класса. Следовательно, дли-

тельность ожидания требованиями второго класса в очереди возрастает и вместе с этим возрастает общая длительность пребывания требований второго класса в системе.

Приоритетом обслуживания можно объяснить также и возрастание вероятности отказа. Для требований первого класса рост данной величины менее интенсивен, чем для требований второго класса, которые дольше ожидают в очереди и, следовательно, быстрее её заполняют, и вновь поступающие требования получают отказ.

Также может быть поставлена задача по выявлению оптимального приоритета обслуживания исходя из интенсивностей входящих потоков. Данная задача была численно решена в том случае, когда целевой характеристикой являлось математическое ожидание длительности пребывания требований в системе. Полученные результаты можно видеть в табл. 1. Здесь каждому значению интенсивности соответствуют два значения. Первое значение соответствует математическому ожиданию длительности пребывания требований в системе в случае приоритета требований первого класса, второе значение – в случае приоритета требований второго класса.

Таблица 1

Зависимость длительности пребывания требований в системе от приоритета обслуживания и интенсивностей входящих потоков требований

λ_2 / λ_1	**0.5**		**0.8**		**1.1**		**1.4**		**1.7**		**2**	
0.5	0.872	0.886	1.313	1.326	2.341	2.288	3.867	4.106	4.905	6.341	5.391	8.174
0.8	0.929	0.948	1.555	1.526	2.948	2.742	4.307	4.698	4.963	6.792	5.252	8.426
1.1	1.014	1.033	1.908	1.758	3.490	3.146	4.468	5.071	4.862	7.001	5.041	8.519
1.4	1.145	1.148	2.390	2.028	3.847	3.502	4.462	5.312	4.701	7.095	4.817	8.541
1.7	1.357	1.304	2.926	2.330	4.014	3.802	4.380	5.459	4.529	7.117	4.606	8.521
2	1.704	1.512	3.370	2.647	4.054	4.036	4.272	5.529	4.368	7.081	4.421	8.464

Заключение

В настоящей работе исследована split-merge система с двумя классами требований и потерями. В результате исследования получено стационарное распределение вероятностей состояний системы, а также вычислены её основные характеристики. Помимо этого, в результате численного эксперимента была выявлена зависимость длительности пребывания требований в системе и вероятности отказа для первого и второго класса от интенсивности входящего потока требований первого класса, а также решена задача по выявлению оптимального приоритета обслуживания исходя из интенсивностей входящих потоков. Кроме того, для анализа и исследования данной системы была построена имитационная модель, позволяющая рассмотреть систему в случае очередей неограниченной вместимости.

ЛИТЕРАТУРА

1. *Клейнрок Л.* Теория массового обслуживания. Пер. с англ. / Пер. И. И. Грушенко; ред. В. И. Нейан. – М.: Машиностроение, 1979. 477 с.

2. *Narahari Y., Sundarrajan P.* Performability Analysis of Fork-Join Queueing Systems, Journal of the Operational Research Society 46 (10) (1995) 1237–1249. doi:10.1057/jors.1995.171.

3. *Thomasian A.* Analisys of Fork/Join and Related Queueing Systems, ACM Computing Surveys 47 (2) (2014) 17:1–17:71. doi:10.1145/2628913.

4. *Rizk A., Poloczek F., Ciucu F.* Stochastic bounds in Fork-Join queueing systems under full and partial mapping. Queueing Systems 83, 261–291 (2016). https://doi.org/10.1007/s11134-016-9486-x

5. *Кельтон В., Лоу А.* Имитационное моделирование. Классика CS. 3-е изд. – СПб.: Питер: Киев: Издательская группа BHV, 2004. 847 с.

ИССЛЕДОВАНИЕ ДОПОЛНИТЕЛЬНО ФОРМИРУЕМОГО ПОТОКА В СИСТЕМЕ С ЭКСПОНЕНЦИАЛЬНЫМ ОБСЛУЖИВАНИЕМ И НЕОГРАНИЧЕННЫМ ЧИСЛОМ ПРИБОРОВ МЕТОДОМ МАРКОВСКОГО СУММИРОВАНИЯ

Даммер Д.Д., Федерягина П.В.
Томский государственный университет
di.dammer@yandex.ru, polina.federagina@gmail.com

Введение

Для исследования экономических [1,2], производственных [3,4] систем часто применяются методы массового обслуживания [5,6]. При таком исследовании могут анализироваться входящие и выходящие потоки заявок, процесс изменения во времени числа занятых приборов и др. Анализ работ по данному вопросу показывает необходимость развития методов исследования таких систем. В работе предлагается и реализуется подход для получения характеристик сгенерированных (дополнительных) потоков. Примерами таких потоков могут быть потоки страховых случаев клиентов, заключивших договр на оказание страховых услуг [1]; Бартлетт применял такие дополнительные потоки для моделирования транспортных потоков [7], П. Льюис рассматривал их в качестве модели отказов вычислительных машин [8].

В данной работе рассмотрена система массового обслуживания с неограниченным числом приборов, простейшим потоком входящих заявок с параметром λ и с обслуживанием, которое является случайной величиной с экспоненциальным законом с параметром μ. В течение времени обслуживания каждая заявка с интенсивностью γ генерирует дополнительные события.

Будем называть локальным *d*-потоком последовательность моментов наступления событий, сформированных одной поступившей в систему заявкой; суммарным *d*-потоком – последовательность моментов наступления событий, сформированных всеми поступившими в систему заявками.

В работе решается задача нахождения характеристической функции числа событий *d*-потока применением двух методов: исследованием двумерного процесса числа занятых приборов и числа событий *d*-потока, и применением метода Марковского суммирования.

На практике решение может быть применимо к модели страховой фирмы или веб-приложению, например, службе вызова такси и интернет магазину. В этом случае мы найдем распределение суммарного числа требований на страховые выплаты или числа вызовов, поступающих в службу вызова такси. Зная это распределение, можно, например, вычислить распределение общей суммы страховых выплат.

1. Описание модели и постановка задачи

Рассмотрим (рис. 1) систему массового обслуживания с неограниченным числом приборов и экспоненциальным обслуживанием с параметром μ. На вход системы поступает простейший с параметром λ поток заявок. В течение времени обслуживания каждая заявка с интенсивностью γ формирует дополнительные события, совокупность которых от всех обслуживаемых заявок будем называть d-потоком.

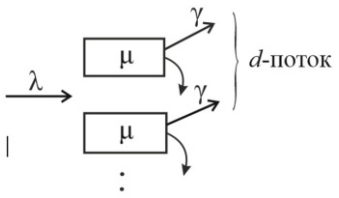


Рис. 1. Модель системы с неограниченным числом приборов и *d*-потоком

В работе решается задача нахождения вероятностных характеристик числа событий *d*-потока потока, наступивших за время T на интервале $[0,T]$ при условии, что в начальный момент времени $t=0$, система свободна.

Пусть $i(t)$ – число занятых приборов, $n(t)$ – число событий *d*-потока, $\{i(t),n(t)\}$ – двумерный случайный процесс числа заявок в системе в момент времени t и числа событий *d*-потока, наступивших на интервале $[0,t]$, $P\{i(t)=i,n(t)=n\}=P(i,n,t)$ – распределение вероятностей двумерного случайного процесса $\{i(t),n(t)\}$.

2. Исследование двумерного процесса числа занятых приборов и числа событий *d*-потока

Применяя формулу полной вероятности, запишем допредельные равенства с помощью Δt-метода [9]:

$$P(i,n,t+\Delta t)=P(i,n,t)(1-\lambda\Delta t-i\mu\Delta t-i\gamma\Delta t)+P(i-1,n,t)\lambda\Delta t+$$
$$+P(i+1,n,t)(i+1)\mu\Delta t+P(i,n-1,t)i\gamma\Delta t+o(\Delta t).$$

Тогда систему дифференциальных уравнений можем запись в виде:

$$\frac{\partial P(i,n,t)}{\partial t}=P(i,n,t)(-\lambda-i\mu-i\gamma)+P(i-1,n,t)\lambda+P(i+1,n,t)(i+1)\mu+$$
$$+P(i,n-1,t)i\gamma. \qquad (1)$$

Определим характеристическую функцию

$$H(u_1,u_2,t)=\sum_{i=0}^{\infty}\sum_{n=0}^{\infty}e^{ju_1i}e^{ju_2n}P(i,n,t). \qquad (2)$$

Запишем из (1) дифферециальное уравнение для характеристической функции (2):

$$\frac{\partial H}{\partial t}=\lambda H\left(e^{ju_1}-1\right)+\frac{\gamma}{j}\frac{\partial H(u_1,u_2,t)}{\partial u_1}\left(e^{ju_2}-1\right)+\frac{\mu}{j}\frac{\partial H(u_1,u_2,t)}{\partial u_1}\left(e^{-ju_1}-1\right)=$$
$$=-\lambda H(u_1,u_2,t)\left(1-e^{ju_1}\right)+j\frac{\partial H(u_1,u_2,t)}{\partial u_1}\left[\mu\left(1-e^{-ju_1}\right)+\gamma\left(1-e^{ju_2}\right)\right]. \qquad (3)$$

Решение дифференциального уравнения (3) первого порядка в частных производных определяется решением системы обыкновенных дифференциальных уравнений для характеристических кривых [10]:

$$\frac{dt}{1}=\frac{dH}{H\lambda\left(e^{ju_1}-1\right)}=\frac{du}{-j\left(\mu+\gamma-\mu e^{-ju_1}-\gamma e^{ju_2}\right)}.$$

Начальные условия

$$H(u,v,0)=1. \qquad (4)$$

Решение уравнения (3) при начальном условии (4) имеет вид:

$$H(u,v,t)=\exp\left(\frac{\lambda}{A}\left(\frac{B-(A\left(e^{ju}-1\right)+B)e^{-At}}{A}-Bt+e^{ju}-1\right)\right), \qquad (5)$$

где $A=\mu+\gamma\left(1-e^{jv}\right)$, $B=\gamma\left(1-e^{jv}\right)$, $u_1=u$, $u_2=v$.

Из (5) найдем характеристическую функцию числа $n(t)$ событий *d*-потока при $t=T$:

$$G(v,T)=H(u,v,T)\Big|_{u=0}=\exp\left\{\frac{\lambda\gamma\left(1-e^{jv}\right)}{\mu+\gamma\left(1-e^{jv}\right)}\left[\frac{1-e^{-\left(\mu+\gamma\left(1-e^{jv}\right)\right)T}}{\mu+\gamma\left(1-e^{jv}\right)}-T\right]\right\}. \qquad (6)$$

3. Метод Марковского суммирования

Для применения метода Марковского суммирования введем обозначия: $\xi(t)$ при $[0,T]$ – число событий *d*-потока сформированных на интервале $[0,T]$ заявкой входящего потока, поступившей в систему в момент времени *t*. Также обозначим $n(t)$ – суммарное число событий *d*-потока сформированных на интервале $[0,T]$ всеми заявками входящего потока, поступившими в систему за время *t* на интервале $[0,t]$.

Так как случайные величины $\xi(t)$ для различных моментов *t* поступления заявок входящего потока независимы (времена обслуживания независимы), $n(t)$ – сумма случайного числа независимых случайных величин. В силу того, что поток входящих заявок простейший, случайный процесс $n(t)$ – марковский. Методом марковского суммирования будем называть подход, применяемый ниже для решения поставленной задачи.

Для заяки входящего потока, поступившей в систему в момент времени $t\in[0,T]$, обозначим $r(i,t)=P(\xi(t)=i)$ и найдем это распределение вероятностей. Поступившие в систему заявки условно можно разделить на два типа (рис. 2).

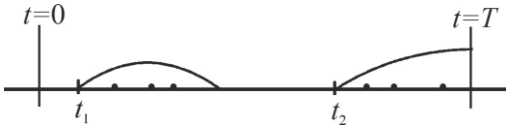


Рис. 2. Схема наступлений событий в *d*-потоке

Пусть t_1 и t_2 – моменты поступления двух заявок входящего потока. Для заявки, поступившей в момент времени t_1, время обслуживания заканчивается до момента $t=T$; для заявки входящего потока, поступившей в момент t_2, время обслуживания заканчивается после момента *T*. В первом случае все события *d*-потока, сформированные соответствующей заявкой, принадлежат интервалу $[0,T]$. Для заявки второго типа интервалу $[0,T]$ принадлежат только те события *d*-потока, которые наступают на интервале $[t,T]$.

При фиксированном значении *x* времени обслуживания заявки число событий локального *d*-потока, формируемого этой заявкой, является пуассоновским с параметром γx. Для заявки первого типа время *x* является значением случайной величины, имеющей экспоненциальное распределение с параметром μ. Для заявок второго типа время *x* является детерминированной величиной равной $T-t$.

По формуле полной вероятности можем записать выражение для распределения, учитывая типы заявок :

$$r(i,t)=\int_0^{T-t}\frac{(\gamma x)^i}{i!}e^{-\gamma x}\mu e^{-\mu x}dx+e^{-\mu(T-t)}\frac{\left(\gamma(T-t)\right)^i}{i!}e^{-\gamma(T-t)}.$$

Характеристическую функцию этого распределения представим в виде:

$$g(u,t)=M\left\{e^{ju\xi(t)}\right\}=\sum_{i=0}^{\infty}e^{jui}r(i,t)=$$

$$=\int_0^{T-t}\exp\left\{\left(e^{ju}-1\right)\gamma x\right\}d\left(1-e^{-\mu x}\right)+\left(1-\left(1-e^{-\mu(T-t)}\right)\right)\exp\left\{\left(e^{ju}-1\right)\gamma(T-t)\right\}=$$

$$=\exp\left\{\left(e^{ju}-1\right)\gamma(T-t)-\mu(T-t)\right\}\frac{\gamma\left(1-e^{ju}\right)}{\mu+\gamma\left(1-e^{ju}\right)}+\frac{\mu}{\mu+\gamma\left(1-e^{ju}\right)}.$$

Так как для рассматриваемой системы входящий поток простейший с параметром λ и величины, определяемые распределением $r(i,t)$, независимы, то число $n(t)$ событий суммарного *d*-потока является цепью Маркова.

Для марковского процесса n(t) обозначим $P(n,t)=P\{n(t)=n\}$ и запишем равенства:

$$P(n,t+\Delta t)=P(n,t)(1-\lambda\Delta t)+\lambda\Delta t\sum_{i=0}^{n}P(n-i,t)r(i,t)+o(\Delta t),$$

откуда получаем систему дифференциальных уравнений:

$$\frac{\partial P(n,t)}{\partial t}=\lambda\left(\sum_{i=0}^{n}P(n-i,t)r(i,t)-P(n,t)\right).$$

Определим характеристическую функцию процесса $n(t)$:

$$H(u,t)=M\left\{e^{jun(t)}\right\}=\sum_{n=0}^{\infty}e^{jun}P(n,t). \qquad (7)$$

Дифференциальное уравнение для характеристической функции (7) имеет вид:

$$\frac{\partial H(u,t)}{\partial t}=\lambda H(u,t)\left(g(u,t)-1\right). \qquad (8)$$

Совместно с начальным условием $H(u,0)=1$ уравнение (8) определяет вид характеристической функции при $t=T$:

$$H(u,T)=\exp\left\{\lambda\int_0^T\left(g(u,t)-1\right)dt\right\}=\exp\left\{\lambda\frac{\gamma\left(1-e^{ju}\right)}{\mu+\gamma\left(1-e^{ju}\right)}\left(\frac{1-e^{-\left(\mu+\gamma\left(1-e^{ju}\right)\right)T}}{\mu+\gamma\left(1-e^{ju}\right)}-T\right)\right\}. \qquad (9)$$

4. Характеристики числа событий *d*-потока

Таким образом, сравнивая результаты, полученные применением многомерных марковских процессов (6) и методом марковского суммирования (9), делаем вывод, что полученные характеристические функции числа событий суммарного *d*-потока совпадают.

Зная характеристическую функцию, можно получить математическое ожидание и дисперсию числа событий *d*-потока:

$$M\{n(t)\}=\frac{\lambda\gamma}{\mu^2}\left(e^{-\mu t}+\mu t-1\right),\ D\{n(t)\}=\frac{\lambda\gamma}{\mu^3}\left(\mu^2 t-\mu-4\gamma+4\gamma e^{-\mu t}+\mu e^{-\mu t}+2\gamma\mu t e^{-\mu t}\right).$$

5. Численные эксперименты

В результате численного эксперимента с помощью системы Mathcad был получен графический вид распределения как обратное преобразование Фурье при различных значениях парметров. Графические результаты представлены ниже (рис. 3–5).

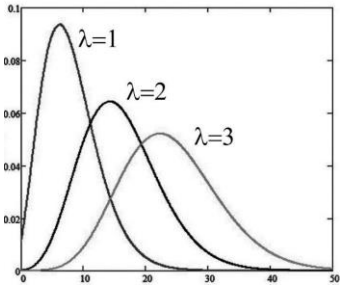


Рис. 3. Распределение числа событий *d*-потока при различных значениях λ и $\mu = 0.5$, $\gamma = 0.5$, $t = 10$

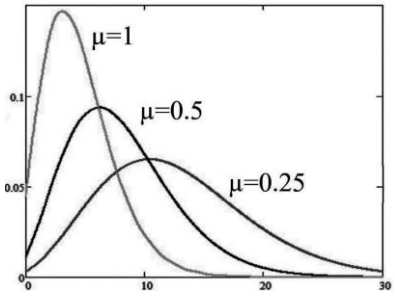


Рис. 4. Распределение числа событий *d*-потока при различных значениях μ и $\lambda = 1$, $\gamma = 0.5$, $t = 10$

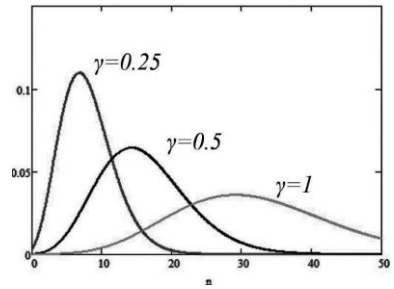


Рис. 5. Распределение числа событий *d*-потока при различных значениях γ и $\lambda = 2$, $\mu = 0.5$, $t = 10$

Также были найдены значения математического ожидания и дисперсии, представлены в табл. 1–3.

Таблица 1

Значения среднего и дисперсии числа $n(t)$ при $\lambda = 1$

	$\mu = 0.25$	$\mu = 0.5$	$\mu = 1$
$\gamma = 0.25$	$\alpha = 6.328$, $D = 13.283$	$\alpha = 4.007$, $D = 7.054$	$\alpha = 2.25$, $D = 3.25$
$\gamma = 0.5$	$\alpha = 12.657$, $D = 40.477$	$\alpha = 8.013$, $D = 20.202$	$\alpha = 4.5$, $D = 8.5$
$\gamma = 1$	$\alpha = 25.313$, $D = 136.594$	$\alpha = 16.027$, $D = 64.782$	$\alpha = 9$, $D = 25.001$

Таблица 2

Значения среднего и дисперсии числа $n(t)$ при $\mu = 0.5$

	$\gamma = 0.25$	$\gamma = 0.5$	$\gamma = 1$
$\lambda = 1$	$\alpha = 4.007$, $D = 7.054$	$\alpha = 8.013$, $D = 20.202$	$\alpha = 16.027$, $D = 64.782$
$\lambda = 2$	$\alpha = 8.013$, $D = 14.108$	$\alpha = 16.027$, $D = 40.404$	$\alpha = 32.054$, $D = 129.563$
$\lambda = 3$	$\alpha = 12.02$, $D = 21.162$	$\alpha = 24.04$, $D = 60.606$	$\alpha = 48.081$, $D = 194.345$

Таблица 3

Значения среднего и дисперсии числа $n(t)$ при $\gamma = 0.5$

	$\mu = 0.25$	$\mu = 0.5$	$\mu = 1$
$\lambda = 1$	$\alpha = 12.657$, $D = 40.477$	$\alpha = 8.013$, $D = 20.202$	$\alpha = 4.5$, $D = 8.5$
$\lambda = 2$	$\alpha = 23.313$, $D = 80.954$	$\alpha = 16.027$, $D = 40.404$	$\alpha = 9$, $D = 17.001$
$\lambda = 3$	$\alpha = 37.97$, $D = 121.431$	$\alpha = 24.04$, $D = 60.606$	$\alpha = 13.5$, $D = 25.501$

Заключение

В данной работе была получено выражение для характеристическаой функции числа $n(T)$ событий дополнительного потока, формируемого заявками, поступившими в систему на интервале $[0,T]$. Также были выведены формулы для характеристик числа $n(T)$ событий *d*-потока. Получен вид распределения вероятностей $P(n,t)$ в рамках численного эксперимента при различных значениях параметров, найдены значения математического ожидания и дисперсии при заданных параметрах распределений. Полученные результаты могут быть использованы для анализа деятельности различных экономических систем.

ЛИТЕРАТУРА

1. *Даммер Д.Д.* Исследование математической модели страховой компании в виде системы массового обслуживания в случайной среде и с учетом единовременных страховых выплат // Компьютерные науки и информационные технологии. Материалы Международной научной конференции 2018. С. 117 – 121.

2. *Жидкова Л.А., Моисеева С.П.* Математическая модель потоков покупателей двухпродуктовой торговой компании в виде системы массового обслуживания с повторным обращением к блокам // Известия Томского политехнического университета. 2013. Том 322, №6, С. 5 – 9.

3. *Balsamo S., De Nitti Persone V., Inverardi P.* A review on Queueing Network Models with Finite Capacity Queues for Software Architectures Performance Prediction // Performance Evaluat. 2003. V.51. Iss. 2. P. 269--288.

4. *Матальцкий М.А., Станкевич С.Э.* НМ-сети как новые стохастические модели прогнозирования доходов различных объектов // Вестник ГрГУ. Сер.5 Экономика. 2009. №1. С. 107 – 115.

5. *Narayan Bhat U.* An Introduction to Queueing Theory: Modeling and Analysis in Applications. Boston: Birkhauser, 2008.

6. *Клейнрок Л.* Теория массового обслуживания /Пер. И. И. Грушко; ред. В. И. Нейман. М.: Машиностроение, 1979.

7. *Bartlett M.S.* The spectral analysis of point processes // J. Royal Stat. Soc. 1963. B 25. P.264—296.

8. *Cox D.R., Lewis P.A.W.* The statistical analysis of series of events. London: Methuen, 1966.

9. *Назаров А.А., Терпугов А.Ф.* Теория вероятностей и случайных процессов. Томск: Изд-во НТЛ, 2010.

10. *Назаров А.А., Терпугов А.Ф.* Теория массового обслуживания. Томск: Изд-во НТЛ, 2010.

РАЗРАБОТКА ФРЕЙМВОРКА ДИСКРЕТНО-СОБЫТИЙНОГО МОДЕЛИРОВАНИЯ

Заварзин А.С., Осипов О.А.

Саратовский государственный университет

zavarzin.27100@mail.ru, oleg.alex.osipov@gmail.com

Введение

За последнее время имитационное моделирование стало одним из основных и наиболее распространённых инструментов исследования сложных систем. Роль имитационного моделирования в промышленном и информационном мире велика. Благодаря возможности построения имитационной модели реальной системы исследователи могут проводить большое количество экспериментов, анализировать поведение системы, собирать статистические данные и делать соответствующие выводы. Эксперименты над имитационной моделью, в отличие от экспериментов над реальной системой, позволяют сэкономить временные и денежные ресурсы. Таким образом, имитационное моделирование является важным инструментом исследования и анализа поведения реальных систем, который в последнее время активно развивается.

Для создания имитационных моделей используются различные подходы:

- создание компьютерных моделей с помощью универсальных языков программирования (Java, C++, Delphi);
- разработка компьютерных моделей с применением специализированных языков моделирования (SIMULA, AnyLogic);
- построение компьютерных моделей и проведение имитационных экспериментов при помощи специализированных компьютерных сред с графическим редактором (AnyLogic, Arena, SIMUL8, NetLogo);
- включение средств имитационного моделирования в стандартные математические компьютерные системы (пакет Simulink системы MATLAB).

Большинство компьютерных сред имитационного моделирования [1,2] обладают некоторыми недостатками. Например, многие продукты вынуждают пользователя использовать свой собственный язык программирования. Некоторые системы являются достаточно гибкими и многофункциональными (AnyLogic, SIMUL8), но обладают высокой стоимостью. Остальные же системы примитивны или сложны в использовании.

В данной работе представлен фреймворк, разработанный на универсальном языке программирования Java для дискретно-событийного моделирования систем. Данный фреймворк позволяет конструировать модели систем и осуществлять имитационное моделирование с вычислением основных характеристик.

1. Дискретно-событийное моделирование

Дискретно-событийное моделирование [3,4] используется для построения модели, отражающей развитие системы во времени, когда состояния переменных меняются мгновенно в конкретные моменты времени. В такие моменты времени происходят *события*, при этом событие определяется как мгновенное возникновение, которое может изменить состояние системы. Динамическая природа дискретно-событийных имитационных моделей обязует следить за текущим имитационным временем по мере функционирования имитационной модели.

Необходимо также иметь механизм для продвижения имитационного времени от одного значения к другому. В имитационной модели переменная, обеспечивающая текущее значение модельного времени, называется *часами модельного времени*. Существует два основных способа продвижения модельного времени: продвижение времени от события к событию и продвижение времени с постоянным шагом. При использовании подхода продвижения времени от события к событию часы модельного времени в исходном состоянии устанавливаются в 0 и определяется время возникновения будущих событий. После этого часы переходят на время возникновения ближайшего события и в этот момент обновляется состояние системы с учетом произошедшего события, а также обновляются сведения о времени возникновения будущих событий. Затем часы продвигаются ко времени возникновения следующего ближайшего события, обновляется состояние системы и определяется время будущих событий, и т.д. Процесс продвижения модельного времени от времени возникновения одного события ко времени возникновения другого продолжается до тех пор, пока не будет достигнуто какое-либо условие останова, указанное до начала имитационного моделирования.

