

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МАТЕРИАЛЫ

**VIII Международной молодежной научной
конференции**

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

Томск, 26–30 мая 2021 г.

Под общей редакцией И.С. Шмырина

Томск
Издательство Томского государственного университета
2021

ББК 22.17–22.19
УДК 519.2, 519.7, 519.8
Т78

**ЧЛЕНЫ КОЛЛЕГИИ, РУКОВОДИТЕЛИ НАУЧНЫХ РЕДАКЦИЙ
ПО НАПРАВЛЕНИЯМ:**

д-р техн. наук, проф. **А.А. Глазунов** – научная редакция «Механика, математика»; д-р техн. наук, проф. **Э.Р. Шрагер** – научная редакция «Механика, математика»; д-р техн. наук, проф. **А.М. Горцев** – научная редакция «Информатика и кибернетика»; д-р техн. наук, проф. **С.П. Сущенко** – научная редакция «Информатика и кибернетика»; д-р физ.-мат. наук, проф. **В.Г. Багров** – научная редакция «Физика»; д-р физ.-мат. наук, проф. **А.И. Потекаев** – научная редакция «Физика»; д-р биол. наук, проф. **С.П. Кулижский** – научная редакция «Биология»; д-р геол.-минер. наук, проф. **В.П. Парначев** – научная редакция «Науки о Земле, химия»; канд. хим. наук, доц. **Ю.Г. Слижов** – научная редакция «Науки о Земле, химия»; д-р филол. наук, проф. **Т.А. Демешкина** – научная редакция «История, филология»; д-р ист. наук, проф. **В.П. Зиновьев** – научная редакция «История, филология»; д-р экон. наук, проф. **В.И. Канов** – научная редакция «Юридические и экономические науки»; д-р юрид. наук, проф. **В.А. Уткин** – научная редакция «Юридические и экономические науки»; д-р ист. наук, проф. **Э.И. Черняк** – научная редакция «Философия, социология, психология, педагогика, искусствознание»; д-р психол. наук, проф. **Э.В. Галажинский** – научная редакция «Философия, социология, психология, педагогика, искусствознание»

НАУЧНАЯ РЕДАКЦИЯ ТОМА:

к.ф.-м.н., доц. **Бакланова О.Е.**, д.т.н., проф. **Замятин А.В.**, д.ф.-м.н., доц. **Дмитриев Ю.Г.**, д.т.н., проф. **Горцев А.М.**, к.ф.-м.н. **Бронер В.И.**, д.т.н., проф. **Назаров А.А.**, д.ф.-м.н., доц. **Моисеев А.Н.**, к.т.н., доц. **Останин С.А.**, к.т.н. **Шмырин И.С.**

Т78 Труды Томского государственного университета. – Т. 306. Серия физико-математическая: Математическое и программное обеспечение информационных, технических и экономических систем : материалы VIII Международной молодежной научной конференции. Томск, 26–30 мая 2021 г. / под общ. ред. И.С. Шмырина. – Томск : Издательство Томского государственного университета, 2021. – 234 с.

ISBN 978-5-907442-42-9

Сборник содержит материалы VIII Международной молодежной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем», проводившейся 26–30 мая 2021 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 978-5-907442-42-9

© Томский государственный университет, 2021

I. ИММЕРСИВНЫЕ ТЕХНОЛОГИИ, ТЕХНИЧЕСКОЕ ЗРЕНИЕ И ВИДЕОАНАЛИТИКА

DOI: 10.17223/978-5-907442-42-9-2021-1

РУКОВОДСТВО ПО САМОСТОЯТЕЛЬНОМУ ПРОЕКТИРОВАНИЮ И СБОРКЕ МАСШТАБНОЙ МОДЕЛИ С АВТОНОМНЫМ УПРАВЛЕНИЕМ НА ПРИМЕРЕ KOENIGSEGG ONE:1

Дорожкин А.Е.

Томский государственный университет
doliman100@gmail.com

Введение

Масштабная модель – это уменьшенная копия реального автомобиля. Бывают статические и радиоуправляемые модели. На данный момент можно купить либо готовую модель, либо готовый набор для самостоятельной сборки. Но что делать, если желаемая модель находится в недостижимом ценовом диапазоне или не существует вовсе?

С появлением 3D-печати стало возможным производство подобных вещей в домашних условиях. Но на данный момент печать крупномасштабных моделей автомобилей слабо распространена, и найти какую-либо информацию по этому направлению очень сложно. Мы хотим предложить решение, которое поможет любому желающему, не обладающему навыками 3D моделирования, самостоятельно спроектировать и собрать свою масштабную модель. В данном руководстве изложены базовые техники и принципы, которые позволят достичь желаемого результата. Также здесь рассмотрены разные варианты достижения поставленных целей, описаны их преимущества и недостатки.

1. Постановка задачи

Предлагается реализовать модель со следующими свойствами в масштабе 1:8:

1. Сходство с оригиналом.
2. Большой уровень детализации.
3. Интерактивность – наличие подвижных частей, например, возможность открыть дверь или крышку моторного отсека.
4. Автономное управление.

В автомобиле можно выделить несколько ключевых элементов: кузов, подвеска и электронная схема. Всё это проектируется самостоятельно.

Немаловажным является выбор прототипа. Т.к. создание масштабной модели – это долгий процесс, то для рассмотрения был выбран автомобиль, на который будет не жалко затраченного времени. У Koenigsegg One:1 имеются уникальные дверные петли [1], и некоторое время ему принадлежал рекорд 0–300–0 км/ч среди серийных автомобилей. Кроме того, всего было выпущено шесть таких автомобилей, но изготовлено их было семь. В руководстве рассматривается седьмой автомобиль, который принадлежал компании-производителю, и использовался для экспериментов.

2. Кузов

Кузов является самой главной частью автомобиля, т.к. именно он отражает его облик. Отсюда возникает необходимость сделать его не отличимым от настоящего автомобиля.

2.1. Исходные данные

Широко распространён метод самостоятельного моделирования кузова по проекциям. Данный метод требует наличия выдающихся способностей в области 3D-моделирования. Кроме того, происхождение таких проекций зачастую неизвестно. Все находящиеся в открытом доступе проекции описанного здесь автомобиля основаны на проекциях, созданных дизайнером MillergoCG.

Самый достоверный способ передать облик настоящего автомобиля – это взять готовую 3D-модель. Для этого требуется только наличие достаточного количества времени на исправление модели. Никакого навыка 3D-моделирования не понадобится, т.к. в видимую геометрию кузова нельзя вносить никакие изменения.

Самый достоверный и полный источник моделей автомобилей – это видеоигры. Крупные разработчики заключают контракты с производителями автомобилей для включения эксклюзивных автомобилей в игру. Производители заинтересованы в том, чтобы их автомобили выглядели как можно более достоверно, поэтому они предоставляют реальные автомобили для точного 3D-сканирования [2].

Для подведения итога по двум этим методам нужно сказать, что проекции Millergo сняты с модели, которую он изготовил по заказу Electronic Arts для игры Need for Speed: Rivals.

Т.к. технология создания моделей для игр показана на примере видеоигры Forza Horizon 4, данное руководство основывается на модели из этой игры (рис. 1).

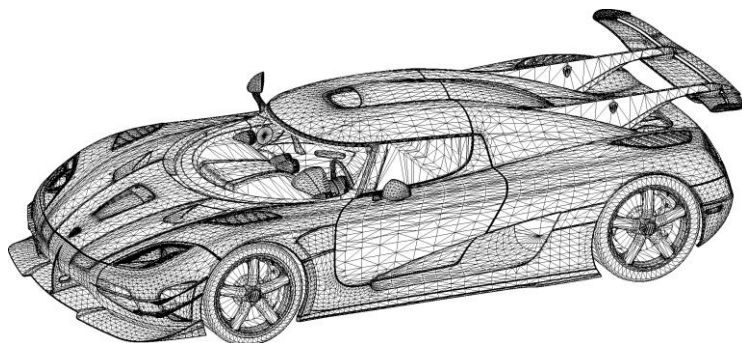


Рис. 1. Модель кузова Koenigsegg One:1

Модели из игр обычно извлекаются из дампов памяти видеокарты, сохранённых во время игры. Для извлечения моделей из серии игр Forza существует специальная программа 3DSimED.

2.2. Обработка модели

Извлечённая модель представляет из себя полигональную сетку. Для работы с такими моделями используется программа 3ds Max. Очень важно при обработке модели не затрагивать внешнюю геометрию кузова. Обработка заключается в вырезании отдельных деталей из сетки, объединении разбитых на разные части деталей в одну, проведение булевых операций над деталями с пересекающимися поверхностями и придание оболочке кузова толщины. В конечном счёте, получается набор панелей, подобных тем, из которых собираются полномасштабные реплики из стекловолокна.

2.3. Параметрическая модель

Моделирование внутренних деталей происходит в программе параметрического моделирования Fusion 360. К таким деталям относятся: дверные петли, подвеска и различные крепления. Кузова реальных автомобилей также параметрические, но такие модели являются коммерческой тайной. Т.к. Fusion 360 плохо справляется с полигональными сетками, лицевая и внутренняя части объединяются в программе 3ds Max.

2.4. Изготовление

Для полноразмерных реплик обычно используется трудоёмкая технология изготовления кузовов из стекловолокна. Преимущество данного метода заключается в том, что он не требует дорогостоящего оборудования.

При создании масштабной модели есть возможность распечатать готовую модель на 3D-принтере из пластика.

Распечатывание лучше начать с какой-то небольшой и несложной детали, чтобы отладить процесс, а на подготовку какой-то большой детали может уйти много времени. Например, можно начать с заготовки диска для колеса. Печать происходит в несколько этапов, самые первые модели нужны просто для понимания происходящего, они не должны обладать совершенной формой или продуманными креплениями, т.к. не будут использоваться в финальной сборке.

3. Подвеска

Конструкция подвески у гоночных радиоуправляемых моделей, доступных на рынке, схожа. Она направлена исключительно на повышение ходовых характеристик модели. С другой стороны, сделать функциональную копию подвески реального автомобиля в масштабе невозможно, в виду наличия очень мелких деталей.

Таким образом, для того, чтобы подвеска внешне напоминала свой прототип, но при этом обладала ходовыми характеристиками, не уступающими самым дешёвым моделям на рынке, при проектировании нужно принять во внимание обе конструкции.

3.1. Исходные данные

На сайте производителя [3] есть некоторое количество измеримых характеристик, таких, как колёсная база, дорожный просвет, радиус поворота и размеры колёс. Модель кузова выступает в качестве отправной точки, для того, чтобы детали подвески не пересекались с кузовными панелями.

В качестве источника информации могут выступать работы студентов, проходивших практику на заводе изготовителя, т.к. компания предоставляет им доступ к техническим моделям. В работе [4] одного из таких студентов мы нашли проекции задней подвески. Стоит отметить, что они совпадают, и даже дополняют модель из Need for Speed: Rivals, что говорит о высоком уровне достоверности обоих источников.

Некоторые детали изготавливать самостоятельно слишком сложно и нецелесообразно, например, амортизаторы и шаровые соединения. Вместо этого приобретаются стандартные заводские запасные части. Таким образом, их размеры тоже являются исходной информацией.

Некоторые характеристики можно косвенно вычислить из имеющихся. Так, угол поворота колёс находится из радиуса поворота, с учётом правила Аккермана.

3.2. Проектирование

Базовый принцип, который лежит в проектировании функциональной подвески, заключается в том, что все размеры должны быть обоснованными. Для примера возьмём проектирование задних рычагов, т.к. о них имеется достаточно много информации.

Сначала создаётся модель колёсного диска. Далее создаётся примерная модель задней подвески. После этого происходит подстройка деталей по ситуации. В данном случае в задней подвеске есть две детали, от которых можно оттолкнуться. В корпусе дифференциала заготовлены крепления под амортизаторы, длина и ход которых известны. При подборе длины рычагов следует руководствоваться принципом "чем больше – тем лучше".

Нижний рычаг ни от чего не зависит, поэтому он должен быть настолько длинным, насколько это возможно. Верхний рычаг связан с амортизатором, т.е. нужно учитывать длину плеч, возникающих со стороны амортизатор, и со стороны колеса. Если

плечо амортизатора будет короче плеча колеса, то на него будет воздействовать повышенное усилие, что приведёт к его низкой эффективности.

Нахождение оптимальной формы рычагов происходит путём использования тестовыхборок. Их суть заключается в создании неполной сборки модели, содержащей только исследуемые части. В данном случае сборка содержит только подрамник, рычаги, амортизатор и обод колеса. Сборка дублируется таким образом, чтобы в одном случае амортизатор находился в полностью сжатом, а в другом случае – в полностью разжатом состоянии, т.к. он определяет ход колеса. Если в одном из крайних положений поверхность рычага пересекается с поверхностью обода, необходимо изменить значения динамических параметров. Параметрами может быть как длина рычагов, так и их точка крепления к поворотному кулаку.

3.3. Изготовление

Для изготовления модель преобразуется в чертежи. В программе Fusion 360 для этого предусмотрен удобный функционал.

Для изготовления деталей подвески 3D-печать не подходит, т.к. на подвеску приходится большие усилия. При изготовлении из пластика все узлы должны были бы быть очень толстыми, или даже монолитными [5], что существенно влияет на внешний вид. Поэтому подвеска изготавливается преимущественно из листового металла.

Сначала сгибается набор маленьких металлических деталей. Затем, путём использования оправок, сваривается в такие пространственные конструкции, как передний подрамник. Отверстия в детали сверлятся на координатном столе после того, как деталь сварена, для обеспечения соосности отверстий.

Отдельно стоит выделить процесс изготовления колёс, т.к. они одновременно являются и частью силовой конструкции, и важным элементом внешнего вида. Необходимость в изготовлении колёс возникает ввиду того, что колёса для радиоуправляемых моделей стандартизированы, и их существует всего несколько разновидностей. Колесо состоит из двух частей – диск и покрышка.

Диск изготавливается из двух частей. Силовая часть вытачивается из металла, а сверху накладывается накладка, распечатанная на принтере. Таким образом, достигается высокая прочность с сохранением внешнего вида.

Покрышки изготавливаются путём отливки из полиуретана с использованием силиконовой формы. Данная методика широко распространена среди владельцев гоночных RC-моделей ввиду дороговизны заводских комплектов покрышек. В данном случае силиконовая матрица снимается не с настоящей покрышки, а с твёрдой заготовки из пластика, распечатанной на 3D-принтере.

4. Электронная схема

Модель оснащается большим количеством иллюминации и сервоприводами, приводящими в движение такие части, как заднее антикрыло. Кроме того, сама модель приводится в движение мощным электромотором и имеет на борту видео камеру. Все эти компоненты требуется как-то контролировать и подавать на них требуемое напряжение.

Для повышения компактности данного комплекса предлагается разработать свою плату расширения, на которой будут расположены все необходимые управляющие компоненты.

Т.к. задача обработки изображения требует вычислительной мощности, недоступной на микроконтроллерах, предлагается использовать для этой цели одноплатный компьютер Raspberry Pi 4 Model B. Таким образом, получается конструкция, занимающая мало места, но обладающая большим потенциалом.

5. Автономное управление

Заключительная часть работы над масштабной моделью – это написание программы управления. Рассмотрим вариант с автономным управлением. Т.к. автономное управление предполагает решение какой-то задачи в автономном режиме, сначала нужно поставить эту задачу. Для примера берётся задача, поставленная в рамках фестиваля по образовательной робототехнике «Hello, Robot» [6]. Задача заключается в том, что роботу необходимо двигаться вдоль кривой, нанесённой на белое поле краской, не отражающей свет.

В качестве входных данных используется только изображение с камеры, закреплённой в передней части модели. Для взаимодействия с изображением используется библиотека OpenCV.

Алгоритм обработки состоит из нескольких этапов.

Первым делом нужно найти на изображении саму линию. Для этого анализируется полоса пикселей, находящаяся в нижней части изображения. Анализ заключается в преобразовании цветов пикселей в градации серого. Для устранения шумов в виде плавных переходов между белым и чёрным цветом, например, лёгких теней, используется фильтр, находящий разницу между соседними пикселями. Таким образом, в местах резкого перехода между цветами получаются всплески, а там, где цвета перетекают плавно, всплески (даже если и есть) – небольшие. В качестве отправной точки берётся точка, находящаяся между самым большим положительным и отрицательным всплесками.

Преимущество камеры перед датчиками цвета, закреплёнными под передним бампером, заключается в прогнозировании движения [7], поэтому после нахождения отправной точки аналогичным образом анализируются пиксели, находящиеся на заданном расстоянии от неё, т.е. теперь анализируется не прямая, а окружность. Найденная на данном шаге точка является продолжением линии, найденной на предыдущем шаге.

Если повторить предыдущий шаг, получим набор точек, отражающий положение линии во всём поле обзора камеры. Таким образом, можно выполнить прогнозирование движения и заранее подготовиться к вхождению в поворот.

Данные шаги позволяют справиться с поставленной задачей. Теперь можно её расширить, запустив модель на «восьмёрке» – это тип замкнутой трассы с перекрёстком в центре. Таким образом, представленного алгоритма оказывается недостаточно, т.к. появляется необходимость различать пути, образующие перекрёсток, и выбирать тот, что приведёт к наименьшему отклонению от текущего курса.

Для решения данной проблемы, когда в процессе анализа окружности возникает ситуация, при которой её диаметра оказывается недостаточно, она расширяется, т.е. теперь анализируется значительно большая область, в которую помещается сразу весь перекрёсток.

Заключение

В качестве доказательства эффективности автономной программы управления можно сказать, что по результатам соревнования, из которого было взято задание, данная модель заняла первое место в старшей категории гонок по линии. Также данная модель принимала участие во всероссийском этапе чемпионата RoboCup Russia Open 2017, который проходил в рамках форума молодых учёных U-NOVUS, по итогам которого в рамках дисциплины Rescue Line заняла шестое место.

В рамках данного проекта было построено две модели. Первая модель использовалась для тестирования алгоритма автономного управления, она и принимала участие в соревнованиях. Вторая модель разрабатывается в соответствии с изложенными здесь техниками. На данный момент изготовлена действующая подвеска из металла, на которую установлены заводские колёса.

ЛИТЕРАТУРА

1. *Koenigsegg C. Door hinge.* [Электронный ресурс] // Google Patents. URL: <https://patents.google.com/patent/US10662687B2/en> (дата обращения: 11.06.2021).
2. *The Making of Forza Motorsport 4: Car Sourcing* [Электронный ресурс] // Официальный YouTube канал Turn 10 Studios. URL: <https://www.youtube.com/watch?v=CWb7cCS16TM> (дата обращения: 11.06.2021).
3. *Характеристики Koenigsegg One:1* [Электронный ресурс] // Официальный сайт Koenigsegg. URL: <https://www.koenigsegg.com/car/one1/> (дата обращения: 11.06.2021).
4. *Oliva M. Rear Cross Frame Re-Design and Optimization in Carbon Fiber.* [Электронный ресурс] // База знаний Политехнического университета Каталонии. URL: <https://upcommons.upc.edu/handle/2099.1/24276> (дата обращения: 11.06.2021).
5. *Sköld O. The 3D-printed Koenigsegg Agera RS* [Электронный ресурс] // The blog of Olle Sköld. URL: <http://depronized.com/2019/04/29/the-3d-printed-koenigsegg-agera-rs/> (дата обращения: 11.06.2021).
6. *Положение о проведении Фестиваля "Hello, Robot".* [Электронный ресурс] // Официальный сайт центра "Планирование карьеры". URL: <http://cpc.tomsk.ru/priglashaem-k-uchastiyu-v-festivale-po-obrazovatelnoy-robototekhnike-hello-robot/> (дата обращения: 11.06.2021).
7. *Дорожкин А.Е., Дорожкин Е.П., Изюмов А.А.* Использование набора Lego Mindstorms 9797 для решения задачи "цветная сортировка" // Научная сессия ТУСУР–2017 : материалы Международной научно-технической конференции студентов, аспирантов и молодых ученых, посвященной 55-летию ТУСУРа. Томск, 10–12 мая 2017 г. – В-Спектр, 2017. – Ч. 1. – С. 263–266.

DOI: 10.17223/978-5-907442-42-9-2021-2

ДЕТЕКТИРОВАНИЕ СТАНДАРТНЫХ ДОРОЖНЫХ ЗНАКОВ НА ИЗОБРАЖЕНИЯХ

Рудов В.А., Приступа А.В., Скворцов А.В.

Томский государственный университет
rudov.vlad2013@gmail.com, andreyp@mail.ru

Введение

Задача распознавания дорожных знаков на фото- и видеоизображениях достаточно часто возникает в самых разных отраслях: в дорожном и железнодорожном хозяйстве при паспортизации и оценке уровня содержания дорог, в беспилотных автомобилях для коррекции цифровой модели дороги при движении по заданному навигационному маршруту [1].

После принятия в 1968 г. Венской конвенции о дорожном движении основная масса знаков в мире была стандартизирована [1], и только небольшое число знаков допускалось оформлять нестандартно текстовыми надписями и стрелками. Такие знаки в России называются знаками индивидуального проектирования (ЗИП) [2]. И если распознавание таких знаков ещё алгоритмически возможно, то их автоматизированная интерпретация весьма сомнительна. Именно поэтому в основном решается задача распознавания стандартных знаков.

Существующие методы решения задачи детектирования объектов на изображении можно разделить на 3 категории [3]:

- опирающиеся на цветовые признаки;
- опирающиеся на форму и контур;
- на основе нейронных сетей.

За последние годы методы, опирающиеся на цветовые признаки, форму и контур объектов, в реальных условиях показали свою недостаточно высокую эффективность, т.к.:

- объект на изображении может находиться не под прямым углом или может оказаться повернут вокруг своей оси;
- цвета на изображении могут быть искажены ввиду различных погодных условий или освещения;
- знак может частично перекрываться другим объектом.

Именно поэтому основные исследования последнего десятилетия нацелены на применение нейронных сетей: как правило, исследователи предлагают различные варианты свёрточных нейронных сетей.

Задача распознавания знаков обычно решается в два этапа: поиск областей на изображении с потенциальными знаками (задача сегментации) и распознавание знака в указанной области (задача идентификации).

В свою очередь, т.к. знаки имеют разную форму (круг, треугольник, квадрат, ромб) и основной цвет (белый, синий, жёлтый), то уже на этапе сегментации можно выделять регионы соответствующей формы (например, региональными свёрточными нейронными сетями серии R-CNN [4]) и существенно сужать последующий поиск в задаче идентификации.

В настоящей работе предлагается единый алгоритм, последовательно решающий всю задачу распознавания стандартных дорожных знаков: от предварительной подготовки изображения с помощью OpenCV [5], предварительного поиска областей методом максимально стабильных экстремальных областей (MSER) [6] и их фильтрации до последующей идентификации знаков с помощью сети ResNet [7]. Для сравнения полученного решения с аналогами использовался набор реальных изображений, а также автоматически подготовленные синтетические изображения знаков.

1. Предлагаемый алгоритм

Общий алгоритм решения поставленной задачи в общем виде представлен на рис. 1. На вход подаётся изображение дороги, которое после предварительной обработки поступает на вход детектора MSER для сегментации изображения. Полученные на выходе детектора регионы проходят фильтрацию и поступают на вход свёрточной нейронной сети, которая выделяет лишь те регионы, которые содержат дорожный знак.

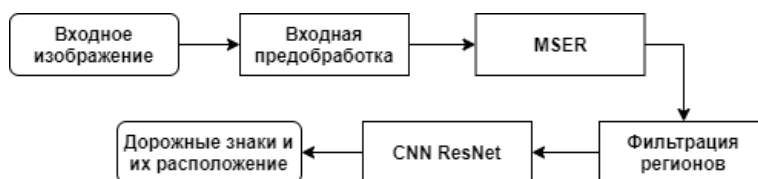


Рис. 1. Схема работы алгоритма детектирования

Для увеличения контрастности входного изображения используется алгоритм улучшения контрастности CLAHE (Contrast Limited Adaptive Histogram Equalization) [8]. В отличие от других алгоритмов он более устойчив к порождению излишнего шума, при этом его реализация доступна в библиотеке OpenCV.

Детектор MSER выделяет множество различных регионов с экстремальными свойствами функции интенсивности внутри региона и на его внешней границе.

В статье Кристиана Миколайджека [9] приведён сравнительный анализ шести детекторов областей, в том числе и детектора MSER, показавший, что:

- MSER обнаруживает наибольшее количество небольших областей при изменении точки обзора;
- MSER занял второе место по обнаружению при изменении масштаба и вращению в плоскости;
- MSER продемонстрировал наибольшую повторяемость результатов при изменении освещения сцены.

Таким образом, для предварительного поиска областей, потенциально содержащих дорожные знаки, в данной работе принято решение использовать OpenCV реализацию MSER-детектора.

Одним из недостатков детектора MSER является обнаружение достаточно большого количества регионов. Чтобы отсеять области, не содержащие дорожные знаки, был

разработан фильтр выходных регионов. Из выделенных MSER областей отбирались лишь те, соотношение сторон которых не превышает 1.5 (т.е. достаточно близкие по форме к квадрату; в действительности же есть знаки с другими пропорциями, но к ним требуется иной подход), а их длина и ширина лежит в пределах от 35 до 200 пикселей. Так отсеиваются области некорректной формы (слишком малые или слишком крупные для дорожного знака).

Пример работы этапа фильтрации представлен на рис. 2. В приведённом примере алгоритмом MSER было выделено 588 областей, а после применения фильтрации осталось 159 областей. Таким образом, применение выходной фильтрации может уменьшить количество выходных регионов в разы.

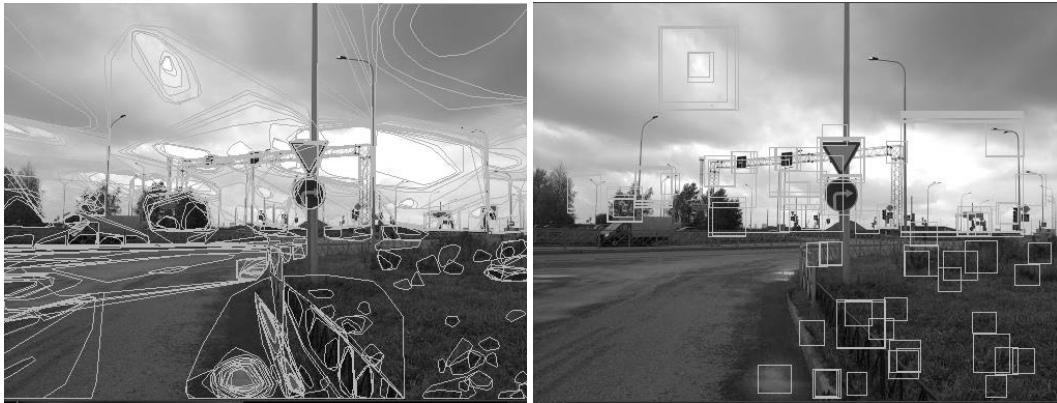


Рис. 2. Результат сегментации изображения на регионы алгоритмом MSER

После работы алгоритма MSER сегментированные регионы необходимо классифицировать. В данной работе предлагается использовать ResNet (Residual Network – «остаточная сеть») – уже классическая свёрточная нейронная сеть, используемая в качестве основы для многих задач компьютерного зрения.

Основная идея нейронной сети ResNet заключается в добавлении так называемых «соединений быстрого доступа» – связей между несмежными слоями (рис. 3). При этом обучение такой сети занимает большее количество времени, но и растёт точность классификации при существенном увеличении глубины.

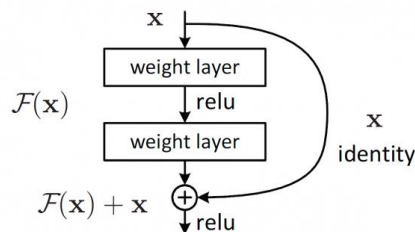


Рис. 3. Соединения быстрого доступа

Существует несколько классических разновидностей сетей ResNet, которые отличаются числом слоёв. Для исследовательских целей (для распознавания небольшого подмножества типов знаков) нами была выбрана сеть ResNet34: она сходится быстрее, чем более глубокая ResNet50, и при этом обеспечивает лучший результат обучения по сравнению с более мелкой сетью ResNet18. При реализации же коммерческого решения (для распознавания сотен видов и подвидов знаков) более предпочтительным является использование ResNet101 или ResNet152.

2. Тестовые наборы данных

Для проверки работоспособности предлагаемого подхода было выбрано подмножество знаков (табл. 1), которые будут распознаваться. Среди них можно выделить как группы знаков, имеющие схожие признаки (круглые, квадратные, треугольные), так и уникальные знаки (ромб, перевёрнутый треугольник).

Таблица 1

Типы детектируемых дорожных знаков

3.1	2.4	3.24			3.20	5.19.1 5.19.2	2.1	6.4	5.20	1.25	1.23	1.17	1.11.1 1.11.2

Для обучения нейронной сети использовались реальные (фотографии дорожных знаков) и синтетические данные (программно сгенерированные изображения).

В качестве реальной основы для обучения и тестирования сверточной нейронной сети взят набор российских дорожных знаков (Russian Traffic Sign Dataset, RTSD). Эта выборка содержит кадры, снятые в различных условиях освещения, погодных условиях и временах года, и содержит 156 классов знаков [10]. На рис. 4 представлены примеры данных из RTSD.



Рис. 4. Примеры знаков 5.19.2 и 1.25 из RTSD

Из всего набора данных RTSD были выделены только требуемые для реализации системы классы дорожных знаков. С целью оценки качества используемого набора данных была построена гистограмма распределения его тренировочных и тестовых данных (рис. 5).

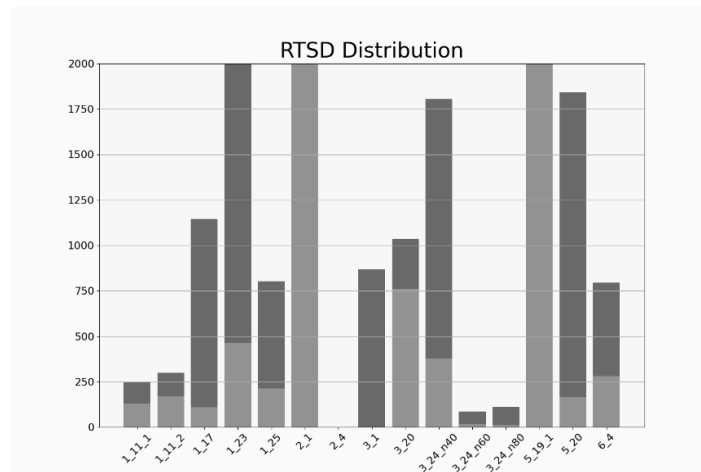


Рис. 5. Диаграмма распределения данных по классам в исходном наборе RTSD, темным цветом обозначены тренировочные данные, светлым – тестовые

Полученное распределение образцов оказалось неравномерным: некоторые данные отсутствуют, а количество изображений знаков 3.24 (60 шт.) и 3.24 (80 шт.) критически мало. Для решения этой проблемы было решено дополнить группы с наименьшим содержанием дорожных знаков (менее 600 шт.) реальными данными из сторонних источников и синтетическими данными, а также удалить образцы размером менее 35x35 пикселей. После проделанных манипуляций распределение данных приняло следующий вид (рис. 6).

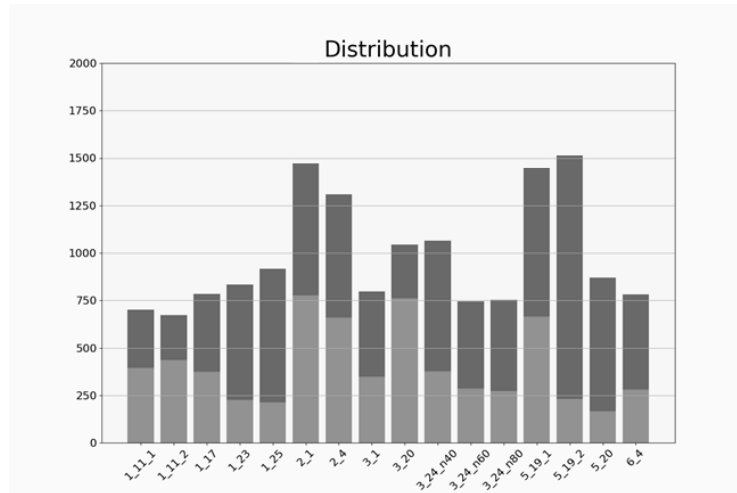


Рис. 6. Конечное распределение используемого набора данных

Полученное распределение является относительно равномерным: в каждом классе присутствует минимум 600 образцов.

Дополнительно в датасет были добавлены изображения, не содержащие дорожных знаков (отрицательные прецеденты). Для генерации таких областей использовалась следующая процедура:

- 1) к набору изображений из тренировочных данных применялся детектор MSER с последующей фильтрацией регионов;
- 2) среди полученных регионов выбирались области, не содержащие дорожный знак, но при этом находящиеся относительно недалеко от него.

Количество таких изображений составило около 20% от общего объёма выборки.

Для дополнения существующей выборки была сгенерирована вспомогательная синтетическая выборка дорожных знаков. В качестве основы для генерации был использован набор векторных изображений дорожных знаков размером 512×512 пикселей из специализированной системы проектирования знаков дорожного движения IndorRoadSigns [11]. К этим векторным изображениям применялись следующие преобразования:

- 1) аффинные преобразования и изменение перспективы (рис. 7);



Рис. 7. Применение аффинных и проективных преобразований

- 2) добавление случайного фона, вырезанного из реальных изображений (рис. 8);



Рис. 8. Примеры изображений с добавленным фоном

3) различного рода искажения, такие, как размытие по Гауссу, размытие в движении; гауссовский шум, эффект дождя, эффект тумана, изменение яркости компонент изображения в цветовом пространстве HSV (рис. 9).



Рис. 9. Примеры изображений с искажениями

Идея применения именно этих способов искажения обусловлена исследованием Шахуро В.И. [12], где полученная на их основе синтетическая выборка показала достойные результаты. Сгенерированная нами синтетическая выборка состоит из 16 классов знаков и включает в себя $350 \times 16 = 5600$ изображений.

3. Реализация свёрточной нейронной сети ResNet34 на Keras

С помощью библиотеки Keras на языке Python нами была реализована архитектура сети ResNet34 со следующими изменениями:

- 1) выходной слой 1000-d fc SoftMax был заменён на полносвязный выходной слой 17-d fc SoftMax. Данная размерность соответствует размерности заданного множества для классификации (16 классов дорожных знаков и 1 класс отрицательных прецедентов);
- 2) в модуль обучения нейронной сети были добавлены планировщик скорости обучения, который помогает модели лучше учиться, и проверка контрольной точки (сохранение модели с наилучшей точностью).

После задания архитектуры полученная модель была скомпилирована со следующими параметрами:

- 1) метод оптимизации: Adam;
- 2) функция потерь: categorical_crossentropy.

Перед использованием изображений в качестве входных данных необходимо произвести их предварительную обработку. Для этого использовался метод библиотеки Keras: `tf.keras.preprocessing.image_dataset_from_directory` [13]. Данный метод позволяет перейти от необработанных данных на диске к объекту `tf.data.Dataset`, который можно использовать для обучения модели. В результате изображения из выборок были приведены к размеру 96×96 пикселей, перемешаны и разделены на партии по 16 экземпляров. Кроме того, из обучающей выборки было выделено 20% данных для выборки валидации, чтобы оценить ожидаемый уровень соответствия модели.

Затем была произведена нормализация значений тензора из интервала от 0 до 255 к интервалу от 0 до 1, т.к. нейронные сети лучше работают с маленькими входными значениями. Обучение производилось в течение 35 эпох и заняло около 1 часа. График процесса обучения представлен на рис. 10.

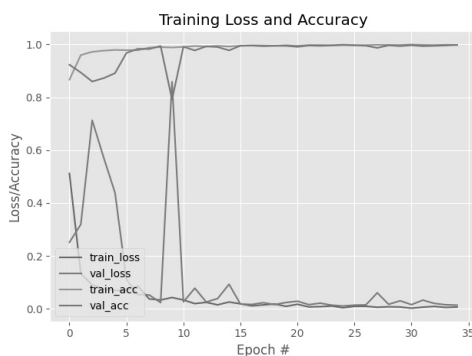


Рис. 10. График обучения нейронной сети

Достигнутая точность на тестовой выборке составила 0,96, что является достаточно хорошим результатом. После обучения модель была сохранена в формате двоичных данных HDF5 и в дальнейшем может использоваться без необходимости обучать её снова.

4. Тестирование и оценка точности

Для того чтобы оценить результат работы системы, было произведено её тестирование на изображениях реальных дорожных сцен. При этом производился подсчёт характеристик *False negative* (число случаев, когда алгоритм не выделил дорожные знаки, а они были) и *False positive* (число случаев, когда алгоритм выделил дорожные знаки там, где их не было).

Количество тестовых изображений составило 100 шт., на которых изображено 187 знаков дорожного движения. Результаты тестирования представлены в табл. 2.

Таблица 2

Результаты тестирования

Общее количество	Детектировано корректно	False negative	False positive
187	173	14	16

Полученные результаты свидетельствуют о том, что предлагаемый алгоритм вполне пригоден для практического применения. Ниже представлено несколько характерных примеров работы алгоритма (рис. 11.1 – 11.6).



Рис. 11.1. Все знаки локализованы корректно



Рис. 11.2. Большое количество знаков локализовано корректно



Рис. 11.3. Знаки, расположенные под углом, локализованы корректно



Рис. 11.4. Знак, заслонённый препятствием, локализован корректно



Рис. 11.5. Вместе со знаком 3.1 локализован объект, схожий с дорожным знаком



Рис. 11.6 – Алгоритм локализовал не все изображенные знаки

Заключение

В работе предложен способ автоматического распознавания стандартных дорожных знаков, основанный на совместном применении алгоритма сегментации областей MSER и детектирования знаков свёрточной нейронной сетью ResNet. Для обучения нейронной сети использован открытый набор данных RTSD и собственный синтезированный набор изображений дорожных знаков с различными искажениями. Для этого в специализированном редакторе дорожных знаков IndorRoadSigns создавались эталонные изображения, а на их основе генерировались различные синтетические изображения с помощью аффинных и проективных преобразований, различных растровых эффектов, а также добавления случайного заднего фона, вырезанного из реальных изображений.

Полученный программный модуль способен определять на изображениях некоторые стандартные знаки дорожного движения из указанного в работе списка в различных условиях. Для детектирования большего количества знаков обучение нейросети необходимо проводить на большем наборе данных, включающем в себя недостающие знаки. Для распознавания нестандартных знаков индивидуального проектирования требуются иные подходы, выходящие за рамки данной статьи.

ЛИТЕРАТУРА

1. Венская конвенция о дорожном движении. – 1968. – 19 с.
2. ГОСТ 52290-2004. Технические средства организации дорожного движения. Знаки дорожные. Общие технические требования. – 194 с.
3. Saadna Y., Behloul A. An overview of traffic sign detection and classification methods. // Int. J. Multimed Info Retr. – 2017. – V. 6. – P. 193–210.
4. Gandhi R. R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms // Towards Data Science. – 2018. – July 9.
5. OpenCV [Электронный ресурс] // URL: <http://www.opencv.org>
6. Matas J., Chum O., Urban M., Pajdla T. Robust wide-baseline stereo from maximally stable extremal regions // Image and Vision Computing. – 2004. – V. 22(10). – P. 761–767.
7. He K., Zhang X.; Ren Sh., Sun J. Deep Residual Learning for Image Recognition". arXiv: 1512.03385. 2015-12-10.
8. Pizer S.M., Amburn E.P., Austin J.D. et al. Adaptive Histogram Equalization and Its Variations // Computer Vision, Graphics and Image Processing. – 1987. – V. 39. – P. 355–368.
9. Mikolajczyk K., Tuytelaars T., Schmid C., Zisserman A., Matas J., Schaffalitzky F., Kadir T., Gool L.V. A comparison of affine region detectors // International Journal of Computer Vision. – 2005. – V. 65. – P. 43–72.
10. Shakhuro V.I., Konushin A.S. Russian traffic sign images dataset // Computer Optics. – 2016. – V. 40. – № 2. – P. 294–300.
11. Система проектирования дорожных знаков IndorRoadSigns [Электронный ресурс] // URL: <https://indorsoft.ru/products/roadsigns>
12. Шахуро В.И. Распознавание редких дорожных знаков с использованием синтетических обучающих выборок: диссертация на соискание учёной степени кандидата компьютерных наук. Высшая школа экономики, Москва, 2020. URL: <https://www.hse.ru/sci/diss/426009543>
13. TensorFlow image_dataset documentation [Электронный ресурс] // URL: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image_dataset_from_directory

ОБНАРУЖЕНИЕ КАЛЬЦИНАТОВ В ЛЁГКИХ С ПОМОЩЬЮ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Сыч М.Б., Аксёнов С.В.

*Томский государственный университет
msych98@gmail.com*

Введение

На сегодняшний день существует множество работ, связанных с применением компьютерного зрения в медицине, чаще всего решается задача сегментации – выделения множеств пикселей из исходного изображения. Например, в [1] авторы обнаруживают паренхиму легких на КТ-изображениях с использованием сверточной нейронной сети, обученной на наборе данных, предварительно обработанном с помощью алгоритма кластеризации. В [2] авторы с помощью глубоких нейронных сетей проводили сегментацию изображений человеческих ооцитов. В [3] авторы создали систему для обнаружения рака легких в наборе данных, собранных из иракских больниц, с использованием метода сверточной нейронной сети с архитектурой AlexNet. Такое количество исследований, часть из которых применяются на практике, говорит о том, что компьютерное зрение востребовано в медицине. Мы не нашли исследований, связанных с обнаружением кальцинатов в лёгких при помощи компьютерного зрения, что обосновывает актуальность данной работы.

Кальцинаты – это плотные скопления солей кальция в мягких тканях организма. Зачастую причиной появления кальцинатов в легких является туберкулез. Но также они могут появляться как результат перенесенной пневмонии, наличия инородных тел в легких, глистных инвазий, опухолевых образований, нарушения обмена кальция и т.д.. Чаще всего кальцинаты в лёгких не представляют угрозы для здоровья и дыхательной функции, однако важно установить причину их образования с целью профилактики дальнейшего роста и уточнения статуса первичного заболевания. Важно оценивать их объём с течением времени. Вручную это сделать довольно сложно и займёт большое количество времени врачей.

1. Постановка задачи

Целью данной работы является создание программы, которая позволит обнаруживать кальцинаты на снимках компьютерной томографии легких и оценивать их объём. Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить похожие работы и тематическую литературу.
2. Получить снимки КТ лёгких с кальцинатами и произвести их разметку.
3. Выбрать наиболее подходящую архитектуру нейронной сети с помощью экспериментов.
4. Провести качественную оценку полученных результатов.
5. На основе полученной оценки изменять архитектуру с целью получения лучших результатов.

2. Подготовка данных

Для разметки данных использовалась программа MITK, она позволяет работать с различными форматами КТ, например, dcm и nii. После того как мы получили снимки, мы просмотрели их и разделили снимки пациентов с кальцинатами, без кальцинатов и с медицинскими инструментами. Это было сделано, чтобы в дальнейшем было удобно оценивать модель.

Каждый снимок представляет собой трёхмерный тензор с длиной и шириной 512 пикселей и глубиной от 113 до 137 срезов, каждое значение которого – значение на

шкале Хаунсфилда. Шкала Хаунсфилда – количественная шкала рентгеновской плотности, включает 4096 значений от -1024 до $+3071$ единиц Хаунсфилда (HU). Экран монитора может отображать максимум 256 оттенков серого. Для отображения всего диапазона на одну градацию серого цвета пришлось бы $4096/256 = 16$ HU. Это означает, что невозможно было бы отличить по яркости ткани, разность плотностей которых менее 16 HU (например, серое и белое вещество головного мозга). С помощью окна визуализации осуществляется регулировка яркости и контрастности КТ-изображений таким образом, чтобы на экране монитора в шкале серого цвета были видны только ткани требуемого диапазона плотностей – определенного участка шкалы Хаунсфилда. Ткани с меньшей плотностью будут перекрыты черным цветом, ткани с большей плотностью – белым. Окно визуализации характеризуется параметрами ширины и центра. Ширина окна – это величина разности наибольшего и наименьшего значений КТ-чисел, отображаемых на экране монитора в шкале серого цвета. Центр, или уровень окна – это значение КТ-числа, расположенного по середине между наибольшим и наименьшим значениями отображаемого диапазона КТ чисел.