Поскольку в дискретно-событийной имитационной модели все изменения происходят только во время возникновения событий, периоды простоя системы пропускаются. Длительность интервала продвижения модельного времени от одного события к другому может быть различной.

Все дискретно-событийные имитационные модели включают ряд общих компонентов. В частности, дискретно-событийная имитационная модель, которая использует механизм продвижения модельного времени от события к событию содержит следующие основные компоненты:

- *состояние системы* – совокупность переменных состояния, необходимых для описания системы в определенный момент времени;

- *часы модельного времени* – переменная, указывающая текущее значение модельного времени;
- *список событий* – список, содержащий время возникновения следующих событий;
- *статистические счётчики* – переменные, предназначенные для хранения статистической информации о характеристиках системы;
- *синхронизирующая программа* – подпрограмма, которая определяет следующее событие в списке событий и затем переводит часы модельного времени на время возникновения этого события;
- *программа обработки событий* – подпрограмма, обновляющая состояние системы, когда наступает определённое событие;
- *генераторы случайных чисел* – для моделирования случайных величин, определённых в имитационном моделировании;
- *генератор отчётов* – для расчётов характеристик системы и вывода отчётов по окончании моделирования;
- *основная программа* – подпрограмма, которая задаёт последовательность вызова остальных подпрограмм, и именно основная программа реализует имитационное моделирование системы.

2. Архитектура фреймворка дискретно-событийного моделирования

В данном разделе будет представлено краткое описание архитектуры разработанного ядра фреймворка дискретно-событийного моделирования.

Архитектурным подходом при разработке фреймворка стала архитектура, управляемая событиями (*event-driven architecture*, EDA) [5]. Эта архитектура завязана на производителях и потребителях событий. Главная идея состоит в том, чтобы разделить части системы так, чтобы каждая из частей активизировалась, когда необходимое событие происходит в другой. Производитель события не знает за каким из событий наблюдает какой из потребителей. Также и другие потребители не знают, кто из них за каким событием наблюдает. Таким образом, главная идея заключается в расщеплении частей системы.

Компоненты данного фреймворка спроектированы с учётом дальнейшей расширяемости, ядро фреймворка содержит минимально необходимый набор интерфейсов, абстрактных классов и базовых реализаций, что позволяет пользователю в дальнейшем расширить фреймворк под свои задачи. При разработки был поставлен упор на простоту и гибкость использования. Поведение компонентов самой системы моделируется с помощью событий и их обработчиков. Все компоненты модели находятся в специально отведенном для них окружении (**Environment**), фактически оно отражает текущее состояние модели. Компоненты могут быть связаны между собой и события одних компонентов могут влиять на состояния других (например, событие поступления требования генерирует в источнике новое требование и отправляет его в очередь системы обслуживания). Для имитационного моделирования необходимо объявить часы модельного времени (**Clock**). У каждого события (**Event**) в системе должен присутствовать обработчик, он будет выполнять действие, которое должно произойти при наступлении события, т.е. логику события.

Работа имитационной модели прекращается при выполнении некоторого условия останова. Это может быть достижение некоторого заданного времени (**TimeStopCondition**), или отсутствие новых событий в системе (**EmptyStopCondition**). Также пользователь фреймворка может сам определить дополнительные условия останова реализовав интерфейс **Predicate**.

Перечислим основные интерфейсы и базовые имплементации в разработанном фреймворке:

- **Event** – интерфейс события, содержит методы отсрочки события (postpone), метод активации события (activate), который необходимо реализовать в дочерних классах;
- **AbstractEvent** – абстрактный класс, имплементирует интерфейс **Event**, содержит время активации события;
- **HandlerEvent** – наследует **AbstractEvent**, позволяет определить множество обработчиков, которые применяются для обработки событий, что позволяет разбить логику обработки на части;
- **EventProvider** – интерфейс контейнера для событий, содержит базовые методы работы с контейнерами (add, addAll, peek, getNext, remove, count, isEmpty), где метод getNext возвращает ближайшее событие, удаляя его из контейнера;
- **EventProviderImpl** – базовая реализация интерфейса **EventProvider**, использующая ArrayList<**Event**> в качестве контейнера;
- **EventHandler** – функциональный интерфейс для обработки событий;
- **TimeOut** – класс имплементирующий интерфейс **EventHandler**, содержит логику откладывания события;
- **Environment** – интерфейс окружения для имитационной модели;
- **EnvironmentImpl** – базовая реализация интерфейса **Environment**;
- **SimulationModel** – интерфейс для имитационной модели, содержит метод запуска (run) и метод для перехода к следующему состоянию (step);
- **AbstractSimulationModel** – абстрактный класс, реализует интерфейс **SimulationModel**, является базовой реализацией для всех моделей;
- **SimulationModelImpl** – наследует класс **AbstractSimulationModel**, не добавляет никакой новой логики, также является базовой реализацией для всех моделей;
- **TimerSimulationModel** – наследует класс **AbstractSimulationModel**, описывает модель таймера;
- **SimulationContext** – интерфейс контекста имитационной модели;
- **SimulationContextImpl** – базовая реализация интерфейса **SimulationContext**, содержит компоненты для системы (**Environment**, **Clock**, **EventProvider**) имитационного моделирования.

На данный момент реализованы следующие модули, изображённые на рис. 1.

- *simulation-core* – модуль, содержащий основные компоненты для дискретно-событийной модели (события, контейнеры для событий, обработчики событий, часы модельного времени и т.д.);
- *random-variable* – модуль, содержащий датчики псевдослучайных чисел с различными распределениями (экспоненциальное, эрланговское, нормальное и т.д.);
- *queueing-component-library* – модуль, содержащий необходимый набор компонентов для создания имитационных моделей процессов обслуживания;
- *examples* – примеры имитационных моделей.

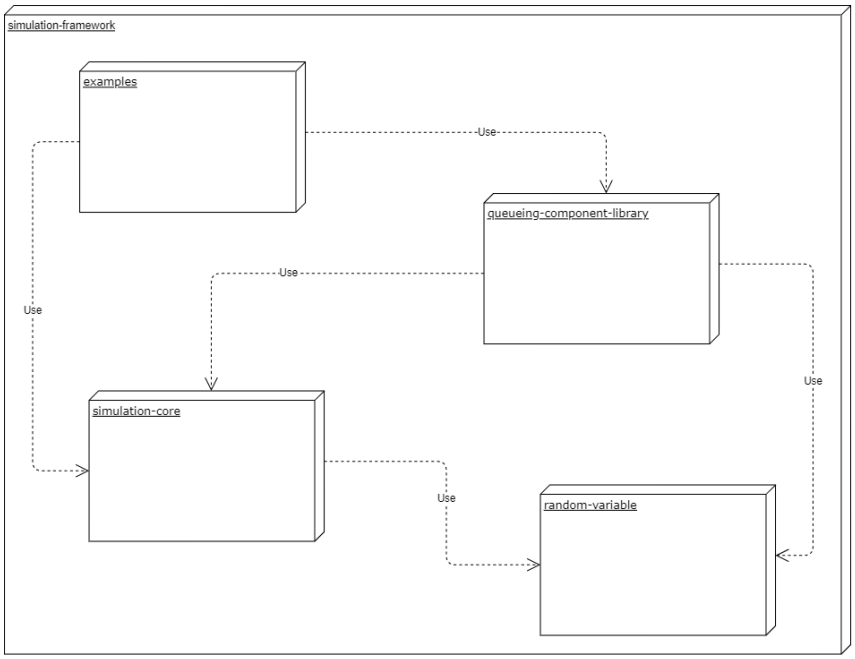


Рис. 1. Модули фреймворка дискретно-событийного моделирования

3. Пример использования

Рассмотрим пример использования фреймворка дискретно-событийного моделирования. Для начала, чтобы сконфигурировать модель, необходимо создать объекты, отвечающие за контекст модели, т.е. окружение, где будут находиться все объекты системы; часы модельного времени, отвечающие за синхронизацию и продвижение времени; провайдер (контейнер) для событий, позволяющий получать события в порядке их наступления. Данное действие можно описать так:

```
Environment env = new EnvironmentImpl();
Clock clock = new ClockImpl();
EventProvider eventProvider =
   new EventProviderImpl(Collections.emptyList());
SimulationContext simulationContext =
   new SimulationContextImpl(env, clock, eventProvider);
```

Затем необходимо инициализировать события и логику их обработки. Для данного примера были выбраны два таймера. Интервалы между срабатываниями первого таймера имеют экспоненциальное распределение. Срабатывание второго таймера происходят через фиксированные интервалы. Логика обработки событий передаётся через лямбда-выражение в функцию addHandler:

```
HandledEvent randomPeriodic =
   new HandledEvent.HandledEventBuilder(simulationContext)
      .periodic(new ExponentialRandomVariable(new Random(), 1))
      .addHandler(event -> System.out.println(
         "Message from periodic random event: " +
            event.getActivateTime()))
      .build();

HandledEvent constPeriodic =
     new HandledEvent.HandledEventBuilder(simulationContext)
```

```
.periodic(3)
.addHandler(event -> System.out.println(
   "Message from periodic const event: " +
      event.getActivateTime()))
.build();
```

После этого помещаем эти два события в провайдер, созданный ранее:

```
eventProvider.add(randomPeriodic);
eventProvider.add(constPeriodic);
```

Остаётся только инициализировать модель созданным выше контекстом, добавить условие останова и запустить имитационную модель:

```
SimulationModelImpl model =
   new SimulationModelImpl(simulationContext);
model.setStopCondition(new TimeStopCondition(10));
model.run();
```

Заключение

В данной работе был представлен разработанный фреймворк дискретно-событийного моделирования. В дальнейшем планируется доработка фреймворка, добавление компонентов для имитационного моделирования систем и сетей массового обслуживания, создание удобного пользовательского интерфейса для быстрого конструирования систем, добавление автоматизированного расчёта необходимой статистики и вывода результатов.

ЛИТЕРАТУРА

1. *Журавлёв С.С.* Краткий обзор методов и средств имитационного моделирования производственных систем // Журнал: Проблемы информатики // Рубрика: Имитационное моделирование технических систем и технологических процессов, 2009. 47 – 53.

2. *Рыжиков Ю.И., Соколов Б.В., Юсупов Р.М.* Проблемы теории и практики имитационного моделирования // Сб. докл. III Всерос. науч.-прак. конф. «Имитационное моделирование. Теория и практика» (ИММОД-2007). Санкт-Петербург, 17-19 окт. 2007. Т. 1. 58 – 70.

3. *Кельтон В., Лоу А.* Имитационное моделирование. Классика CS. 3-е изд. – СПб.: Питер: Киев: Издательская группа BHV, 2004. 847 с.

4. *Wagner G.* AOR modelling and simulation: Towards a general architecture for agent-based discrete event simulation // International Bi-Conference Workshop on Agent-Oriented Information System. – Springer, Berlin, Heidelberg, 2003. 174 – 188.

5. *Welsh M.* The staged event-driven architecture for highly-concurrent server applications // University of California, Berkeley. – 2000.

ИССЛЕДОВАНИЕ ЦИКЛИЧЕСКОЙ СИСТЕМЫ С ПОВТОРНЫМИ ВЫЗОВАМИ

Ключникова П.Н., Пауль С.В.

Томский государственный университет

polya.klyuch@gmail.com, paulsv82@mail.ru

Введение

В настоящее время телекоммуникационные системы, такие как компьютерные, телефонные сети, мобильная связь, call-центры, играют всё большую роль в нашей жизни. Отметим, что системы массового обслуживания являются адекватными математическими моделями вышеперечисленных реальных систем [1,2].

На данный момент есть много работ о моделировании различных телекоммуникационных систем [3,4,5], но большинство из них не рассматривает возможность повторного обращения к прибору. Системы массового обслуживания с повторными вызовами

(RQ-системы) [6] как раз позволяют исследовать подобные системы. Исследования различных RQ-систем доступны для изучения в работах [7,8,9,10].

В данной статье рассматривается циклическая система с повторными вызовами. Для исследования используется метод асимптотического анализа [11], применение которого позволяет найти распределение вероятностей числа заявок на орбитах в таких системах.

1. Математическая модель и постановка задачи

Рассмотрим две RQ-системы (рис. 1). На вход каждой системы поступает простейший поток событий с интенсивностью λ_n, $n=1,2$. Заявки каждого потока формируют свою орбиту неограниченного объема. Один прибор обходит две RQ-системы в циклическом порядке, начиная с первой и заканчивая второй, потом цикл повторяется. Время нахождения прибора у n-й RQ-системы имеет экспоненциальную функцию распределения с параметром αn, $n=1,2$. Если поступившая заявка входящего потока обнаруживает прибор занятым или не подключенным, она мгновенно уходит на соответствующую орбиту, где осуществляет случайную задержку в течение экспоненциального времени с параметром σ_n, $n=1,2$, после которой вновь обращается к прибору. В течение этого времени прибор обслуживает заявки, которые поступают из входящего потока и с орбиты. Время обслуживания заявок имеет экспоненциальную функцию распределения с параметрами μ_n, $n=1,2$.

Если заявок на орбите к моменту прихода прибора нет или он обслужил все заявки, которые находились на орбите, и из входящего потока больше не поступили новые заявки, прибор все равно остается подключенным к RQ-системе, пока не истечет время подключения. Методом исследования циклической системы является ее декомпозиция и исследование систем с прогулками прибора.

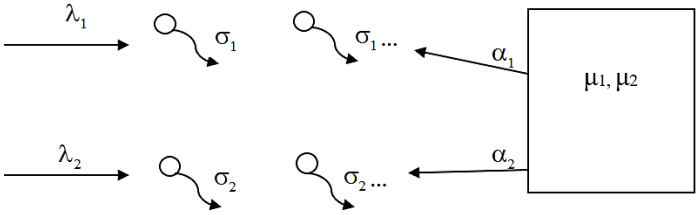


Рис. 1. Циклическая RQ-система

1.1. Система с прогулками

Для исследования циклической RQ-системы, перейдём к системе с прогулками (рис. 2). Переход к системе с прогулками, является одним из методов исследования циклических систем [12].

Рассмотрим RQ-систему с одним обслуживающим прибором и орбитой с неограниченным числом мест для ожидания. В систему поступает простейший поток заявок с интенсивностью λ. Система функционирует в циклическом режиме, цикл которой состоит из двух последовательных интервалов. В течение первого интервала прибор обслуживает заявки, которые поступают из входящего потока с экспоненциальной функцией распределения с параметром μ. Если поступившая заявка из входящего потока обнаруживает прибор занятым, она мгновенно уходит на орбиту, где осуществляет случайную задержку в течение экспоненциального времени с параметром σ, после которой вновь обращается к прибору.

Если заявок на орбите нет к моменту начала этого интервала или прибор обслужил все заявки, которые на этом интервале находились на орбите, то прибор все равно остается в этом режиме, ожидая прихода заявок. От момента окончания этого интервала прибор уходит на «прогулку», в течение второго интервала указанного цикла. Во время

прогулки, пришедшие в систему, заявки накапливаются на орбите и ждут, когда прибор вернется на обслуживание.

Пусть продолжительности этих интервалов случайные и определяются экспоненциальными функциями распределения с параметрами α_1 и α_2 соответственно. Будем рассматривать системы без дообслуживания заявок. Когда прибор уходит на прогулку, недообслуженная заявка уходит обратно на орбиту.

Найдем распределение вероятностей числа заявок на орбите в момент времени t в системе с прогулками прибора, тем самым определим распределение вероятностей числа заявок на орбите в выделенной первой подсистеме циклической RQ-системы.

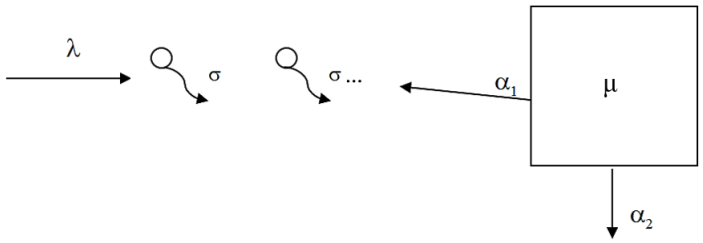


Рис. 2. RQ-система с прогулками прибора

1.2. Система дифференциальных уравнений Колмогорова

Обозначим $i(t)$ – число заявок на орбите в момент времени t, $k(t)$ – состояние прибора: 0 – свободен, 1 – прибор обслуживает заявку, 2 – прибор на прогулке.

Рассмотрим двумерный Марковский процесс $\{k(t), i(t)\}$, для распределения вероятностей $P_k(i,t) = P_k\{i(t) = i\}$ которого составим систему дифференциальных уравнений Колмогорова:

$$\begin{aligned}
\frac{\partial P_0(i,t)}{\partial t} &= -(\lambda + i\sigma + \alpha_1) P_0(i,t) + \mu P_1(i,t) + \alpha_2 P_2(i,t),\\
\frac{\partial P_1(i,t)}{\partial t} &= -(\lambda + \mu + \alpha_1) P_1(i,t) + \lambda P_0(i,t) + (i+1)\sigma P_0(i+1,t) + \lambda P_1(i-1,t),\\
\frac{\partial P_2(i,t)}{\partial t} &= -(\lambda + \alpha_2) P_2(i,t) + \alpha_1 P_1(i-1,t) + \alpha_1 P_0(i,t) + \lambda P_2(i-1,t).
\end{aligned} \qquad (1)$$

Систему (1) запишем в стационарном режиме:

$$\begin{gathered}
-(\lambda + i\sigma + \alpha_1) P_0(i) + \mu P_1(i) + \alpha_2 P_2(i) = 0,\\
-(\lambda + \mu + \alpha_1) P_1(i) + \lambda P_0(i) + (i+1)\sigma P_0(i+1) + \lambda P_1(i-1) = 0,\\
-(\lambda + \alpha_2) P_2(i) + \alpha_1 P_1(i-1) + \alpha_1 P_0(i) + \lambda P_2(i-1) = 0.
\end{gathered}$$

1.3. Метод частичных характеристических функций

Введем частичные характеристические функции: $H_k(u) = \sum_{i=0}^{\infty} e^{jui} P_k(i)$, $k = \overline{0,2}$. Получим систему уравнений для частичных характеристических функций:

$$\begin{aligned}
&-(\lambda + \alpha_1) H_0(u) + j\sigma H_0'(u) + \mu H_1(u) + \alpha_2 H_2(u) = 0,\\
&-(\lambda + \mu + \alpha_1) H_1(u) + \lambda H_0(u) - j\sigma e^{-ju} H_0'(u) + \lambda e^{ju} H_1(u) = 0,\\
&-(\lambda + \alpha_2) H_2(u) + \alpha_1 e^{ju} H_1(u) + \alpha_1 H_0(u) + \lambda e^{ju} H_2(u) = 0.
\end{aligned} \qquad (2)$$

Просуммируем уравнения системы (2):

$$e^{-ju} j\sigma H_0'(u) + (\lambda + \alpha_1) H_1(u) + \lambda H_2(u) = 0. \qquad (3)$$

Систему (2)–(3) будем решать асимптотическим методом в предельном условии $\sigma \to 0$.

1.4. Асимптотика первого порядка

Обозначим $\sigma = \varepsilon$ и выполним в (2) и (3) замены:

$$u = \varepsilon w,\ H_k(u) = F_k(w,\varepsilon),\ k = \overline{0,2}. \tag{4}$$

Запишем систему уравнений для функций $F_k(w,\varepsilon)$:

$$-(\lambda+\alpha_1)F_0(w,\varepsilon) + j\frac{\partial F_0(w,\varepsilon)}{\partial w} + \mu F_1(w,\varepsilon) + \alpha_2 F_2(w,\varepsilon) = 0,$$

$$-(\lambda+\mu+\alpha_1)F_1(w,\varepsilon) + \lambda F_0(w,\varepsilon) - je^{-j\varepsilon w}\frac{\partial F_0(w,\varepsilon)}{\partial w} + \lambda e^{j\varepsilon w}F_1(w,\varepsilon) = 0,$$

$$-(\lambda+\alpha_2)F_2(w,\varepsilon) + \alpha_1 e^{j\varepsilon w}F_1(w,\varepsilon) + \alpha_1 F_0(w,\varepsilon) + \lambda e^{j\varepsilon w}F_2(w,\varepsilon) = 0,$$

$$e^{-ju}j\frac{\partial F_0(w,\varepsilon)}{\partial w} + (\lambda+\alpha_1)F_1(w,\varepsilon) + \lambda F_2(w,\varepsilon) = 0.$$

Устремим $\varepsilon \to 0$ и получим систему:

$$\begin{aligned}
&-(\lambda+\alpha_1)F_0(w) + jF_0'(w) + \mu F_1(w) + \alpha_2 F_2(w) = 0,\\
&-(\mu+\alpha_1)F_1(w) + \lambda F_0(w) - jF_0'(w) = 0,\\
&-\alpha_2 F_2(w) + \alpha_1 F_1(w) + \alpha_1 F_0(w) = 0,\\
&jF_0'(w) + (\lambda+\alpha_1)F_1(w) + \lambda F_2(w) = 0,
\end{aligned} \tag{5}$$

решение которой будем искать в виде:

$$F_k(w) = r_k\Phi(w), \tag{6}$$

где r_k – распределение вероятностей состояний прибора. Подставим выражение (6) в систему (5) и разделим полученную систему на $\Phi(w)$:

$$\begin{aligned}
&-(\lambda+\alpha_1)r_0 + jr_0\frac{\Phi'(w)}{\Phi(w)} + \mu r_1 + \alpha_2 r_2 = 0,\\
&-(\mu+\alpha_1)r_1 + \lambda r_0 - jr_0\frac{\Phi'(w)}{\Phi(w)} = 0,\\
&-\alpha_2 r_2 + \alpha_1 r_1 + \alpha_1 r_0 = 0,\\
&jr_0\frac{\Phi'(w)}{\Phi(w)} + (\lambda+\alpha_1)r_1 + \lambda r_2 = 0.
\end{aligned} \tag{7}$$

Так как отношение $\dfrac{\Phi'(w)}{\Phi(w)}$ не зависит от w, то скалярная функция $\Phi(w)$ имеет вид $\Phi(w) = \exp\{jw\kappa_1\}$, тогда $j\dfrac{\Phi'(w)}{\Phi(w)} = -\kappa_1$. Подставим это значение в систему (7):

$$\begin{aligned}
&-(\lambda+\alpha_1)r_0 - r_0\kappa_1 + \mu r_1 + \alpha_2 r_2 = 0,\\
&-(\mu+\alpha_1)r_1 + \lambda r_0 + r_0\kappa_1 = 0,\\
&-\alpha_2 r_2 + \alpha_1 r_1 + \alpha_1 r_0 = 0,\\
&-r_0\kappa_1 + (\lambda+\alpha_1)r_1 + \lambda r_2 = 0.
\end{aligned} \tag{8}$$

Учитывая условие нормировки для распределения r_k, $k = \overline{0,2}$: $r_0 + r_1 + r_2 = 1$, получим систему уравнений:

$$\begin{aligned}
&-(\lambda+\alpha_1)r_0 - r_0\kappa_1 + \mu r_1 + \alpha_2 r_2 = 0,\\
&-(\mu+\alpha_1)r_1 + \lambda r_0 + r_0\kappa_1 = 0,
\end{aligned}$$

$$-\alpha_2 r_2 + \alpha_1 r_1 + \alpha_1 r_0 = 0,$$
$$-r_0\kappa_1 + (\lambda + \alpha_1) r_1 + \lambda r_2 = 0,$$
$$r_0 + r_1 + r_2 = 1.$$

Решим данную систему:

$$r_2 = \frac{\alpha_1}{\alpha_1 + \alpha_2},\ r_0 = \frac{\mu\alpha_2 - \lambda\alpha_2 - \lambda\alpha_1}{\mu(\alpha_1 + \alpha_2)},\ r_1 = \frac{\lambda}{\mu},\ \kappa_1 = \frac{\lambda\mu\alpha_1 + \lambda\alpha_1\alpha_2 + \lambda\alpha_1^2 + \lambda^2\alpha_2 + \lambda^2\alpha_1}{\mu\alpha_2 - \lambda\alpha_2 - \lambda\alpha_1}, \quad (9)$$

(9) – решение данной системы.