Для того, чтобы правильно произвести разметку, необходимо было для каждого снимка устанавливать одинаковые параметры окна. Процесс разметки происходил следующим образом. Изначально все снимки были в формате dcm, для удобства дальнейшей работы мы сохраняли их в формат nii, сам снимок это не изменяло. Для разметки использовалась ширина окна 1, центр – 800 HU, что позволяло исключить ошибки разметки, т.к рисунок будет иметь всего 2 оттенка серого (чёрный и белый).

После этого каждый белый пиксель в области лёгкого размечался вручную. Кальцинаты имеют широкий диапазон плотностей, схожий с плотностью костей. Чтобы ошибочно не разметить кости, каждый срез снимка дополнительно проверялся с шириной окна 2500 и центром 800 HU, это позволяло более детально рассмотреть срез, т.к. снимок будет представлен 256 оттенками серого.

В результате было размечено 12 снимков пациентов с кальцинатами, 2 снимка пациентов с медицинскими инструментами и 2 снимка пациентов без кальцинатов.

3. Подготовка обучающих выборок

Поскольку часть снимка не несет полезной информации и ресурсы компьютера ограничены, было решено нарезать снимки на фрагменты. Такой подход часто применяется в глубоком обучении, например, в [1] исследователи выделили нужную область лёгкого с помощью алгоритма кластеризации, возможно, мы тоже попробуем применить этот подход, но на текущий момент используется упрощенная версия. Также можно использовать подход нарезки на трёхмерные фрагменты, но для этого нужно использовать другую архитектуру нейронной сети.

Изначально для каждого снимка устанавливались следующие параметры окна: центр 900 и ширина 2200 HU, затем каждый снимок нарезался на срезы 512 на 512 пикселей и сохранялся в формате png. Для каждого снимка число срезов разное, в нашей выборке число варьировалось от 113 до 137.

На полученных срезах наблюдалось, что существенная область вокруг туловища просто чёрная и не несёт в себе полезной информации. Как упоминалось ранее, из таких срезов можно извлечь область, содержащую только лёгкие, как в [1]. Но мы пошли более простым путем и для каждого снимка просто вырезали сверху и снизу полосы размера 512 на 64 пикселей, что позволило избавиться от 25% пикселей. После этого мы нарезали каждый срез на фрагменты 128x128 пикселей, в результате с каждого среза получилось по 12 фрагментов. Для масок выполнялась аналогичная нарезка, за исключением этапа с параметрами окна. На рис. 1 продемонстрирован результат нарезки для снимков, на рис. 2 – для масок.

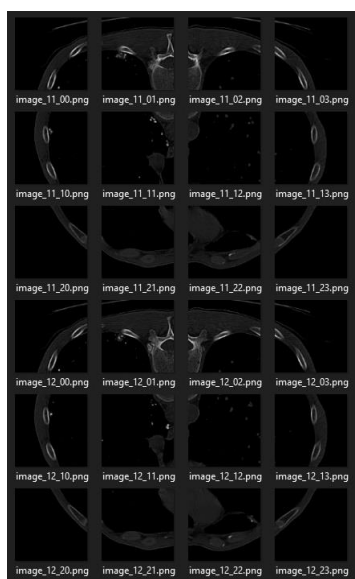


Рис 1. Пример нарезки для снимков



Рис. 2. Пример нарезки для масок

Если посмотреть на маски на рис. 2, соответствующие фрагментам срезов, то можно заметить, что подавляющее большинство масок просто чёрные и не содержат белых пятен разметки. Из этого можно сделать вывод, что обучающая выборка сильно несбалансирована. Для обучения и оценки модели потребуется три выборки: обучающая, валидационная и тестовая. Тестовая и валидационные выборки представляют собой часть фрагментов, каждая из которых была получена из двух различных снимков КТ. Один снимок – пациента с медицинским инструментом в области грудной клетки, чтобы проверить будет ли модель ошибочно классифицировать инструменты, как кальцинаты, и один снимок – пациента с кальцинатами. Таким образом, в валидационной выборке получилось 2340 фрагментов, а в тестовой – 2274.

Для тренировочной выборки возможно использовать все оставшиеся размеченные снимки пациентов с кальцинатами и без медицинских инструментов, это более 8000 фрагментов, но, как отмечалось ранее, большая часть масок – это просто чёрные изображения без белых пятен. Поэтому, было решено сделать три тренировочные выборки и сравнить для них результаты. Первая выборка – без чёрных масок, содержит только те изображения, для которых маски имеют белые пятна. Вторая выборка – смешанная, берётся равное количество изображений с кальцинатами и без них, получается выборка вдвое больше, чем выборка без чёрных масок. Третья выборка содержит все фрагменты исходных снимков.

После разделения выборок на тренировочную, валидационную и тестовую все фрагменты переводились в двумерные массивы целых чисел от 0 до 255. Затем проводилась аугментация данных, это практика часто используется в глубоком обучении. Аугментация позволила немного расширить обучающий набор данных. После аугментации все значения массива делились на 255, чтобы они находились в диапазоне от 0 до 1, соответственно, теперь все массивы содержат числа с плавающей точкой. Аугментация реализовывалась с помощью библиотеки Keras.

4. Архитектура нейронной сети

В 2015 г. была предложена архитектура нейронной сети U-Net для сегментации биомедицинских изображений [4], с того момента появилось множество работ [1,2,3,5], где эта архитектура успешно применяется и изменяется для конкретной задачи. Оригинальная архитектура U-Net представлена на рис. 3.

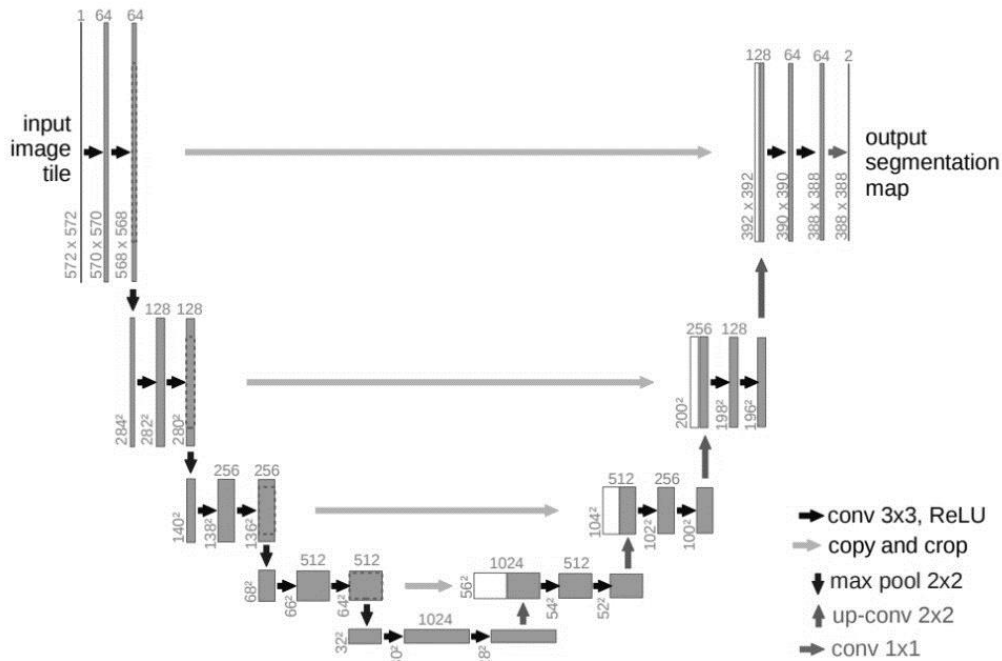


Рис 3. Оригинальная архитектура U-Net

Архитектура U-Net имеет свёрточную часть слева и развёрточную часть справа. Свёрточная часть состоит из блоков свёртки, каждый блок свёртки состоит из повторяющихся применений двух свёрток с ядром 3x3 и функцией активации ReLU; стоит обратить внимание на то, что перед применением свёртки не используется эффект дополнения. Выходная карта признаков блока свёртки сохраняется для дальнейшего использования, после чего к ней применяется операция выбора максимальных значений из соседних с ядром 2 (max pooling), каждый такой блок удваивает количество фильтров в выходной карте признаков и уменьшает в два раза длину и ширину карты признаков. Оригинальная сеть содержит 4 блока свёртки, 5-й блок повторяет предыдущие, но там отсутствует операция выбора максимальных значений из соседних. Назовём этот слой базовым. Начиная с базового блока, операция выбора максимальных значений заменяется операцией повышающей дискретизацией (upsampling). Возможно также использовать операцию Conv2DTranspose, все операции реализованы в библиотеки Keras.

В ходе работы было опробовано несколько архитектур с различными настройками гиперпараметров. Для всех моделей было добавлены дополнения перед операцией свёртки и изменён размер входа. Среди опробованных вариантов были следующие: U-Net классическая с добавлением прореживания (dropout) между блоками, U-Net классическая, U-Net классическая, но в качестве функции потерь использовалась отрицательная мера Жаккара, U-Net классическая, но вместо UpSampling использовалось Conv2DTranspose. Последняя модель показала себя лучше всего, её архитектуру можно посмотреть на рис. 4.

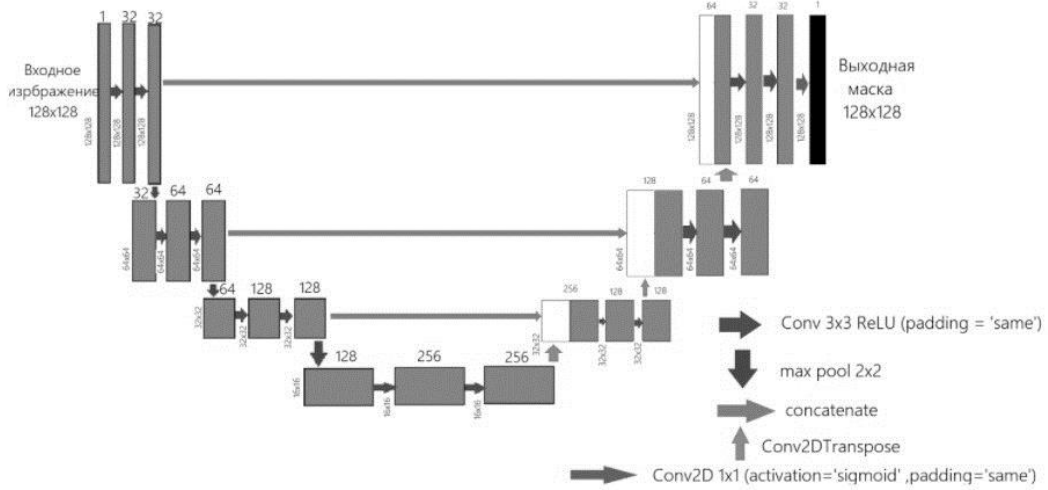


Рис 4. Архитектура U-Net, используемая в данной работе

5. Обучение и оценка

После подготовки тренировочных, валидационных и тестовых выборок и построения архитектуры модели необходимо правильно выбрать метрику оценки качества. Если выбрать неподходящую метрику, то можно получить высокую оценку модели, даже если в действительности она будет работать плохо. Далее в формулах используются следующие обозначения: TP, TN, FP и FN (true positive, true negative, false positive, false negative). TP – результаты положительны, метка модели совпала с реальной, TN – результаты отрицательны, метка модели совпала с реальной, FP – модель предсказала положительный класс, на самом деле он отрицательный, FN – модель предсказала отрицательный класс, на самом деле он положительный.

Рассмотрим формулы accuracy и IOU.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \quad (1)$$

$$\text{IOU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \quad (2)$$

Рассмотрим пример предсказания модели и сравнение его с реальными данными для нашей модели на рис. 5.

TP = 32 TN = 16348 FP = 0 FN = 4

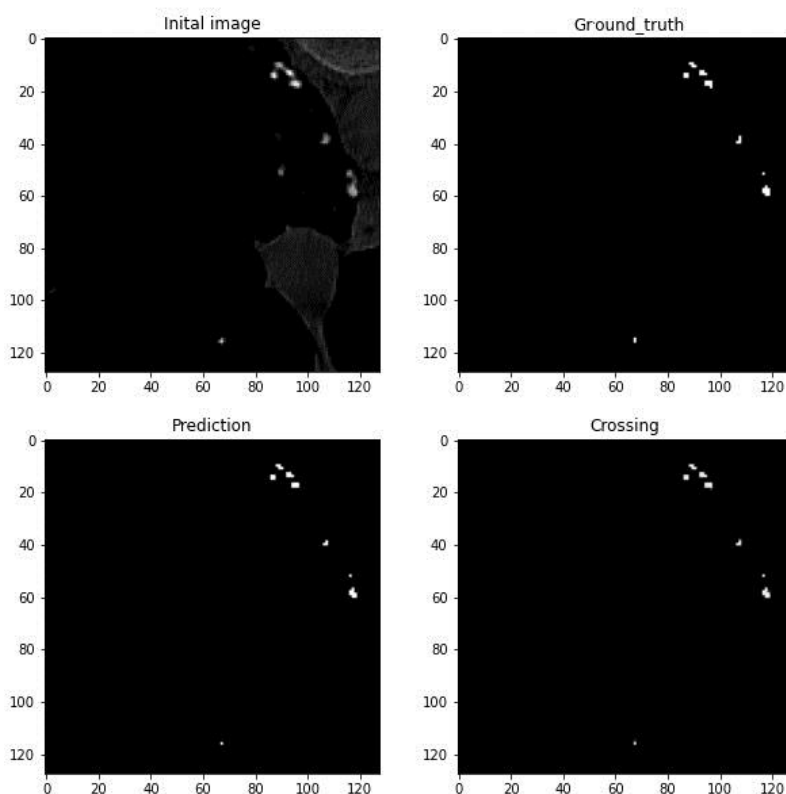


Рис. 5. Сравнение и пересечение предсказаний модели с реальной маской

В некоторых задачах сегментации метрика (1) не подойдет. Если посмотреть на (1), то можно легко посчитать, что даже если модель всегда будет предсказывать чёрный квадрат и никогда не предсказывать целевой класс, то в случае с примером из рис. 5 TP, TN, FP, FN будут соответственно равны 0, 16352, 0, 32, и метрика ассурасу будет равна 0,998, почти 100% для модели, которая совсем не выполняет поставленную задачу. Поэтому для оценки модели нельзя использовать (1).

Часто для оценки моделей сегментации используют метрику IOU (мера Жаккара) – бинарная мера сходства, предложенная Полем Жаккаром в 1901 г. [6]. Если рассмотреть предыдущий пример для модели, которая всегда предсказывает чёрный квадрат, то IOU для неё будет равно 0. Если же рассмотреть исходный пример из рис. 5, то IOU будет равно 0.889, что более правдоподобно.

6. Результаты обучения и экспериментов

В ходе экспериментов было протестировано 4 различные архитектуры U-Net. U-Net классическая с добавлением прореживания (dropout), U-Net классическая, U-Net классическая, но в качестве функции потерь использовалась отрицательная мера Жаккара показали результаты IOU ниже 10% на тестовой выборке. Для модели из рис. 4 результаты экспериментов приведены в табл. 1.

Таблица 1

Модель	IOU	Количество эпох	Выборка
U-Net Conv2D	0.66845	25	Смешанная
U-Net Conv2D	0.62066	50	Смешанная
U-Net Conv2D	0.71550	25	Полная

Модель	IOU	Количество эпох	Выборка
U-Net Conv2D	0.66559	50	Без чёрных масок
U-Net Conv2D	0.71424	100	Без чёрных масок
U-Net Conv2D	0.63126	25	Без чёрных масок
U-Net Conv2D	0.74064	50	Без чёрных масок
U-Net Conv2D	0.70785	75	Без чёрных масок

Заключение

В результате получилось создать модель, которая показывает хорошие результаты на тестовой выборке и не классифицирует ошибочно медицинские инструменты и кости. Это первый этап работы, и в дальнейшем планируется расширить обучающую выборку, провести эксперименты с другими архитектурами моделей и попробовать гистограммные методы сегментации.

ЛИТЕРАТУРА

1. *Xu M., Qi S., Yue Y. et al.* Segmentation of lung parenchyma in CT images using CNN trained with the clustering algorithm generated dataset // BioMed Eng OnLine. 2019 // [Электронный ресурс] <https://doi.org/10.1186/s12938-018-0619-9>
2. *Targosz A., Przystalka P., Wiaderkiewicz R. et al.* Semantic segmentation of human oocyte images using deep neural networks. BioMed Eng OnLine 20, 40. 2021 // [Электронный ресурс] <https://doi.org/10.1186/s12938-021-00864-w>
3. *Hamdalla F. Al-Yasriy, Muayed S. AL-Husieny, Furat Y. Mohsen, Enam A. Khalil, Zainab S. Hassan.* Diagnosis of Lung Cancer Based on CT Scans Using CNN // [Электронный ресурс] <https://iopscience.iop.org/article/10.1088/1757-899X/928/2/022035/pdf>
4. *Ronneberger O., Fischer P., Brox T.* U-Net: Convolutional Networks for Biomedical Image Segmentation // [Электронный ресурс] <https://arxiv.org/pdf/1505.04597.pdf>
5. *Xiao Z., Liu B., Geng L., Zhang F., Liu Y.* Segmentation of Lung Nodules Using Improved 3D-UNet Neural Network // [Электронный ресурс] <https://iopscience.iop.org/article/10.1088/1757-899X/928/2/022035/pdf>
6. *Jaccard P.* Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines // Bull. Soc. Vaudoise sci. Natur. – 1901. – V. 37. –Bd. 140. – P. 241–272.

II. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, МАШИННОЕ ОБУЧЕНИЕ, БОЛЬШИЕ ДАННЫЕ

DOI: 10.17223/978-5-907442-42-9-2021-4

КОМИТЕТНЫЕ МЕТОДЫ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧЕ РАСПОЗНАВАНИЯ РУКОПЕЧАТНЫХ СИМВОЛОВ

Авдеев Н.Г.

Томский политехнический университет
ndb1@tpu.ru

Введение

Контроль знаний, умений и навыков является важным звеном учебного процесса. От его организации во многом зависит результат обучения. В процессе контроля выявляются как достоинства, так и недостатки знаний и умений учащихся, что позволяет управлять учебным процессом, совершенствуя формы и методы обучения. Массовые тестирования проводятся по регламентируемым процедурам и требуют особого подхода к организации и обработке. К такому методу контроля знаний относится и Государственная итоговая аттестация (ГИА) – форма оценки степени и уровня освоения обучающимися основных образовательных программ, соответствующим требованиям федерального государственного образовательного стандарта. Для ознакомления учеников 9 и 11 классов с заданиями ГИА, а также его процедурой проведения проводятся диагностические работы ОГЭ и ЕГЭ. Существующий формат проведения данного мероприятия ставит перед организаторами задачу анализа информации, представленной в виде изображений – произвести так называемое off-line распознавание уже написанного на бумаге текста. Задача обработки и распознавания изображений относится к разряду трудно формализуемых задач и является одной из наиболее важных на сегодняшний день.

1. Выбор технологии распознавания

Для поставленной задачи были выбраны сверточные нейронные сети. Выбор сверточных нейронных сетей обусловлен следующими факторами:

- начиная с 2012 г. сверточные сети занимают первые места на конкурсе ImageNet;
- сверточные нейронные сети устойчивы к аффинным преобразованиям;
- сверточные нейронные сети учитывают топологию изображений [1].

2. Подготовка обучающей выборки

Выборка подготовлена из бланков ЕГЭ прошлых лет. Исходная выборка состоит из 47850 монохромных изображений размером 32x32 (по 1450 изображений для каждого класса буквы). Примеры изображений продемонстрированы на рис. 1.

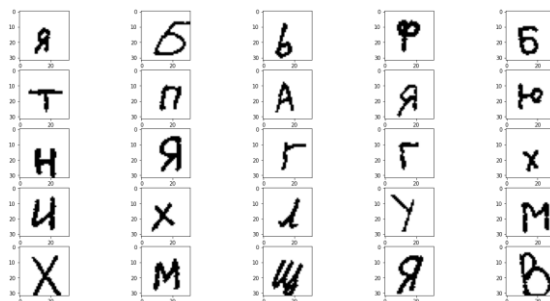


Рис. 1. Примеры изображений букв

Выборка была разделена на тренировочную, валидационную и тестовую в соотношении 70%, 20% и 10% соответственно [2].

3. Вычислительные эксперименты

Для экспериментов была выбрана модель сверточной нейронной сети LeNet5 [3]. Количество нейронов на выходном слое было увеличено до 33 (по количеству классов). Данная нейронная сеть хорошо себя зарекомендовала в задачах распознавания рукописных символов. Архитектура LeNet5 для распознавания рукописных цифр продемонстрирована на рис. 2.

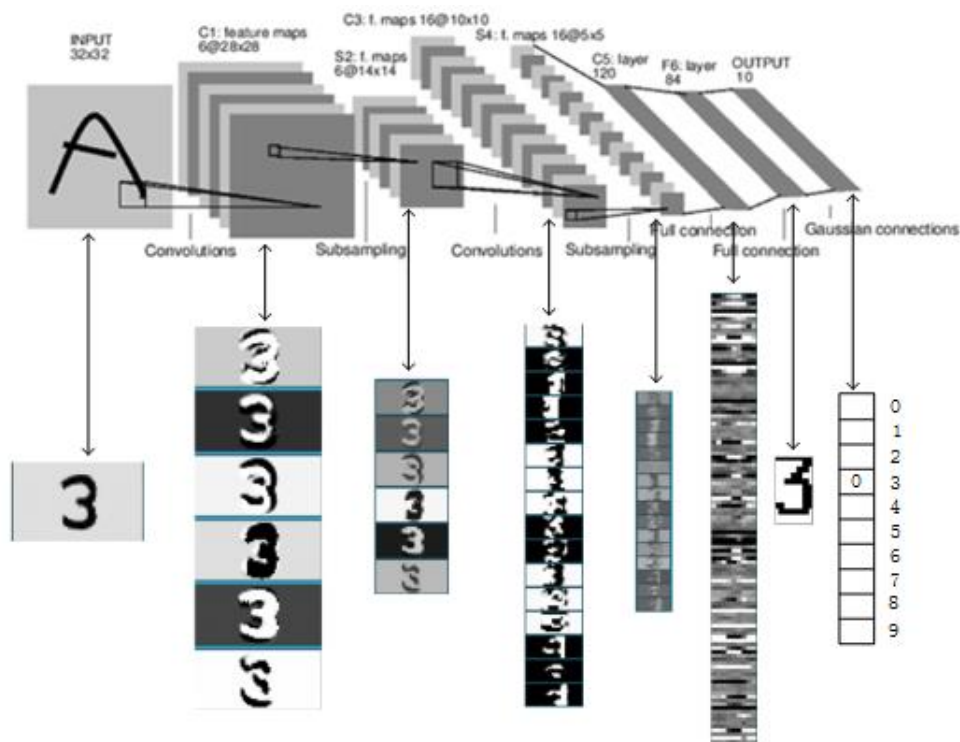


Рис. 2. Архитектура сверточной сети LeNet5

Нейронная сеть была обучена на тренировочной выборке. Точность обученной нейронной сети составляет 94.5%.

4. Метод бэггинга

Для повышения качества распознавания символов предлагается использовать композицию нейронных сетей с архитектурой LeNet5. Одним из способов получения композиции классификаторов является использование бэггинга (bootstrap aggregation). Данный метод заключается в независимом обучении нескольких моделей на подвыборках, полученных из исходных с помощью статистического метода под названием бутстрэп. Этот статистический метод заключается в генерации выборок размера B (так называемых бутстрэп-выборок) из исходного набора данных размера N путем случайного выбора элементов с повторениями в каждом из наблюдений B . Метод бэггинга позволяет получить ансамблевую модель с меньшим разбросом, чем ее отдельные компоненты [4].

Результирующее предсказание классификатора может быть получено одним из следующих способов:

- консенсус: если все элементарные классификаторы присвоили объекту одну и ту же метку, то относим объект к выбранному классу;
- простое большинство: объекту метка того класса, который определило для него большинство элементарных классификаторов;
- взвешивание классификаторов: при голосовании голос классификатора умножается на его вес, отражающий его качество [5].

На практике консенсус достижим очень редко и это приводит к уменьшению точности распознавания, поэтому было решено реализовать методы простого большинства и взвешивания классификаторов. В методе взвешивания классификаторов использовался подход, заключающийся в нахождении результирующего значения классификации на основании точности распознавания сети каждого из классов. еще одним настраиваемым параметром является размер бутстрэп-выборок. Будем выбирать размер бутстрэп-выборки равным от 10% до 90% от тестовой выборки с шагом, равным 10%. Для каждой из бутстрэп-выборок будем находить точность для ансамблей, состоящих из разного количества нейронных сетей (от 1 до 10). На рис. 3 продемонстрированы результаты вычислительного эксперимента.

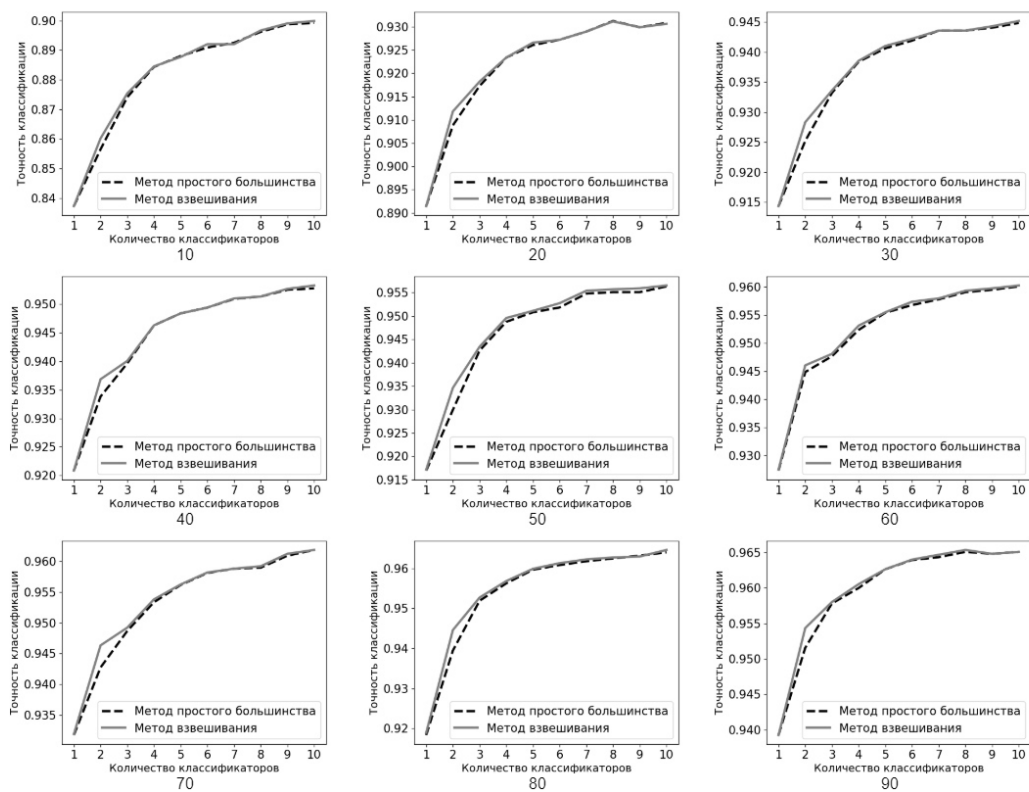


Рис.3. Результаты эксперимента для бэггинга

Каждый график на рис. 3 соответствует вычислительному эксперименту с бутстрэп-выборкой определенного размера (подпись под графиком). По оси абсцисс отложены значения размера ансамбля нейронных сетей, по оси ординат – значения точности распознавания для данного ансамбля. Результирующее значение точности вычислялось как среднее по 100 запускам. Серой линии на графике соответствует метод взвешивания классификаторов, черной пунктирной – метод простого большинства.

Наибольшая точность классификации, равная 96.5%, была достигнута при размере обучающей выборки для классификаторов, равном 90% от количества примеров в обучающей выборке, и количестве классификаторов, равном 8.

5. Метод бустинга

Другим способом получения композиции классификаторов является бустинг. Бустинг заключается в построении композиции алгоритмов, когда каждый последующий алгоритм стремится компенсировать недостатки предыдущих алгоритмов. Одними из самых известных алгоритмов бустинга на сегодняшний день являются градиентный бустинг, который используется в основном для решающих деревьев (и его различные модификации), и AdaBoost. Принцип работы алгоритма AdaBoost заключается в том, что обучающим примерам присваиваются веса, в зависимости от того, корректно или некорректно они классифицировались. Существует два алгоритма для вычисления веса обучающего примера: SAMME и SAMME.R. SAMME.R использует оценки вероятности принадлежности к классам, SAMME использует только факт верной/неверной классификации. Большие веса получают неверно классифицированные примеры. Примеры с большими весами имеют больший шанс попадания в обучающую выборку для следующего классификатора [6].

От количества эпох обучения зависит то, насколько хорошо сеть «запомнила» обучающие примеры, и, следовательно, сколько примеров попадет в обучающую выборку для следующего классификатора. Слишком большое количество эпох обучения может привести к переобучению сети, слишком маленькое – к недообучению. Будем выбирать количество эпох от 10 до 40 с шагом 10. Еще одним оптимизируемым параметром является количество классификаторов. На рис. 4 продемонстрированы результаты вычислительного эксперимента.

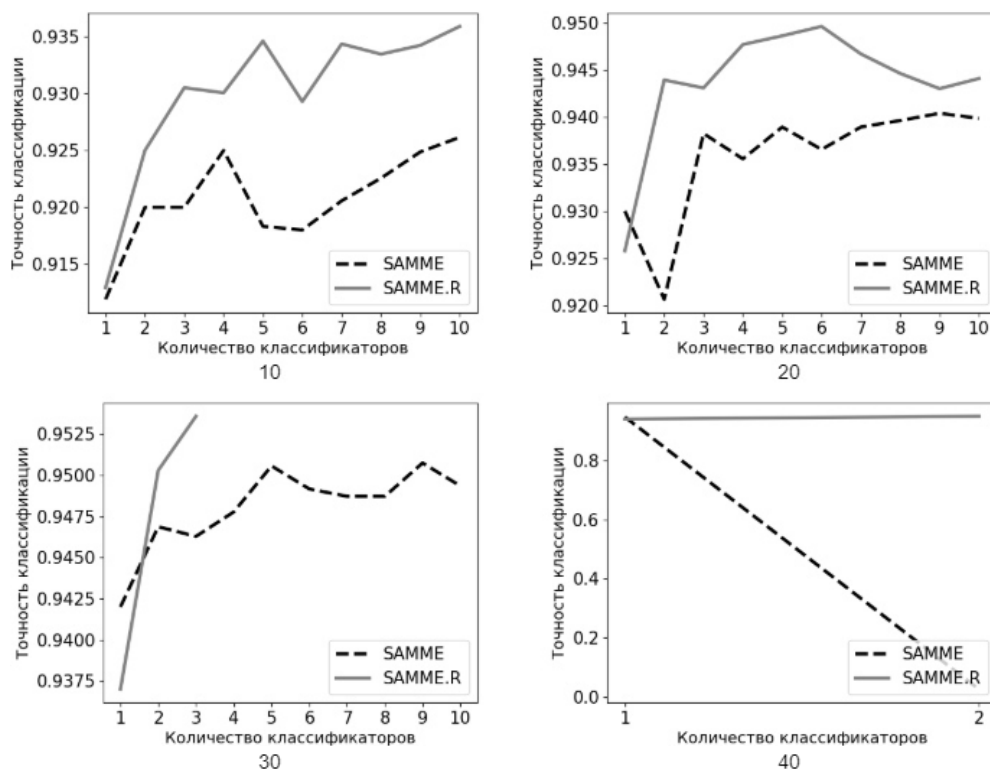


Рис. 4. Результаты эксперимента для бустинга

Каждому графику на рис. 4 соответствует вычислительный эксперимент для определенного количества эпох (подпись под графиком). По оси абсцисс отложены значения размера ансамбля нейронных сетей, по оси ординат – значения точности распознавания для данного ансамбля. Результирующее значение точности вычислялось как среднее по 100 запускам. Серой линии на графике соответствует алгоритм SAMME.R, черной пунктирной – алгоритм SAMME. Алгоритм прекращает работу, когда ошибка на тренировочной выборке становится равной 0, что продемонстрировано на графиках для количества эпох, равных 30 и 40.

Как видно из результатов эксперимента, алгоритм SAMME.R демонстрирует большую точность, чем алгоритм SAMME. Наибольшая точность (95.4%) была достигнута для трех классификаторов в комитете, количества эпох, равных 30, и алгоритма SAMME.R.

Заключение

В результате работы была реализована композиция классификаторов сверточных нейронных сетей архитектуры LeNet5 на основании методов бэггинга и бустинга. Работа композиции оценивалась на выборке рукописных символов, полученных с бланков ЕГЭ. В результате работы были сделаны следующие выводы:

1. Метод бустинга позволил достичь максимальной точности классификации 95.4% на основе композиции из 3-х классификаторов сверточных нейронных сетей архитектуры LeNet5.
2. Алгоритм SAMME.R демонстрирует большую точность, чем алгоритм SAMME.
3. Метод бэггинга позволил достичь максимальной точности классификации 96.5% на основе композиции из 8-ми классификаторов сверточных нейронных сетей архитектуры LeNet5.
4. Предложенная композиция увеличила точность распознавания на 2% по сравнению с одним классификатором LeNet5.
5. Вместе с размером бутстрэп-выборки растет значение точности распознавания ансамблей.
6. Метод взвешенных классификаторов демонстрирует незначительно большую точность для ансамблей, состоящих из двух классификаторов.

ЛИТЕРАТУРА

1. Сверточная нейронная сеть, часть 1: структура, топология, функции активации и обучающее множество // [Электронный ресурс]. URL: <https://habr.com/ru/post/348000/> (Дата обращения: 14.06.2021).
2. The Train, Validation, Test Split and Why You Need It. // [Электронный ресурс]. URL: <https://blog.roboflow.com/train-test-split/> (Дата обращения: 14.06.2021).
3. Understanding and Implementing LeNet-5 CNN Architecture (Deep Learning) // [Электронный ресурс]. URL: <https://towardsdatascience.com/understanding-and-implementing-lenet-5-cnn-architecture-deep-learning-a2d531ebc342> (Дата обращения: 14.06.2021).
4. Ансамблевые методы: бэггинг, бустинг и стекинг // [Электронный ресурс]. URL: <https://neurohive.io/ru/osnovy-data-science/ansamblevye-metody-begging-busting-i-steking/> (Дата обращения: 14.06.2021).
5. Бэггинг // [Электронный ресурс]. URL: <http://www.machinelearning.ru/wiki/index.php?title=Бэггинг> (Дата обращения: 14.06.2021).
6. Multi-class AdaBoost // [Электронный ресурс]. URL: <https://web.stanford.edu/~hastie/Papers/samme.pdf> (Дата обращения: 14.06.2021).

НЕЧЁТКИЙ АНАЛИЗ ФОРМАЛЬНЫХ ПОНЯТИЙ: ОБЗОР ПОДХОДОВ НА ПРИМЕРЕ

Гольшев В.К.

Сибирский федеральный университет
valeriygolyshev@mail.ru

Введение

Классический анализ формальных понятий работает с бинарными контекстами в виде $(0,1)$ -матриц. На практике при измерении свойств объектов возникают ситуации, при которых оценка истинности утверждения «объект g обладает признаком m » является расплывчатой. Для оценивания таких суждений привлекают математический аппарат различных теорий неопределённости, таких, как теория нечётких множеств, грубых множеств, теории вероятностей или их комбинаций [1,2]. Внедрение нечёткой логики [3,4,5] в классический анализ формальных понятий привело к возникновению и активному развитию нового направления интеллектуального анализа данных – нечёткий анализ формальных понятий (сокр. НАФП).

1. Нечёткий анализ формальных понятий

В нечётком анализе формальных понятий можно выделить восемь основных подходов [1,2], различающихся по

- структуре степеней истинности и сопряжённым операциям,
- способам определения нечёткого формального контекста,
- способам обобщения соответствий Галуа,
- способам определения нечёткого формального понятия.

Сравнение подходов по выделенным различиям представлено в табл. 1 и 2 (значок волны над множеством обозначает его нечёткость).

Таблица 1

Сравнение подходов по структурам степеней истинности и способу задания нечёткого контекста

№	Подход	Автор, год	Контекст	Структура степеней истинности
1	Решётка нечётких понятий	Burusco и Fuentes-Gonzalez, 1994 Pollandt, 1997 Belohlavek, 1998	(G, M, \tilde{I})	Полная резидуальная решётка
2	Решётка нечётких понятий с некоммутативной конъюнкцией	Georgescu 2002, 2003	(G, M, \tilde{I})	Обобщённая резидуальная решётка
3	Односторонняя решётка понятий	Yahia и Jaoua, 2001, Krajci, 2003	(G, M, \tilde{I})	Частично упорядоченное множество, полная решётка, полная резидуальная решётка
4	Решётка чётко сгенерированных понятий	Belohlavek, 2005	(G, M, \tilde{I})	Полная резидуальная решётка
5	Обобщённая решётка понятий	Krajci, 2005	$(G, M, \tilde{I}, \otimes)$	Две полные решётки для объектов и признаков, частично упорядоченное множество степеней истинности с конъюнкцией \otimes
6	Мультисопряжённая решётка	Medina, 2007, 2009	$(G, M, \tilde{I}, \sigma)$	Мультисопряжённый фрейм
7	Решётка с порогами	Elloumi, 2004, Zhang, 2007	(G, M, \tilde{I})	Полная резидуальная решётка
8	Решётка с замыканиями	Belohlavek и Vychodil, 2005, 2007, 2012	(G, M, \tilde{I})	Полная резидуальная решётка

Сравнение подходов по способу обобщения соответствий Галуа
и способу определения нечёткого формального понятия

№	Подход	Стрелочные операторы (соответствия Галуа)	Нечёткое формальное понятие
1	Нечёткая решётка понятий	$\tilde{A}^\uparrow(m) = \bigwedge_{g \in G} (\tilde{A}(g) \rightarrow \tilde{I}(g, m)),$ $\tilde{B}^\downarrow(g) = \bigwedge_{m \in M} (\tilde{B}(m) \rightarrow \tilde{I}(g, m))$	$(\tilde{A}, \tilde{B}):$ $\tilde{A}^\uparrow = \tilde{B},$ $\tilde{B}^\downarrow = \tilde{A}$
2	Решётка нечётких понятий с некоммутативной конъюнкцией	$\tilde{A}^\uparrow(y) = \bigwedge_{g \in G} (\tilde{A}(g) \rightarrow \tilde{I}(g, m)),$ $\tilde{A}^\uparrow(y) = \bigwedge_{g \in G} (\tilde{A}(g) \Rightarrow \tilde{I}(g, m)),$ $\tilde{B}^\downarrow(y) = \bigwedge_{m \in M} (\tilde{B}(m) \rightarrow \tilde{I}(g, m)),$ $\tilde{B}^\downarrow(y) = \bigwedge_{m \in M} (\tilde{B}(m) \Rightarrow \tilde{I}(g, m))$	$(\tilde{A}, \tilde{B}_1, \tilde{B}_2):$ $\tilde{A}^\uparrow = \tilde{B}_1,$ $\tilde{A}^\uparrow = \tilde{B}_2,$ $\tilde{B}_1^\downarrow = \tilde{A},$ $\tilde{B}_2^\downarrow = \tilde{A}$
3	Односторонняя решётка понятий	$f(A)(m) = \bigwedge_{g \in A} \tilde{I}(g, m),$ $h(\tilde{B}) = \{g \in G \mid \forall m \in M : \tilde{B}(m) \leq \tilde{I}(g, m)\}$	$(A, \tilde{B}):$ $f(A) = \tilde{B},$ $h(\tilde{B}) = A$
4	Решётка чётко сгенерированных понятий	$\tilde{A}^\uparrow(m) = \bigwedge_{g \in G} (\tilde{A}(g) \rightarrow \tilde{I}(g, m)),$ $\tilde{B}^\downarrow(g) = \bigwedge_{m \in M} (\tilde{B}(m) \rightarrow \tilde{I}(g, m)).$	$(\tilde{A}, B):$ $\exists B_c \subseteq M$ такое, что $\tilde{A} = B_c^\downarrow$ и $B = B_c^\uparrow$
5	Обобщённая решётка понятий	$\tilde{A}^\uparrow(m) = \sup\{b \in L_M \mid \forall g \in G : \tilde{A}(g) \otimes b \leq \tilde{I}(g, m)\},$ $\tilde{B}^\downarrow(g) = \sup\{a \in L_G \mid \forall m \in M : a \otimes \tilde{B}(m) \leq \tilde{I}(g, m)\}$	$(\tilde{A}, \tilde{B}):$ $\tilde{A}^\uparrow = \tilde{B},$ $\tilde{B}^\downarrow = \tilde{A}$
6	Мультисопряжённая решётка	$\tilde{A}^{\uparrow\sigma}(g) = \inf\{\tilde{I}(g, m) \square^{\sigma(m)} \tilde{A}(m) \mid m \in M\},$ $\tilde{B}^{\downarrow\sigma}(m) = \inf\{\tilde{I}(g, m) \square_{\sigma(g)} \tilde{B}(g) \mid g \in G\}$	$(\tilde{A}, \tilde{B}):$ $\tilde{A}^{\uparrow\sigma} = \tilde{B},$ $\tilde{B}^{\downarrow\sigma} = \tilde{A}$
7	Решётка с порогами	$C^* = \{m \in M \mid \inf_{g \in G} (C(g) \rightarrow \tilde{I}(g, m)) \geq \delta\},$ $D^* = \{g \in G \mid \inf_{m \in M} (D(m) \rightarrow \tilde{I}(g, m)) \geq \delta\}$	$(C, D):$ $C^* = D,$ $D^* = C$
		$C^\square(m) = (\delta \rightarrow \inf_{g \in C} \tilde{I}(g, m)),$ $\tilde{B}^\square = \{g \in G \mid \inf_{m \in M} (\tilde{B}(m) \rightarrow \tilde{I}(g, m)) \geq \delta\}$	$(C, \tilde{B}):$ $C^\square = \tilde{B},$ $\tilde{B}^\square = C$
		$\tilde{A}^\square = \{m \in M \mid \inf_{g \in G} (\tilde{A}(g) \rightarrow \tilde{I}(g, m)) \geq \delta\},$ $D^\square(g) = (\delta \rightarrow \inf_{m \in D} \tilde{I}(g, m))$	$(\tilde{A}, D):$ $\tilde{A}^\square = D,$ $D^\square = \tilde{A}$
8	Решётка с замыканиями	$\tilde{A}^\uparrow(m) = \inf_{g \in G} (\tilde{A}(g)^* \rightarrow \tilde{I}(g, m)),$ $\tilde{B}^\downarrow(g) = \inf_{m \in M} (\tilde{B}(m)^* \rightarrow \tilde{I}(g, m))$	$(\tilde{A}, \tilde{B}):$ $\tilde{A}^\uparrow = \tilde{B},$ $\tilde{B}^\downarrow = \tilde{A}$

Основополагающим подходом стал подход №1, положивший начало активному развитию НАФП [6,7]. Подход №2 имеет место в ситуациях, когда важен порядок между членами конъюнкции, т.к. опускает требование её коммутативности [8]. Односторонние решётки [9,10] содержат нечёткие понятия, состоящие из объёмов, представленных чёткими множествами, и из содержаний в виде нечётких множеств. Подход №4 использует определение чётко сгенерированного понятия [11]. Полученная решётка изоморфна односторонней решётке с нечёткими объемами и чёткими содержаниями.

Обобщённые решётки понятий [12,13] включают в себя некоторые другие подходы (подход №1, №3, №8). Мультисопряжённые решётки используют так называемые бирезидуальные структуры [14,15] и включают в себя подход №1, а также обобщённые решётки понятий. Решётки с порогами [16,17,18] используют параметр, являющийся некоторой степенью истинности и использующийся для контроля числа извлекаемых понятий. Решётки с замыканиями [2,5,19] используют функцию истинности унарной логической связки «очень», также позволяющую контролировать число получаемых нечётких понятий. Включает в себя подход №7 с порогами.