1.5. Асимптотика второго порядка

В системе уравнений (4)–(5) сделаем замены: $H_k(u) = \exp\left(j\frac{u}{\sigma}\kappa_1\right) H_k^{(2)}(u)$, получим систему:

$$\begin{aligned}
&-(\lambda + \alpha_1 + \kappa_1) H_0^{(2)}(u) + j\sigma \frac{dH_0^{(2)}(u)}{du} + \mu H_1^{(2)}(u) + \alpha_2 H_2^{(2)}(u) = 0,\\
&-\left(\mu + \alpha_1 + \lambda\left(1 - e^{ju}\right)\right) H_1^{(2)}(u) + \left(\lambda + \kappa_1 e^{-ju}\right) H_0^{(2)}(u) - j\sigma e^{-ju} \frac{dH_0^{(2)}(u)}{du} = 0,\\
&-\left(\alpha_2 + \lambda\left(1 - e^{ju}\right)\right) H_2^{(2)}(u) + \alpha_1 e^{ju} H_1^{(2)}(u) + \alpha_1 H_0^{(2)}(u) = 0,\\
&-\kappa_1 e^{-ju} H_0^{(2)}(u) + j\sigma e^{-ju} \frac{dH_0^{(2)}(u)}{du} + (\lambda + \alpha_1) H_1^{(2)}(u) + \lambda H_2^{(2)}(u) = 0.
\end{aligned} \quad (10)$$

Обозначим $\sigma = \varepsilon^2$ и в системе (10) сделаем замены: $u = \varepsilon w$, $H_k^{(2)}(u) = F_k^{(2)}(w, \varepsilon)$, $k = \overline{0,2}$, получим систему уравнений:

$$\begin{aligned}
&-(\lambda + \alpha_1 + \kappa_1) F_0^{(2)}(w,\varepsilon) + j\varepsilon \frac{\partial F_0^{(2)}(w,\varepsilon)}{\partial w} + \mu F_1^{(2)}(w,\varepsilon) + \alpha_2 F_2^{(2)}(w,\varepsilon) = 0,\\
&-\left(\mu + \alpha_1 + \lambda\left(1 - e^{j\varepsilon w}\right)\right) F_1^{(2)}(w,\varepsilon) + \left(\lambda + \kappa_1 e^{-j\varepsilon w}\right) F_0^{(2)}(w,\varepsilon) - j\varepsilon e^{-j\varepsilon w} \frac{\partial F_0^{(2)}(w,\varepsilon)}{\partial w} = 0,\\
&-\left(\alpha_2 + \lambda\left(1 - e^{j\varepsilon w}\right)\right) F_2^{(2)}(w,\varepsilon) + \alpha_1 e^{j\varepsilon w} F_1^{(2)}(w,\varepsilon) + \alpha_1 F_0^{(2)}(w,\varepsilon) = 0,\\
&-\kappa_1 e^{-j\varepsilon w} F_0^{(2)}(w,\varepsilon) + j\varepsilon e^{-j\varepsilon w} \frac{\partial F_0^{(2)}(w,\varepsilon)}{\partial w} + (\lambda + \alpha_1) F_1^{(2)}(w,\varepsilon) + \lambda F_2^{(2)}(w,\varepsilon) = 0.
\end{aligned}$$

В полученную систему подставим разложение: $F_k^{(2)}(w,\varepsilon) = \Phi_2(w)\{r_k + j\varepsilon w f_k\} + O(\varepsilon^2)$, получим систему уравнений:

$$\begin{aligned}
&\left(-(\lambda + \alpha_1 + \kappa_1) r_0 + \mu r_1 + \alpha_2 r_2\right)\Phi_2(w) + j\varepsilon r_0 \frac{d\Phi_2(w)}{dw} +\\
&\quad + j\varepsilon w\left(-(\lambda + \alpha_1 + \kappa_1) f_0 + \mu f_1 + \alpha_2 f_2\right)\Phi_2(w) = O(\varepsilon^2),\\
&\left(-(\mu + \alpha_1) r_1 + r_0\lambda + r_0\kappa_1\right)\Phi_2(w) +\\
&\quad + j\varepsilon w\left(-(\mu + \alpha_1) f_1 + \lambda r_1 - r_0\kappa_1 + f_0(\lambda + \kappa_1)\right) - j\varepsilon r_0 \frac{d\Phi_2(w)}{dw} = O(\varepsilon^2),\\
&\left(-\alpha_2 r_2 + \alpha_1 r_1 + \alpha_1 r_0\right)\Phi_2(w) +\\
&\quad + j\varepsilon w\left(-(\alpha_2 f_2 - r_2\lambda) + \alpha_1 r_1 + \alpha_1 f_1 + \alpha_1 f_0\right)\Phi_2(w) = O(\varepsilon^2),\\
&\left(-r_0\kappa_1 + (\lambda + \alpha_1) r_1 + \lambda r_2\right)\Phi_2(w) +\\
&\quad + j\varepsilon w\left(\kappa_1 r_0 - \kappa_1 f_0 + (\lambda + \alpha_1) f_1 + \lambda f_2\right)\Phi_2(w) + j\varepsilon r_0 \frac{d\Phi_2(w)}{dw} = O(\varepsilon^2).
\end{aligned} \quad (11)$$

В систему (11) подставим (8), получим:

$$\begin{aligned}
&j\varepsilon r_0 \frac{d\Phi_2(w)}{dw} + j\varepsilon w\left(-\left(\lambda+\alpha_1+\kappa_1\right)f_0 + \mu f_1 + \alpha_2 f_2\right)\Phi_2(w) = O\left(\varepsilon^2\right),\\
&j\varepsilon w\left(-\left(\mu+\alpha_1\right)f_1 + \lambda r_1 - r_0\kappa_1 + f_0\left(\lambda+\kappa_1\right)\right) - j\varepsilon r_0 \frac{d\Phi_2(w)}{dw} = O\left(\varepsilon^2\right),\\
&j\varepsilon w\left(-(\alpha_2 f_2 - r_2\lambda) + \alpha_1 r_1 + \alpha_1 f_1 + \alpha_1 f_0\right)\Phi_2(w) = O\left(\varepsilon^2\right),\\
&j\varepsilon w\left(\kappa_1 r_0 - \kappa_1 f_0 + \left(\lambda+\alpha_1\right)f_1 + \lambda f_2\right)\Phi_2(w) + j\varepsilon r_0 \frac{d\Phi_2(w)}{dw} = O\left(\varepsilon^2\right).
\end{aligned} \tag{12}$$

В системе (12) разделим уравнения на $j\varepsilon w$ и устремим $\varepsilon \to 0$, имеем:

$$\begin{aligned}
&-\left(\lambda+\alpha_1+\kappa_1\right)f_0 + \mu f_1 + \alpha_2 f_2 + r_0 \frac{\Phi_2'(w)}{w\Phi_2(w)} = 0,\\
&-\left(\mu+\alpha_1\right)f_1 + \lambda r_1 - r_0\kappa_1 + f_0\left(\lambda+\kappa_1\right) - r_0 \frac{\Phi_2'(w)}{w\Phi_2(w)} = 0,\\
&-\left(\alpha_2 f_2 - r_2\lambda\right) + \alpha_1 r_1 + \alpha_1 f_1 + \alpha_1 f_0 = 0,\\
&\left(\kappa_1 r_0 - \kappa_1 f_0 + \left(\lambda+\alpha_1\right)f_1 + \lambda f_2\right) + r_0 \frac{\Phi_2'(w)}{w\Phi_2(w)} = 0.
\end{aligned} \tag{13}$$

Так как отношение $\dfrac{\Phi_2'(w)}{w\Phi_2(w)}$ не зависит от w, то скалярная функция $\Phi_2(w)$ имеет вид $\Phi_2(w) = \exp\left\{\dfrac{(jw)^2}{2}\kappa_2\right\}$, тогда $\dfrac{\Phi_2'(w)}{w\Phi_2(w)} = -\kappa_2$. Подставим это значение в систему (13):

$$\begin{aligned}
&-\left(\lambda+\alpha_1+\kappa_1\right)f_0 + \mu f_1 + \alpha_2 f_2 = r_0\kappa_2,\\
&\left(\lambda+\kappa_1\right)f_0 - \left(\mu+\alpha_1\right)f_1 = r_0\kappa_1 - r_0\kappa_2 - \lambda r_1,\\
&\alpha_1 f_0 + \alpha_1 f_1 - \alpha_2 f_2 = -r_2\lambda - \alpha_1 r_1,\\
&-\kappa_1 f_0 + \left(\lambda+\alpha_1\right)f_1 + \lambda f_2 = \kappa_2 r_0 - \kappa_1 r_0.
\end{aligned} \tag{14}$$

Система (14) – неоднородная система линейных алгебраических уравнений для f_k, $k = \overline{0,2}$. Определитель матрицы коэффициентов системы равен 0, при этом ранг расширенной матрицы равен рангу матрицы коэффициентов, т.е. система совместна и имеет множество решений.

Рассмотрим однородную систему уравнений (8) и неоднородную систему (14). Сравнивая эти системы, можно увидеть, что система (8) является однородной системой для неоднородной системы (14). Тогда решение системы (14) можно записать в виде $f_k\left(\kappa_2\right) = Cr_k + g_k\left(\kappa_2\right)$, $k = \overline{0,2}$, где C – константа, вероятности r_k определены выше, а величины $g_k\left(\kappa_2\right)$ являются частными решениями неоднородной системы (14), которое удовлетворяет условию $\sum_{k=0}^{2} g_k\left(\kappa_2\right) = 0$, т.е.

$$
\begin{aligned}
&-(\lambda+\alpha_1+\kappa_1)\,g_0(\kappa_2)+\mu g_1(\kappa_2)+\alpha_2 g_2(\kappa_2)=r_0\kappa_2,\\
&(\lambda+\kappa_1)\,g_0(\kappa_2)-(\mu+\alpha_1)\,g_0(\kappa_2)=r_0\kappa_1-r_0\kappa_2-\lambda r_1,\\
&\alpha_1 g_0(\kappa_2)+\alpha_1 g_1(\kappa_2)-\alpha_2 g_2(\kappa_2)=-r_2\lambda-\alpha_1 r_1,\\
&-\kappa_1 g_0(\kappa_2)+(\lambda+\alpha_1)\,g_1(\kappa_2)+\lambda g_2(\kappa_2)=\kappa_2 r_0-\kappa_1 r_0,\\
&\sum_{k=0}^{2} g_k(\kappa_2)=0.
\end{aligned}
\qquad (15)
$$

Решив систему (15), полученные величины $g_k(\kappa_2)$ подставим в выражения для $f_k(\kappa_2)$, а затем в уравнение для нахождения величины κ_2: $-\kappa_1 f_0+(\lambda+\alpha_1)f_1+\lambda f_2=r_0\kappa_2-r_0\kappa_1$. Из последнего уравнения мы найдем величину κ_2.

Заключение

В данной работе ставилась задача нахождения распределения вероятностей числа заявок на орбите в выделенной подсистеме с повторными вызовами в циклической системе. Задача решена классическим методом «систем с прогулками прибора». Получено распределение вероятностей числа заявок на орбите в системе с прогулками прибора методом асимптотического анализа. Найденное асимптотическое распределение вероятностей числа $i(t)$ является гауссовским с асимптотическим средним κ_1/σ и дисперсией κ_2/σ. Определив гауссовскую плотность распределения вероятностей с этими параметрами мы получим распределение вероятностей числа заявок на орбите в RQ-системе с прогулками прибора в асимптотическом условии большой задержки заявок на орбите. Найденное распределение позволяет выполнить исследование числа заявок на орбите в циклических системах с двумя входящими потоками, в которых время прогулки прибора от одной очереди равно продолжительности периода занятости прибора обслуживанием заявок второй пары потока и орбиты.

ЛИТЕРАТУРА

1. *Alexandre Deslauriers.* Markov chain models of a telephone call center with call blending / Alexandre Deslauriers, Pierre L'Ecuyer, Juta Pichitlamken [et al.]. // Computers & operations research. – 2007. – Vol. 34, no. 6. – P. 1616–1645.

2. *Gilmore Audrey.* Call centres: how can service quality be managed? / Gilmore Audrey, Moreland Lesley. // Irish Marketing Review. – 2000. – Vol. 13, no. 1. – P. 3.

3. *Гнеденко Б.В.* Введение в теорию массового обслуживания / Б. В. Гнеденко, И. Н. Коваленко. – М.: Наука, 1987. – 336 с.

4. *Назаров А.А.* Теория массового обслуживания: [учебное пособие по специальностям 010200 (010501)"Прикладная математика и информатика 061800 (080116)"Математические методы в экономике"]. / А. А. Назаров, А. Ф. Терпугов. – Томск: Изд-во НТЛ, 2010. – 228 с.

5. *Хинчин А.Я.* Работы по математической теории массового обслуживания. / А. Я. Хинчин. – Москва, 1963. – Т. 236.

6. *Artalejo Jesus` R.* Retrial Queueing Systems: A Computational Approach / Artalejo Jesus` R., Gomez-Corral´ Antonio. – Springer-Verlag Berlin Heidelberg, 2008. – 318 с.

7. *Пауль С.В.* Анализ RQ-системы M/GI/GI/1/1 с вызываемыми заявками, ненадежным прибором и дообслуживанием прерванных заявок // Информационные технологии и математическое моделирование (ИТММ-2018) : материалы XVII Международной конференции имени А. Ф. Терпугова, 10-15 сентября 2018 г. / С. В. Пауль, А. А. Назаров. – Томск, 2018. – С. 139-145.

8. *Назаров А.А.* Асимптотический анализ RQ-системы с N типами вызываемых заявок в предельном условии большой задержки заявок на орбите // Вестник Томского государственного университета. Серия: Управление, вычислительная техника и информатика / Назаров А. А., Пауль С. В., Лизюра О. Д. – Томск, 2019. – С. 13–20.

9. *Назаров А.А.* Исследование RQ-системы M|M|1|1 с вызываемыми заявками методом асимптотически-диффузионного анализа // Распределенные компьютерные и телекоммуникационные сети: управление, вычисление, связь (DCCN2019) / А. А. Назаров, С. В. Пауль, О. Д. Лизюра. – Томск, 2019. – С. 148-155.

10. *Лизюра О.Д.* Асимптотический анализ RQ-системы MMPP|M|1 с N типами вызываемых заявок в условии предельно редких изменений состояний входящего потока // Математическое и программное обеспечение информационных, технических и экономических систем / О. Д. Лизюра, А. А. Назаров, С. В. Пауль. – Томск, 2019. – С. 241-246.

11. *Лопухова С.В.* Асимптотические и численные методы исследования специальных потоков однородных событий : дис. . . . канд. наук / С. В. Лопухова. – Том. гос. ун-та, 2008. – 167 с.

12. *Назаров А.А.* Исследование системы массового обслуживания с "прогулками" прибора, управляемой Т-стратегией // Теория вероятностей, случайные процессы, математическая статистика и приложения : материалы Международной научной конференции, посвященной 80-летию профессора, доктора физико-математических наук Геннадия Алексеевича Медведева, Минск, 23-26 февраля 2015 г. / А. А. Назаров, С. В. Пауль. – Минск, 2015. – 202-207 с.

МОДЕЛИ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ СВЯЗИ В ВИДЕ СИСТЕМ С ПОВТОРНЫМИ ВЫЗОВАМИ И ВЫЗЫВАЕМЫМИ ЗАЯВКАМИ

Морозова М.А., Пауль С.В., Назаров А.А.
Томский государственный университет
morozova_maria_a@mail.ru, paulsv82@mail.ru, anazarov@fpmk.tsu.ru

Введение

В настоящее время теория массового обслуживания является обширной областью для проведения исследований. Она очень востребована во многих областях науки и продолжает стремительно развиваться.

Чтобы получить представление о случайных процессах, протекающих в системах массового обслуживания, прибегают к их моделированию. Особой популярностью обладают модели call-центров с повторными звонками. Повторные звонки – это повторные обращения к оператору. Системы с повторными звонками называются системами с орбитой (RQ-системы) [1].

Однако call-центры занимаются не только приемом входящих звонков от клиентов, но и производят исходящие вызовы с целью рекламы, проведения опросов или анкетирования (вызываемые заявки). RQ-системы с вызываемыми заявками впервые упомянул Фалин [2]. Такие модели очень популярны в виду их гибкости.

Немало работ посвящено исследованию RQ-систем с вызываемыми заявками: в работе [3] разрабатывают рекуррентный алгоритм для нахождения распределения вероятностей числа заявок на орбите, в [4] исследование проводится методом векторно-матричной алгебры.

В данной статье рассматриваются RQ-системы с вызываемыми заявками вида M|M|1|1. Исследование проводится с помощью метода асимптотического анализа [5] при условии большой задержки заявок на орбите, который позволяет найти характеристики системы.

1. Описание математической модели и постановка задачи

Рассмотрим систему с повторными вызовами (RQ-систему) с одним обслуживающим прибором, на вход которой поступает простейший поток заявок с интенсивностью λ (рис. 1). Если прибор свободен, то поступившая заявка встает на прибор и обслуживается в течение экспоненциально-распределенного времени с параметром μ_1. Если заявка застает прибор занятым, она мгновенно отправляется на орбиту, где после случайной экспоненциально-распределенной задержки с параметром σ снова пытается встать на прибор. Когда прибор свободен, он может вызывать заявки извне с интенсивностью α, которые обслуживаются в течение времени, распределенного по экспоненциальному закону с параметром μ_2.

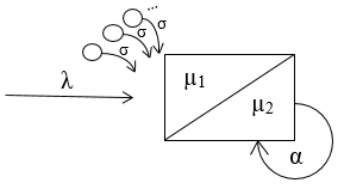


Рис. 1. RQ-система M|M|1|1 с вызываемыми заявками

Введем следующие обозначения: процесс $k(t)$ – состояние прибора в момент времени *t*: 0, если прибор свободен; 1, если прибор занят обслуживанием заявки входящего потока; 2, если прибор обслуживает вызываемую заявку; процесс $i(t)$ – число заявок, находящихся на орбите в момент времени *t*.

Ставится задача нахождения распределения вероятностей числа заявок $i(t)$ на орбите.

2. Система дифференциальных уравнений Колмогорова

Обозначим вероятности $P_k(i,t) = P\{k(t) = k, i(t) = i\}$, $k = \overline{0,2}$. Случайный процесс $\{k(t), i(t)\}$, $k = \overline{0,2}$ является марковским, поэтому для распределения вероятностей составим систему дифференциальных уравнений Колмогорова

$$\frac{\partial P_0(i,t)}{\partial t} = -(\lambda + i\sigma + \alpha)P_0(i,t) + \mu_1 P_1(i,t) + \mu_2 P_2(i,t),$$

$$\frac{\partial P_1(i,t)}{\partial t} = \lambda P_0(i,t) + (i+1)\sigma P_0(i+1,t) - (\mu_1 + \lambda)P_1(i,t) + \lambda P_1(i-1,t),$$

$$\frac{\partial P_2(i,t)}{\partial t} = \alpha P_0(i,t) - (\mu_2 + \lambda)P_2(i,t) + \lambda P_2(i-1,t).$$

Перепишем данную систему в стационарном виде

$$-(\lambda + i\sigma + \alpha)P_0(i) + \mu_1 P_1(i) + \mu_2 P_2(i) = 0,$$

$$\lambda P_0(i) + (i+1)\sigma P_0(i+1) - (\lambda + \mu_1)P_1(i) + \lambda P_1(i-1) = 0,$$

$$\alpha P_0(i) - (\lambda + \mu_2)P_2(i) + \lambda P_2(i-1) = 0.$$

Чтобы решить систему, перейдем к частичным характеристическим функциям вида $H_k(u) = \sum_{i=0}^{\infty} e^{jui} P_k(i)$, $k = \overline{0,2}$. Имеем следующую систему уравнений:

$$\begin{aligned}
&-(\lambda + \alpha)H_0(u) + j\sigma\frac{dH_0(u)}{du} + \mu_1 H_1(u) + \mu_2 H_2(u) = 0,\\
&\lambda H_0(u) - j\sigma e^{-ju}\frac{dH_0(u)}{du} + \left(\lambda\left(e^{ju} - 1\right) - \mu_1\right)H_1(u) = 0,\\
&\alpha H_0(u) - \left(\lambda\left(1 - e^{ju}\right) + \mu_2\right)H_2(u) = 0,\\
&j\sigma e^{-ju}\frac{dH_0(u)}{du} + \lambda H_1(u) + \lambda H_2(u) = 0.
\end{aligned} \qquad (1)$$

Основным содержанием работы является решение системы (1) методом асимптотического анализа в предельном условии большой задержки на орбите, когда $\sigma \to 0$. Характеристическая функция $H(u)$ числа заявок на орбите и распределение вероятностей состояний прибора r_k достаточно просто выражаются через частичные характери-

стические функции $H_k(u)$ следующими равенствами: $H(u) = Me^{juli(t)} = H_0(u) + H_1(u) + H_2(u)$, $r_k = H_k(0)$, $k = \overline{0,2}$.

3. Метод асимптотического анализа

Систему уравнений (1) будем решать методом асимптотического анализа в условии большой задержки, когда среднее значение времени задержки требования на орбите неограниченно возрастает.

Обозначим $\sigma = \varepsilon u = \varepsilon w$, $H_k(u) = F_k(w,\varepsilon)$, $k = \overline{0,2}$, перепишем (1) с учетом введенных обозначений

$$-(\lambda+\alpha)F_0(w,\varepsilon) + j\varepsilon\frac{\partial F_0(w,\varepsilon)}{\partial w} + \mu_1 F_1(w,\varepsilon) + \mu_2 F_2(w,\varepsilon) = 0,$$

$$\lambda F_0(w,\varepsilon) - j\varepsilon e^{-jw\varepsilon}\frac{\partial F_0(w,\varepsilon)}{\partial w} + \left(\lambda\left(e^{jw\varepsilon}-1\right)-\mu_1\right)F_1(w,\varepsilon) = 0,$$

$$\alpha F_0(w,\varepsilon) + \left(\lambda\left(e^{jw\varepsilon}-1\right)-\mu_2\right)F_2(w,\varepsilon) = 0,$$

$$j\varepsilon e^{-jw\varepsilon}\frac{\partial F_0(w,\varepsilon)}{\partial w} + \lambda F_1(w,\varepsilon) + \lambda F_2(w,\varepsilon) = 0.$$

Выполним в данной системе уравнений предельный переход при $\varepsilon \to 0$ и получим систему для функций $F_k(w)$

$$\begin{aligned}
&-(\lambda+\alpha)F_0(w) + j\frac{dF_0(w)}{dw} + \mu_1 F_1(w) + \mu_2 F_2(w) = 0,\\
&\lambda F_0(w) - j\frac{dF_0(w)}{dw} - \mu_1 F_1(w) = 0,\\
&\alpha F_0(w) - \mu_2 F_2(w) = 0,\\
&j\frac{dF_0(w)}{dw} + \lambda F_1(w) + \lambda F_2(w) = 0,
\end{aligned} \tag{2}$$

решение которой будем искать в виде

$$F_k(w) = r_k\Phi(w),\ k = \overline{0,2}, \tag{3}$$

где r_k – вероятности состояний прибора, для которых выполняется равенство $\sum_{k=0}^{2} r_k = 1$.

Подставим выражение (3) в систему (2), получим

$$-(\lambda+\alpha)r_0 + jr_0\frac{\Phi'(w)}{\Phi(w)} + \mu_1 r_1 + \mu_2 r_2 = 0,$$

$$\lambda r_0 - jr_0\frac{\Phi'(w)}{\Phi(w)} - \mu_1 r_1 = 0,$$

$$\alpha r_0 - \mu_2 r_2 = 0,$$

$$jr_0\frac{\Phi'(w)}{\Phi(w)} + \lambda r_1 + \lambda r_2 = 0.$$

Т.к. отношение $\frac{\Phi'(w)}{\Phi(w)}$ не зависит от w, то скалярная функция $\Phi(w)$ имеет вид $\Phi(w) = \exp\{jw\kappa_1\}$. Тогда $j\frac{\Phi'(w)}{\Phi(w)} = -\kappa_1$. Подставим это значение в систему:

$$
\begin{aligned}
&-(\lambda+\alpha)r_0-\kappa_1 r_0+\mu_1 r_1+\mu_2 r_2=0,\\
&\lambda r_0+\kappa_1 r_0-\mu_1 r_1=0,\\
&\alpha r_0-\mu_2 r_2=0,\\
&-\kappa_1 r_0+\lambda r_1+\lambda r_2=0.
\end{aligned}
\qquad (4)
$$

Решим следующую систему, заменив первое уравнение, являющееся суммой второго и третьего уравнений, условием нормировки $r_0+r_1+r_2=1$:

$$
r_0=\frac{\mu_2(\mu_1-\lambda)}{\mu_1(\mu_2+\alpha)},\ r_1=\frac{\lambda}{\mu_1},\ r_2=\frac{\alpha(\mu_1-\lambda)}{\mu_1(\mu_2+\alpha)},\ \kappa_1=\frac{\lambda(\lambda\mu_2+\alpha\mu_1)}{\mu_2(\mu_1-\lambda)}.
$$

Таким образом, мы получили асимптотическое среднее значения числа заявок на орбите κ_1/σ в допредельной ситуации ненулевых значений σ. Для более детального исследования числа $i(t)$ заявок на орбите рассмотрим асимптотику второго порядка.