2. Демонстрация работы соответствий Галуа в различных подходах на примере

Для примера был взят нечёткий формальный контекст с тремя объектами $\{1,2,3\}$ и тремя признаками $\{a,b,c\}$ (табл. 3).

Таблица 3

Пример нечёткого формального контекста

	a	b	c
1	1	0	0.3
2	0	0.8	0.5
3	0.6	0.1	0.5

В качестве сопряжённых операций рассматривались нечёткая конъюнкция и импликация Лукасевича:

$$a \otimes b = \max\{0, a + b - 1\}, \quad a \rightarrow b = \min\{1, 1 - a + b\}.$$

Для множества объектов были взяты нечёткое множество $\{2|_{0.5}, 3|_{0.1}\}$ и множество $\{2,3\}$ в зависимости от требований подхода. С помощью обобщений Галуа находились множества признаков (табл. 4, столбец 4), после чего данные множества отображались обратно во множества объектов (табл. 4, столбец 5). В столбце 6 табл. 4 указывается, составляет ли полученная пара множеств объектов и признаков нечёткое формальное понятие в смысле рассматриваемого подхода.

Таблица 4

Примеры обобщённых соответствий Галуа в различных подходах НАФП

№	Подход	\tilde{A}	$\tilde{B} = \tilde{A}^\uparrow$	$\tilde{B}^\downarrow = \tilde{A}^{\uparrow\downarrow}$	(\tilde{A}, \tilde{B}) – нечёткое понятие
1	Решётка нечётких понятий	$\{2 _{0.5}, 3 _{0.1}\}$	$\{a _{0.5}, b, c\}$	$\{2 _{0.5}, 3 _{0.1}\}$	да
2	Решётка нечётких понятий с некоммутативной конъюнкцией	$\{2 _{0.5}, 3 _{0.1}\}$	$\tilde{B}_1 = \tilde{A}^\uparrow = \{a _{0.5}, b, c\},$ $\tilde{B}_2 = \tilde{A}^\uparrow = \{a _{0.5}, b, c\}$	$\tilde{B}_1^\downarrow = \{2 _{0.5}, 3 _{0.1}\},$ $\tilde{B}_2^\downarrow = \{2 _{0.5}, 3 _{0.1}\}$	да
3	Односторонняя решётка понятий	$\{2,3\}$	$\{b _{0.1}, c _{0.5}\}$	$\{2,3\}$	да
4	Решётка чётко сгенерированных понятий	$\{2 _{0.5}, 3 _{0.1}\}$	$\{a _{0.5}, b, c\}$	$\{2 _{0.5}, 3 _{0.1}\}$	нет
5	Обобщённая решётка понятий	$\{2 _{0.5}, 3 _{0.1}\}$	$\{b, c\}$	$\{2 _{0.5}, 3 _{0.1}\}$	да
6	Мультисопряжённая решётка понятий	$\{2 _{0.5}, 3 _{0.1}\}$	$\{b, c\}$	$\{2 _{0.5}\}$	нет
7	Решётка с порогами, $\delta = 0.3$	$\{2,3\}$	$D = C^* = \{c\}$	$D^* = \{1,2,3\}$	нет
		$\{2,3\}$	$\tilde{B} = C^\square = \{a _{0.7}, b _{0.8}, c\}$	$\tilde{B}^\square = \{2,3\}$	да
		$\{2 _{0.5}, 3 _{0.1}\}$	$D = \tilde{A}^\square = \{a, b, c\}$	$D^\square = \{1 _{0.7}, 2 _{0.7}, 3 _{0.8}\}$	нет
8	Решётка с замыканиями	$\{2 _{0.5}, 3 _{0.1}\}$	идентичность $\tilde{B} = \tilde{A}^\uparrow = \{a _{0.5}, b, c\}$	идентичность $\tilde{B}^\downarrow = \{2 _{0.5}, 3 _{0.1}\}$	да
		$\{2 _{0.5}, 3 _{0.1}\}$	глобализация $\tilde{B} = \tilde{A}^\uparrow = \{a, b, c\}$	глобализация $\tilde{B}^\downarrow = \{3 _{0.1}\}$	нет

Продemonстрируем процесс нахождения обобщённых соответствий Галуа в подходе №3, предполагая, что множество объектов $A = \{2, 3\}$ – чёткое (табл. 4, подход №3). Результат отображения $f(A)$ есть нечёткое множество объектов $\tilde{B} = f(A) = \{2|_{0.1}, 3|_{0.5}\}$. Действительно,

$$\begin{aligned} f(A)(a) &= \bigwedge_{g \in A} \tilde{I}(g, a) = \tilde{I}(2, a) \wedge \tilde{I}(3, a) = 0 \wedge 0.6 = 0, \\ f(A)(b) &= \bigwedge_{g \in A} \tilde{I}(g, b) = \tilde{I}(2, b) \wedge \tilde{I}(3, b) = 0.8 \wedge 0.1 = 0.1, \\ f(A)(c) &= \bigwedge_{g \in A} \tilde{I}(g, c) = \tilde{I}(2, c) \wedge \tilde{I}(3, c) = 0.5 \wedge 0.5 = 0.5. \end{aligned}$$

Далее вычислим $h(\tilde{B})$:

$$\begin{aligned} g = 1: & \tilde{B}(a) \leq \tilde{I}(1, a), \quad 0 \leq 1, \\ & \tilde{B}(b) \leq \tilde{I}(1, b), \quad 0.1 > 0, \\ g = 2: & \tilde{B}(a) \leq \tilde{I}(2, a), \quad 0 \leq 0, \\ & \tilde{B}(b) \leq \tilde{I}(2, b), \quad 0.1 \leq 0.8, \\ & \tilde{B}(c) \leq \tilde{I}(2, c), \quad 0.5 \leq 0.5, \\ g = 3: & \tilde{B}(a) \leq \tilde{I}(3, a), \quad 0 \leq 0.6, \\ & \tilde{B}(b) \leq \tilde{I}(3, b), \quad 0.1 \leq 0.1, \\ & \tilde{B}(c) \leq \tilde{I}(3, c), \quad 0.5 \leq 0.5. \end{aligned}$$

Значит, множество $h(\tilde{B}) = \{2, 3\}$. Поскольку $A = h(\tilde{B})$, пара множеств (A, \tilde{B}) является нечётким формальным понятием в смысле подхода №3.

Заключение

В данной работе произведен обзор подходов НАФП, для них приведены сравнительные таблицы. Для демонстрации работы обобщённых соответствий Галуа представлен простой пример на небольшом нечётком контексте, результаты вычислений собраны в таблицу, показывающую различия работы стрелочных операторов. Кроме того, для подходов №1, №8 реализованы алгоритмы, позволяющие находить все нечёткие формальные понятия заданного нечёткого контекста. Однако применение НАФП ограничено в связи с высокой вычислительной сложностью, что говорит об актуальности исследований по разработке эффективных алгоритмов решения задачи нахождения всех нечётких формальных понятий.

ЛИТЕРАТУРА

1. *Poelmans J., Ignatov D.I., Kuznetsov S.O., Dedene G.* Fuzzy and rough formal concept analysis: a survey // International Journal of General Systems. – 2014. – P. 105–134.
2. *Belohlavek R., Vychodil V.* What is a fuzzy concept lattice? // Proc. CLA 2005. – 3rd Int. Conference on Concept Lattices and Their Applications. – 162. – P. 34–45.
3. *Zadeh L.A.* Fuzzy sets // Information and control. – 1965. – 8. – P. 338–353.
4. *Новак В., Перфильева И., Мочкорж И.* Математические принципы нечеткой логики // Пер. с англ.; под ред. Аверкина А.Н. – М.: ФИЗМАТЛИТ, 2006. – 352 с.
5. *Гольшев В.К.* Методы нечёткой логики в анализе формальных понятий // Математическое и программное обеспечение информационных, технических и экономических систем: Материалы VII Международной молодежной научной конференции. – Томск, 2019. – С. 269–276.
6. *Belohlavek R.* Fuzzy Concepts and Conceptual Structures: Induced Similarities // Joint Conf. Inf. Sci. 1998, Durham, NC. – 1998. – V.I. – P. 179–182.
7. *Belohlavek R.* Fuzzy Galois Connections // Mathematical Logic Quarterly – 1995. 45. – P. 497–504.
8. *Georgescu G., Popescu A.* Concept Lattices and Similarity in Non-Commutative Fuzzy Logic // Fund. Inform. – 2002. – 55 (1). – P. 23–54.

9. *Yahia S.B., Jaoua A.* Discovering Knowledge from Fuzzy Concept Lattice // Data Mining and Computational Intelligence. Studies in Fuzziness and Soft Computing. – Physica, Heidelberg. – 2001. – V. 68. – P. 167–190.
10. *Krajci S.* Cluster based efficient generation of fuzzy concepts // Neural Network World. – 2003. – 13. – P. 521–530.
11. *Belohlavek R., Sklenar V., Zaczal J.* Crisply Generated Fuzzy Concepts // B. Ganter et al. eds. ICFCA, LNAI. – 2005. – P. 269–284.
12. *Krajci S.* A generalized concept lattice // Logic J. IGPL. – 2005. – 13(5). – P. 543–550.
13. *Medina J., Ojeda-Aciego M., Ruiz-Calvino J.* Relating Generalized Concept Lattices and Concept Lattices for Non-Commutative Conjunctions // Applied Mathematics Letters. – 2008. – 2. – P. 1296–1300.
14. *Medina J., Ojeda-Aciego M., Ruiz-Calvino J.* On Multi-adjoint Concept Lattices: Definition and Representation Theorem // ICFCA. LNAI. – 2007. – 4390. – P. 197–209.
15. *Medina J., Ojeda-Aciego M., Ruiz-Calvino J.* Formal Concept Analysis via Multi-Adjoint Concept Lattices // Fuzzy Sets and Systems. – 2009. – 160. – P. 130–144.
16. *Elloumi S., Jaam J., Hasnah A., Jaoua A., Nafkha I.* A Multi-Level Conceptual Data Reduction Approach Based on the Lukasiewicz Implication // Information Sciences. – 2004. – 163. – P. 253–262.
17. *Shao M.W., Zhang W.X.* Information Granularity Lattices // Proc. of the Sixth IEEE Int. Conf. on Machine Learning and Cybernetics. – Hong Kong, 2007.
18. *Belohlavek R.* A note on variable threshold concept lattices: Threshold-based operators are reducible to classical concept-forming operators // Information Sciences. – 2007. – 177. – P. 3186–3191.
19. *Belohlavek R., Vychodil V.* Fuzzy concept lattices constrained by hedges // Journal of Advanced Computational Intelligence and Intelligent Informatics. – 2007. – 11(6). – P. 536–545.

DOI: 10.17223/978-5-907442-42-9-2021-6

ПРИМЕНЕНИЕ МЕТОДОВ ГЛУБОКОГО ОБУЧЕНИЯ К ЗАДАЧЕ КЛАССИФИКАЦИИ ВРЕМЕННЫХ РЯДОВ

Дуля И.С., Мурзагулов Д.А.

Томский государственный университет
idulya7@gmail.com, murzagulovdamir@gmail.com

Введение

В эпоху Индустрии 4.0 компании все больше сосредотачиваются на разработке технологических решений и стремятся максимально перестроить существующие бизнес-процессы под новую концепцию, основанную на сборе и анализе данных. Большая часть собираемых данных на предприятиях представлена в виде сигналов (данные, динамически изменяющиеся во времени). Например, электрические импульсы, данные с акселерометров, давление в устройствах и т.д. В целях компьютерной обработки исходные аналоговые сигналы дискретизируют и представляют в виде временных рядов. Далее к данным применяют статистические модели и методы машинного обучения, позволяющие извлечь ценные знания для бизнеса, которые используются в деятельности предприятия и оптимизации производства. В данном контексте классификация временных рядов была и остаётся одной из наиболее популярных задач интеллектуального анализа данных [1].

С ростом числа открытых наборов данных были предложены сотни подходов и алгоритмов решения задачи классификации временных рядов. Можно выделить три основные группы подходов к решению задачи классификации временных рядов. Первый подход самый очевидный: на вход статистической модели подаются значения временного ряда. Здесь существует ограничение на то, что временные ряды должны быть одинаковой длины, кроме того, малейшее смещение может привести к существенному искажению результата классификации. В основе второго подхода лежит использование метрических моделей (например, модель k -ближайших соседей) с «эластичной» метрикой расстояния. Третий подход предполагает переход из временного пространства в пространство признаков. В рамках подхода исследователь самостоятельно подбирает характеристики, полезные в решении конкретной задачи. Ключевая проблема состоит в том, что разные аналитики для одной задачи подберут разные наборы характеристик. Отсюда возникает проблема: невозможно определить является ли предложенный пере-

чень характеристик полным и достаточным для решения задачи. Субъективность данного этапа подталкивает к поиску такого решения, которое позволило бы систематизировать или упростить данную процедуру.

Успех моделей глубокого обучения в области компьютерного зрения послужил мотивацией к разработке подходов к классификации временных рядов на основе глубоких сверточных сетей [2]. В рамках недавнего крупного исследования [3] было оценено 18 алгоритмов классификации временных рядов на 85 наборах данных, ни один из которых не был моделью глубокого обучения. Это показывает, что научное сообщество не стремится активно изучать эффективность моделей глубокого обучения для решения задачи классификации временных рядов [4].

Преимущество применения методов глубокого обучения состоит в том, что оно позволяет справиться с несистемностью этапа подготовки данных в задаче классификации временных рядов [5]. Подходы на основе глубокого обучения применяются в связке с обратимыми преобразованиями, которые позволяют представить временной ряд в виде матрицы (например, оконное преобразование Фурье или вейвлет-преобразование). Следует заметить, что в результате таких преобразований информация практически не теряется, т.к. существует обратное преобразование, с помощью которого можно восстановить исходный временной ряд. Далее глубокие сверточные нейронные сети извлекают латентные признаки, на основе которых производится классификация с помощью полносвязной сети прямого распространения.

1. Классификация на основе непрерывного вейвлет-преобразования

Непрерывное вейвлет-преобразование (Continuous Wavelet Transform, CWT) – интегральное преобразование, которое представляет собой свертку вейвлет-функции с сигналом [6]. Вейвлет-преобразование переводит сигнал из временного представления в частотно-временное. Непрерывное вейвлет-преобразование можно описать уравнением

$$\text{cwt}(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} x(t) \psi\left(\frac{t - \tau}{s}\right) dt,$$

где τ – сдвиг, s – масштаб, $\psi(\cdot)$ – вейвлет. Описанное преобразование основано на концепциях масштабирования и смещения. Масштабирование представляет собой растяжение или сжатие вейвлета на коэффициент масштабирования. Сдвиг представляет перемещение вейвлетов с различным масштабом от начала до конца сигнала.

В качестве основного набора данных для исследования эффективности применения непрерывного вейвлет-преобразования для извлечения признаков из временных рядов в задаче классификации временных рядов было решено взять популярный датасет Human Activity Recognition (HAR). Выбранный набор данных содержит данные с девяти акселерометров/гироскопов. Каждому набору из девяти временных рядов (данные с 9 датчиков) сопоставлены активности носителя устройства, считывающего движения.

Для решения задачи классификации временных рядов на основе вейвлет-преобразования необходимо решить две задачи:

1. Преобразовать исходные сигналы в матрицу коэффициентов вейвлет-преобразования;
2. Привести полученную матрицу в квадратную форму.

Для задачи было решено использовать вейвлет Морле с диапазоном масштабов от 1 до 65. Далее полученная матрица коэффициентов размер 64x128 преобразуется в квадратную форму (64x64) по правилу среднего. Этот шаг не является необходимым, но позволяет сэкономить вычислительные ресурсы, кроме того, сверточные нейронные сети принято обучать на квадратных изображениях.

Также необходимо определиться: как подавать сигналы с разных датчиков на вход нейронной сети? Было решено подавать сигнал с каждого отдельного датчика в виде отдельного канала (по аналогии с RGB-каналами цветного изображения). Соответственно, все зависимости между сигналами с различных датчиков могут учитываться одновременно, что очень важно.

В качестве основы свёрточной сети было решено взять архитектуру LeNet-5 и внести в неё несколько улучшений (рис. 1).

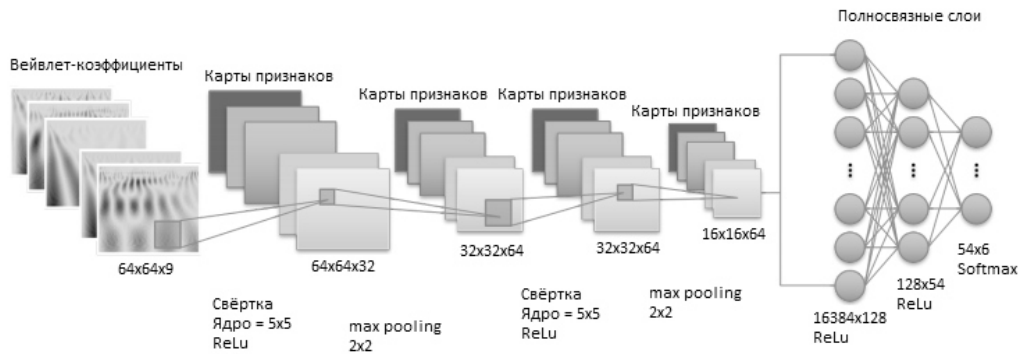


Рис. 1. Свёрточная нейронная сеть, основанная на архитектуре LeNet-5

Было решено использовать max pooling вместо average pooling, что позволит обеспечить более высокую вычислительную эффективность и извлекать признаки, полезные для обнаружения резких изменений в сигналах. Также функция активации гиперболического тангенса (Tanh) была заменена на Rectified linear unit (ReLU) для ускорения обучения и достижения лучшей производительности.

Далее была обучена и протестирована заданная модель. На рис. 2 представлена матрица несоответствий для тестового множества после 20 эпох обучения на тренировочном наборе данных.

Действительные метки	Ходьба	Подъем по ступенькам	Спуск по ступенькам	Сидение	Стояние	Лежание
Ходьба	480	3	0	0	0	0
Подъем по ступенькам	4	458	6	0	2	0
Спуск по ступенькам	12	1	414	0	0	0
Сидение	0	8	0	416	47	0
Стояние	0	0	0	75	483	0
Лежание	0	1	0	0	0	537
Предсказанные метки						

Рис. 2. Матрица несоответствий

Наибольшее число ошибок связано с тем, что модель путает активности «стояние» и «сидение». Обе активности малоподвижные и сигналы с разных датчиков очень сильно похожи друг на друга, с чем и связано большое число ошибок. В целом, модель

качественно осуществляет классификацию активностей по данным с датчиков (акселерометров, гироскопов). Точность, рассчитанная на тестовом множестве, составляет 94.6%. Этого вполне достаточно для практических задач (например, учёт активности и расчёт потраченных калорий). Полученный результат следует рассматривать как доказательство того, что комбинация непрерывного вейвлет-преобразования и свёрточной сети является эффективным методом классификации многомерных нестационарных временных рядов.

2. Сравнение метода с базовыми методами классификации временных рядов

Для подтверждения качества рассмотренного метода необходимо его сравнить с другими моделями/методами классификации временных рядов на наборах данных различной природы. Помимо рассмотренного датасета Human activity recognition были рассмотрены еще четыре набора данных: Waffer [7], FordA [8], StarlightCurves [9] и синтетический набор данных с 10-тью видами нестационарных сигналов.

В качестве базовых методов классификации были взяты три метода: классификация по исходным временным рядам без извлечения признаков, классификация по признаковому представлению временных рядов и классификация на основе метрической модели (метод k-ближайших соседей) с «эластичной» метрикой расстояния DTW. Для первого и второго базовых методов использовалась модель случайного леса в качестве классификатора. В табл. 1 приведены сводные результаты оценки точности моделей на тестовом множестве для различных наборов данных.

Таблица 1

Сводные результаты оценки точности моделей (Accuracy)

	HAR	Waffer	FordA	SLC	Synthetic
WT + CNN	0.924	0.984	0.922	0.951	0.986
Raw TS + RF	0.776	0.957	0.705	0.934	0.869
Feature-based TS + RF	0.858	0.954	0.745	0.978	0.937
KNN + DTW	0.605	0.860	0.500	0.911	0.922

Рассматривая значение метрики Accuracy, можно заметить, что в четырех из пяти датасетов классификация на основе скалограмм вейвлет-преобразования и свёрточной нейронной сети дала лучшие результаты. Только в рамках задачи определения положения небесного объекта (StarLightCurves) основной подход уступил классификации на основе признакового представления. Если упорядочить оставшиеся подходы, то самым мощным окажется классификация на основе признакового представления, далее идёт классификация на основе исходного временного ряда и самые плохие результаты у подхода, основанного на метрической модели с «эластичной» метрикой расстояния.

Следует заметить, что на основании этого эксперимента нельзя утверждать, что для любой задачи классификации временных рядов основной подход будет давать более высокие результаты. Это связано с тем, что предметных областей и задач, в которых применяется анализ временных рядов, крайне обширен, соответственно, временные ряды существенно отличаются между собой по свойствам и типам генеративных процессов. Это исследование лишь показывает, что во многих задачах классификация на основе скалограмм вейвлет-преобразования может дать более качественный результат, чем другие подходы.

3. Тестирование в условиях ограниченного набора обучающих данных

В задачах классификации критическую роль играет объем обучающей выборки. В настоящих прикладных задачах существуют ограничения на её размер, возникающие по различным причинам (стоимость, небольшое число наблюдаемых объектов и т.д.). Чем больше параметров содержит модель, тем больше обучающих наблюдений необходимо, чтобы найти оценки этих параметров. Нейросетевые классификационные мо-

дели содержат достаточно большое количество параметров, для глубоких свёрточных сетей их число может достигать до нескольких миллионов. С этим связано ограничение в применении нейросетевых классификаторов в реальных задачах. Часто предпочтение отдают более простым моделям.

В связи с этим, критически важно показать, что основной метод может работать и в условиях ограниченного объема обучающих данных. Для каждого из пяти наборов данных были созданы обучающие выборки с числом наблюдений от 50 до 1000. Полученные результаты приведены на рис. 3.

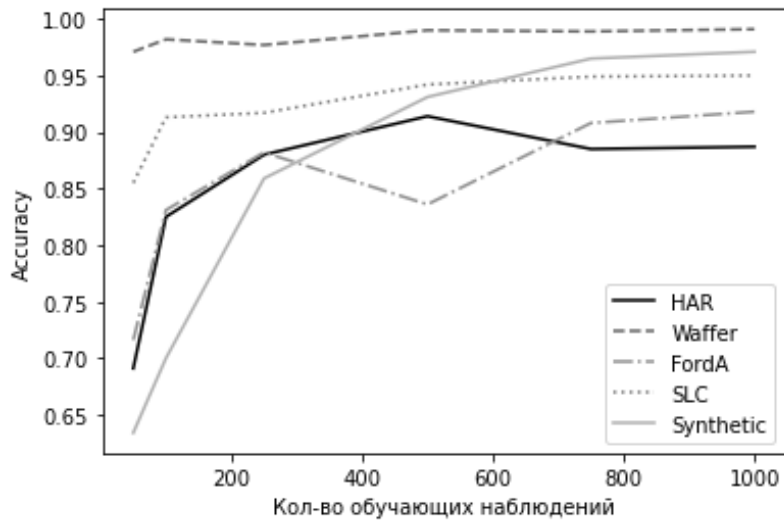


Рис. 3. Зависимость Accuracy от числа обучающих наблюдений

Несмотря на то, что выбранные свёрточные модели имеют большое число параметров, для их обучения требуется сравнительно небольшое число наблюдений. В среднем на различных наборах данных достаточно 250 обучающих наблюдений, для некоторых задач (Waffer) достаточно и 50 размеченных временных рядов для получения требуемой точности. Полученный результат позволяет утверждать, что подход на основе глубокого обучения можно использовать в условиях ограниченного объема данных, когда доступно всего лишь несколько сотен обучающих наблюдений.

Заключение

Появление новых открытых наборов данных, развитие моделей и методов машинного обучения, а также активное внедрение достижений четвёртой научно-технической революции подогревает интерес к задаче классификации временных рядов. Аналитики пытаются найти новые более эффективные и универсальные решения задачи. Прогресс в области глубокого обучения и задачах компьютерного зрения служит мотивацией к разработке подходов к классификации временных рядов на основе свёрточных нейронных сетей.

Подходы на основе глубокого обучения обладают существенным преимуществом перед базовыми подходами – высокой универсальностью и системностью. Если в базовых подходах качество решения существенно зависит от того, насколько удачно аналитик подобрал перечень признаков, характеризующих временные ряды, то подходы на основе глубокого обучения используют единую процедуру подготовки данных. На этапе подготовки используются обратимые преобразования (оконное преобразование Фурье, вейвлет-преобразование и др.), сохраняющие большую часть информации в результате преобразования, т.к. существует обратное преобразование, с помощью которого можно восстановить исходный временной ряд. Далее глубокие свёрточные нейрон-

ные сети извлекают латентные признаки, на основе которых производится классификация с помощью полносвязной сети прямого распространения. Также глубокое обучение позволяет работать с многомерными временными рядами. В сверточных нейронных сетях сигнал с каждого отдельного датчика подается в виде отдельного канала (по аналогии с RGB-каналами цветного изображения). Соответственно, все зависимости между сигналами с различных датчиков могут учитываться одновременно, что очень важно при работе с многомерными временными рядами.

В данной работе было проведено исследование эффективности применения глубоких сверточных сетей в связке с непрерывным вейвлет-преобразованием для решения задачи классификации временных рядов. Был описан алгоритм подготовки данных и обучения моделей на примере задачи распознавания человеческой активности. Было проведено сравнение подхода с базовыми алгоритмами решения задач классификации временных рядов. Полученные результаты показывают, что глубокое обучение может обеспечить существенный прирост точности для задач из разных предметных областей по сравнению с базовыми алгоритмами решения задачи классификации временных рядов (увеличение точности в четырех задачах из пяти). В результате исследования было обнаружено, что для обучения модели требуется относительно немного обучающих наблюдений для получения необходимой точности. На рассмотренных наборах данных требуемое число обучающих наблюдений варьируется от 50 до 250. Можно предполагать, что для обучения модели с несколькими миллионам параметров потребуются значительно больше обучающих наблюдений. Соответственно, подход, основанный на глубоком обучении и непрерывном вейвлет-преобразовании, легко можно применять в реальных задачах, когда существенно ограничен размер обучающей выборки.

В дальнейшей работе полученные результаты в области классификации временных рядов планируется перенести на задачу обнаружения точек перехода. Под точками перехода понимаются моменты времени, когда изучаемый объект меняет своё состояние. В контексте задачи распознавания человеческой активности это моменты времени, когда человек меняет одну активность на другую. Так, планируется использовать глубокую нейронную сеть для извлечения латентных признаков, на основе которых будет происходить обнаружение точек перехода между состояниями.

ЛИТЕРАТУРА

1. *Yang Q., Wu X.* 10 challenging problems in data mining research // *Information Technology & Decision Making*. – 2006. – № 5. – P. 597–604.
2. *Wang Z., Yan W., Oates T.* Time series classification from scratch with deep neural networks: a strong baseline // *International Joint Conference on Neural Networks*. – 2017. – P. 1578–1585.
3. *Bagnall A., Lines J., Bostrom A., Large J., Keogh E.* The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances // *Data Mining and Knowledge Discovery*. – 2017. – № 31. – P. 606–660.
4. *Lines J., Taylor S., Bagnall A.* HIVE-COTE: The hierarchical vote collective of transformation-based ensembles for time series classification // *ACM Transactions on knowledge discovery from data*. – 2018. – № 12.
5. *Li T., Zhao Z., Sun C., et al.* WaveletKernelNet: an interpretable deep neural network for industrial intelligent diagnosis // *Transactions on systems, man, and cybernetics: systems*. – 2021. – P. 1–11.
6. *Hurley C., Mclean J.* *Wavelet, Analysis and Methods*. / Waltham Abbey: ED-Tech press, 2018. – 280 p.
7. *Olszewski R., Maxilon R., Siewiorek D.* Generalized feature extraction for structural pattern recognition in time-series data / USA: Carnegie mellon university, 2001. – 111 p.
8. Dataset: FordA [Электронный ресурс]: набор данных / Time-series classification repository. – URL: <http://www.timeseriesclassification.com/description.php?Dataset=FordA> (дата обращения 20.04.2021)
9. *Rebbapragada U., Protopapasand P., Brodley C., Alcock C.* Finding anomalous periodic time series // *Machine Learning*. – 2009. – № 74. P. 281–313.

О ДВУХ ЗАДАЧАХ КЛАСТЕРИЗАЦИИ ЗНАКОВОГО ГРАФА

Ибрагимова Э.И.

Сибирский федеральный университет
ibragimovaei@mail.ru

Введение

В настоящее время широкое применение знаковых графов в социологии, психологии, биоинформатике и кластерном анализе данных вызывает растущий интерес к исследованиям математических моделей проблем знакового баланса и поиска эффективных методов решения этих проблем. В связи с широким применением знаковых графов в социологии, психологии, биоинформатике и кластерном анализе актуальны исследования, направленные на разработку алгоритмов решения задач, связанных с ними. Центральное место в теории знаковых графов занимает поиск эффективных методов решения проблемы знакового баланса и связанных с ней задач [1,2].

Знаковым графом называется пара $\Sigma = (G, \sigma)$, где $G = (V, E)$ – неориентированный граф ($n = |V| \geq 2$, $m = |E| \geq 1$), на ребрах которого задана функция знака $\sigma: E \rightarrow \{+, -\}$. Знаковый граф называется k -кластеризуемым (k -сбалансированным), если множество его вершин можно разбить на k попарно непересекающихся непустых кластеров так, что все положительные рёбра находятся внутри кластеров, а отрицательные между кластерами [3,4]. Задача распознавания k -кластеризуемости знакового графа формулируется следующим образом [5].

Условие. Задан знаковый граф $\Sigma = (G, \sigma)$, где $G = (V, E)$ – простой неориентированный граф, $n = |V| \geq 2$, $m = |E| \geq 1$.

Вопрос. При каких k граф $\Sigma = (G, \sigma)$ является k -кластеризуемым?

Т.к. знаковые графы не всегда являются k -кластеризуемыми, возникает задача поиска такого разбиения на k попарно непересекающихся непустых кластеров, чтобы некоторый функционал ошибки кластеризации был минимальным.

1. Задача k -кластеризации знакового графа

Будем говорить, что система множеств $C_k = \{V_1, V_2, \dots, V_k\}$ образует разбиение множества вершин V на k подмножеств, если подмножества непустые ($V_i \neq \emptyset$, $i = \overline{1, k}$), попарно не пересекаются ($V_i \cap V_j = \emptyset$, $i \neq j$), и их объединение равно

$V: V = V_1 \cup V_2 \cup \dots \cup V_k$. Обозначим $\Phi = \bigcup_{k=1}^n \Phi_k$ – множество всех возможных разбиений

V , где Φ_k – множество возможных разбиений V ровно на k подмножеств. Пусть C_k – какое-либо разбиение графа. Под отрицательной ошибкой $N(C_k)$ будем понимать число отрицательных ребер внутри подмножеств, под положительной $P(C_k)$ – число положительных ребер между подмножествами. Критерием качества разбиения будет являться функция суммарной ошибки – выпуклая комбинация положительной и отрицательной ошибок:

$$E_\alpha(C_k) = \alpha N(C_k) + (1 - \alpha) P(C_k), \quad \alpha \in [0, 1]. \quad (1)$$

Тогда задача k -кластеризации знакового графа может быть сформулирована следующим образом [5].

Условие. Задан знаковый граф $\Sigma = (G, \sigma)$, где $G = (V, E)$ – простой неориентированный граф, $n = |V| \geq 2$, $m = |E| \geq 1$.

Вопрос. Для заданного $\alpha \in [0, 1]$ требуется найти такое разбиение множества вершин V знакового графа Σ , для которого суммарная ошибка минимальна.

Решением задачи будет множество кластеров $C^* = \arg \min_{C \in \Phi} E_\alpha(C)$ и $k = |C^*|$ – количество кластеров. Известно, что задача минимизации суммарной ошибки является NP-полной [2].

В работе исследуются два частных случая этой задачи при $\alpha = 0$ и при $\alpha = 1$.

Случай 1. При $\alpha = 0$ ищется такое разбиение C^* , для которого минимизируется положительная ошибка, т.е.

$$C^* = \arg \min_{C \in \Phi} P(C). \quad (2)$$

Очевидно, что положительная ошибка достигает минимума на разбиениях, для которых $P(C^*) = 0$.

Случай 2. При $\alpha = 1$ ищется такое C^* , для которого минимизируется отрицательная ошибка, т.е.:

$$C^* = \arg \min_{C \in \Phi} N(C). \quad (3)$$

Аналогично, предыдущему случаю отрицательная ошибка достигает минимума на разбиении, для которого $N(C^*) = 0$.

2. Алгоритмы k -кластеризации знакового графа

Для решения задач (2) и (3) разработаны алгоритмы *SGClust0* и *SGClust1* соответственно.

Алгоритм *SGClust0* решает задачу минимизации положительной ошибки (случай 1). В ходе работы данного алгоритма для заданного графа Σ с помощью удаления отрицательных ребер строится граф Σ^+ . Искомое разбиение C^* будет состоять из компонент связности Σ^+ . Доказана теорема 1, устанавливающая вычислительную сложность алгоритма *SGClust0*.

Теорема 1. Алгоритм *SGClust0* находит решение задачи k -кластеризации знакового графа при $\alpha = 0$ за время, не превышающее $O(n + m)$.

Схема работы алгоритма *SGClust1*, решающего задачу минимизации отрицательной ошибки ($\alpha = 1$), приведена на рис. 1.

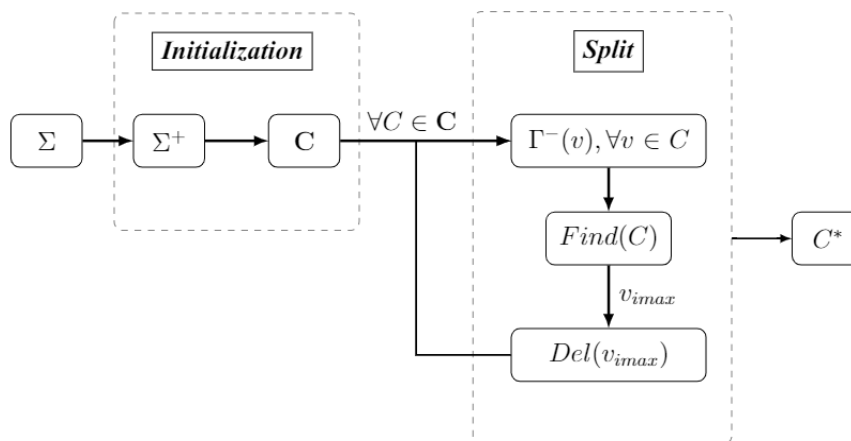


Рис. 1. Схема работы алгоритма *SGClust1*

В схеме используются следующие обозначения: Σ – исходный граф, Σ^+ – граф, полученный из Σ удалением отрицательных ребер, C – множество компонент связности в Σ^+ . Алгоритм на вход получает граф Σ и находит решение в виде множества кластеров C^* . Алгоритм *SGClust1* состоит из двух основных блоков. В блоке *Initialization* выделяются компоненты связности C в Σ^+ . В блоке *Split* формируется множество кластеров C^* . Функция $\Gamma^-(v)$ вычисляется для каждой вершины v из C и определяет количество инцидентных отрицательных ребер вершине v . Функция *Find*(C) находит вершину $v_{i_{\max}}$ с наибольшим количеством инцидентных отрицательных ребер в компоненте c согласно формуле $v_{i_{\max}} = \arg \max_{v \in C} \Gamma^-(v)$. Функция *Del*($c_{i_{\max}}$) выделяет вершину $v_{i_{\max}}$ в отдельный кластер, т.е. $P = P \cup \{c_{i_{\max}}\}$ и удаляет ее из c . Блок *Split* выполняется пока в компоненте c есть отрицательные ребра. Вычислительная сложность алгоритма *SGClust1* установлена в теореме 2.

Теорема 2. Алгоритм *SGClust1* находит решение задачи k -кластеризации знакового графа при $\alpha = 1$ за время, не превышающее $O(n^2 + nm)$.

Пример. На рис. 2–4 приведены примеры работы алгоритмов *SGClust0* и *SGClust1*. На рис. 2 изображен исходный граф Σ . Результат работы алгоритма *SGClust0* приведен на рис. 3. Получены три кластера между которыми нет положительных ребер, но внутри кластера $\{6,7,8,9\}$ есть отрицательное ребро, что является допустимым. На рис. 4 изображено разбиение на 4 кластера, полученное с помощью алгоритма *SGClust1*. Т.к. наличие отрицательных ребер внутри кластера не является допустимым, вершина 7 выделена в отдельный кластер.

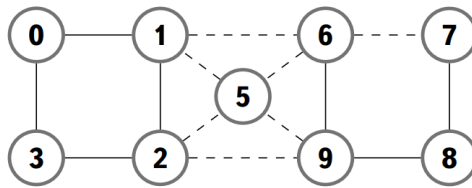


Рис. 2. Граф Σ

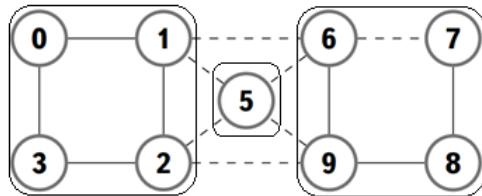


Рис. 3. Результат работы алгоритма *SGClust0*

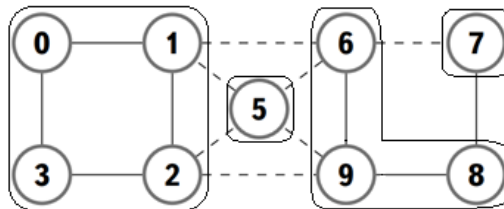


Рис. 4. Результат работы алгоритма *SGClust1*

3. Вычислительные эксперименты

Для оценки эффективности алгоритмов были проведены вычислительные эксперименты, результаты которых приведены в табл. 1 и 2. Для проверки алгоритма *SGClust0* использовались графы, для которых было известно k , т.е. количество кластеров, для которого граф k -сбалансирован. Алгоритм *SGClust1* проверялся на графах, которые не являются k -кластеризуемым. Эксперименты проводились на ПК с процессором Intel® Core™ i7-10510U CPU @ 1.80ГГц и оперативной памятью объемом 8ГБ.

Таблица 1

Алгоритм *SGClust0*

Число вершин	Число ребер	Сбалансированность	Количество кластеров	Время работы
9	9	1- и 3-сбалансирован	1, 2 или 3	менее 1 мс
25	50	5-сбалансирован	5	менее 1 мс
36	90	несбалансирован	2	1 мс
30	30	1- или 10-сбалансирован	1, 2, ..., 10	1 мс
30	435	10- сбалансирован	10	1 мс
30	435	несбалансирован	10	1 мс

Таблица 2

Алгоритм *SGClust1*

Число вершин	Число ребер	Количество кластеров	Время работы
4	4	2	менее 1 мс
5	10	5	менее 1 мс
6	6	3	менее 1 мс
7	8	3	менее 1 мс
10	45	5	1 мс

Результаты вычислительных экспериментов показывают, что алгоритмы *SGClust0* и *SGClust1* работают корректно.

Заключение

В работе исследуются два частных случая задачи кластеризации знакового графа. Разработаны алгоритмы *SGClust0* для кластеризации в случае минимизации положительной ошибки и *SGClust1* для кластеризации в случае минимизации отрицательной ошибки. Приведены теоремы, показывающие, что разработанные алгоритмы находят решение за полиномиальное время.

Проведенные вычислительные эксперименты подтвердили эффективность разработанных алгоритмов. Перспективным направлением исследований является разработка алгоритма для решения задачи при произвольных $0 < \alpha < 1$.

ЛИТЕРАТУРА

1. *Kartwright D., Harary F.* Structural balance: a generalization of Heider's theory // *The Psychological Review.* – 1956. – V.63. – No.5. – P. 277–293.
2. *Nikhil Bansal, Avrim Blum, Shuchi Chawla* Correlation clustering // *Machine Learning.* – 2004. – V.56. – P. 89–113.
3. *Doreian P., Mrvar A.* Structural balance and partitioning signed graphs // In A. Ferligoj & A. Kramberger (Eds.), *Developments in data analysis (Metodoloskizvezki 12, pp. 195-208).* Ljubljana, Slovenia: FDV.
4. *Zaslavsky T.* Balance and clustering in signed graphs // Unpublished manuscript, Binghamton University, New York, 2010.
5. *Ибрагимова Э.И., Семенова Д.В.* Распознавание k -кластеризуемости знаковых графов // Информационные технологии и математическое моделирование (ИТММ-2020). Материалы XIX Международной конференции имени А.Ф. Терпугова. – 2020. – С. 468–472.

АНАЛИЗ ЗАВИСИМОСТИ ТОЧНОСТИ БИНАРНОЙ КЛАССИФИКАЦИИ ТЕКСТОВ ОТ ПРИМЕНЕНИЯ МЕТА- ФУНКЦИЙ ДЛЯ РАЗЛИЧНЫХ АЛГОРИТМОВ КЛАССИФИКАЦИИ

Павлюченко М.В., Кабанова Т.В.

Томский государственный университет

pavluchenkomaria@gmail.com, tvk@bk.ru

Введение

Классификация текстов с помощью имеющихся в открытом доступе библиотек и статей стала достаточно популярным в своей области инструментом, используемым как в различных исследованиях, так и просто в процессе производства. Однако у пользователей часто отсутствует возможность проверить результаты применяемой классификации, что в свою очередь ведет к отсутствию полного доверия к инструменту. Внедрение мета-пространства над процессом классификации – это шаг к решению данной проблемы путем использования методов различных областей математики для получения новой полезной дискриминационной информации. Целью данного исследования является выявление, экспериментальная проверка и анализ возможных подходов к применению мета-обучения для задач бинарной текстовой классификации.

Мета-пространство – это совокупность результатов преобразования мета-функциями исходных векторизованных текстовых данных. Одним из первых подходов к работе с мета-функциями был кластерный метод. Это попытка создать метакarakterистики на основе расстояния с использованием кластеров, полученных на предыдущем этапе кластеризации, с учетом как размеченных, так и неразмеченных данных [1,2]. Другим подходом к работе с мета-функциями является совокупность метода k ближайших соседей и использования центроидов категорий в качестве инструмента для генерации мета-признаков [3,4].

Существуют различные подходы к применению мета-обучения для задачи автоматической текстовой классификации. В [3,5,6] исследовалась идея создания групп мета-функций «вручную» – на основе анализа данных – с целью извлечения дополнительной информации из распределения расстояний между документами. Каждая группа использует какой-то определенный аспект, к примеру, оценки сходства, распределение классов, энтропию и внутриклассовая сплоченность, наблюдаемые у k-ближайших соседей данного тестового документа. В [7] предлагается использовать генетические алгоритмы, которые сокращают количество мета-признаков, повышая эффективность за счет удаления зашумленных мета-признаков. Генетические алгоритмы предоставляют альтернативу традиционным методам оптимизации за счет использования направленного случайного поиска для поиска оптимальных решений в сложных условиях. В [8–10] предлагаются различные авторские алгоритмы для решения данной задачи.

В рамках данной публикации будут использованы мета-функции, предложенные в [6] Кануто, он с другими исследователями в своей работе проводил обширное исследование по использованию мета-функций на основе метода k-ближайших соседей – метрического алгоритма для автоматической классификации объектов или регрессии.

Задачей данной работы является реализация и анализ различных стратегий использования мета-функций на основе ранее проведенного исследования о применении мета-функций для задач уменьшения количества ошибок в процессе бинарной текстовой классификации.

1. Пути применения мета-функций на практике

Первый вариант – преобразовать мета-функциями векторизованное множество текстов и результат этого преобразования использовать как исходное пространство

ввода для классификатора при дальнейших экспериментах. Второй вариант – формировать с помощью мета-функций системы уравнений, которые будут реализовывать различные условия, полученные путем предварительного анализа и методами оптимизации. Важное отличие второго варианта от первого заключается в том, что на вход классификатору будут поступать оригинальные вектора текстов, а не трансформированные в мета-пространство, как в первом варианте. Рассмотрим данные варианты подробнее.