В систему (1) подставим следующее выражение: $H_k(u)=\exp\left(j\frac{u}{\sigma}\kappa_1\right)H_k^{(2)}(u)$, $k=\overline{0,2}$, после обозначим $\sigma=\varepsilon^2$, $u=\varepsilon w$, $H_k^{(2)}(u)=F_k^{(2)}(w,\varepsilon)$, $k=\overline{0,2}$, и получим систему

$$
-(\lambda+\alpha+\kappa_1)F_0^{(2)}(w,\varepsilon)+j\varepsilon\frac{\partial F_0^{(2)}(w,\varepsilon)}{\partial w}+\mu_1 F_1^{(2)}(w,\varepsilon)+\mu_2 F_2^{(2)}(w,\varepsilon)=0,
$$

$$
\lambda F_0^{(2)}(w,\varepsilon)+\kappa_1 e^{-j\varepsilon w}F_0^{(2)}(w,\varepsilon)-j\varepsilon e^{-j\varepsilon w}\frac{\partial F_0^{(2)}(w,\varepsilon)}{\partial w}+\left(\lambda\left(e^{j\varepsilon w}-1\right)-\mu_1\right)F_1^{(2)}(w,\varepsilon)=0,
$$

$$
\alpha F_0^{(2)}(w,\varepsilon)-\left(\lambda\left(1-e^{j\varepsilon w}\right)+\mu_2\right)F_2^{(2)}(w,\varepsilon)=0,
$$

$$
-\kappa_1 e^{-j\varepsilon w}F_0^{(2)}(w,\varepsilon)+j\varepsilon e^{-j\varepsilon w}\frac{\partial F_0^{(2)}(w,\varepsilon)}{\partial w}+\lambda F_1^{(2)}(w,\varepsilon)+\lambda F_2^{(2)}(w,\varepsilon)=0.
$$

Подставим в нее следующее разложение: $F_k^{(2)}(w,\varepsilon)=\Phi_2(w)\{r_k+j\varepsilon w f_k\}+O(\varepsilon^2)$, и перепишем систему, учитывая систему алгебраических уравнений (4):

$$
-(\lambda+\alpha+\kappa_1)f_0+f_1\mu_1+f_2\mu_2+\frac{\Phi_2'(w)}{w\Phi_2(w)}r_0=0,
$$

$$
f_0(\lambda+\kappa_1)-\mu_1 f_1-\kappa_1 r_0+\lambda r_1-\frac{\Phi_2'(w)}{w\Phi_2(w)}r_0=0,
$$

$$
f_0\alpha-\mu_2 f_2=-\lambda r_2,
$$

$$
-\kappa_1 f_0+\lambda f_1+\lambda f_2=-\frac{\Phi_2'(w)}{w\Phi_2(w)}r_0-\kappa_1 r_0.
$$

Т.к. отношение $\frac{d\Phi_2(w)/dw}{w\Phi_2(w)}$ не зависит от w, то скалярная функция $\Phi_2(w)$ имеет вид $\Phi_2(w)=\exp\left\{\frac{(jw)^2}{2}\kappa_2\right\}$, тогда $\frac{\Phi_2'(w)}{w\Phi_2(w)}=-\kappa_2$. Подставим это значение в систему:

$$
-(\lambda+\alpha+\kappa_1)f_0+f_1\mu_1+f_2\mu_2=\kappa_2 r_0,
$$

$$
f_0(\lambda+\kappa_1)-\mu_1 f_1=\kappa_1 r_0-\lambda r_1-\kappa_2 r_0,
$$

$$
f_0\alpha-\mu_2 f_2=-\lambda r_2,
$$

$$
-\kappa_1 f_0+\lambda f_1+\lambda f_2=\kappa_2 r_0-\kappa_1 r_0.
$$

Решим ее и получим: $\kappa_2=\frac{\lambda^3\mu_2^2+\lambda^3\mu_2\alpha+\lambda^2\alpha\mu_1^2-\lambda^3\alpha\mu_1}{\mu_2^2(\mu_1-\lambda)^2}+\kappa_1$.

Асимптотика второго порядка показывает, что асимптотическое распределение вероятностей числа $i(t)$ заявок на орбите является гауссовским с асимптотическим средним κ_1/σ и дисперсией κ_2/σ, что позволяет для допредельного распределения построить аппроксимацию.

4. Численные реализации

Рассмотрим два случая построения аппроксимаций. В частности, построим аппроксимацию $P^{(1)}(i)$ вида

$$P^{(1)}(i) = \left(L(i+0.5) - L(i-0.5)\right)\left(1 - L(-0.5)\right)^{-1}, \tag{5}$$

где $L(x)$ – функция нормального распределения с параметрами κ_1/σ и κ_2/σ.

Второй случай получения аппроксимации основан на обратном преобразовании Фурье от характеристической функции $H(u)$ числа заявок на орбите, которая выражаются через частичные характеристические функции $H_k(u)$ следующими равенствами: $H(u) = Me^{jui(t)} = H_0(u) + H_1(u) + H_2(u)$. Тогда распределение вероятностей числа заявок на орбите является обратным преобразованием Фурье по переменной *u* от характеристической функции и имеет вид

$$P^{(2)}(i) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-jui} H(u)du\,. \tag{6}$$

Аппроксимации сравним с полученной ранее допредельной характеристической функций числа заявок на орбите к предложенной системе [6].

$$H(u) = \frac{1-\rho\left(e^{ju}-1\right)}{1+\nu_1}\left(\left(1+\nu_1\left(\frac{1-p}{1-pe^{ju}}\right)^{-1}\right)\left(\frac{1-\rho}{1-\rho e^{ju}}\right)^{\frac{\lambda}{\sigma}(1+\nu_2)+1}\left(\frac{1-p}{1-pe^{ju}}\right)^{\frac{\alpha(\theta-\lambda)}{\sigma\theta}}\right),$$

где $\rho = \dfrac{\lambda}{\mu_1}$, $p = \dfrac{\lambda}{\mu_2+\lambda}$, $\theta = \lambda + \mu_2 - \mu_1$, $\nu_1 = \dfrac{\alpha}{\mu_2}$, $\nu_2 = \dfrac{\alpha}{\theta}$. Ее так же преобразуем с помощью обратного преобразования Фурье. Точность полученных аппроксимаций определим расстоянием Колмогорова $\Delta_l = \max\limits_{0\le i\le\infty}\left|\sum\limits_{v=0}^{i}\left(P(v) - P^{(l)}(v)\right)\right|$, $l = 1,2$ между распределениями $P(i)$ и $P^{(l)}(i)$, где распределение $P(i)$ определяется реализацией численного алгоритма, а аппроксимации $P^{(l)}(i)$ построены на основе второй асимптотики. $P^{(1)}(i)$ – гауссовская аппроксимация на основе формулы (5), а $P^{(2)}(i)$ получена с помощью обратного преобразования Фурье (6).

В табл. 1 приведены значения этих расстояний Δ_1 и Δ_2 для различных параметров σ и $\rho = \lambda/\mu_1$ (загрузка системы). В табл. 1 полагаем $\mu_1 = 1$, $\mu_2 = 2$ и $\alpha = 1$.

Таблица 1

Расстояние Колмогорова

σ	ρ = 0.5	ρ = 0.6	ρ = 0.7	ρ = 0.8	ρ = 0.9	
1	0.137	0.089	0.110	0.131	0.157	Δ_1
	0.170	0.136	0.134	0.128	0.112	Δ_2
0.5	0.084	0.067	0.081	0.098	0.118	Δ_1
	0.115	0.086	0.074	0.062	0.076	Δ_2
0.1	0.028	0.033	0.040	0.048	0.057	Δ_1

σ	ρ = 0.5	ρ = 0.6	ρ = 0.7	ρ = 0.8	ρ = 0.9	
	0.025	0.031	0.039	0.047	0.057	Δ_2
0.05	0.022	0.024	0.028	0.034	0.040	Δ_1
	0.022	0.024	0.028	0.034	0.040	Δ_2

Табл. 1 содержит значения величин Δ_1 и Δ_2 в зависимости от загрузки системы ρ и интенсивности повторных вызовов σ. Из таблицы видно, что точность аппроксимаций растет с уменьшением параметров ρ и σ. Обе аппроксимации применимы для значений $\sigma < 0.05$, когда относительная погрешность, в виде расстояния Колмогорова, не превышает 0.05.

Кроме того, из таблицы видно, что аппроксимация, полученная с помощью обратного преобразования Фурье, при значениях $0.05 \le \sigma < 0.1$ дает более точные результаты, чем гауссовская аппроксимация. При $\sigma \le 0.05$ расстояния Колмогорова для аппроксимаций становятся равны между собой, поэтому при условии большой задержки на орбите обе аппроксимации применимы.

На рисунках сплошной линией показано допредельное распределение числа заявок на орбите, точками – аппроксимация $P^{(1)}(i)$ и пунктиром – аппроксимация $P^{(2)}(i)$, полученная с помощью обратного преобразования Фурье. Значения параметров: $\mu_1 = 1$, $\mu_2 = 2$, $\alpha = 1$.

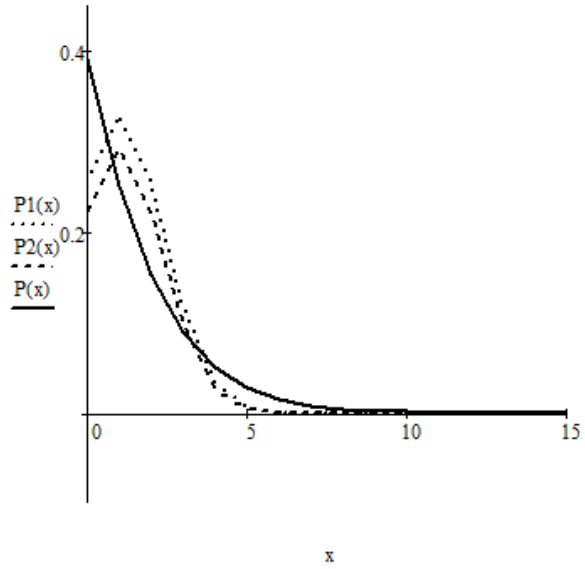


Рис. 2. Распределение вероятностей числа заявок на орбите σ = 1, ρ = 0.5

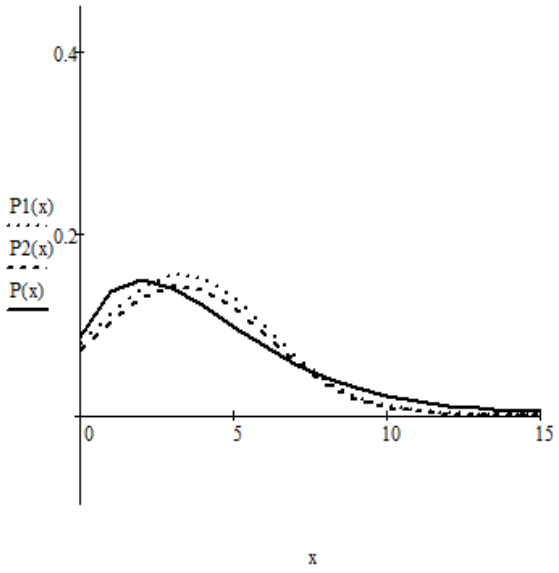


Рис. 3. Распределение вероятностей числа заявок на орбите σ = 0.5, ρ = 0.6

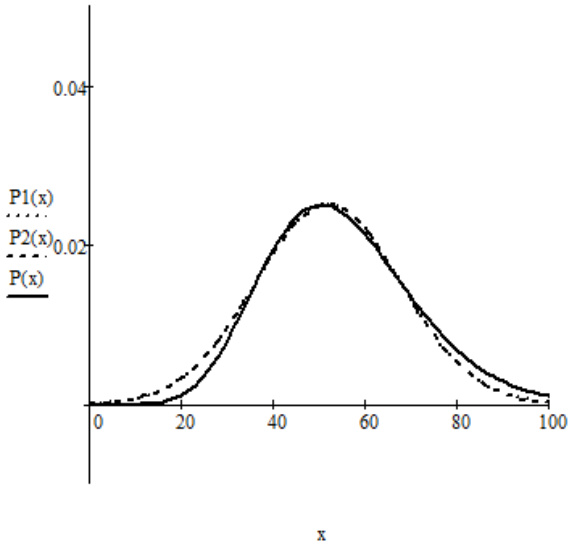


Рис. 4. Распределение вероятностей числа заявок на орбите σ = 0.1, ρ = 0.8

Заключение

В данной курсовой работе мы рассмотрели RQ-систему M|M|1 с вызываемыми заявками. Были найдены асимптотики первого и второго порядков числа заявок на орбите в асимптотическом условии низкой интенсивности повторного вызова заявок с орбиты. На основе полученных асимптотик были построены гауссовские аппроксимации распределения вероятностей числа заявок на орбите двумя способами. При сравнении с допредельным распределением показано, что точность аппроксимаций растет с уменьшением параметров σ и загрузки системы.

ЛИТЕРАТУРА

1. *Deslauriers A., L'Ecuyer P., Pichitlamken J. ,Ingolfsson A., Avramidis A.N.* Markov chain models of a telephone call center with call blending // Computers and Operations Research. 2007. V. 34. P. 1616-1645.

2. *Falin G.I., Artalejo J.R., Martin M.* On the single server retrial queue with priority customers // Queueing Systems. 1993. V. 14. P. 439-455.

3. *Artalejo J.R., Phung-Duc T.* Markovian retrial queues with two way communication // Journal of Industrial & Management Optimization. – 2012. – Vol. 8, no. 4. – P. 781–806.

4. *Phung-Duc Tuan, Rogiest Wouter.* Two way communication retrial queues with balanced call blending // International Conference on Analytical and Stochastic Modeling Techniques and Applications / Springer. – 2012. – P. 16–31.

5. *Назаров А.А., Моисеева С.П.* Метод асимптотического анализа в теории массового обслуживания. – Томск: Изд-во НТЛ. 2006. – 117 с.

6. *Nazarov A., Paul S., Lizyura O.* Heavy outgoing call asymptotics for retrial queue with two way communication and multiple types of outgoing calls //Discrete and Continuous Models and Applied Computational Science. 2019. Vol. 27, № 1. P. 5-20.

ИССЛЕДОВАНИЕ RQ-СИСТЕМЫ С ОБРАТНОЙ СВЯЗЬЮ И НЕОРДИНАРНЫМ ПУАССОНОВСКИМ ВХОДЯЩИМ ПОТОКОМ

Назаров А.А.[1], Рожкова С.В.[1,2], Титаренко Е.Ю.[1,2]

[1]*Томский государственный университет*

[2]*Томский политехнический университет*

nazarov.tsu@gmail.com, rozhkova@tpu.ru, teu@tpu.ru

Введение

Системы массового обслуживания с повторными вызовами (RQ-системы) широко используются в различных областях. В таких системах запрос, поступивший в систему и заставший прибор занятым, покидает систему на некоторое случайное время (уходит на орбиту), а затем повторяет попытку попасть на обслуживание [1,2].

На практике довольно часто встречаются СМО, в которых уже получившая обслуживание заявка требует повторного сервиса в зависимости от качества полученного обслуживания, внешних факторов и т.д. Подобные ситуации имеют место в мультиагентных системах, где получившая удовлетворительное обслуживание заявка требует повторного сервиса у этого же агента. Функционирование таких систем достаточно точно описывается СМО с обратной связью [3,4]. Среди моделей обратной связи выделяют два типа: мгновенная и отсроченная. В первом случае некоторые заявки после первичного обслуживания мгновенно возвращаются на повторное, во втором случае заявки, прежде чем поступить на повторное обслуживание, ожидают на орбите.

В данной работе исследуется одноканальная RQ-система массового обслуживания с экспоненциальным обслуживанием, неординарным пуассоновским входящим потоком, с мгновенной и отсроченной обратной связью.

1. Постановка задачи

Рассмотрим RQ-систему массового обслуживания $M^n/M/1$ (рис. 1), на вход которой поступает пуассоновский неординарный поток заявок с параметром λ и заданными вероятностями q_ν появления ν заявок в группе, при этом $\nu>1$, $q_0=0$, $\sum_{\nu=1}^{\infty} q_\nu = 1$. Если обслуживающий прибор свободен, то одна заявка из группы поступает на обслуживание, остальные переходят на орбиту, туда же попадают заявки, поступившие в момент, когда прибор занят. Продолжительность обслуживания заявки имеет экспоненциальное распределение с параметром μ.

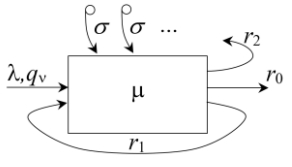


Рис. 1. Система массового обслуживания $M^n/M/1$ с обратной связью

После обслуживания заявка с вероятностью r_0 покидает систему, с вероятностью r_1 мгновенно поступает на повторное обслуживание, с вероятностью r_2 переходит на орбиту. Очевидно, что $r_0 + r_1 + r_2 = 1$. На орбите заявки ожидают повторного обслуживания в течение времени, распределенного по экспоненциальному закону с параметром σ, после чего повторяют попытку занять прибор. В случае неудачной попытки заявки остаются на орбите.

Обозначим $i(t)$ – число заявок на орбите в момент времени t. Т.к. полученный случайных процесс $\{i(t)\}$ немарковский, то марковизируем его, введя дополнительную переменную $n(t)$, которая определяет состояние прибора следующим образом:

$$n(t) = \begin{cases} 0, \text{если прибор свободен;} \\ 1, \text{если прибор занят.} \end{cases}$$

Тогда двумерный процесс $\{i(t), n(t)\}$ будет марковским. Ставится задача исследования числа заявок на орбите в момент времени t.

2. Система дифференциальных уравнений Колмогорова

Обозначим $P_n(i,t) = P\{i(t) = i, n(t) = n\}$, $n = 0,1$, $i = 0,1,2\ldots$, тогда для распределения вероятностей получаем систему дифференциальных уравнений Колмогорова

$$\frac{\partial P_0(i,t)}{\partial t} = -(\lambda + i\sigma)P_0(i,t) + \mu r_0 P_1(i,t) + \mu r_2 P_1(i-1,t),$$

$$\frac{\partial P_1(i,t)}{\partial t} = (i+1)\sigma P_0(i+1,t) + (\mu r_1 - \mu - \lambda)P_1(i,t) + \sum_{\nu=1}^{i+1} \lambda q_\nu P_0(i-\nu+1,t) + \sum_{\nu=1}^{i} \lambda q_\nu P_1(i-\nu,t).$$

Для стационарного распределения вероятностей $P_n(i) \equiv P_n(i,t)$ перепишем систему в виде

$$\begin{aligned} &-(\lambda + i\sigma)P_0(i) + \mu r_0 P_1(i) + \mu r_2 P_1(i-1) = 0, \\ &(i+1)\sigma P_0(i+1) + (\mu r_1 - \mu - \lambda)P_1(i) + \sum_{\nu=1}^{i+1} \lambda q_\nu P_0(i-\nu+1) + \sum_{\nu=1}^{i} \lambda q_\nu P_1(i-\nu) = 0. \end{aligned} \quad (1)$$

3. Решение системы методом характеристической функции

Введем частичные характеристические функции числа заявок на орбите $H_n(u) = \sum_{i=0}^{\infty} e^{jui} P_n(i)$ и характеристическую функцию числа заявок в группе $h(u) = \sum_{\nu=1}^{\infty} e^{ju\nu} q_\nu$, где $j = \sqrt{-1}$, и, учитывая, что

$$\frac{\partial H_n(u)}{\partial u} = \sum_{i=0}^{\infty} ij e^{jui} P_n(i),$$

$$\sum_{i=0}^{\infty}\sum_{\nu=1}^{i} q_\nu e^{jui} P_1(i-\nu) = \sum_{\nu=1}^{\infty} q_\nu e^{ju\nu} \sum_{i=0}^{\infty} e^{jui} P_1(i) = h(u)H_1(u),$$

$$\sum_{i=0}^{\infty}\sum_{\nu=1}^{i+1} q_\nu e^{jui} P_0(i-\nu+1) = e^{-ju} \sum_{\nu=1}^{\infty} q_\nu e^{ju\nu} \sum_{i=0}^{\infty} e^{jui} P_0(i) = e^{-ju} h(u)H_0(u),$$

преобразуем систему (1) к виду

$$
\begin{aligned}
&-\lambda H_0(u)+\mu r_0 H_1(u)+\mu r_2 e^{ju} H_1(u)-\frac{\sigma}{j}\frac{\partial H_0(u)}{\partial u}=0,\\
&\left(\mu r_1-\mu-\lambda\right)H_1(u)+\frac{\sigma}{j}e^{-ju}\frac{\partial H_0(u)}{\partial u}+\lambda e^{-ju}h(u)H_0(u)+\lambda h(u)H_1(u)=0.
\end{aligned}
\tag{2}
$$

Данную систему можно решить аналитически. Для этого выразим $H_1(u)$ через $H_0(u)$: $H_1(u)=\dfrac{\lambda(h(u)-1)H_0(u)}{\mu r_0\left(e^{ju}-1\right)-\lambda e^{ju}(h(u)-1)}$, и подставив $H_1(u)$ в первое уравнение системы (2), получим

$$
\left(\frac{\lambda\left(\mu r_0+\mu r_2 e^{ju}\right)(h(u)-1)}{\mu r_0\left(e^{ju}-1\right)-\lambda e^{ju}(h(u)-1)}-\lambda\right)H_0(u)-\frac{\sigma}{j}\frac{\partial H_0(u)}{\partial u}=0\,.
$$

Решением этого дифференциального уравнения будет функция

$$
H_0(u)=R_0\exp\left\{\frac{\lambda j}{\sigma}\int_0^u\frac{e^{ju}(h(u)-1)\left(\lambda+\mu r_2\right)+\mu r_0\left(h(u)-e^{ju}\right)}{\mu r_0\left(e^{ju}-1\right)-\lambda e^{ju}(h(u)-1)}du\right\},
$$

где R_0 – некоторая константа, для нахождения которой запишем характеристическую функцию

$$
\begin{aligned}
H(u)=H_0(u)+H_1(u)&=\frac{\lambda(h(u)-1)+\mu r_0\left(e^{ju}-1\right)-\lambda e^{ju}(h(u)-1)}{\mu r_0\left(e^{ju}-1\right)-\lambda e^{ju}(h(u)-1)}R_0\times\\
&\times\exp\left\{\frac{\lambda j}{\sigma}\int_0^u\frac{e^{ju}(h(u)-1)\left(\lambda+\mu r_2\right)+\mu r_0\left(h(u)-e^{ju}\right)}{\mu r_0\left(e^{ju}-1\right)-\lambda e^{ju}(h(u)-1)}du\right\},
\end{aligned}
$$

и, учитывая условие нормировки $H(0)=1$, а также $h(0)=1$, $h'(0)=j\sum_{\nu=1}^{\infty}\nu q_\nu=j\bar{\nu}$, получим

$$
1=\lim_{u\to 0}H(u)=\lim_{u\to 0}\frac{\lambda(h(u)-1)+\mu r_0\left(e^{ju}-1\right)-\lambda e^{ju}(h(u)-1)}{\mu r_0\left(e^{ju}-1\right)-\lambda e^{ju}(h(u)-1)}R_0\,,
$$

отсюда $R_0=\dfrac{\mu r_0-\lambda\bar{\nu}}{\mu r_0}$.

Обозначив $\rho=\dfrac{\lambda\bar{\nu}}{\mu r_0}$ – коэффициент загрузки системы, получим

$$
\begin{aligned}
H(u)&=(1-\rho)\cdot\frac{\left(\bar{\nu}-\rho(h(u)-1)\right)\left(e^{ju}-1\right)}{\bar{\nu}\left(e^{ju}-1\right)-\rho e^{ju}(h(u)-1)}\times\\
&\times\exp\left\{\frac{\lambda j}{\sigma}\int_0^u\frac{e^{ju}(h(u)-1)\left(\lambda+\mu r_2\right)+\mu r_0\left(h(u)-e^{ju}\right)}{\mu r_0\left(e^{ju}-1\right)-\lambda e^{ju}(h(u)-1)}du\right\},
\end{aligned}
\tag{3}
$$

где $\bar{\nu}=\sum_{\nu=1}^{\infty}\nu q_\nu$ – среднее число заявок в группе. Отсюда видно, что условием существования стационарного режима является $\rho<1$.

В случае ординарного потока

$$
q_\nu=\begin{cases}1,\ \nu=1;\\ 0,\ \nu\neq 1;\end{cases}\quad\bar{\nu}=1\,,\ h(u)=\sum_{\nu=1}^{\infty}e^{ju\nu}q_\nu=e^{ju}
$$

характеристическая функция будет иметь вид $H(u)=\left(1-\rho_0\left(e^{ju}-1\right)\right)\cdot\left(\frac{1-\rho_0}{1-\rho_0 e^{ju}}\right)^{\frac{1}{\sigma}(\mu r_2+\lambda+\sigma)}$, где коэффициент загрузки системы $\rho_0=\frac{\lambda}{\mu r_0}$.

4. Численные результаты

Для нахождения распределения вероятностей числа заявок на орбите $P(i)$ достаточно применить обратное преобразование Фурье к (3): $P(i)=\frac{1}{2\pi}\int_{-\pi}^{\pi}e^{-jui}H(u)du$.

Выражения для нахождения основных характеристик системы, а именно математического ожидания и дисперсии числа заявок на орбите, имеют вид $M\{i(t)\}=-jH'(0)$, $D\{i(t)\}=-H''(0)+\left(\mathrm{H}'(0)\right)^2$.

Рассмотрим систему с параметрами $\lambda=1$, $\sigma=1$, $r_0=0.5$, $r_1=0.3$, $r_2=0.2$, $\nu_1=0.5$, $\nu_2=0.3$, $\nu_3=0.1$, $\nu_4=0.1$, $\nu_5=0$, $\nu_6=0$ ... . Параметр μ подберем так, чтобы коэффициент загрузки системы ρ был равен 0.5, 0.7 и 0.9. Полученные распределения вероятностей числа заявок на орбите показаны на рис. 2.

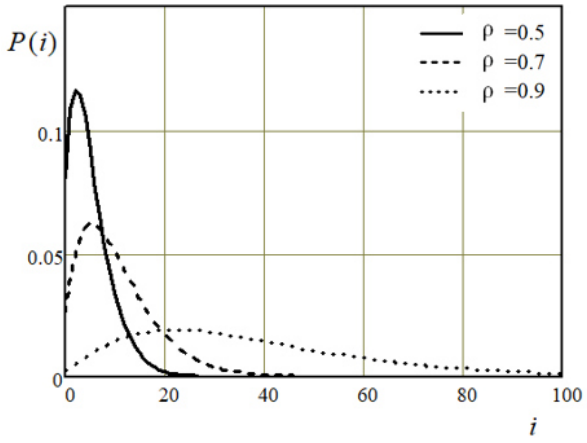


Рис. 2. Распределение вероятностей числа заявок на орбите

В табл. 1 представлены математическое ожидание и дисперсия числа заявок на орбите для рассматриваемой системы с разными коэффициентами загрузки.

Таблица 1

Основные характеристики системы

ρ	*M*{*i*(*t*)}	*D*{*i*(*t*)}
0.5	5.207	18.117
0.7	10.589	61.963
0.9	38.300	656.086

Заключение

В работе исследована RQ-система с обратной связью и неординарным пуассоновским входящим потоком. Получено выражение для характеристической функции, что позволяет определить основные вероятностные характеристики числа заявок на орбите. Кроме того, представлены численные результаты и получено стационарное распределение вероятностей числа заявок на орбите.

ЛИТЕРАТУРА

1. *Назаров А.А.* Теория массового обслуживания / А. А. Назаров, А. Ф. Терпугов. Томск: Изд-во НТЛ, 2005. с. 228.

2. *Клименок В.И., Тарамин О.С.* Двухфазная система обслуживания с групповым марковским потоком и повторными вызовами // Автоматика и телемеханика. 2010. Т. 71, № 1. С. 3–17.

3. *Шкленник М.А., Моисеева С.П.* Исследование потоков в неоднородной бесконечнолинейной системе массового обслуживания с обратной связью //Марчуковские научные чтения – 2017. Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук. Новосибирск. 25 июня – 14 июля 2017 г. Новосибирск: Омега Принт, 2017. С. 161.

4. *Klimenok V., Kim C.S., Tsarenkov G.V., Breuer L., Dudin A.N.* The BMAP/G/1 → ·/PH/1/M tandem queue with feedback and losses // Performance Evaluation. 2007. V. 64. P. 802–818.

РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ ПЛАТФОРМЫ ИНТЕРНЕТА ВЕЩЕЙ «MICRAN IOT»

Рачис В.А.

Томский политехнический университет

seva-ra4is@mail.ru

Введение

Индустриальный Интернет Вещей (IIoT) – интернет вещей для корпоративного применения, т.е. система объединенных компьютерных сетей и подключенных промышленных объектов с датчиками и ПО для сбора и обмена данными, с возможностью удаленного контроля и управления в автоматизированном режиме, без участия человека (рис. 1) [1].

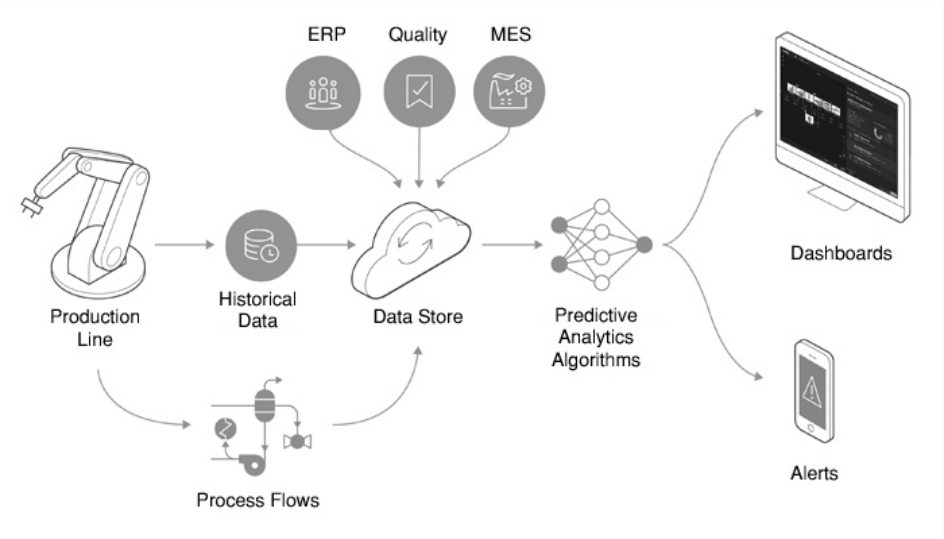


Рис. 1. Индустриальный Интернет вещей (IIoT) [2]

Принцип работы технологии заключается в следующем: первоначально устанавливаются датчики, исполнительные механизмы, контроллеры, человеко-машинные интерфейсы на ключевые части оборудования, после чего осуществляется сбор информации позволяющей получить оценку состояния предприятия. Могут быть предотвращены внеплановые простои, поломки, сокращение техобслуживания и сбоев в управлении цепочками поставок, тем самым позволяя предприятию функционировать более эффективно.

1. Постановка целей и задач

Цель: реализовать автоматизированную информационную платформу промышленного интернета вещей – «Micran IoT». Данный программный комплекс должен использоваться для визуализации возможностей компании в сфере умных устройств во время предоставления подобных услуг заказчику.

Задачи:

- Изучение системы
- Настойка сервера
- Выбор средств разработки
- Развертывание и настройка инфраструктуры
- Установка и настройка необходимых сервисов
- Создание макет страниц сайта
- Верстка frontend
- Написание backend

Комплекс должен содержат следующую функциональность:

- Авторизация (с учёта различных прав доступа)
- Просмотр списка устройств
- Для группы пользователей возможность изменения и удаления устройств
- Просмотр информации с датчиков (изначально ИК-камеры, затем добавили манометры, но система должна быть гибкая в этом плане):
  - Тепловая карта (только для ИК-камеры)
  - Текущий показатель (ИК-камеры – максимальная температура, манометр - давление)
  - RSSI
  - Уровень заряда батареи
  - График показателей
- Карта устройств (только для ИК-камер с возможностями:
  - Отображения устройств
  - Визуальное отделение
    - Отсутствующих (место для камеры есть, а камеры нет)
    - Неактивных
    - С нормальными показателями
    - С превышенными показателями
  - Возможность перехода к просмотру информации с устройства
  - Запуск аудиосигнала в случае проблем с каким-то устройством

2. Выбранный инструментарий разработки

- Python 3.7 (фреймворк Django 3) – написание backend.
- HTML / CSS / JavaScript – верстка frontend
- SQLite3 – БД для хранения данных о устройствах и пользователях
- InfluxDB – хранение данных с устройств (первые версии)
- MongoDB – хранение данных с устройств

3. Описание системы

Рассмотрим потоки данных в проекте (рис. 2).

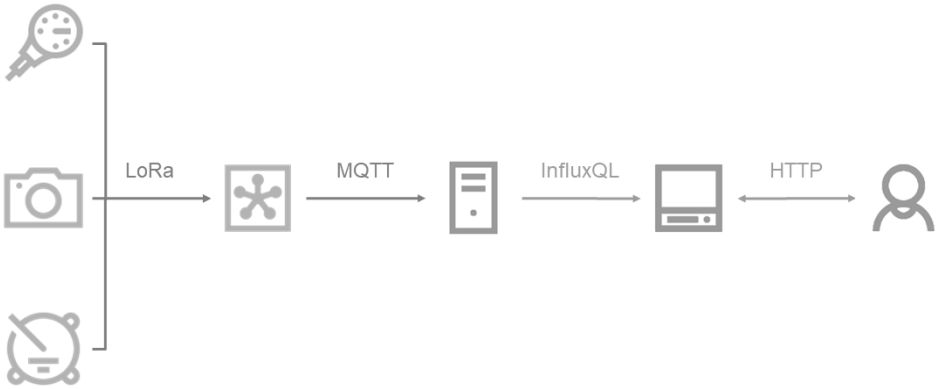


Рис. 2. Потоки данных

Информация с IoT-датчиков (ИК-камеры, манометры и другие) по протоколу LoRa передаются на шлюз. Шлюзом в данном проекте выступает базовая станция LoRa IoT Kerlink, однако это может быть любое другое устройство, имеющее схожий функционал.

Затем эти данные по MQTT передаются на сервер. Сервером является компьютер, под управлением Unix-подобной ОС, с установленным комплектом программ, включающим MQTT-брокер. Там информация обрабатывается сервисами LoraServer (loraserver и lora-app-server) и преобразуются в пакеты, также они отвечают за первичную визуализацию. В первых версиях программы эти же сервисы отвечают за занесение показателей в базу данных InfluxDB, однако позже был написал скрипт, который заносит полеченные данные в необходимую СУБД (MongoDB).

Из этой базы данных показатели считывает развернутое на клиентском компьютере веб-приложение (тип и формат запросов зависят от базы данных, для InfluxDB – InfluxQL, а для MongoDB – ad-hoc-запросы).

С этим приложением взаимодействует пользователь через браузер.

Умные устройства и шлюз, протоколы их взаимодействия, а также комплект программного обеспечения для преобразования полученных данных изменить нельзя, поэтому в качестве изначального источника информации взяты пакеты полученные от loraserver.

4. Потоки данных на сервере

Рассмотрим подробнее протоколы данных на сервере (рис. 3).

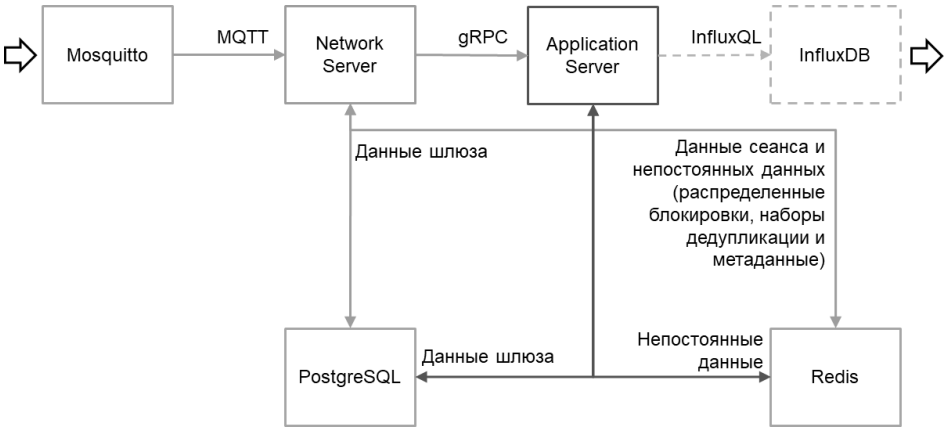


Рис. 3. Потоки данных на сервере

Все компоненты, кром InfluxDB, необходимы для правильной работы системы loraserver. Рассмотрим их [3]:

- Mosquitto – брокер сообщений с открытым исходным кодом, реализующий протокол MQTT. Mosquitto подходит для использования на всех устройствах от маломощных одноплатных компьютеров до полноценных серверов.
- loraserver (network server) – реализация сетевого сервера LoRaWAN® с открытым исходным кодом. Ответственность компонента сетевого сервера заключается в устранении дублирования полученных кадров шлюзами, а также в их обработке.
- lora-app-server (application server) – отвечает за “инвентаризацию” устройств, обработку запросов на соединение и анализ, шифрование полезных нагрузок. Предлагает веб-интерфейс, для управления, организациями, приложением и устройствами.
- PostgreSQL – мощная объектно-реляционная система баз данных с открытым исходным кодом, имеющая более чем 30-летний опыт активной разработки и заслужившая высокую репутацию за надежность, надежность функций и производительность.
- Redis – хранилище структур данных с открытым исходным кодом в памяти, используемое в качестве базы данных, кэша и брокера сообщений. Поддерживает такие структуры данных, как строки, хэши, списки, наборы и многое другое.

5. ORM-классы

ORM – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных». Существуют как проприетарные, так и свободные реализации этой технологии. Рассмотри созданные классы (рис. 4), также используются встроенный класс «пользователь», однако рассматривать его нет смысла [4].

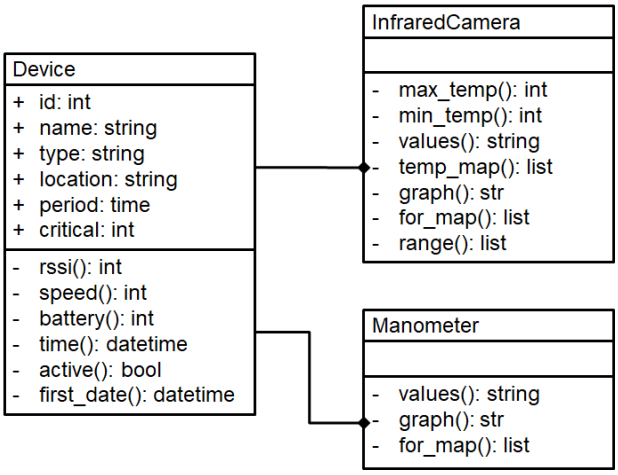


Рис. 4. ORM-классы приложения

Имеется три класса:

- Device (Устройство) – абстрактный класс, является родителем конкретных устройств
  + id – уникальный идентификатор устройства в БД
  + name – имя (совпадает с именем в веб-интерфейсе lora-app-server)
  + type – тип (ИК-камера, манометр…)
  + location – расположение, необходимое для карты устройств
  + period – время обновления, необходимое для отслеживания активности

+ critical – критическое значение (ИК-камеры – макс. темп, для манометра – давление)
– rssi – возвращает данные RSSI (уровень принимаемого сигнала)
– speed – скорость передачи данных
– battery – показатель батареи
– time – время последних полученных данных
– active – активность устройства
– first_date – дата первого получения данных, необходимое для графиков

- InfraredCamera (Инфракрасная камера / ИК-камера) – класс ИК-камеры
  – max_temp – возвращает максимальную температуру
  – min_temp – минимальная температура
  – values – строка с показателями мин. и макс. температуры
  – temp_map – двумерный массив для отрисовки тепловой карты
  – graph – двумерный массив для отрисовки графика данных
  – for_map – массив для отрисовки карты устройств
  – range – массив значений для подстановки в легенду карты глубины
- Manometer (Устройство) – класс манометра
  – values – строка с показателем давления
  – graph – двумерный массив для отрисовки графика данных
  – for_map – массив для отрисовки карты устройств

Отметим, что атрибуты классов храниться в SQLite3, а методы используют данные из InfluxDB, куда они были записаны сервисами loraserver. Для лучшего понимания смысла ORM рассмотрим пример передачи данных между компонентами программы в Django-проекте (рис. 5).

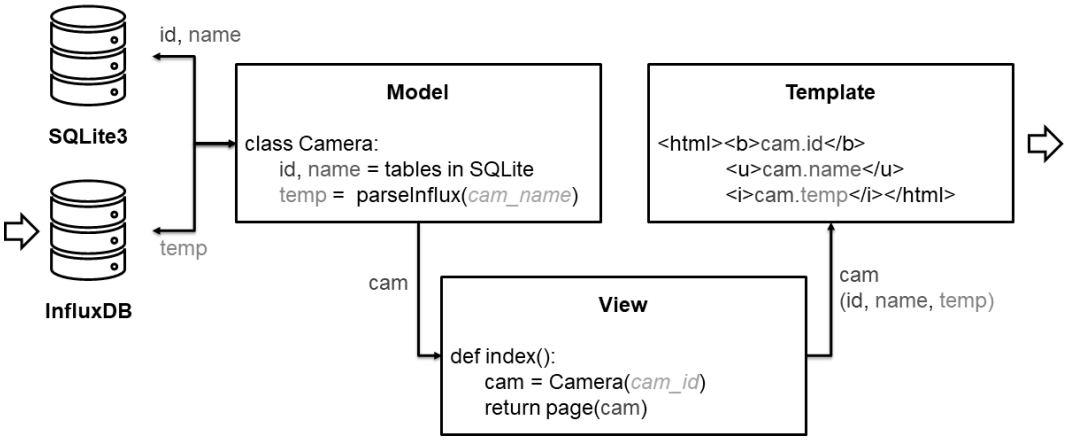


Рис. 5. Передача данных между компонентами программы в Django-проекте

Предположим, пользователь открыл сайт ИК-камеры, где указан только id, name и temp. Когда пользователь переходит на такую страницу, происходит вызов функции в компоненте View. В функции происходит получение объекта класса Camera по cam_id из компонента Model. В классе Camera имеется три данных: id, name (берутся с БД SQLite3, куда были записаны при создании объекта камеры) и temp (происходит обработка данных полученных из InfluxDB). После того как данные были занесены, этот объект класса вызывает отрисовку HTML-документа, подставляя в его шаблон конкретные значение объекта (cam.id, ca.name, cam.temp).

Данный способ хранения информации, по сравнению с обычными внешними функциями, имеет следующие преимущества:

- Компонент Model содержит всю информацию о устройствах, что позволяет легче разбираться в коде

- Код во View стал намного короче и его проще масштабировать (выводить информацию не о одном устройстве, а о нескольких)
- Повысилась читаемость верстки Template, т.к. теперь нет путаницы в устройствах (если их несколько).

6. Реализация

Главная страница представлена на рис. 6 (все скриншоты обреза по вертикали, чтобы не занимать лишнее место).

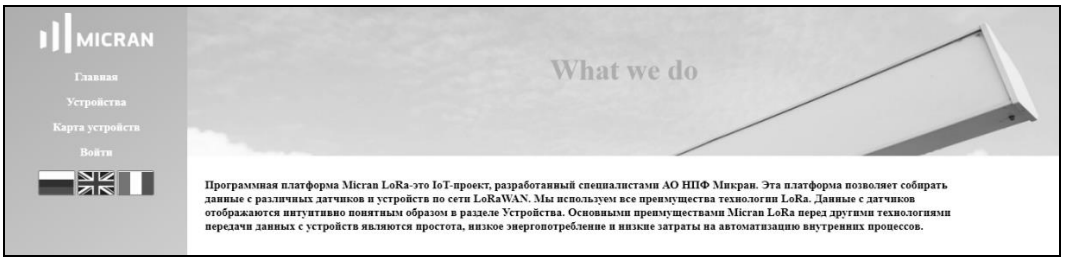


Рис. 6. Главная страница

Слева располагается меню, вверху шапка с картинкой, а по центру текст с описанием комплекса. Сайт сам определяет язык (из cookie), однако его нажав переключить, нажав соответствующую кнопку флага (рис. 7).

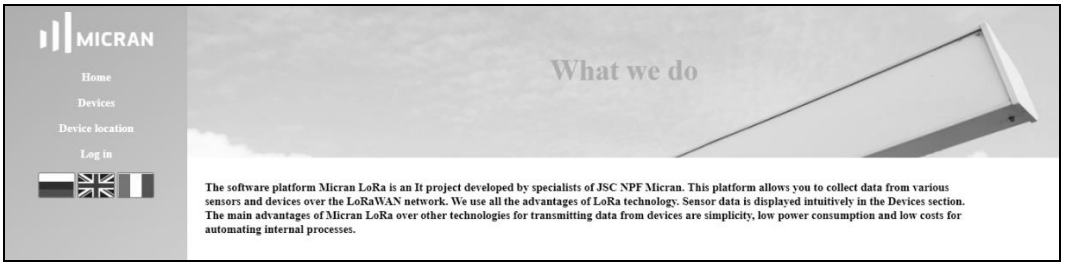


Рис. 7. Английская версия

Войдем в аккаунт. Для этого нажимаем на соответствующую кнопку и вводим логин с паролём в форму (рис. 8).

Рис. 8. Форма входа

Вернемся на русский язык и перейдём на страницу списка устройств (рис. 9).

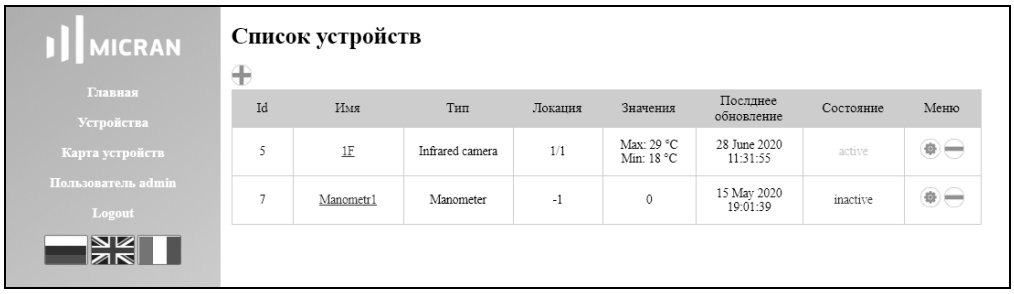


Рис. 9. Список устройств (версия admin)

Если пользователь является администратором, но он может добавлять и редактировать (рис. 10) устройства.

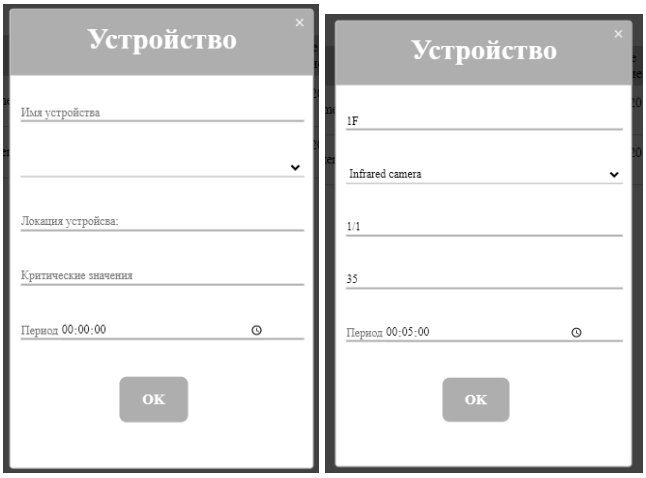


Рис. 10. Добавление нового устройства

Обычный же пользователем не имеет данных возможностей (рис. 11).

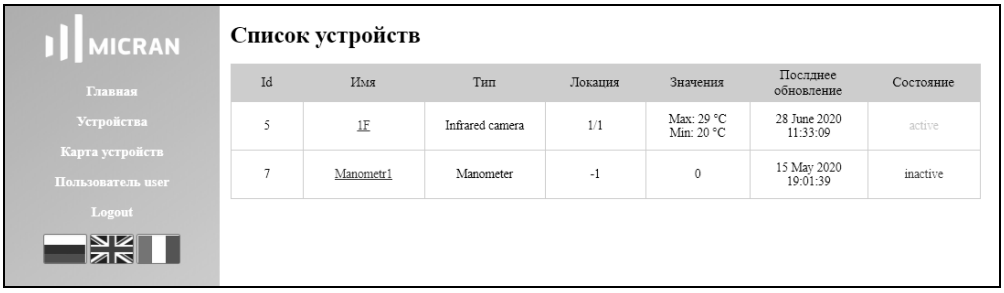


Рис. 11. Список устройств (версия пользователя)

Далее перейдём нас страницу ИК-камеры (рис. 12). На ней имеется карта температур, показатель максимальной температуры, RSSI, скорость, батарея, а также график, для которого можно указывать время.

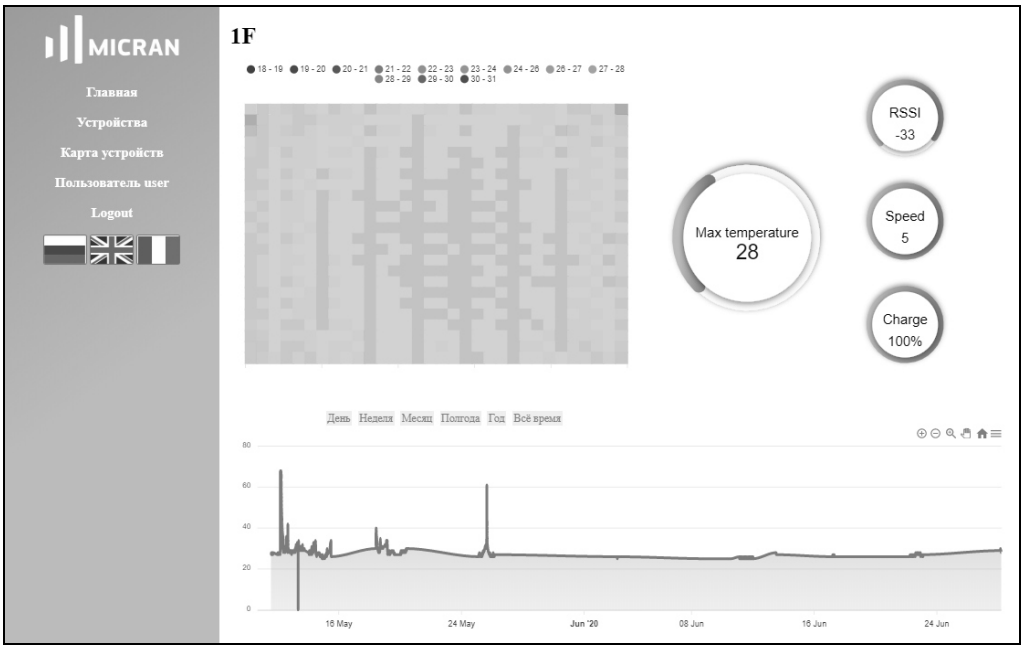


Рис. 12. Страница ИК-камеры

Последней страницей является карат устройств (рис. 13). На которой показаны камеры в локациях, а также их состояние.

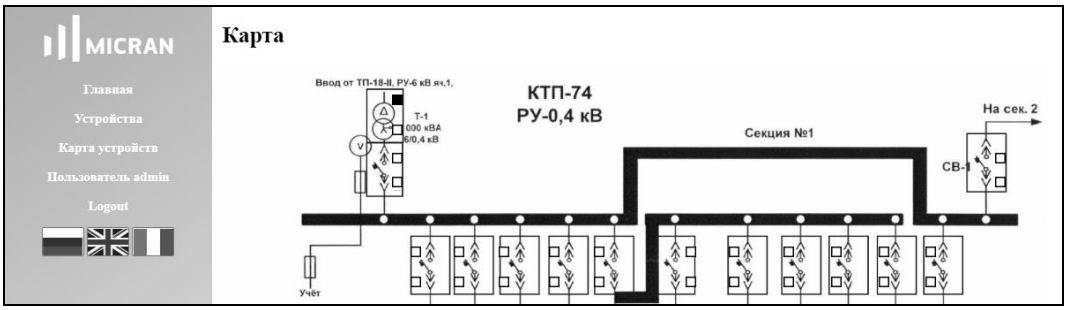


Рис. 13. Карта устройств

Заключение

В рамках работы над проектом была реализована автоматизированная информационная платформа промышленного интернета вещей – «Micran IoT». Платформа готова к внедрению, однако окончательного внедрения с подписанием документов не было. Платформа введена в тестовую эксплуатацию на одном из предприятии СИБУР. «Micran IoT» проходил опытную эксплуатацию в Сибуре (Воронеж и Томск), были высказаны несколько замечаний, но решением заинтересовались. Также демонстрировали работу системы в Связьтранснефть, им решение тоже интересно, однако о покупке речь пока не идёт.