Реализация идеи использования мета-функций как исходного пространства ввода (первый вариант) будет заключаться в выполнении следующих этапов:

1. Перевод векторного пространства текстов в мета-пространство, представляющее собой конкатенацию числовых векторов мета-функций, определенных для каждого наблюдения и категории. Используемые мета-функции, а также их особенности описаны выше.
2. Разделение полученного векторного пространства на тренировочную и тестовую выборки в заданном соотношении.
3. Обучение бинарного классификатора на тренировочной выборке.
4. Оценка правильности работы классификатора на тестовой выборке.

Подразумевается, что в начале существует некоторая известная выборка текстов, на которой будет происходить обучение классификатора по пунктам, описанным выше. Если поступили новые данные, имеющие заданный в эксперименте формат, т.е. векторизованные указанным выше способом в процессе предобработки, то последовательность действий будет такова:

1. Тексты будут переведены с помощью мета-функций в мета-пространство.
2. Полученное мета-пространство данных будет использовано уже обученным ранее на известной выборке данных классификатором для получения предсказаний класса.

Для идеи использования мета-функций как системы условий исходную выборку векторизованных текстов стоит поделить на две – тренировочную и тестовую.

1. Для тренировочной выборки, в свою очередь, производится разделение на тренировочную и тестовую, на тренировочной происходит обучение классификатора, а для тестовой – формирование прогнозов. Параллельно происходит преобразование данных мета-функциями. Далее для этой выборки строится система условий в виде неравенств. Цель этой системы – отбор текстов, для которых результат классификации лучше согласно заданным метрикам.

2. Для тестовой выборки проводится преобразование данных мета-функциями, на результат этого преобразования накладываются полученные условия. Для всех текстов строится прогноз обученным на шаге 1 классификатором, однако для текстов, которые прошли заданные условия, вводится дополнительный маркер – «метка уверенности».

Процесс формирования системы условий и задания граничных значений описан в [11].

2. Отбор наиболее результативных мета-функций

В предыдущем пункте обозначилась общая идея применения мета-функций для автоматической текстовой классификации. Однако остался один очень важный вопрос: стоит использовать все реализованные мета-функции или все-таки нужно ограничиться выбором некоторых из них. Важной деталью здесь является то, что использование большого числа мета-функций, конечно, может дать максимум информации, однако важно отметить, что это не подразумевает, что полученная информация будет качественной, а также количество затраченных на ее получение ресурсов будет адекватно результатам, которые эта информация может дать. Т.е., нужно найти такую комбинацию мета-функций, которая бы сводила к минимуму влияние нерелевантных и зашумленных данных и максимизировала бы объем информации, с помощью которой можно выделить определенный класс.

Полный перебор

Согласно названию, данный подход позволяет оценить результаты всех возможных комбинаций мета-функций. Применимость данного метода напрямую – для поиска группы мета-функций – не имеет смысла в силу требуемых мощностей для реализации алгоритма, однако его можно использовать как способ вычисления границы Парето, которая представляет собой компромисс между максимизацией эффективности и минимизацией количества функций.

Жадный алгоритм

Использование жадного алгоритма относительно работы с мета-функциями было предложено в [6]. Его стратегия заключается в следующем:

1. Пусть множество M – множество анализируемых мета-функций. Присвоить M множество всех реализованных мета-функций.
2. Найти такое подмножество M_1 множества M , которое бы при удалении из M приносило бы максимальное повышение эффективности. Если с помощью удаления невозможно получить статистически значимое улучшение эффективности, следует остановиться. Иначе перейти к шагу 3.
3. Выполнить вычитание множества M_1 из множества M и перейти к шагу 2.

Построение границы Парето

Для поиска наиболее информативных комбинаций происходит оценивание противоречивых целей. Первая цель – это улучшение результатов классификации, повышение эффективности классификации. Вторая цель касается мета-функций и заключается в уменьшении их количества для уменьшения затрат на производительность и временных затрат на их обработку и реализацию. В частности, происходит анализ оптимальности по Парето, которую можно описать как такое состояние некоторой системы, при котором значение каждого частного показателя, характеризующего систему, не может быть улучшено без ухудшения других.

Идея разработки этого подхода находится в [12]. Введем в рассмотрение график того, как могла бы выглядеть зависимость эффективности классификации (пока безотносительно какой-либо из метрик) от количества используемых мета-функций на рис. 1. Каждая из точек на границе обозначает некоторый набор мета-функций, который невозможно заменить более эффективным набором или же набором с меньшим количеством мета-функций. К примеру, если комбинация мета-функций имеет низкую эффективность, но состоит из малого числа функций, ее нельзя исключить из границы Парето, если нет другой комбинации, которая была бы более эффективной и использовала бы, по крайней мере, такое же количество мета-функций. На графике линией представлена граница Парето, маленькими кругами представлены различные комбинации мета-функций заданного размера (размер – количество используемых мета-функций), размер указывается по оси абсцисс, значение метрики – по оси ординат.



Рис. 1. Пример графика зависимости эффективности по Парето от количества используемых мета-функций

3. Результаты экспериментов

Результаты всех экспериментов представлены в табл. 1 – для использования мета-функций как входа для классификатора и для использования мета-функций для составления систем условий соответственно. Первые три строки в таблицах –полученные значения метрик, вторые три строки –сравнение в процентах значений, полученных при использовании мета-функций, с результатами, полученными без использования мета-функций. Если знака нет, то подразумевается знак "плюс", т.е. улучшение результатов, иначе – знак "минус". При анализе данных таблиц можно сделать следующие выводы:

1. При использовании всех реализуемых мета-функций нашлись метрики, по которым произошло ухудшение результата. Это наблюдение доказывает небезосновательность идеи о внедрении принципов отбора мета-функций.

2. При применении мета-функций как исходного пространства ввода каждый способ отбора находил комбинации, улучшающие исходный результат для всех метрик и всех классификаторов. Самыми часто встречаемыми сочетаниями были сочетания трех мета-функций, для каждого из классификаторов состав тройки был своим.

3. При применении мета-функций для составления условий наблюдалась ситуация, при которой улучшение в качестве согласно метрикам было весьма значительное, однако при этом доля прошедших отбор текстов, на которой мерились значения, становилась в процессе отбора совсем невелика.

Результаты экспериментов

Таблица 1

Метрики	Применение мета-функций как исходного пространства ввода																						
	Без мета-функций				Все				Полный перебор				Жадный алгоритм										
	SVM	LR	TREE		SVM	LR	TREE		SVM	LR	TREE		SVM	LR	TREE		SVM	LR	TREE				
f1	0,3312	0,3071	0,3325		0,3564	0,3705	0,303		0,3813	0,3788	0,3866		0,3778	0,3928	0,3827		0,3928	0,3778	0,3827		0,3928	0,3778	0,3859
precision	0,2377	0,25	0,231		0,2214	0,2508	0,2479		0,2548	0,2583	0,272		0,2551	0,2529	0,2543		0,2636	0,2583	0,2543		0,2636	0,2583	0,2732
recall	0,6548	0,3978	0,5927		0,6599	0,7083	0,3898		0,8508	0,7675	0,8656		0,7675	0,8898	0,8589		0,8337	0,7675	0,8589		0,8337	0,7675	0,8629
f1	Сравнение результатов с результатами "без мета-функций"				7,608696	20,6447	-8,8722		15,12681	23,34744	16,271		14,07005	27,90622	15,09774		18,59903	23,02182	15,09774		18,59903	23,02182	16,06015
precision					-6,85738	0,32	7,31602		7,193942	3,32	17,749		7,320151	1,16	10,08658		10,89609	3,32	10,08658		10,89609	3,32	18,2684
recall					0,778864	78,0543	-34,233		29,9328	92,93615	46,044		17,21136	123,6802	44,91311		27,32132	92,93615	44,91311		27,32132	92,93615	45,58799

Метрики	Применение мета-функций как системы условий																						
	Без мета-функций				Все				Полный перебор				Жадный алгоритм										
	SVM	LR	TREE		SVM	LR	TREE		SVM	LR	TREE		SVM	LR	TREE		SVM	LR	TREE		SVM	LR	TREE
f1	0,3312	0,3071	0,3325		0,1294	0,6	0,9524		0,915	0,6	0,96		0,915	0,6	0,96		0,915	0,6	0,96		0,915	0,6	0,96
precision	0,2377	0,25	0,231		0,0746	0,4286	0,9195		0,875	0,4286	0,931		0,875	0,4286	0,931		0,875	0,4286	0,931		0,875	0,4286	0,931
recall	0,6548	0,3978	0,5927		0,4897	1	0,9877		0,9589	1	0,9908		0,9589	1	0,9908		0,9589	1	0,9908		0,9589	1	0,9908
f1	Сравнение результатов с результатами "без мета-функций"				-60,93	95,3761	186,436		176,2681	95,3761	188,72		176,2681	95,3761	188,7218		176,2681	95,3761	188,7218		176,2681	95,3761	188,7218
precision					-68,6159	71,44	298,052		268,1111	71,44	303,03		268,1111	71,44	303,0303		268,1111	71,44	303,0303		268,1111	71,44	303,0303
recall					-25,2138	151,383	66,6442		46,44166	151,3826	67,167		46,44166	151,3826	67,1672		46,44166	151,3826	67,1672		46,44166	151,3826	67,1672

Заключение

В данной работе было предложено два подхода к использованию мета-обучения: применение мета-функций как исходного пространства ввода для классификатора и применение мета-функций для создания системы условий. В результате экспериментов можно сделать следующий вывод: при поставленной задаче улучшения значений метрик применение мета-функций даст стабильно положительные результаты как пространство для ввода классификатора. К тому же данный процесс не требует дополнительного участия эксперта в процессе работы. Однако если стоит задача проанализировать данные, то мета-пространство лучше использовать для построения системы условий, и как в процессе поиска граничных значений экспертом, так и в процессе применения этой системы условий к данным, можно извлечь новую информацию о данных.

В дополнение к изучению данных подходов в работе был изучен вопрос об оптимальности использования не всех реализуемых мета-функций, а какой-то их подгруппы. Для этого были предложены следующие процедуры: полный перебор, жадный алгоритм и построение границы Парето. Полный перебор очень прост с точки зрения реализации и не требует введения дополнительных программных модулей или библиотек, однако обработка данных может быть достаточно продолжительной по времени. Использование полного перебора позволяет увидеть картину целиком. С помощью жадного алгоритма можно получить однозначный вывод о том, какие мета-функции для какого классификатора дают лучший результат. Это сокращает время на анализ полученных сведений в отличие от метода полного перебора. Благодаря введению порогового значения в процессе исследования можно контролировать переход на следующий уровень алгоритма, т.е. удаление используемых мета-функций. Стоит отметить, что реализация данного алгоритма оказалась не сложнее, чем реализация полного перебора, а благодаря графическому построению границы Парето анализировать исходы, полученные для каждой из комбинаций мета-функций, становится гораздо проще, это сокращает время на формулирование выводов и принятие дальнейших решений, связанных с задачей.

Перспективами данного исследования могут быть эксперименты на новых данных, внедрение дополнительных мета-функций или реализация подходов, указанных в литературном обзоре.

ЛИТЕРАТУРА

1. Kyriakopoulou A., Kalamboukis T. Using clustering to enhance text classification // SIGIR. – 2007. – P. 805–806.
2. Ferra H., Raskutti B., Kowalczyk A. Using unlabelled data for text classification through addition of cluster parameters // ICML. – 2002. – P. 514–521.
3. Gopal S., Yang Y. Multilabel classification with meta-level features // SIGIR. – 2010. – P. 315–322.
4. Jin H., Pang G., Jiang S. Cenkn: a scalable and effective text classifier // DMKD. – 2015. – V. 29. – No. 3. – P. 593–625.
5. Santos W., Canuto S., Marcos G. Efficient and scalable metafeature-based document classification using massively parallel computing // SIGIR. – 2015. – P. 333–342.
6. Gon M., Canuto S., Salles T. On efficient meta-level features for effective text classification // CIKM. – 2014. – P. 1709–1718.
7. Srinivas M., Patnaik L.M. Genetic algorithms: a survey // IEE CS. – 1994. – P. 17–26.
8. Morales E., Madrid J., Escalante H. Meta-learning of Textual Representations // ECML PKDD. – 2019. – P. 57–67.
9. Brazdil P., Ferreira M. Workflow recommendation for text classification with active testing method // ICML. – 2018.
10. Lai K., Lam W. A Meta-Learning Approach for Text Categorization // DBLP. – 2001. – P. 303–309.
11. Павлюченко М.В., Кабанова Т.В. Анализ ошибок бинарного классификатора текстов с применением мета-признаков // Материалы Международной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем». – 2020. – С. 57–66.
12. Gon M., Canuto S., Sousa D. A Thorough Evaluation of Distance-Based Meta-Features for Automated Text Classification // IEEE Transactions on Knowledge and Data Engineering. – 2018. – P. 2242–2256.

III. МАТЕМАТИЧЕСКИЕ МЕТОДЫ И МОДЕЛИ ЦИФРОВОЙ ЭКОНОМИКИ

DOI: 10.17223/978-5-907442-42-9-2021-9

АНАЛИЗ ДИНАМИКИ ПОКАЗАТЕЛЕЙ ОБОРАЧИВАЕМОСТИ ОБОРОТНЫХ СРЕДСТВ ПРОИЗВОДСТВЕННОГО ПРЕДПРИЯТИЯ Г. ТОМСКА

Андриевская А.А., Черняк Э.М.
Томский государственный университет
anna_andriev@mail.ru, eduard.chernyak.00@mail.ru

Введение

Одной из важнейших составляющих деятельности предприятия является повышение качества управления оборотными активами в целях обеспечения непрерывности процесса производства и минимизации вложений в оборотные средства. Рациональное использование оборотного капитала создаёт все необходимые условия для стабильного роста прибыли компании, а также формирует необходимые накопления для последующего процесса производства. В связи с этим возрастает необходимость в оптимизации объёмов вложений в оборотные средства, что приводит к повышению экономических показателей компании.

В данной работе на примере крупного производственного предприятия г. Томска исследуется динамика таких показателей оборачиваемости как коэффициент оборачиваемости и оборот, с помощью регрессионной модели строятся прогнозные значения показателей на следующий период.

1. Оценка эффективности управления оборотными средствами предприятия

Повышение эффективности использования оборотных средств предприятия приобретает все более актуальное значение, т.к. высвобождаемые при этом финансовые ресурсы представляют собой дополнительные внутренние источники будущих инвестиций. Рациональное и эффективное оперирование оборотными средствами благоприятствует увеличению финансовой устойчивости предприятия и его платежеспособности. В этих условиях предприятие вовремя и в полной мере выполняет свои расчетно-платежные обязательства, что позволяет успешно осуществлять коммерческую деятельность и увеличивать прибыль [1,2].

Эффективность использования оборотных средств описывается системой экономических показателей, главным образом – оборачиваемостью оборотных средств. В определение оборачиваемости оборотных средств вкладывается смысл продолжительности полного кругооборота средств с момента приобретения оборотных средств (создание запасов, покупка сырья, материалов и т.д.) до момента реализации готовой продукции. Скорость оборачиваемости оборотных средств исчисляется с помощью следующих взаимосвязанных показателей [3]:

1. *Коэффициент оборачиваемости (КО)*, или скорость оборота, показывает, сколько раз за данный период возвращаются вложенные в оборотные средства деньги:

$$КО = \frac{TR}{Об_{ср}},$$
 где TR – это суммарный объём выручки за выбранный период, а $Об_{ср}$ –

средняя стоимость оборотных средств за выбранный период.

2. *Оборот (O)*, или продолжительность одного оборота в днях, равна частному от деления числа дней за анализируемый период (30, 90, 365) к оборачиваемости оборот-

ных средств: $O = \frac{n \text{ (дни)}}{KO}$. Оборот показывает, на сколько дней в среднем инвестируются финансовые средства в размере Ob_{cp} в оборотные активы предприятия [4–6].

2. Анализ управления оборотными средствами (запасами) ЗАО «Томское Пиво»

На основе реальных данных о выручке и стоимости запасов крупного производственного предприятия ОАО «Томское Пиво» за 2012–2018 гг., представленных предприятием в сети Интернет, были рассчитаны значения KO и O , результаты которого приведены в табл. 1.

Таблица 1

Значения показателей оборачиваемости предприятия А

Год t	KO	O	Выручка, тыс. руб.	Стоимость запасов на начало года, тыс. руб.
2012	4.015	90.915	3 090 989	792 777
2013	4.332	84.259	3 091 435	747 044
2014	5.032	72.538	3 236 202	680 248
2015	7.337	49.745	4 762 381	606 032
2016	7.165	50.941	5 095 643	692 073
2017	6.546	55.760	4 820 754	730 280
2018	7.060	51.702	5 234 816	742 622
2019				740 399

По итогам расчётов можно сделать вывод, что предприятие успешно справляется с задачей улучшения качества управления оборотным капиталом, т.к. очевиден существенный рост коэффициента оборачиваемости и сокращение оборота, при этом основным фактором, влияющим на оборачиваемость, является рост выручки; тем не менее, общее сокращение уровня запасов тоже благоприятно повлияло на результат, особенно в 2015 г.

С помощью коэффициента корреляции Пирсона выявлена линейная связь t с коэффициентом оборачиваемости KO и оборота O : $r_{iKO} = 0.849572$, $r_{iO} = -0.87609$. Заметим, что значения KO и O связаны друг с другом обратной зависимостью, тем не менее, была проанализирована динамика обоих показателей. Методом наименьших квадратов найдены уравнения линейной регрессии, отражающие тенденции роста KO и сокращения O , которые определяются следующим образом: $KO \approx 0.5606t - 1123.6$, $O \approx -7.0083t + 14187$, где t – год.

На рис. 1, 2 приведены графики полученных моделей исходных данных.

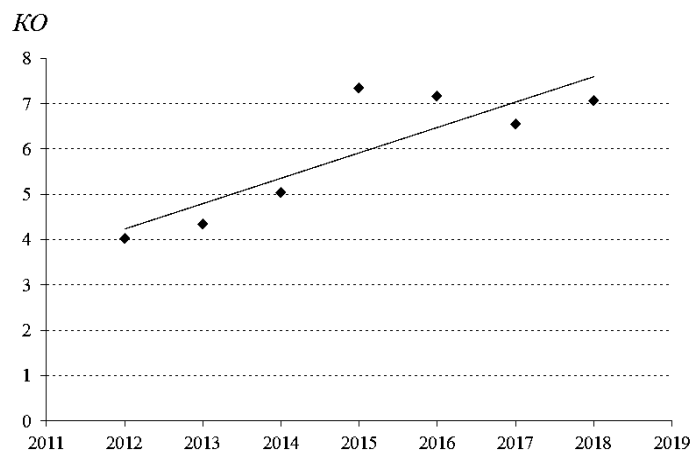


Рис. 1. Зависимость KO от времени

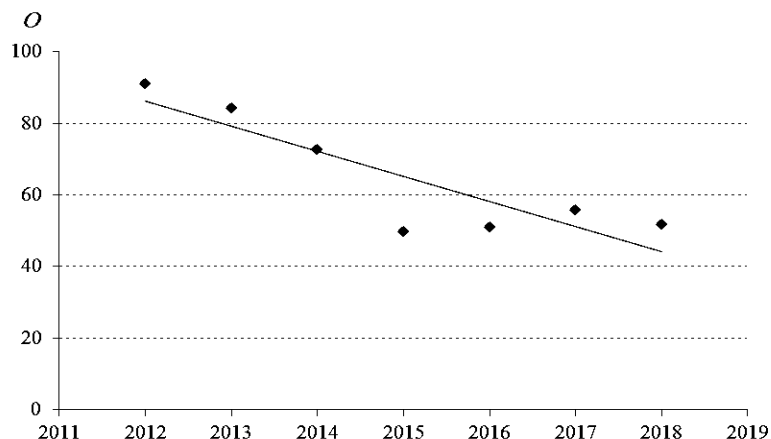


Рис. 2. Зависимость O от времени

Коэффициенты детерминации моделей $R_{KO}^2 = 0.721771916$, $R_O^2 = 0.7675$. При проверке гипотезы о слабости линейной связи критерием Фишера получили, что достигнутые уровни значимости $p\text{-value}_{KO} = 0.015518559$, $p\text{-value}_O = 0.058851$, это позволяет принять альтернативную гипотезу о том, что линейная связь между переменными сильная. Гипотеза о равенстве коэффициентов линейной регрессионной модели нулю отвергнута, т.к. $p\text{-value}_{KO} = 0.016$, $p\text{-value}_O = 0.0097$. Также была рассчитана средняя абсолютная ошибка аппроксимации $A = \frac{1}{N} \sum_{i=1}^N \frac{|\Phi_{AKT_i} - \text{МОДЕЛЬ}_i|}{\Phi_{AKT_i}} \cdot 100\%$, где наблюдаемые значения $\Phi_{AKT_i} > 0$, $i = \overline{1, N}$, $N = 8$, при этом $A_{KO} = 10.45\%$, $A_O = 11.34\%$.

Нормальность остатков моделей проверялась с помощью различных критериев [7]. Значения достигнутых уровней значимости $p\text{-value}$ показаны в табл. 2, на рис. 3, 4 приведены графики на нормальной вероятностной бумаге для остатков моделей. Результаты расчётов позволяют принять гипотезу о нормальном распределении остатков обеих моделей.