Однако предстоит ещё реализовать следующее:

- Упаковка приложения в docker-контейнер
- Интеграция других баз данных для хранения данных с датчиков
- Развертывание инфраструктуры loraserver в облаке
- Интеграция OSM сервера (карты в привычном их понимании)
- Написание конфигурационных файлов
- Документирование

ЛИТЕРАТУРА

1. Industrial Internet of Things - IIoT Промышленный интернет вещей // tadviser URL: https://www.tadviser.ru/index.php/Статья:IIoT_-_Industrial_Internet_of_Things_(Промышленный_интернет_вещей) (дата обращения: 28.06.2020).

2. IIoT Applications // Internet of Things Wiki URL: https://internetofthingswiki.com/moving-from-preventive-to-predictive-maintenance-in-iiot-projects/1348/iiot-applications/ (дата обращения: 28.06.2020).

3. ChirpStack, open-source LoRaWAN® Network Server stack URL: https://www.chirpstack.io/ (дата обращения: 30.08.2020).

4. ORM (Object-Relational Mapping) // Национальная библиотека им. Н. Э. Баумана Bauman National Library URL: https://ru.bmstu.wiki/ORM_(Object-Relational_Mapping)#:~:text=ORM%20(Object-Relational%20Mapping)%20–%20технология%20программирования%2C%20которая,«виртуальную%20объектную%20базу%20данных». (дата обращения: 30.08.2020).

МОДЕЛЬ ОБСЛУЖИВАНИЯ ТРАФИКА ОДНОАДРЕСНЫХ СОЕДИНЕНИЙ В БЕСПРОВОДНОЙ СЕТИ НА БАЗЕ ТЕХНОЛОГИИ "НОВОЕ РАДИО"*

Удодова А.Э., Бесчастный В.А., Острикова Д.Ю.

Российский университет дружбы народов

alina.udodova1998@yandex.ru, {beschastnyy-va, ostrikova-dyu}@rudn.ru

Введение

Телекоммуникационные системы пятого поколения «Новое Радио», использующие для передачи данных миллиметровый диапазон длин волн, позволят достичь значительного увеличения пропускной способности беспроводных сетей. Однако, из-за проблем [1], возникающих по мере изучения рода технологий «Новое Радио», возникает потребность в интеллектуальных механизмах управления лучом. На стабильное обслуживание абонентов существенно влияют подвижные препятствия (например, люди или машины) [2], временно блокирующие распространение радиосигнала. В случае выхода абонента из зоны обслуживания применяется предложенный консорциумом 3rd Generation Partnership Project (3GPP) механизм «множественных соединений» [3], в соответствии с которым одновременно формируются несколько активных каналов между абонентом и размещенными рядом базовыми станциями (БС), что позволяет сессии в случае возникновения препятствия сменить обслуживающую точку доступа. Таким образом, зона покрытия БС разделяется на две части: внутренний круг с центром в положении БС, в котором соединение возможно вне зависимости от условий прямой видимости; и внешнее кольцо, в котором заблокированные соединения переключаются на соседние БС [4–6].

Для расчета и анализа вероятности сброса сессий в высокочастотной сети 5G были построены математические модели в виде ресурсной системы массового обслуживания (РСМО) с обслуживанием фазового типа и переменными требованиями к ресурсу.

1. Постановка задачи

Рассмотрим модель функционирования базовой станции (БС) высокочастотной сети 5G. БС покрывает зону внутри окружности радиуса d, максимальное значение которого d_{LoS}^{E} рассчитывается при помощи модели распространения сигнала в миллиметровом диапазоне. Предполагается, что абоненты равномерно распределены по зоне покрытия БС в соответствии с пуассоновским точечным процессом с параметром ρ. Таким образом, процесс поступления запросов абонентов на установление соединений

* Публикация подготовлена при поддержке Программы РУДН «5-100» (Удодова А.Э., разработка математической модели; Бесчастный В.А., численный эксперимент). Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-37-70079 (Острикова Д.Ю., постановка задачи исследования).

является пуассоновским с интенсивностью $\lambda = \Lambda\rho\pi d^2$, где Λ – параметр экспоненциально распределенных интервалов между двумя последовательными запросами от одного абонента, а ρ – плотность абонентов в зоне покрытия.

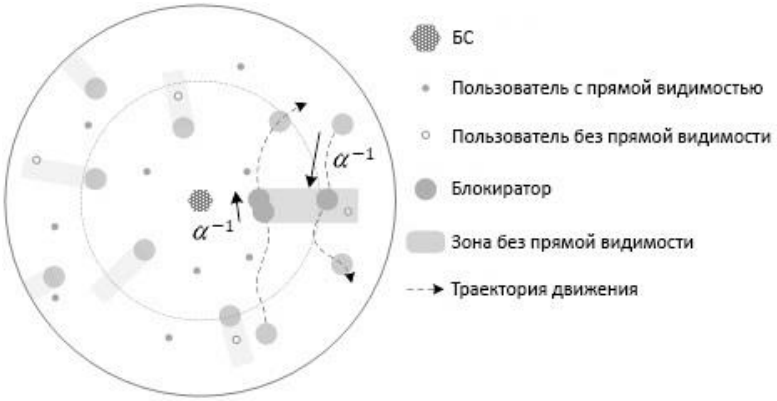


Рис. 1. Модель движения препятствий

Для обслуживания запросов абонентов выделяется радиоресурс, при этом объем ресурса является случайной величиной и зависит от местоположения абонентов. В соответствии с требованиями к объему ресурса область покрытия разбивается на две зоны – ближний к точке доступа круг, в центре которого расположена базовая станция, и оставшуюся зону вплоть до границ зоны покрытия, которая имеет форму кольца (рис. 1). Радиус круга определяет модель распространения сигнала как максимальное расстояние d_{LoS}^{E}, на котором можно установить соединение в условиях блокировки прямой видимости. Внутренний радиус кольца равен радиусу круга, а внешний радиус кольца рассчитывается по формуле $d = \min\left(d_{ISD} - d_{nLoS}^{E}, d_{LoS}^{E}\right)$, где d_{ISD} – расстояние от целевой до ближайшей соседней БС.

Будем считать, что требования к объему ресурса для обслуживания запросов абонентов, находящихся в круге, ниже соответствующих требований абонентов в кольце. Это предположение определяется простейшей моделью распространения сигнала, согласно которой объем ресурса обратно пропорционален расстоянию между оборудованием абонента и точкой доступа.

При появлении препятствия на линии прямой видимости между точкой доступа и абонентом в круге соединение не прерывается, но увеличивается объем ресурса, выделенный точкой доступа для его поддержания. Когда аналогичная ситуация возникает для абонента в кольце, качество соединения в радиоканале падает настолько, что для его поддержания необходимо сменить опорную БС.

Далее в работе разработаны математические модели обслуживания одноадресного трафика высокочастотной сети 5G с механизмом «множественных соединений» и без него для анализа вероятности сброса сессии и средней доли занятого ресурса.

2. Разработка математических моделей системы

Для анализа основных показателей эффективности высокочастотной сети 5G были построены две математические модели в виде РСМО с обслуживанием фазового типа и переменными требованиями к ресурсу, первая из которых – без реализации механизма «множественные соединения», т.е. обслуживание осуществляется только основной БС, а вторая – с реализацией механизма «множественные соединения», т.е. при возникновении блокировки прямой видимости в кольце, сессии переходят на обслуживание на соседнюю БС.

На систему, состоящую из бесконечного числа приборов, и содержащую ресурс конечного объема *R*, поступает пуассоновский поток заявок с интенсивностью λ. α и β – параметры экспоненциальных распределений периодов между последовательными блокировками прямой видимости и длительностей блокировок, соответственно. Длительность обслуживания заявок также имеет экспоненциальное распределение и может зависеть от текущей фазы обслуживания.

Рассмотрим первый случай модели без механизма «множественных соединений». Состояние системы в момент *t* описывается однородным марковским процессом $X(t)=\left(X_1(t),X_2(t)\right)$, где $X_i(t)$ – количество заявок, обслуживаемых в *i*-й фазе. Процесс $X(t)$ имеет конечное число состояний: $X=\left\{\left(n_1,n_2\right): n_1b_1+n_2b_2\le R\right\}$, где n_1 – число сессий в условиях прямой видимости, а n_2 – при отсутствии прямой видимости.

На рис. 2 изображена схема модели распределения ресурса при поступлении и обслуживании одноадресных сессий в круге.

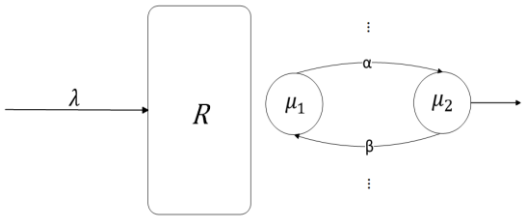


Рис. 2. Схема СМО без реализации механизма «множественные соединения»

Система уравнений равновесия (СУР) имеет следующий вид:

$$
\begin{cases}
\lambda p_{00}=\mu p_{10}+\mu p_{10}\\
(\lambda+\mu+n\alpha)\,p_{n,0}=\lambda p_{n-1,0}+(n+1)\mu p_{n+1,0}+\beta p_{n-1,1}+\mu p_{n,1},\ n=\overline{1,\left\lfloor\frac{R}{b_1}\right\rfloor-1},\\
\left\lfloor\frac{R}{b_1}\right\rfloor\mu p_{\left\lfloor\frac{R}{b_1}\right\rfloor,0}=\lambda p_{\left\lfloor\frac{R}{b_1}\right\rfloor-1,0},\\
(\lambda+n\mu+n\alpha+m\mu+\mu\beta)\,p_{n,m}=\lambda p_{n-1,m}+(n+1)\mu p_{n+1,m}+(n+1)\alpha p_{n+1,m-1}+(m+1)\mu p_{n,m+1}+\\
\quad+(m+1)\beta p_{n-1,m+1},\ m=\overline{1,\left\lfloor\frac{R}{b_2}\right\rfloor-1},\ n=\overline{1,\left\lfloor\frac{R-mb_2}{b_1}\right\rfloor-1},\\
\left(\left\lfloor\frac{R}{b_2}\right\rfloor\mu+\left\lfloor\frac{R}{b_2}\right\rfloor\beta+\lambda\right)p_{0,\left\lfloor\frac{R}{b_2}\right\rfloor}=\mu p_{1,\left\lfloor\frac{R}{b_2}\right\rfloor}+\alpha p_{1,\left\lfloor\frac{R}{b_2}\right\rfloor-1},\\
(m\beta+m\mu+n\mu+n\alpha)\,p_{n,m}=\lambda p_{n-1,m}+(n+1)\alpha p_{n+1,m-1}+(m+1)\beta p_{n-1,m+1},\\
\quad m=\overline{1,\left\lfloor\frac{R}{b_2}\right\rfloor-1},\ n=\overline{1,\left\lfloor\frac{R=mb_2}{b_1}\right\rfloor-1}.
\end{cases}
$$

Для расчета стационарных вероятностей был применен метод Гаусса.

Перейдем к описанию математической модели с реализацией механизма «множественных соединений». Будем считать, что длительности обслуживания заявок не зависят от поступающего потока, независимы в совокупности, и имеют функции распределения фазового типа $B(x)=1-\mathbf{a}\exp(x\mathbf{M})\mathbf{u}$. Здесь $\mathbf{u}$ – единичный вектор,

$\mathbf{a} = (a_1, 0, a_3, 0)$ – вектор-строка распределения вероятностей начальной фазы, $\mathbf{M} = \left[\mu_{i,j}\right]$ – матрица интенсивностей переходов между фазами обслуживания:

$$\mathbf{M} = \left[\mu_{i,j}\right] = \begin{pmatrix} -(\alpha+\mu) & \alpha & 0 & 0 \\ \beta & -(\beta+\mu) & 0 & 0 \\ 0 & 0 & -(\alpha+\mu) & \alpha \\ 0 & 0 & \beta & -(\beta+\mu) \end{pmatrix}.$$

Смена фазы обслуживания моделирует изменение требований заявок к ресурсам в течение времени обслуживания. Для обслуживания заявки на *i*-й фазе требуется b_i, $i = \overline{1,4}$ единиц ресурса. Заявка теряется, если при поступлении или в момент смены фазы обслуживания выяснится, что объём требуемого заявке ресурса превышает доступный объём незанятого ресурса.

Состояние системы описывается случайным процессом $Y(t) = (Y_1(t), Y_2(t), Y_3(t), Y_4(t))$ над пространством состояний $Y = \left\{(n_1, n_2, n_3, n_4) : n_1 b_1 + n_2 b_2 + n_3 b_3 + n_4 b_4 \le R\right\}$, где n_1 и n_3 – число сессий из кольца/круга в условиях прямой видимости, n_2 и n_4– число сессий из кольца/круга в условиях блокировки прямой видимости.

На рис. 3 изображена схема модели распределения ресурса при поступлении и обслуживании одноадресных сессий в круге и в кольце.

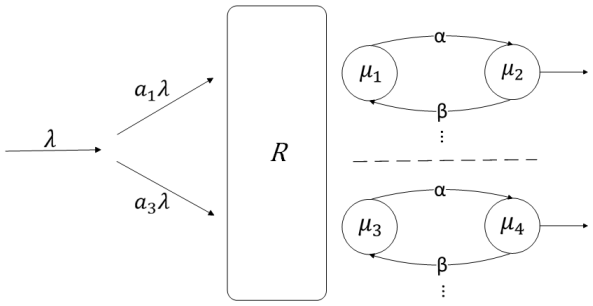


Рис. 3. Схема СМО с реализацией механизма «множественные соединения»

Пусть $\mu_{i0} = -\sum_{j=1}^{4} \mu_{ij}$, $i = \overline{1,4}$ – интенсивность завершения обслуживания заявки после *i*-фазы. СУР для процесса $Y(t)$ имеет следующий вид:

$$\left[\lambda\left(I\left(\sum_{i=1}^{4} n_i b_i + b_1 \le R\right)a_1 + I\left(\sum_{i=1}^{4} n_i b_i + b_3 \le R\right)a_3\right) + \sum_{i=1}^{4} n_i \mu_i\right] p(\mathbf{n}) =$$

$$= \lambda\left(I(n_1 > 0)\, p(\mathbf{n} - \mathbf{e}_1)\, a_1 + I(n_3 > 0)\, p(\mathbf{n} - \mathbf{e}_3)\, a_3\right) + \sum_{i=1}^{4} p(\mathbf{n} + \mathbf{e}_i)(n_i + 1)\mu_{i0} +$$

$$+ \sum_{\substack{i=1 \\ j \ne i}}^{4} \sum_{j=1}^{4} I\left(\sum_{k=1}^{4} n_k b_k + b_j > R\right) p(\mathbf{n} + \mathbf{e}_i)(n_i + 1)\mu_{ij} +$$

$$+ \sum_{\substack{i=1 \\ j \ne i}}^{n} \sum_{j=1}^{n} I(n_j > 0)\, I\left(\sum_{k=1}^{4} n_k b_k + b_j \le R\right) p(\mathbf{n} + \mathbf{e}_i - \mathbf{e}_j)(n_i + 1)\mu_{ij},$$

где $I(s)$ – функция индикатор утверждения *s*, $\mathbf{e}_i$ – нулевой вектор с единицей в *i*-й позиции. СУР решается численно методом последовательных приближений, описанном в [4,5].

3. Пример численного анализа

Численный анализ функционирования БС системы 5G «Новое Радио» проводится с точки зрения ключевых характеристик качества обслуживания – вероятности сброса сессий и средней доли занятого ресурса точкой доступа.

Рассмотрим сначала численный анализ упрощенной модели без механизма «множественных соединений». Исследуются разные типы услуг (video, audio, filesharing, web, social networking, M2M (англ. Machine-to-Machine), e-mail). Для примера используется рабочая частота 28 ГГц с доступной полосой частот в 1 ГГц и коэффициентом затухания в 2,1. Мощность передающей антенны равна 0,2 Вт, а ее усиление 2,58 дБ. Препятствия обозначают цилиндры с радиусом 0,4 м и высотой 1,7 м, имитирующие тело человека [6].

Размер выделяемого ресурса равен 667 RB. Для «video» отводится 70% от этого числа, т.е. 460 RB. Оставшиеся 30% разделяются поровну между остальными услугами (по 5% каждому типу, что соответствует 33 RB). Это обуславливается тем, что услуга «video» требует больше ресурса, и с целью снижения количества блокировок, для ее обслуживания выделяется больший ресурс.

Проведем численный анализ основных характеристик качества обслуживания функционирования БС системы 5G «Новое Радио» – вероятности сброса сессий $B = \sum_{\substack{\mathbf{n}\in X \\ n_1 b_1 + n_2 b_2 + b_1 > R}} p(\mathbf{n})$ и средней доли занятого ресурса $UTIL = \sum_{n\in X} p(\mathbf{n})\left(n_1 b_1 + n_2 b_2\right)$.

В табл. 1 представлены исходные значения параметров для рассматриваемой системы. Количество требуемых ресурсных блоков рассчитано из усредненных показателей спектральной эффективности в соответствии со скоростями, представленными в [7].

Таблица 1

Исходные значения для различных типов услуг

Тип услуг	μ^{-1}, с	b_1, RB	b_2, RB	Скорость, Кбит/с
Video	3600	16	158	50000
Audio	90	1	2	12,2
FileSharing	2	4	32	1024
Web	3	2	15	500
Social Networking	8	2	12	384
M2M	1	1	6	200
E-mail	1	1	3	100

Диаметр круга равен 400 м, $\alpha = \frac{1}{30}\text{с}^{-1}$, $\beta = \frac{1}{3}\text{с}^{-1}$ [8].

На рис. 4 и рис. 6 представлены графики зависимости вероятности сброса сессий от интенсивности поступления запросов пользователей. Бóльшая вероятность сброса наблюдается у типа услуг «video» и «audio», как самых длительных.

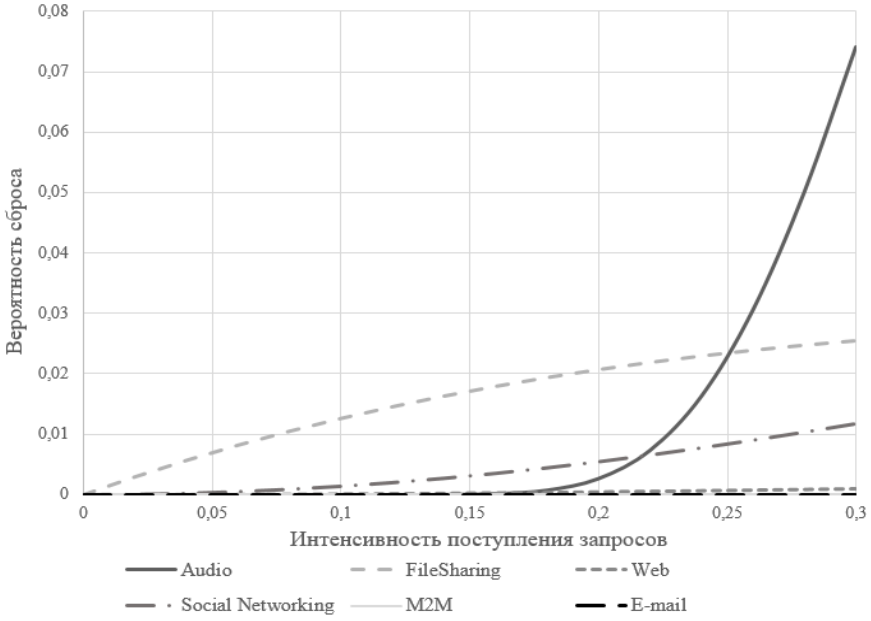


Рис. 4. Вероятность сброса сессий

Таким образом, можно выделить два эффекта. Первый заключается в том, что при обслуживании длительных услуг ресурс не успевает освобождаться под вновь пришедшие запросы. Второй свидетельствует о том, что чем больше длительность услуги, тем чаще возникает блокировка прямой видимости, что, как будет показано далее для модели с механизмом «множественных соединений», в наибольшей степени влияет на вероятность сброса сессии.

Аналогичный эффект наблюдается и на рис. 5 для средней доли занятого ресурса.

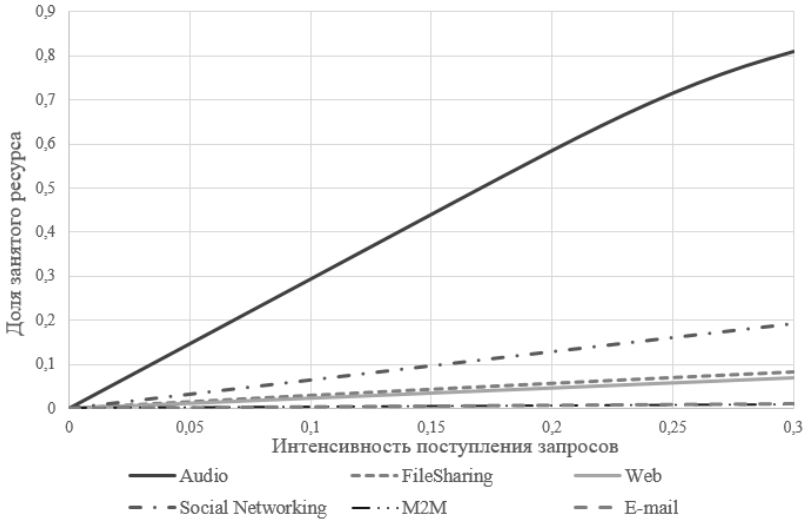


Рис. 5. Средняя доля занятого ресурса

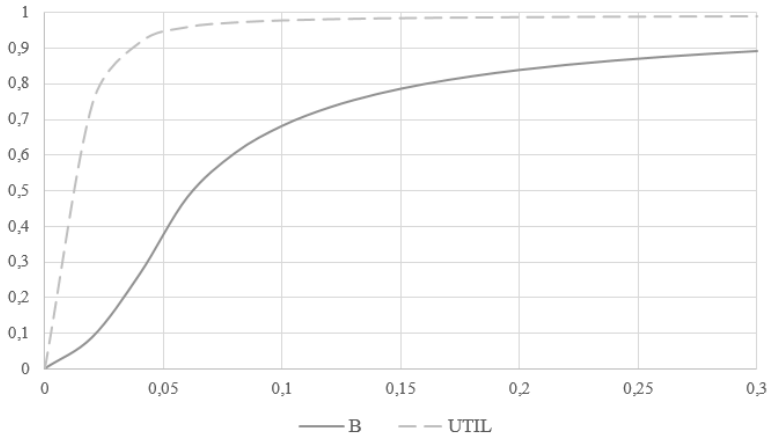


Рис. 6. Вероятность сброса сессий для услуги «видео»

Перейдем к численному анализу второй математической модели, в которой при возникновении блокировки прямой видимости в кольце, сессии переходят на обслуживание на соседнюю БС. Предполагается, что все пользователи получают один тип услуг – «video», со скоростью передачи данных в 50 Мбит/с. Прочие параметры эквивалентны предыдущей модели.

Ниже представлены формулы для расчета вероятности сброса сессий и средней доли занятого ресурса:

$$B = a_1 \sum_{\substack{\mathbf{n}\in X \\ \sum_{j=1}^{4} n_j b_j + b_1 > R}} p(\mathbf{n}) + a_3 \sum_{\substack{\mathbf{n}\in X \\ \sum_{j=1}^{4} n_j b_j + b_3 > R}} p(\mathbf{n}) \,,\; UTIL = \sum_{\mathbf{n}\in X} p(\mathbf{n}) \sum_{i=1}^{4} n_i b_i \,.$$

Исходные данные для заданных значений параметров системы представлены в табл. 2 [8].

Таблица 2

Входные параметры для численного анализа

Обозначение	Описание	Значение
d	радиус внешней зоны	400 м
d_{ISD}	расстояние между соседними точками доступа	600 м
R	количество ресурсных блоков в выделенной полосе частот	667
b_1	требования сессий в кольце при прямой видимости	31 RB
b_2	требования сессий в кольце при возникновении препятствия	0 RB
b_3	требования сессий в круге при прямой видимости	16 RB
b_4	требования сессий из круга при возникновении препятствия	158 RB
λ	интенсивность поступления запросов от одного абонента	10^{-3} с$^{-1}$
μ	интенсивность обслуживания сессий	1/30 с$^{-1}$
α	параметр распределения периодов между последовательными блокировками прямой видимости	0,033 с$^{-1}$
β	параметр распределения длительностей блокировок	0,33 с$^{-1}$

На рис. 7 изображены графики зависимости средней доли занятого ресурса и вероятности сброса сессии от числа абонентов в соте. Видно, что вероятность сброса сессий из кольца значительно выше, чем сессий из круга, т.к. их общее число и требования к ресурсу выше. Наиболее высокая вероятность сброса наблюдается в результате блокировки прямой видимости для сессий из круга. Это обусловлено тем, что в результате блокировки требования к ресурсу возрастают почти в 10 раз. А поскольку при блокировке прямой видимости у сессий из кольца есть возможность переключиться на другую БС, то блокировки для них в принципе отсутствуют, что свидетельствует о надежности таких систем.

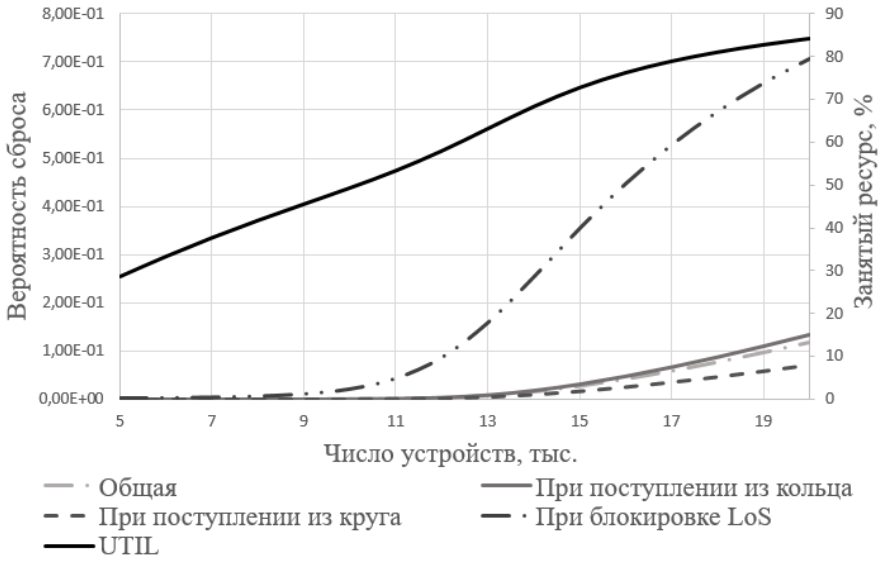


Рис. 7. Графики зависимости вероятности сброса сессий
и средней доли занятого ресурса от количества устройств в соте

Заключение

В работе рассмотрены особенности телекоммуникационных систем пятого поколения «Новое Радио» с точки зрения процесса выделения радио ресурсов точкой достура с учетом временных блокировок прямой видимости подвижными препятствиями, предложены математические модели в виде РСМО с обслуживанием фазового типа и переменными требованиями к ресурсу для анализа основных показателей эффективности – вероятности сброса соединения и средней доли занятого ресурса.

Полученные результаты позволяют сделать вывод о необходимости использования механизма «множественных соединений», который позволил бы существенно снизить вероятность сброса сессий за счёт предоставления на соседних точках доступа дополнительного ресурса.

ЛИТЕРАТУРА

1. *Petrov V., Solomitckii D., Samuylov A., Lema M.A., Gapeyenko M., Moltchanov D., Andreev S., Naumov V., Samouylov K., Dohler M.* Dynamic Multi-connectivity Performance in Ultra-dense Urban mmWave Deployments // IEEE Journal on Selected Areas in Communications, 2017. Vol. 35. No. 9. P. 2038–2055. doi: 10.1109/jsac.2017.2720482.

2. *Gapeyenko M., Samuylov A., Gerasimenko M., Moltchanov D., Singh S.A., Riza M., Aryafar E., Himayat N., Andreev S., Koucheryavy Ye.* 2017. On the Temporal Effects of Mobile Blockers in Urban Millimeter-Wave Cellular Scenarios. IEEE Transactions on Vehicular Technology 66(11):10124–10138. doi: 10.1109/tvt.2017.2754543.

3. 3GPP TS 38.211; NR; Physical channels and modulation (Release 16), Apr 2020.

4. *Naumov V.A., Beschastnyi V.A., Ostrikova D.Yu., Gaidamaka Yu.V.* 5G New Radio System Performance Analysis Using Limited Resource Queuing Systems with Varying Requirements, Lecture Notes in Computer Science. Springer, Vol. 11965, 2019. P. 3-14. doi:10.1007/978-3-030-36614-8_1.

5. *Бесчастный В.А., Острикова Д.Ю., Гайдамака Ю.В.* Анализ производительности систем «Новое радио» сети 5G с помощью СМО с переменными требованиями к ресурсу, Системы и средства информатики, Т. 29, No 4, 2019. – С. 73-83. doi:10.14357/08696527190407.

6. *Samuylov A., Beschastnyi V., Moltchanov D., Ostrikova D., Gaidamaka Yu., Shorgin V.* Modeling Coexistence of Unicast and Multicast Communications in 5G New Radio Systems // 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Istanbul, Turkey, Volume 2019, 8904350. P. 1163-1168. doi: 10.1109/PIMRC.2019.8904350

7. *Khatibi S.* Radio Resource Management Strategies in Virtual Networks // Universidade de lisboa instituto superior técnico, 2016, p. 100.

8. *Kovalchukov R., Moltchanov D., Begishev V., Samuylov A., Andreev S., Koucheryavy Y., Samouylov K.* Improved session continuity in 5G NR with joint use of multi-connectivity and guard bandwidth // 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates. P. 1–7. doi:10.1109/GLOCOM.2018.8647608

АСИМПТОТИЧЕСКИЙ АНАЛИЗ RQ-СИСТЕМЫ M/M/1 С НЕНАДЕЖНЫМ ПРИБОРОМ

Федорова Е.А.[1], Рожкова С.В.[1,2], Воронина Н.М.[2]
[1]Томский государственный университет, [2]Томский политехнический университет
moiskate@mail.ru, rozhkova@tpu.ru, vnm@tpu.ru

Введение

Математические модели RQ-систем широко применяются при анализе и оптимизации различных телекоммуникационных систем, сетей мобильной связи, call-центров и других технических и экономических систем. Характерной чертой таких моделей является возможность повторной передачи искаженных сообщений, когда прибывшая заявка, обнаружив прибор занятым, уходит в зону ожидания (на орбиту) и через некоторое случайное время повторяет попытку получить обслуживание [1].

В данной работе исследуется одноканальная RQ-система массового обслуживания с ненадежным прибором. Системы массового обслуживания называются ненадежными, если их приборы могут время от времени выходить из строя и требовать восстановления (ремонта), только после которого они могут возобновить обслуживание запросов как новые.

На сегодняшний момент исследованию систем массового обслуживания с ненадежными обслуживающими приборами посвящено большое количество работ, обзор которых приведен в [2], в то же время недостатком большинства работ являются весьма жесткие предположения о показательном распределении всех времен, описывающих поведение системы (время между моментами поступления запросов и поломок, время обслуживания запросов, длительность ремонта приборов) [3].

1. Описание модели и постановка задачи

Рассмотрим однолинейную RQ-систему с ненадежным прибором (рис. 1). На вход системы поступает простейший поток заявок с интенсивностью λ. Время обслуживания каждой заявки распределено по экспоненциальному закону с параметром μ_1. Если поступившая заявка находит прибор свободным, то занимает его для обслуживания, в противном случае заявка переходит в источник повторных вызовов (ИПВ) или на орбиту, где ожидает некоторое случайное время с параметром σ, распределенное по экспоненциальному закону. Из ИПВ после случайной задержки заявка вновь обращается к обслуживающему прибору с повторной попыткой его захвата. Если прибор свободен, то заявка из ИПВ занимает его для обслуживания, в противном случае заявка мгновенно возвращается в ИПВ для реализации следующей задержки.

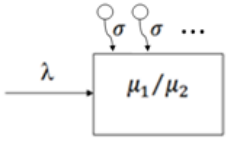


Рис. 1. Модель системы

Предполагается, что сервер ненадежен, т.е. время бесперебойной работы является случайной величиной, экспоненциально распределенной с параметром γ_1, если сервер простаивает, и с параметром γ_2, если он занят обслуживанием. Когда сервер выходит из строя, он сразу же отправляется на ремонт и время восстановления экспоненциально распределено с параметром μ_2.

Когда сервер в нерабочем состоянии, все прибывающие заявки немедленно уходят на орбиту.

Пусть $i(t)$ – число заявок в ИПВ в момент времени t, а $k(t)$ определяет состояние прибора:

$$k(t)=\begin{cases}0, & \text{если прибор свободен,}\\ 1, & \text{если прибор занят,}\\ 2, & \text{если прибор на ремонте.}\end{cases}$$

Процесс $\{k(t), i(t)\}$ изменения состояний данной системы во времени является двумерной цепью Маркова.

Требуется найти стационарное распределение вероятностей числа заявок в источнике повторных вызовов с учетом состояния прибора $P_k(i,t)=P\{k(t)=k,\ i(t)=i\}$, $k=\overline{0,2}$, $i=0,1,2\ldots$.

2. Система дифференциальных уравнений Колмогорова

Для распределения вероятностей $P_k(i,t)$ составим систему дифференциальных уравнений Колмогорова

$$\begin{cases}\partial P_0(i,t)/\partial t=-(\lambda+i\sigma+\gamma_1)P_0(i,t)+\mu_1P_1(i,t)+\mu_2P_2(i,t),\\ \partial P_1(i,t)/\partial t=-(\lambda+\mu_1+\gamma_2)P_1(i,t)+\lambda P_0(i,t)+(i+1)\sigma P_0(i+1,t)+\lambda P_1(i-1,t),\\ \partial P_2(i,t)/\partial t=-(\lambda+\mu_2)P_2(i,t)+\gamma_1P_0(i,t)+\gamma_2P_1(i-1,t)+\lambda P_2(i-1,t).\end{cases}\quad(1)$$

Обозначим $P_k(i)=\lim\limits_{t\to\infty}P_k(i,t)$. Тогда в стационарном режиме система (1) примет вид

$$\begin{cases}-(\lambda+i\sigma+\gamma_1)P_0(i)+\mu_1P_1(i)+\mu_2P_2(i)=0,\\ -(\lambda+\mu_1+\gamma_2)P_1(i)+\lambda P_0(i)+(i+1)\sigma P_0(i+1)+\lambda P_1(i-1)=0,\\ -(\lambda+\mu_2)P_2(i)+\gamma_1P_0(i)+\gamma_2P_1(i-1)+\lambda P_2(i-1)=0.\end{cases}\quad(2)$$

3. Решение системы методом производящих функций

Введем производящие функции $F_k(z)=\sum\limits_{i=0}^{\infty}z^iP_k(i)$, тогда из (2) получаем

$$\begin{cases}-(\lambda+\gamma_1)F_0(z)-\sigma z\partial F_0(z)/\partial z+\mu_1F_1(z)+\mu_2F_2(z)=0,\\ -(\lambda+\mu_1+\gamma_2)F_1(z)+F_0(z)+\sigma\partial F_0(z)/\partial z+\lambda zF_1(z)=0,\\ -(\lambda+\mu_2)F_2(z)+\gamma_1F_0(z)+\gamma_2zF_1(z)+\lambda zF_2(z)=0.\end{cases}\quad(3)$$

Решение системы (3) с учетом условия нормировки $F_1(1)+F_2(1)+F_0(1)=1$ имеет вид $F(z)=F_0(z)+F_1(z)+F_2(z)$, где

$$F_0(z)=R_0\exp\left\{-\frac{\lambda}{\sigma}\int_z^1\left(\frac{(\lambda+\mu_1+\gamma_2-\lambda z)(\lambda+\mu_2-\lambda z+\gamma_1)}{(\mu_1-\lambda z)(\lambda+\mu_2)-\lambda z(\gamma_2+\mu_1-\lambda z)}-1\right)dz\right\},$$

$$R_0=F_0(1)=\frac{\mu_1\mu_2-\mu_2\lambda-\gamma_2\lambda}{\mu_1(\mu_2+\gamma_1)},$$

$$F_1(z)=F_0(z)\frac{\lambda^2+\mu_2\lambda-\lambda^2z+\gamma_1\lambda}{(\mu_1-\lambda z)(\lambda+\mu_2)-\lambda z(\gamma_2+\mu_1-\lambda z)},$$

$$F_2(z)=F_0(z)\frac{\gamma_2z\lambda+\gamma_1\mu_1-\gamma_1\lambda z}{(\mu_1-\lambda z)(\lambda+\mu_2)-\lambda z(\gamma_2+\mu_1-\lambda z)}.$$

Введем частичные характеристические функции: $H_k(u)=\sum_{i=0}^{\infty}e^{jui}P_k(i)$, где $j=\sqrt{-1}$, $k=\overline{0,2}$. Тогда система (2) принимает вид:

$$\begin{cases}-(\lambda+\gamma_1)H_0(u)+j\sigma e^{ju}\dfrac{\partial H_0(u)}{\partial u}+\mu_1H_1(u)+\mu_2H_2(u)=0,\\ -(\lambda+\mu_1+\gamma_2)H_1(u)+\lambda H_0(u)-j\sigma\dfrac{\partial H_0(u)}{\partial u}+\lambda e^{ju}H_1(u)=0,\\ -(\lambda+\mu_2)H_2(u)+\gamma_1H_0(u)+\gamma_2e^{ju}H_1(u)+\lambda e^{ju}H_2(u)=0.\end{cases} \quad (4)$$

Характеристическая функция $H(u)$ числа заявок для системы (4) выражается через частичные характеристические функции $H_k(u)$ следующим равенством: $H(u)=H_0(u)+H_1(u)+H_2(u)$.

Применяя обратное преобразование Фурье к найденной характеристической функции $H(u)$, можно записать распределение вероятностей $P(i)$ в виде $P(i)=\dfrac{1}{2\pi}\int_{-\pi}^{\pi}e^{-jui}H(u)du$

4. Асимптотика первого порядка

Решение системы (4) найдем с помощью метода асимптотического анализа при условии большой задержки на орбите ($\sigma\to 0$). Обозначим $\sigma=\varepsilon$ и сделаем замены: $u=w\varepsilon$, $H_k(u)=F_k(w,\varepsilon)$. Получим следующую систему:

$$\begin{cases}-(\lambda+\gamma_1)F_0(w,\varepsilon)+je^{jw\varepsilon}\dfrac{\partial F_0(w,\varepsilon)}{\partial w}+\mu_1F_1(w,\varepsilon)+\mu_2F_2(w,\varepsilon)=0,\\ -(\lambda+\mu_1+\gamma_2)F_1(w,\varepsilon)+\lambda F_0(w,\varepsilon)-j\dfrac{\partial F_0(w,\varepsilon)}{\partial w}+\lambda e^{jw\varepsilon}F_1(w,\varepsilon)=0,\\ -(\lambda+\mu_2)F_2(w,\varepsilon)+\gamma_1F_0(w,\varepsilon)+\gamma_2e^{jw\varepsilon}F_1(w,\varepsilon)+\lambda e^{jw\varepsilon}F_2(w,\varepsilon)=0.\end{cases} \quad (5)$$

В системе (5) сделаем предельный переход при $\varepsilon\to 0$. Обозначим $\lim_{\varepsilon\to 0}F_k(w,\varepsilon)=F_k(w)$. При $\varepsilon\to 0$ система принимает вид:

$$\begin{cases}-(\lambda+\gamma_1)F_0(w)+j\dfrac{\partial F_0(w)}{\partial w}+\mu_1F_1(w)+\mu_2F_2(w)=0,\\ -(\lambda+\mu_1+\gamma_2)F_1(w)+\lambda F_0(w)-j\dfrac{\partial F_0(w)}{\partial w}+\lambda F_1(w)=0,\\ -(\lambda+\mu_2)F_2(w)+\gamma_1F_0(w)+\gamma_2F_1(w)+\lambda F_2(w)=0.\end{cases}$$

Будем искать решение системы в следующем виде: $F_k(w,\varepsilon)=R_k\Phi(w)$, $\Phi(w)=exp\{G_1jw\}$. Подставим $F_k(w,\varepsilon)$ в систему, получаем:

$$\begin{cases}-(\lambda+\gamma_1)R_0-R_0G_1+\mu_1R_1+\mu_2R_2=0,\\ -(\lambda+\mu_1+\gamma_2)R_1+\lambda R_0+R_0G_1+\lambda R_1=0,\\ -(\lambda+\mu_2)R_2+\gamma_1R_0+\gamma_2R_1+\lambda R_2=0.\end{cases} \quad (6)$$

Выписывая уравнения системы совместно с условием нормировки $R_0 + R_1 + R_2 = 1$, найдем R_0, R_1, R_2: $R_0 = \frac{\mu_1\mu_2 - \mu_2\lambda - \gamma_2\lambda}{\mu_1\mu_2 + \mu_1\gamma_1}$, $R_1 = \frac{\lambda}{\mu_1}$, $R_2 = \frac{\mu_1\gamma_1 + \gamma_2\lambda - \lambda\gamma_1}{\mu_1\mu_2 + \mu_1\gamma_1}$. Подставим R_0, R_1, R_2 в систему (6) и найдем G_1: $G_1 = \frac{\lambda\left(\gamma_2\mu_2 + \mu_1\gamma_1 + \gamma_1\gamma_2 + \mu_2\lambda + \gamma_2\lambda\right)}{\mu_1\mu_2 - \mu_2\lambda - \gamma_2\lambda}$.

Асимптотика первого порядка определяет среднее значение числа поступивших заявок в системе. Для более детального исследования процесса $i(t)$ следует рассмотреть асимптотику второго порядка.

5. Асимптотика второго порядка

Сделаем замены:

$$H_k(u) = exp\left\{\frac{G_1}{\sigma} ju\right\} H_k^{(2)}(u),$$

$$\begin{cases} -(\lambda+\gamma_1)e^{\frac{G_1}{\sigma}ju}H_0^{(2)}(u) + j\sigma e^{ju}\left[e^{\frac{G_1}{\sigma}ju}\frac{G_1}{\sigma}jH_0^{(2)}(u) + e^{\frac{G_1}{\sigma}ju}\frac{\partial H_0^{(2)}(u)}{\partial u}\right] + \\ \quad + \mu_1 e^{\frac{G_1}{\sigma}ju}H_1^{(2)}(u) + \mu_2 e^{\frac{G_1}{\sigma}ju}H_2^{(2)}(u) = 0, \\ -(\lambda+\mu_1+\gamma_2)e^{\frac{G_1}{\sigma}ju}H_1^{(2)}(u) + \lambda e^{\frac{G_1}{\sigma}ju}H_0^{(2)}(u) - \\ \quad - j\sigma\left[e^{\frac{G_1}{\sigma}ju}\frac{G_1}{\sigma}jH_0^{(2)}(u) + e^{\frac{G_1}{\sigma}ju}\frac{\partial H_0^{(2)}(u)}{\partial u}\right] + \lambda e^{ju}e^{\frac{G_1}{\sigma}ju}H_1^{(2)}(u) = 0, \\ -(\lambda+\mu_2)e^{\frac{G_1}{\sigma}ju}H_2^{(2)}(u) + \gamma_2 e^{ju}e^{\frac{G_1}{\sigma}ju}H_1^{(2)}(u) + \lambda e^{ju}e^{\frac{G_1}{\sigma}ju}H_2^{(2)}(u) + \\ \quad + \gamma_1 e^{\frac{G_1}{\sigma}ju}H_0^{(2)}(u) = 0. \end{cases}$$

В предельном условии большой задержки на орбите обозначим $\sigma = \varepsilon^2$, введем следующие замены: $u = w\varepsilon$, $H_k^{(2)}(u) = F_k^{(2)}(w,\varepsilon)$. Получаем систему:

$$\begin{cases} -(\lambda+\gamma_1)e^{\frac{G_1}{\varepsilon}jw}F_0^{(2)}(w,\varepsilon) + je^{jw\varepsilon}e^{\frac{G_1}{\varepsilon}jw}G_1jF_0^{(2)}(w,\varepsilon) + j\varepsilon e^{jw\varepsilon}e^{\frac{G_1}{\varepsilon}jw}\frac{\partial F_0^{(2)}(w,\varepsilon)}{\partial w} + \\ \quad + \mu_1 e^{\frac{G_1}{\varepsilon}jw}F_1^{(2)}(w,\varepsilon) + \mu_2 e^{\frac{G_1}{\varepsilon}jw}F_2^{(2)}(w,\varepsilon) = 0, \\ -(\lambda+\mu_1+\gamma_2)e^{\frac{G_1}{\varepsilon}jw}F_1^{(2)}(w,\varepsilon) + \lambda e^{\frac{G_1}{\varepsilon}jw}F_0^{(2)}(w,\varepsilon) - je^{\frac{G_1}{\varepsilon}jw}G_1jF_0^{(2)}(w,\varepsilon) + \\ \quad + j\varepsilon e^{\frac{G_1}{\varepsilon}jw}\frac{\partial F_0^{(2)}(w,\varepsilon)}{\partial w} + \lambda e^{jw\varepsilon}e^{\frac{G_1}{\varepsilon}jw}F_1^{(2)}(w,\varepsilon) = 0, \\ -(\lambda+\mu_2)e^{\frac{G_1}{\varepsilon}jw}F_2^{(2)}(w,\varepsilon) + \gamma_2 e^{jw\varepsilon}e^{\frac{G_1}{\varepsilon}jw}F_1^{(2)}(w,\varepsilon) + \\ \quad + \lambda e^{jw\varepsilon}e^{\frac{G_1}{\varepsilon}jw}F_2^{(2)}(w,\varepsilon) + \gamma_1 e^{\frac{G_1}{\varepsilon}jw}F_0^{(2)}(w,\varepsilon) = 0. \end{cases} \quad (7)$$

Решение $F_k^{(2)}(w,\varepsilon)$ запишем в виде разложения

$$F_k^{(2)}(w,\varepsilon) = (R_k + jw\varepsilon f_k)\Phi^{(2)}(w) + O(\varepsilon^2). \quad (8)$$

Подставляя (8) в (7) и используя разложение $e^{jw\varepsilon} = 1 + jw\varepsilon + O(\varepsilon^2)$, получаем систему:

$$\begin{cases}\left(-\lambda-\gamma_1-e^{jw\varepsilon}G_1\right)\left(R_0+jw\varepsilon f_0\right)\Phi^{(2)}(w)+j\varepsilon e^{jw\varepsilon}\left(R_0+jw\varepsilon f_0\right)'\Phi^{(2)}(w)+\\ \quad+j\varepsilon e^{jw\varepsilon}\left(R_0+jw\varepsilon f_0\right)\Phi'^{(2)}(w)+\mu_1\left(R_1+jw\varepsilon f_1\right)\Phi^{(2)}(w)+\mu_2\left(R_2+jw\varepsilon f_2\right)\Phi^{(2)}(w)=0,\\ \left(-\lambda-\mu_1-\gamma_2+\lambda e^{jw\varepsilon}\right)\left(R_1+jw\varepsilon f_1\right)\Phi^{(2)}(w)+\left(\lambda+G_1\right)\left(R_0+jw\varepsilon f_0\right)\Phi^{(2)}(w)+\\ \quad+j\varepsilon\left(R_0+jw\varepsilon f_0\right)'\Phi^{(2)}(w)+j\varepsilon\left(R_0+jw\varepsilon f_0\right)\Phi'^{(2)}(w)=0,\\ \left(-\lambda-\mu_2+\lambda e^{jw\varepsilon}\right)\left(R_2+jw\varepsilon f_2\right)\Phi^{(2)}(w)+\gamma_2 e^{jw\varepsilon}\left(R_1+jw\varepsilon f_1\right)\Phi^{(2)}(w)+\\ \quad+\gamma_1\left(R_0+jw\varepsilon f_0\right)\Phi^{(2)}(w)=0.\end{cases} \tag{9}$$

Выполняя несложные преобразования с уравнениями системы (9), получаем:

$$\begin{cases}-G_1 j\varepsilon R_0-\lambda j\varepsilon f_0-\gamma_1 j\varepsilon f_0-G_1 j\varepsilon f_0+\mu_1 j\varepsilon f_1+\mu_2 j\varepsilon f_2+j\varepsilon R_0\dfrac{\Phi'^{(2)}(w)}{w\Phi^{(2)}(w)}=0,\\ \lambda R_1 j\varepsilon-\mu_1 j\varepsilon f_1-\gamma_2 j\varepsilon f_1+\lambda j\varepsilon f_0+G_1 j\varepsilon f_0+j\varepsilon R_0\dfrac{\Phi'^{(2)}(w)}{w\Phi^{(2)}(w)}=0,\\ \lambda R_2 j\varepsilon-\mu_2 j\varepsilon f_2+\gamma_2 R_1 j\varepsilon+\gamma_2 j\varepsilon f_1+\gamma_1 j\varepsilon f_0=0.\end{cases} \tag{10}$$

Полученную систему (10) сократим на общий множитель. Из этой системы можно сделать вывод, что выражение $\dfrac{\Phi'^{(2)}(w)}{w\Phi^{(2)}(w)}$ не зависит от w, следовательно, функцию $\Phi^{(2)}(w)$ можно записать в виде $\Phi^{(2)}(w)=\exp\left\{\dfrac{(jw)^2}{2}G_2\right\}$.

Перепишем систему (10) и получим:

$$\begin{cases}-\left(\lambda+\gamma_1+G_1\right)f_0+\mu_1 f_1+\mu_2 f_2-G_1R_0+R_0G_2=0,\\ (\lambda+G_1)f_0-\left(\mu_1+\gamma_2\right)f_1+\lambda R_1+R_0G_2=0,\\ \gamma_1 f_0+\gamma_2 f_1-\mu_2 f_2+\gamma_2 R_1+\lambda R_2=0.\end{cases} \tag{11}$$

Выписывая уравнения системы (11) совместно с дополнительным условием $f_0+f_1+f_2=0$, найдем f_0, f_1, f_2:

$$f_0=\frac{-G_1R_0\gamma_2-G_1R_0\mu_2-R_1\gamma_2\lambda-R_1\lambda\mu_2-R_2\gamma_2\lambda-2R_2\lambda\mu_1+R_2\lambda\mu_2}{2\left(G_1\gamma_2+G1\mu_2+\gamma_1\gamma_2+\gamma_1\mu_1+\gamma_2\lambda+\gamma_2\mu_2+\lambda\mu_2+\mu_1\mu_2\right)},$$

$$f_1=\frac{G_1R_0\gamma_1+G_1R_0\mu_2-2G_1R_2\lambda+R_1\gamma_1\lambda+R_1\lambda\mu_2-R_2\gamma_1\lambda-2R_2\lambda^2-R_2\lambda\mu_2}{2\left(G_1\gamma_2+G1\mu_2+\gamma_1\gamma_2+\gamma_1\mu_1+\gamma_2\lambda+\gamma_2\mu_2+\lambda\mu_2+\mu_1\mu_2\right)},$$

$$f_2=\frac{-G_1R_0\gamma_1+G_1R_0\gamma_2+2G_1R_2\lambda-R_1\gamma_1\lambda+R_1\gamma_2\lambda+R_2\gamma_1\lambda+R_2\gamma_2\lambda+2R_2\lambda^2+2R_2\lambda\mu_1}{2\left(G_1\gamma_2+G1\mu_2+\gamma_1\gamma_2+\gamma_1\mu_1+\gamma_2\lambda+\gamma_2\mu_2+\lambda\mu_2+\mu_1\mu_2\right)}.$$

Подставим f_0, f_1, f_2 в первое уравнение системы (11) и получим: $G_2=\dfrac{G_1R_0+\left(G_1+\gamma_1+\lambda\right)f_0-\mu_1 f_1-\mu_2 f_2}{R_0}$.

Асимптотика второго порядка показывает, что асимптотическое распределение вероятностей числа поступивших заявок в RQ-системе является гауссовским с параметрами G_1/σ и G_2/σ, что позволяет для распределения $P(i)$ построить аппроксимацию вида $P^{(2)}(i)=\dfrac{(L(i+0{,}5)-L(i-0{,}5))}{(1-L(-0{,}5))}$, где $L(x)$ – функция нормального распределения с параметром G_1/σ и G_2/σ.

6. Численные результаты

Рассмотрим систему с параметрами $\mu_1 = 7$, $\mu_2 = 1$, $\gamma_1 = 0.03$, $\gamma_2 = 0.03$, $\lambda = 3$, $\sigma = 1$. В табл. 1 приведены результаты вычисления математического ожидания числа заявок на орбите для различных значений σ. Для каждого значения σ показаны точное значение $M\{i(t)\}$, асимптотическое значение математического ожидания G_1/σ и относительная погрешность $\Delta = \dfrac{\left| M\{i(t)\} - \dfrac{G_1}{\sigma} \right|}{M\{i(t)\}}$.

Таблица 1

Сравнение асимптотических и аналитических значений математического ожидания

σ	1	0,1	0,01	0,001
$M\{i(t)\}$	3,074	26,075	256,086	2556,197
G_1/σ	2,556	25,557	255,568	2555,678
Δ	0,169	0,020	0,002	0,0002

Отсюда видно, что асимптотический метод можно применять для нахождения среднего числа заявок на орбите при большом времени ожидания на орбите, т.е. при $\sigma < 0,01$.

Заключение

В работе исследована M/M/1 RQ-система с ненадежным прибором. Система уравнений Колмогорова решена методом производящей функции. Получено стационарное распределение вероятностей числа заявок на орбите.

ЛИТЕРАТУРА

1. *Artalejo J.R.* AccessibleBibliographyonRetrialQueues // Progressin 2000–2009 MathematicalandComputer-Modeling. – 2010. – Vol. 51. – P. 1071 – 1081.

2. *Дудин А.Н., Дудин С.А.* Краткий обзор работ в области исследования систем массового обслуживания с ненадежными обслуживающими приборами // В сборнике: Международный конгресс по информатике: информационные системы и технологии, материалы международного научного конгресса. С. В. Абламейко (гл. редактор). 2016. С. 612-616.

3. *Bin Sun, Moon Ho Lee, Sergey A. Dudin, Alexander N. Dudin* Analysis of multiserver queueing system with opportunistic occupation and reservation of servers // Mathematical Problems in Engineering. 2014. ID 178108. P. 1–13.

АСИМПТОТИЧЕСКИЙ АНАЛИЗ RQ-СИСТЕМЫ С ВЫЗЫВАЕМЫМИ ЗАЯВКАМИ И НЕНАДЕЖНЫМ ПРИБОРОМ

Шульгина К.С., Пауль С.В.

Томский государственный университет

shulgina19991999@mail.ru, paulsv82@mail.ru

Введение

RQ-системы – это системы массового обслуживания с повторными вызовами [1,2]. Они характеризуются тем, что поступившая в систему заявка в случае занятости сервера обслуживанием другой заявки, отправляется на орбиту и после случайной задержки пытается вновь встать на прибор. Примером таких систем являются системы обслуживания, как call-центры, моделированию которых посвящены следующие работы [3,4,5].

Мы рассматриваем в качестве математических моделей call-центров, где операторы не только получают вызовы, но и выполняют исходящие вызовы во время простоя прибора – RQ-системы с вызываемыми заявками [6,7,8].

Наряду с этим, рассматриваются RQ-системы с ненадежным прибором, т.е. обслуживающее устройство может выходить из строя и восстанавливаться в течение случай-

ного времени [9,10]. Анализу RQ-систем с ненадежным прибором посвящены статьи [11,12,13].

В данной работе основным методом исследования является метод асимптотического анализа [14,15], который позволяет получить характеристики предложенной системы при выполнении некоторого асимптотического условия.

1. Постановка задачи

Рассмотрим RQ-систему, на вход которой поступает простейший поток заявок с интенсивностью λ. Поступая в систему, заявка входящего потока обнаруживает прибор свободным, занимает его, а прибор начинает обслуживание в течение экспоненциально-распределенного времени с параметром μ_1. Если же заявка, поступая в систему, обнаруживает прибор занятым, она мгновенно уходит на орбиту и осуществляет там случайную задержку в течение экспоненциально-распределенного времени с параметром σ.

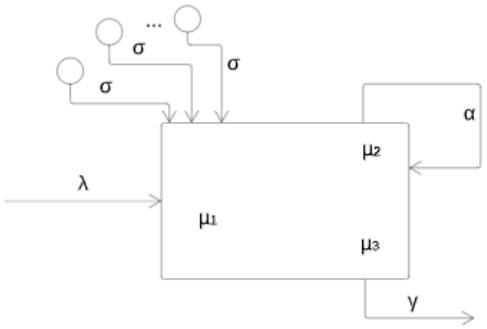


Рис. 1. RQ-система M|M|1 с вызываемыми заявками и ненадежным прибором

Когда прибор не занят обслуживанием (свободен), он вызывает заявку извне с интенсивностью α. Вызываемая заявка встает на прибор и обслуживается в течение экспоненциально-распределенного времени с параметром μ_2.

Будем рассматривать систему с ненадежным прибором, в которой прибор может выходить из строя, когда он свободен или занят обслуживанием заявки входящего потока. Прибор будет находится в состоянии восстановления в течение случайного времени с экспоненциальной функцией распределения с интенсивностью γ выхода из строя прибора, если он находится в *k*-м состоянии, $k = 0,1$. Это означает, что если прибор свободен или занят обслуживанием заявки входящего потока, он выходит из строя с интенсивностью γ. В момент поломки (выхода из строя) прибора обслуживаемая заявка переходит на орбиту.

Когда прибор обслуживает вызываемую заявку или прибор находится в состоянии восстановления, заявки входящего потока уходят на орбиту.

Обозначим процесс $k(t)$ – состояние прибора в момент времени *t*. Этот процесс может принимать следующие значения:

$$k(t) = \begin{cases} 0, \text{ прибор свободен;} \\ 1, \text{ прибор занят обслуживанием заявки входящего потока;} \\ 2, \text{ прибор занят обслуживанием вызываемой заявки;} \\ 3, \text{ прибор находится в состоянии восстановления;} \end{cases}$$

μ_3 – интенсивность восстановления, $i(t)$ – число заявок на орбите в момент времени *t*.

2. Система дифференциальных уравнений Колмогорова

Ставится задача нахождения стационарного распределения числа заявок на орбите. Рассмотрим двумерный марковский процесс $\{k(t), i(t)\}$. Для распределения вероятностей $P\{k(t)=k, i(t)=i\}=P_k(i,t)$, $k=\overline{0,3}$ составим систему уравнений Колмогорова:

$$\begin{aligned}
&-(\lambda+\alpha+i\sigma+\gamma)P_0(i)+\mu_1P_1(i)+\mu_2P_2(i)+\mu_3P_3(i)=0,\\
&-(\lambda+\mu_1+\gamma)P_1(i)+\lambda P_0(i)+\sigma(i+1)P_0(i+1)+\lambda P_1(i-1)=0,\\
&-(\lambda+\mu_2)P_2(i)+\lambda P_2(i-1)+\alpha P_0(i)=0,\\
&-(\lambda+\mu_3)P_3(i)+\lambda P_3(i-1)+\gamma P_0(i)+\gamma P_1(i-1)=0.
\end{aligned} \tag{1}$$

3. Метод характеристических функций.

Введем частичные характеристические функции, обозначив $j=\sqrt{-1}$: $H_k(u)=\sum_{i=0}^{\infty}e^{jui}P_k(i)$, $k=\overline{0,3}$. Перепишем систему (1) для характеристических функций в виде

$$\begin{aligned}
&-(\lambda+\alpha+\gamma)H_0(u)+j\sigma H_0'(u)+\mu_1H_1(u)+\mu_2H_2(u)+\mu_3H_3(u)=0,\\
&-(\lambda+\mu_1+\gamma)H_1(u)+\lambda H_0(u)-j\sigma e^{-ju}H_0'(u)+\lambda e^{ju}H_1(u)=0,\\
&-(\lambda+\mu_2)H_2(u)+\lambda e^{ju}H_2(u)+\alpha H_0(u)=0,\\
&-(\lambda+\mu_3)H_3(u)+\lambda e^{ju}H_3(u)+\gamma H_0(u)+\gamma e^{ju}H_1(u)=0.
\end{aligned} \tag{2}$$

Просуммируем уравнения системы (2):

$$j\sigma e^{-ju}H_0'(u)+\lambda\left(H_1(u)+H_2(u)+H_3(u)\right)+\gamma H_1(u)=0\,.$$

Таким образом, получаем систему из пяти уравнений

$$\begin{aligned}
&-(\lambda+\alpha+\gamma)H_0(u)+j\sigma H_0'(u)+\mu_1H_1(u)+\mu_2H_2(u)+\mu_3H_3(u)=0,\\
&-(\lambda+\mu_1+\gamma)H_1(u)+\lambda H_0(u)-j\sigma e^{-ju}H_0'(u)+\lambda e^{ju}H_1(u)=0,\\
&-(\lambda+\mu_2)H_2(u)+\lambda e^{ju}H_2(u)+\alpha H_0(u)=0,\\
&-(\lambda+\mu_3)H_3(u)+\lambda e^{ju}H_3(u)+\gamma H_0(u)+\gamma e^{ju}H_1(u)=0,\\
&j\sigma e^{-ju}H_0'(u)+\lambda\left(H_1(u)+H_2(u)+H_3(u)\right)+\gamma H_1(u)=0.
\end{aligned} \tag{3}$$

Систему (3) будем решать методом асимптотического анализа в предельном условии $\sigma\to 0$.

4. Асимптотика первого порядка

Обозначим $\sigma=\varepsilon$, сделаем замены: $u=\varepsilon w$, $H_k(u)=F_k(w,\varepsilon)$. Устремим $\varepsilon\to 0$, получим:

$$\begin{aligned}
&-(\lambda+\alpha+\gamma)F_0(w)+jF_0'(w)+\mu_1F_1(w)+\mu_2F_2(w)+\mu_3F_3(w)=0,\\
&-(\lambda+\mu_1+\gamma)F_1(w)+\lambda F_0(w)-jF_0'(w)+\lambda F_1(w)=0,\\
&-(\lambda+\mu_2)F_2(w)+\lambda F_2(w)+\alpha F_0(w)=0,\\
&-(\lambda+\mu_3)F_3(w)+\lambda F_3(w)+\gamma F_0(w)+\gamma F_1(w)=0,\\
&jF_0'(w,\varepsilon)+\lambda\left(F_1(w)+F_2(w)+F_3(w)\right)+\gamma F_1(w)=0.
\end{aligned} \tag{4}$$

Решение системы (4) будем искать в виде $F_k(w)=r_k\Phi(w)$, $k=\overline{0,3}$. Получим

$$-(\lambda+\alpha+\gamma)r_0+jr_0\frac{\Phi'(w)}{\Phi(w)}+\mu_1r_1+\mu_2r_2+\mu_3r_3=0,\ -(\mu_1+\gamma)r_1+\lambda r_0-jr_0\frac{\Phi'(w)}{\Phi(w)}=0,$$

$$-\mu_2r_2+\alpha r_0=0,\ -\mu_3r_3+\gamma r_0+\gamma r_1=0,\ jr_0\frac{\Phi'(w)}{\Phi(w)}+\lambda(r_1+r_2+r_3)+\gamma r_1=0.$$

Т.к. отношение $j\dfrac{\Phi'(w)}{\Phi(w)}$ не зависит от w, то функция $\Phi(w)$ имеет вид $\Phi(w)=\exp\{jw\kappa_1\}$. Здесь $j\dfrac{\Phi'(w)}{\Phi(w)}=-\kappa_1$.

Добавим к системе уравнение $r_0+r_1+r_2+r_3=1$, получим

$$\begin{aligned}
&-(\lambda+\alpha+\gamma+\kappa_1)r_0+\mu_1r_1+\mu_2r_2+\mu_3r_3=0,\\
&-(\mu_1+\gamma)r_1+(\lambda+\kappa_1)r_0=0,\\
&-\mu_2r_2+\alpha r_0=0,\\
&-\mu_3r_3+\gamma r_0+\gamma r_1=0,\\
&-\kappa_1r_0+\lambda(r_1+r_2+r_3)+\gamma r_1=0,\\
&r_0+r_1+r_2+r_3=1.
\end{aligned}\qquad (5)$$

Имеем

$$r_0=\frac{\mu_1\mu_3-\lambda\mu_3-\lambda\gamma}{\mu_1}\cdot\frac{\mu_2}{\mu_2\mu_3+\gamma\mu_2+\alpha\mu_3},\ r_1=\frac{\lambda}{\mu_1},$$

$$r_2=\frac{\mu_1\mu_3-\lambda\mu_3-\lambda\gamma}{\mu_1}\cdot\frac{\alpha}{\mu_2\mu_3+\gamma\mu_2+\alpha\mu_3},\ r_3=\frac{\gamma}{\mu_3}\left(r_0+\frac{\lambda}{\mu_1}\right).$$

Тогда из уравнения $\kappa_1r_0=\lambda(1-r_0)+\gamma r_1$ получим выражение для величины κ_1

$$\kappa_1=\frac{\lambda}{\mu_2}\cdot\frac{\mu_2(\lambda+\gamma)(\mu_3+\gamma)+\alpha\mu_3(\mu_1+\gamma)+\mu_1\mu_2\gamma}{\mu_1\mu_3-\lambda\mu_3-\lambda\gamma}.$$

Величина κ_1 определяет асимптотическое среднее значение κ_1/σ числа заявок на орбите в системе с ненадежным прибором и вызываемыми заявками. Для нахождения распределения вероятностей числа $i(t)$ заявок на орбите рассмотрим асимптотику второго порядка.

5. Асимптотика второго порядка

В системе уравнений (3) сделаем замены $H_k(u)=\exp\left(ju\dfrac{\kappa_1}{\sigma}\right)H_k^{(2)}(u)$, $k=\overline{0,3}$, и в полученной системе сделаем замены $\sigma=\varepsilon^2$, $u=\varepsilon w$, $H_k^{(2)}(u)=F_k^{(2)}(w,\varepsilon)$, $k=\overline{0,3}$, тогда получим систему

$$\begin{aligned}
&-(\lambda+\alpha+\gamma+\kappa_1)F_0^{(2)}(w,\varepsilon)+j\varepsilon F_0^{(2)\prime}(w,\varepsilon)+\mu_1F_1^{(2)}(w,\varepsilon)+\mu_2F_2^{(2)}(w,\varepsilon)+\mu_3F_3^{(2)}(w,\varepsilon)=0,\\
&-\left(\lambda\left(1-e^{j\varepsilon w}\right)+\mu_1+\gamma\right)F_1^{(2)}(w,\varepsilon)+\lambda F_0^{(2)}(w,\varepsilon)+\kappa_1e^{-j\varepsilon w}F_0^{(2)}(w,\varepsilon)-j\varepsilon e^{-j\varepsilon w}F_0^{(2)\prime}(w,\varepsilon)=0,\\
&-\left(\lambda\left(1-e^{j\varepsilon w}\right)+\mu_2\right)F_2^{(2)}(w,\varepsilon)+\alpha F_0^{(2)}(w,\varepsilon)=0,\\
&-\left(\lambda\left(1-e^{j\varepsilon w}\right)+\mu_3\right)F_3^{(2)}(w,\varepsilon)+\gamma F_0^{(2)}(w,\varepsilon)+\gamma e^{j\varepsilon w}F_1^{(2)}(w,\varepsilon)=0,\\
&-\kappa_1e^{-j\varepsilon w}F_0^{(2)}(w,\varepsilon)+j\varepsilon e^{-j\varepsilon w}F_0^{(2)\prime}(w,\varepsilon)+\\
&\quad+\lambda\left(F_1^{(2)}(w,\varepsilon)+F_2^{(2)}(w,\varepsilon)+F_3^{(2)}(w,\varepsilon)\right)+\gamma F_1^{(2)}(w,\varepsilon)=0.
\end{aligned}$$

В последнюю систему подставим разложение $F_k^{(2)}(w,\varepsilon)=\Phi_2(w)\{r_k+j\varepsilon w f_k\}+O(\varepsilon^2)$. Заметим, что скалярная функция $\Phi_2(w)$ определяется в виде $\Phi_2(w)=\exp\left\{\frac{(jw)^2}{2}\kappa_2\right\}$, $\frac{\Phi_2'(w)}{w\Phi_2(w)}=-\kappa_2$. Тогда

$$\begin{aligned}
&-f_0(\lambda+\alpha+\gamma+\kappa_1)+\mu_1 f_1+\mu_2 f_2+\mu_3 f_3=\kappa_2 r_0,\\
&(\lambda+\kappa_1)f_0-f_1(\mu_1+\gamma)=\kappa_1 r_0-\lambda r_1-\kappa_2 r_0,\\
&\alpha f_0-\mu_2 f_2=-\lambda r_2,\\
&\gamma f_0+\gamma f_1-\mu_3 f_3=-\lambda r_3-\gamma r_1,\\
&-\kappa_1 f_0+\lambda(f_1+f_2+f_3)+\gamma f_1=\kappa_2 r_0-\kappa_1 r_0.
\end{aligned}\qquad (6)$$

Рассмотрим отдельно уравнения системы (6)

$$\begin{aligned}
&-f_0(\lambda+\alpha+\gamma+\kappa_1)+\mu_1 f_1+\mu_2 f_2+\mu_3 f_3=\kappa_2 r_0,\\
&(\lambda+\kappa_1)f_0-f_1(\mu_1+\gamma)=\kappa_1 r_0-\lambda r_1-\kappa_2 r_0,\\
&\alpha f_0-\mu_2 f_2=-\lambda r_2,\\
&\gamma f_0+\gamma f_1-\mu_3 f_3=-\lambda r_3-\gamma r_1.
\end{aligned}\qquad (7)$$

Система (7) – неоднородная система линейных алгебраических уравнений для f_k, $k=\overline{0,3}$. Определитель матрицы коэффициентов системы равен 0 (сумма строк матрицы равна нулю), при этом ранг расширенной матрицы равен рангу матрицы коэффициентов, т.е. система совместна и имеет множество решений.

Рассмотрим однородную систему уравнений (5) и неоднородную систему (7). Сравнивая эти системы, можно увидеть, что система (5) является однородной системой для неоднородной системы (7). Тогда решение системы (7) можно записать в виде $f_k(\kappa_2)=Cr_k+g_k(\kappa_2)$, $k=\overline{0,3}$. Здесь C – константа, вероятности r_k определены выше, а величины $g_k(\kappa_2)$ являются частными решениями неоднородной системы (7), удовлетворяющими условию $\sum_{k=0}^{3} g_k(\kappa_2)=0$, т.е.

$$\begin{gathered}
-g_0(\kappa_2)(\lambda+\alpha+\gamma+\kappa_1)+\mu_1 g_1(\kappa_2)+\mu_2 g_2(\kappa_2)+\mu_3 g_3(\kappa_2)=\kappa_2 r_0,\\
(\lambda+\kappa_1)g_0(\kappa_2)-g_1(\kappa_2)(\mu_1+\gamma)=\kappa_1 r_0-\lambda r_1-\kappa_2 r_0,\\
-\mu_2 g_2(\kappa_2)+\alpha g_0(\kappa_2)=-\lambda r_2,\ -\mu_3 g_3(\kappa_2)+\gamma g_0(\kappa_2)+\gamma g_1(\kappa_2)=-\lambda r_3-\gamma r_1,\\
\sum_{k=0}^{3} g_k(\kappa_2)=0.
\end{gathered}$$

Решив последнюю систему, полученные величины $g_k(\kappa_2)$ подставим в выражения для $f_k(\kappa_2)$, а затем в уравнение для нахождения величины κ_2:

$$-\kappa_1 f_0+\lambda(f_1+f_2+f_3)+\gamma f_1=\kappa_2 r_0-\kappa_1 r_0.$$

Таким образом, найденная асимптотика второго порядка показывает, что асимптотическое распределение вероятностей числа $i(t)$ заявок на орбите в рассматриваемой системе с ненадежным прибором является гауссовским с асимптотическим средним κ_1/σ и дисперсией κ_2/σ.

Заключение

В предложенной работе была рассмотрена RQ-система M|M|1 с вызываемыми заявками и ненадежным прибором. Для исследования предложенной модели была по-

строена система дифференциальных уравнений Колмогорова, решая которую в асимптотическом условии большой задержки заявок на орбите, были найдены среднее и дисперсия числа заявок на орбите. Определив гауссовскую плотность распределения вероятностей с такими параметрами, мы получим распределение вероятностей числа заявок на орбите в RQ-системе с вызываемыми заявками и ненадежным прибором в асимптотическом условии большой задержки заявок на орбите.

ЛИТЕРАТУРА

1. *Artalejo J.R.* Retrial queueing systems / J. R. Artalejo, A. Gómez-Corral //Mathematical and Computer Modelling. — 1999. — Vol. 30.

2. *Falin G.* Retrial queues / G. Falin, J. G. C. Templeton. — Chapman and Hall, 1997. — Vol. 75.

3. *Koole Ger* Queueing models of call centers: An introduction / Koole Ger, Mandelbaum Avishai // Annals of Operations Research. — 2002. — Vol. 113, no. 1-4. — P. 41–59.

4. *Deslauriers A.* Markov chain models of a telephone call center with call blending / A. Deslauriers, P. L'Ecuyer, J. Pichitlamken // Computers and operations research. — 2007. — Vol. 34, no. 6. — P. 1616–1645.

5. *Stolletz R.* Performance analysis and optimization of inbound call centers / R. Stolletz // Springer Science and Business Media. — 2003.

6. *Artalejo J.R.* Markovian retrial queues with two way communication / J. R. Artalejo, T. Phung-Duc //Journal of industrial and management optimization. — 2012. — Vol. 8, no. 4. — P. 781-806.

7. *Artalejo J.R.* Single server retrial queues with two way communication / J. R. Artalejo, T. Phung-Duc //Applied Mathematical Modelling. — 2013. — Vol. 37, no. 4. — P. 1811-1822.

8. *Phung-Duc T.* Two-way communication retrial queues with balanced call blending / T. Phung-Duc, W. Rogiest //International Conference on Analytical and Stochastic Modeling Techniques and Applications. — 2012. — P. 16-31.

9. *Nazarov A.* Queueing Theory and Applications / A. Nazarov, J. Sztrik, A. Kvach. — 2017. — Vol. 800. — P. 97–110.

10. *Berczes T.* Information Technologies and Mathematical Modelling / T. Berczes, J. Sztrik, A. Toth, A. Nazarov // Queueing Theory and Applications. — 2017. — Vol. 800. — P. 248–258.

11. *Paul S.* Retrial Queueing Model with Two-Way Communication, Unreliable Server and Resume of Interrupted Call for Cognitive Radio Networks / S. Paul, T. Phung-Duc // Information Technologies and Mathematical Modelling. Queueing Theory and Applications. — Springer, 2018. — P. 213–224.

12. *Nazarov A.* Unreliable Single-Server Queue with TwoWay Communication and Retrials of Blocked and Interrupted Calls for Cognitive Radio Networks / A. Nazarov, T. Phung-Duc, S. Paul // International Conference on Distributed Computer and Communication Networks — Springer, 2018. — P. 276–287.

13. *Muthukrishna S.K.* Performance analysis of an unreliable M/G/1 retrial queue with two-way communication / S. K. Muthukrishnan, D. Aresh, K. Kiseon // Operational Research. — 2018. — P. 1–14.

14. *Nazarov A.* Asymptotic analysis of Markovian retrial queue with two-way communication under low rate of retrials condition / A. Nazarov, S. Paul, I. Gudkova – 2017.

15. *Nazarov A.* Heavy Outgoing Call Asymptotics for MMPP/M/1/1 Retrial Queue with Two-Way Communication / A. Nazarov, T. Phung-Duc, S. Paul //International Conference on Information Technologies and Mathematical Modelling. — Springer, 2017. — P. 28-41.

АВТОРСКИЙ УКАЗАТЕЛЬ

СОДЕРЖАНИЕ

Научное издание

МАТЕРИАЛЫ
Международной научной конференции
«МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ, ТЕХНИЧЕСКИХ И ЭКОНОМИЧЕСКИХ СИСТЕМ»

Томск, 28–30 мая 2020 г.

Под общей редакцией
кандидата технических наук И.С. Шмырина

Издание подготовлено в авторской редакции

Подписано в печать 29.12.2020 г. Формат 70×108 1/16
Печ. л. 20; усл. печ. л. 28.
Тираж 500 экз. Заказ № 4574.

Отпечатано на оборудовании
Издательства Томского государственного университета
634050, г. Томск, пр. Ленина, 36
Тел. 8+(382-2) 52-98-49
Сайт: http://publish.tsu.ru; e-mail: rio.tsu@mail.ru

ISBN 978-5-94621-970-9