Таблица 2

Проверка нормальности остатков различными критериями

Критерий	$p\text{-value}_{KO}$	$p\text{-value}_O$
Колмогорова-Смирнова	0.017	0.115
Крамера-фон Мизеса	0.017	0.277
д'Агостино-Пирсона	0.14	0.37
Шапиро-Уилка	0.019	0.358

Таким образом, полученные модели обладают достаточно высоким качеством и позволяют построить прогноз показателей оборачиваемости на 2019 г.:

$$KO = [8.2514] = 8 \text{ раз/год}, O = [37.2423] = 38 \text{ дней}.$$

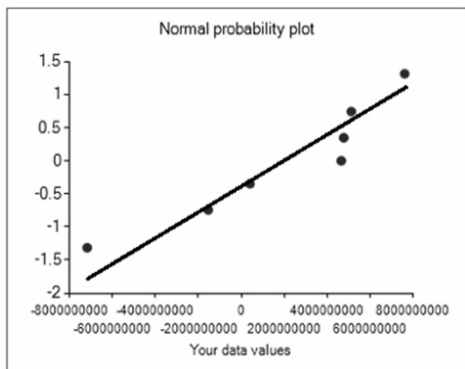


Рис. 4. Проверка нормальности остатков О

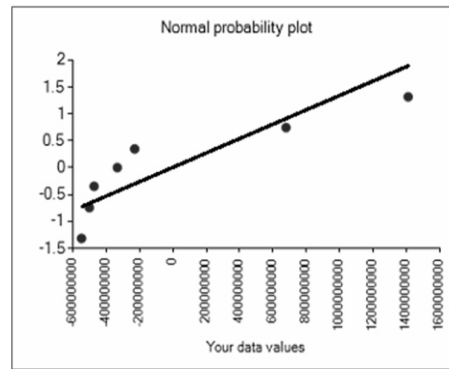


Рис. 5. Проверка нормальности остатков КО

Заключение

В работе на основе реальных данных о выручке и стоимости оборотных активов крупного производственного предприятия за 2012–2018 гг. произведён анализ динамики показателей оборачиваемости оборотных средств, получены линейные модели для коэффициента оборачиваемости и оборота, исследовано качество моделей, получены прогнозные значения на 2019 г. Показано, что предприятие с каждым годом всё более эффективно управляет оборотными средствами, ежегодно наращивая коэффициент оборачиваемости и сокращая оборот – время «замораживания» финансовых средств в оборотные активы.

ЛИТЕРАТУРА

1. *Supriati D., Kananto R., Kusriananda A.* The effects of intellectual disclosures capital, debt to assets ratio, debt equity ratio, company size and assets turnover on company profitability // Proceeding of the 5th Annual International Conference on Accounting Research (AICAR 2018): Advances in Economics, Business and Management Research. – 2019.
2. *Азоев Г. Л. и др.* Управление организацией: учеб. для студентов вузов, обучающихся по специальности. – "Менеджмент организации", Москва, 2007.
3. *Зенкова Ж.Н., Гуров Н.В.* Доверительные интервалы для показателей оборачиваемости оборотных средств предприятия с использованием робастного оценивания // XXI век: итоги прошлого и проблемы настоящего плюс. – 2017. – № 203. – 36-37. – С. 52–57.
4. *Зенкова Ж.Н., Муравлева М.А.* Расчет стоимости оборотных средств предприятия по цензурированным интервалом данным // Логистические системы в глобальной экономике [Электронный ресурс] : материалы Междунар. науч.-практ. конф. (3–4 марта 2014 г., Красноярск). Вып. 1. Электрон. сб. Сиб. гос. аэрокосмич. ун-т. Красноярск, 2014. С. 103–107.
5. *Зенкова Ж.Н., Макеева О.Б.* Применение методов обработки цензурированных данных при анализе оборачиваемости // Вестник науки Казахского агротехнического университета им. С. Сейфуллина. Астана. – 2014. – № 3 (82). – С. 21–30.
6. *Зенкова Ж.Н., Мусони В., Андриевская А.А.* Статистическое оценивание показателей оборачиваемости оборотных средств с учетом дополнительной информации о квантиле // Вестник томского государственного университета. Управление, вычислительная техника и информатика. – № 55. – С. 36-45.
7. *Кобзарь А.И.* Прикладная математическая статистика для инженеров и научных работников. – Москва: ФИЗМАТЛИТ, 2006. – 816 с.

УЛУЧШЕННАЯ ОЦЕНКА ПАРАМЕТРА АВТОРЕГРЕССИОННОГО ПРОЦЕССА С НЕИЗВЕСТНОЙ ДИСПЕРСИЕЙ ШУМОВ

Болдырева К.К.

Томский государственный университет
kristyaboldyreva@yandex.ru

Введение

Чтобы анализировать экономические показатели, необходимы ежегодные, ежедневные данные, которые нужно упорядочить согласно времени их получения, т.е. построить временной ряд. В анализе временных рядов цель состоит в том, чтобы оценить будущую ценность, используя поведение в прошлых данных. В теории временных рядов имеется разнообразие линейных моделей, среди которых выделяются скользящего среднего, авторегрессии и модель скользящего среднего и авторегрессии.

Причина популярности этих моделей кроется, с одной стороны, в их простоте, с другой стороны, в том, что уже с небольшим числом параметров ими можно хорошо аппроксимировать весьма широкий класс стационарных последовательностей. Однако далеко не все временные ряды являются стационарными. Анализ показывает, что часто в статистических данных весьма реально вырисовываются: медленно меняющийся тренд, периодические или непериодические циклы, нерегулярная, флуктуирующая компонента.

1. Постановка задачи

Пусть имеется случайный процесс $\{x_k\}$, который определяется рекуррентным уравнением:

$$x_k = \theta x_{k-1} + \sigma \varepsilon_k, \quad k \geq 1, \quad (1)$$

где ε_k – последовательность независимых одинаково распределенных случайных величин с $E\varepsilon_k = 0$ и $E\varepsilon_k^2 = 1$; параметры θ и σ неизвестны.

Имея два метода оценивания параметра θ [1] и [2], необходимо сравнить качество оценок параметра θ , получаемых алгоритмами оценивания с известной и неизвестной дисперсией для различных распределений шумов.

2. Оценка параметра авторегрессии [1]

Для оценивания θ при неизвестной σ введем для каждого $h > 0$ последовательный план $\tau = \tau(n, h) = \inf \left\{ m \geq n + 1 : \sum_{k=n+1}^m x_{k-1}^2 \geq h A_n R_n \right\}$. Оценка параметра θ находится следующим образом:

$$\theta^*(h, n) = \frac{1}{\tilde{h} A_n R_n} \sum_{k=n+1}^{\tau} \sqrt{\alpha_k} x_{k-1} x_k. \quad (2)$$

Здесь R_n – статистика, заменяющая неизвестную дисперсию σ^2 . Положим $R_n = \sum_{j=1}^n x_{j-1}^2$; $\alpha_k = 1$, если $n < k < \tau$ и $\alpha_k = \beta(h)$; множитель $0 < \beta(h) \leq 1$ находится из

уравнения $\sum_{k=n+1}^{\tau-1} x_{k-1}^2 + \beta(h)x_{\tau-1}^2 = hA_n R_n$; множитель из (2) определяется из

$$\tilde{h}A_n R_n = \sum_{k=n+1}^{\tau} \sqrt{\alpha_k} x_{k-1}^2 \text{ При этом } A_n = \frac{1}{2} \left(\frac{\Gamma\left(\frac{n-1}{2} - p\right)}{\Gamma\left(\frac{n-1}{2}\right)} \right)^{1/p}, \quad n > 2p + 1.$$

3. Улучшенная оценка параметра авторегрессии [2]

Предлагается трехэтапная процедура оценки параметра в модели (1). На первом этапе оцениваем параметр авторегрессии методом наименьших квадратов. На втором этапе строим специальный множитель для компенсации неизвестной дисперсии шума. На третьем этапе оценим параметр авторегрессии.

Чтобы компенсировать неизвестную дисперсию шума, нам понадобится пилотная оценка неизвестного параметра авторегрессии θ . Мы используем обычную оценку

наименьших квадратов:
$$\hat{\theta}_l = \frac{\sum_{k=1}^l x_{k-1} x_k}{\sum_{k=1}^{l-1} x_{k-1}^2}.$$

Вводится величина, являющейся оценкой дисперсии шума, которую будем называть компенсирующим коэффициентом, имеющий следующий

вид: $\Gamma_l = C_f(l) \sum_{k=l+1}^{l+l} (x_k - \hat{\theta} x_{k-1})^2$, где коэффициент $C_f(l)$ определяется функцией плот-

ности шума: $C_f(l) = E \left(\sum_{k=1}^l \varepsilon_k^2 \right)^{-1}$.

Пусть шум имеет двойное экспоненциальное распределение с параметром $\lambda = \sqrt{2}$. Далее с помощью численного моделирования найдется $C_f(l)$. В случае, если шум гауссовский, то $C_f(l) = 1/l - 2$.

На третьем этапе мы строим оценку параметра θ . Для каждого $H > 0$ введем останавливающий случай $\tau = \tau(H) = \inf \left\{ n \geq 1: \sum_{k=l+l+1}^n \frac{x_{k-1}^2}{\Gamma_l} \geq H \right\}$, и определим последовательную оценку по следующей формуле:

$$\theta^*(l, H) = \frac{1}{\tilde{H}} \sum_{k=l+l+1}^{\tau} \sqrt{\beta_k} \frac{x_{k-1} x_k}{\Gamma_l}, \quad (3)$$

где $\beta_k = 1$, если $k < \tau$, и $\beta_\tau = \alpha_\tau$ – поправочный коэффициент, $0 < \alpha_\tau \leq 1$, однозначно

определяется уравнением $\sum_{k=l+l+1}^{\tau-1} \frac{x_{k-1}^2}{\Gamma_l} + \alpha_\tau \frac{x_{\tau-1}^2}{\Gamma_l} = H$.

Множитель из (3) определяется выражением $\tilde{H} = \sum_{k=l+l+1}^{\tau} \sqrt{\beta_k} \frac{x_{k-1}^2}{\Gamma_l}$.

Теорема 1. Среднеквадратичное отклонение оценки (3) ограничено сверху $E(\theta^*(l, H) - \theta)^2 \leq \frac{1}{H}$.

Наконец, мы строим доверительный интервал фиксированной ширины, накрывающий параметр θ с заданной вероятностью α .

Теорема 2. Пусть шумы ε в модели (1) имеют стандартное гауссовское распределение. Тогда для любых $C > 0$ и $0 < \alpha < 1$ доверительный интервал для оценки (3) определяется

$$P\{\theta^*(H) - C \leq \theta < \theta^*(H) + C\} \geq 1 - \alpha, \quad \text{где}$$

$$\alpha = 2 \int_0^{+\infty} (1 - \Phi(C\sqrt{Hy})) \frac{(1/2)^{l/2}}{\Gamma(l/2)} \left(\frac{y}{C_f(l)}\right)^{l/2-1} e^{-\frac{y}{2C_f(l)}} dy, \quad \Gamma(l/2) - \text{гамма-функция.}$$

4. Результаты численного моделирования

4.1. Сравнение точности оценки и среднего времени оценивания параметра θ при разных шумах в авторегрессионной модели AR(1)

В данном разделе рассмотрим результаты численного моделирования метода оценивания параметра авторегрессии при неизвестной дисперсии шумов [2], чтобы убедиться, что достигается гарантированная точность оценивания $1/H$ для двух шумов.

Данные представлены в табл. 1 при $h = 500$, при первых наблюдениях $t = 25$ и последующих $l = 25$. По теореме 1 гарантируем, что точность оценивания должна быть не больше 0.002.

Таблица 1

Точность оценки и среднее время оценивания параметра θ при разных шумах

θ	Гауссовский шум		Двойной экспоненциальный шум	
	$E(\theta - \theta^*)^2$	Среднее время оценивания параметра θ	$E(\theta - \theta^*)^2$	Среднее время оценивания параметра θ
0.1	0.00194	586.086	0.0023	584.006
0.8	0.0020	254.12	0.0019	258.414
0.99	0.0017	88.36	0.0020	89.974
1	0.0017	80.702	0.0019	81.192

Данные представлены в табл. 2 при $h = 1000$, при первых наблюдениях $t = 50$ и последующих $l = 25$. По теореме 1 гарантируем, что точность оценивания должна быть не больше 0.001.

Таблица 2

Точность оценки и среднее время оценивания параметра θ при разных шумах

θ	Гауссовский шум		Двойной экспоненциальный шум	
	$E(\theta - \theta^*)^2$	Среднее время оценивания параметра θ	$E(\theta - \theta^*)^2$	Среднее время оценивания параметра θ
0.1	0.001	1128.913	0.0011	118.594
0.8	0.0010	455.775	0.0011	467.874
0.99	0.0009	130.249	0.00096	132.755
1	0.0009	113.971	0.0011	115.92

Из таблиц видно, что метод оценивания параметра θ гарантирует точность $1/H$ и значения среднего времени оценивания параметра близки при гауссовском и двойном экспоненциальном шумах. Значит, данный метод оценивания подходит не только для конкретного распределения шума в процессе AR(1), он является подходящим для гауссовского и двойного экспоненциального, как видно из табл. 1 и 2.

4.2. Выборочная вероятность непопадания в заданный интервал

Найдем численно выборочную вероятность непопадания в заданный интервал для конкретного значения C и α . Данные представлены в табл. 3 при $\theta = 0.8$ и гауссовском шуме.

Таблица 3

Выборочная вероятность непопадания в заданный интервал

θ	h	C	Интеграл	Выборочная вероятность непопадания
0.03	1260	0.1	0.03	0.001
0.02	1449	0.1	0.02	0.001
0.01	1722	0.1	0.01	0.001
0.002	2310	0.1	0.002	0.0
0.001	2562	0.1	0.001	0.0
0.03	315	0.2	0.03	0.0
0.02	362	0.2	0.0200	0.0
0.01	430	0.2	0.0101	0.0
0.002	557	0.2	0.00220	0.0
0.001	640	0.2	0.0010	0.0

Поскольку интеграл аналитически не считается, он посчитан численно, значение h подбиралось таким образом, чтобы обеспечивалось для заданного значения C заданный уровень ненакрытия интервалом фиксированной ширины a . Чем больше C и a , тем меньше h . В итоге мы видим, что для всех значений C и a гарантируется заданная вероятность непопадания в заданный интервал.

4.3. Сравнение среднего времени оценивания параметра θ при известной и неизвестной дисперсии авторегрессионной модели AR(1) при разных шумах

В данном разделе рассмотрим результаты численного моделирования метода оценивания параметра авторегрессии при неизвестной дисперсии шумов [2], а также метод оценивания того же процесса при известной дисперсии [3], чтоб убедиться, близки ли их времена оценивания параметра.

Данные представлены в табл. 4 при $h = 640$, $\sigma = 1$ и гауссовском шуме, где t и l – первичные и последующие наблюдения.

Таблица 4

Сравнение среднего времени оценивания параметра θ при известной и неизвестной дисперсии авторегрессионной модели AR(1) при разных шумах

t	l	Среднее время оценивания при известной σ^2	Среднее время оценивания при неизвестной σ^2	θ
25	5	935.955	634.124	0.1
25	10	798.68		
5	25	985.004		
10	25	762.855		
25	25	741.946		
25	5	372.985	237.437	0.8
25	10	317.683		
5	25	558.336		
10	25	352.419		
25	25	314.405		
25	5	74.882	53.061	1
25	10	75.733		
5	25	122.406		
10	25	88.533		
25	25	85.227		

Данные представлены в табл. 5 при $h = 640$, $\sigma = 1$, при двойном экспоненциальном шуме, где t и l – первичные и последующие наблюдения.

Таблица 5

Сравнение среднего времени оценивания параметра θ при известной и неизвестной дисперсии авторегрессионной модели AR(1) при разных шумах

t	l	Среднее время оценивания при известной σ^2	Среднее время оценивания при неизвестной σ^2	θ
25	5	868.396	636.102	0.1
25	10	798.021		
5	25	1047.973		
10	25	759.521		
25	25	758.708		
25	5	339.779	237.847	0.8
25	10	324.759		
5	25	791.447		
10	25	371.818		
25	25	313.894		
25	5	70.374	53.636	1
25	10	74.961		
5	25	130.043		
10	25	88.465		
25	25	86.367		

Заметим, что время оценивания параметра зависит от величин первичных и последующих наблюдений, т.е. от количества наблюдений, требующихся для пилотной оценки и оценки дисперсии шумов. Отметим, что если взять малое количество первичных наблюдений, то значение времени оценивания будет во много раз больше, чем при более числе первичных наблюдений.

В случае, когда дисперсия неизвестна, чтобы гарантировать точность оценивания, как при известной дисперсии, приходится использовать больше наблюдений, но, всё-таки, не во много раз больше, как видно, из табл. 4 и 5, при $t = l = 25$ значение среднего времени оценивания при неизвестной дисперсии имеет нерадикальное увеличение от среднего времени оценивания в случае известной дисперсии. Значит, время оценивания параметра при неизвестной дисперсии шумов близка к минимальному значению, следовательно, метод оценивания параметра θ процесса авторегрессии [2] гарантирует, что время оценивания параметра близко к минимальному значению.

4.4. Сравнение точности оценки и среднего времени оценивания параметра θ при известной и неизвестной дисперсии шумов авторегрессионной модели AR(1) при увеличении h

В данном разделе рассмотрим результаты численного моделирования метода оценивания параметра авторегрессии при неизвестной дисперсии шумов [2], а также метод оценивания того же процесса при известной дисперсии [3], чтоб убедиться, что при увеличении h суждения, сказанные ранее, сохраняются, а конкретнее, сохраняется ли утверждение, что время оценивания параметра при неизвестной дисперсии не сильно увеличено от времени оценивания в случае, когда дисперсия шумов известна. Данные, представленные в табл. 6, при $\sigma = 1$, при гауссовском шуме, $t = l = 25$.

Таблица 6

Сравнение точности оценки и среднего времени оценивания параметра θ при известной и неизвестной дисперсии шумов авторегрессионной модели AR(1) при увеличении h

θ	При неизвестной σ^2		При известной σ^2		h
	$E(\theta - \hat{\theta})^2$	Среднее время оценивания параметра θ	$E(\theta - \hat{\theta})^2$	Среднее время оценивания параметра θ	
0.1	0.00192	590.228	0.0018	496.028	500
	0.00098	1120.632		992.497	1000
0.5	0.002	463.289	0.0026	379.149	500
	0.001	872.599		751.974	1000
1	0.00186	81.99	0.009	46.816	500
	0.0009	97.374		65.698	1000

Заметим, что точность оценивания в процессе с неизвестной дисперсией шумов при разных h всегда близка к истине. Аналогично, среднее время оценивания в случае неизвестной дисперсии при увеличении h имеет нерадикальное увеличение от среднего времени оценивания в случае, когда дисперсия известна. Т.е. все суждения, сказанные ранее, сохраняются при увеличении h при разных θ .

4.5. Сравнение качества методов оценивания

Выигрыш оценки состоит в том, что мы для получения оценки с гарантированным качеством можем использовать меньше наблюдений, т.е. мы можем взять сколько угодно данных (мало, много) и получим оценку параметра с точностью $1/N$ и с меньшим временем оценивания. В качестве этого приведем наглядные таблицы, где будем сравнивать оценки при разном числе наблюдений.

Рассмотрим данные в следующих таблицах при гауссовском шуме, $h = 500$, где t и l – первичные и последующие наблюдения. Информация представлена в табл. 7 при $\theta = 0.5$.

Таблица 7

Точность оценки и среднее время оценивания параметра θ при методах оценивания

t	l	$E(\theta - \hat{\theta})^2$		Среднее время оценивания параметра θ	
		Улучшенная оценка	Другая оценка ((I))	Улучшенная оценка	Другая оценка ((I))
25	25	0.0019	0.00163	461.1	541.823
25	50	0.00184	0.00165	482.958	543.109
25	100	0.002	0.0015	521.66	547.698
50	25	0.0021	0.0015	483.063	556.272
100	25	0.0021	0.00148	528.35	604.851
100	100	0.00197	0.0015	583.788	605.904

Информация представлена в табл. 8 при $\theta = 0.9$.

Таблица 8

Точность оценки и среднее время оценивания параметра θ при методах оценивания

t	l	$E(\theta - \hat{\theta})^2$		Среднее время оценивания параметра θ	
		Улучшенная оценка	Другая оценка ((I))	Улучшенная оценка	Другая оценка ((I))
25	25	0.0018	0.00076	168.8	474.98
25	50	0.0019	0.00072	187.946	476.518
25	100	0.0019	0.00076	241.03	416.737
50	25	0.002	0.00055	187.299	511.812
100	25	0.00185	0.00047	236.228	576.15
100	100	0.002	0.00047	306.021	601.599

Рассмотрим данные в табл. 9 при двойном экспоненциальном шуме, $h = 500$, $\theta = 0.9$, где t и l – первичные и последующие наблюдения.

Таблица 9

Точность оценки и среднее время оценивания параметра θ при методах оценивания

t	l	$E(\theta - \hat{\theta})^2$		Среднее время оценивания параметра θ	
		Улучшенная оценка	Другая оценка ((I))	Улучшенная оценка	Другая оценка ((I))
25	25	0.0021	0.0008	171.62	485.405
25	50	0.0020	0.00068	192.213	452.393
25	100	0.00195	0.00086	240.936	483.406
50	25	0.0022	0.000549	187.217	527.385
100	25	0.0023	0.0005	237.427	594.757
100	100	0.0019	0.00045	307.802	599.614

Действительно, обеспечивается точность оценивания $1/N$. В сравнении привели данные оценки [1], и видим, что гарантированная точность наблюдается хуже при росте θ . Также наблюдаем, что среднее время оценивания параметра при улучшенной оценке меньше, чем при другой.

Зачем данное улучшение? Ответ очевиден. Чем больше данных наблюдений мы имеем, тем точнее оценку мы получим. Но не всегда может быть большое количество наблюдений, относительно которых мы оцениваем параметр авторегрессии. Отметим, что намного чаще используются задачи с малым количеством данных. Данная проблема была решена в [2], где была построена оценка параметра авторегрессии, которая гарантирует заданную точность оценивания параметра независимо от количества наблюдений. Также данная оценка обеспечивает минимальное время оценивания параметра, что тоже немало важно.

Заключение

В заключении хочется отметить следующее: оценка параметра авторегрессии [2] лучше оценки [1] в смысле гарантированного среднеквадратического отклонения и времени оценивания, время оценивания параметра улучшенной оценки при неизвестной дисперсии шумов не радикально превышает время оценивания при известной дисперсией шумов, оценки сохраняют свои свойства при увеличении h и при гауссовском и двойном экспоненциальном шумах.

ЛИТЕРАТУРА

1. Дмитриенко А.А., Конев В.В. О гарантированном оценивании параметров авторегрессии при неизвестной дисперсии помех // Автоматика и телемеханика. – 1994. – № 2. – С. 87–99.
2. Vorobeychikov S.E., Burkatovskaya Y.B. Non-asymptotic Confidence Estimation of the Autoregressive Parameter in AR(1) Process with an Unknown Noise Variance // Austrian Journal of Statistics. – 2020. – 49(4). – P. 19–26.
3. Борисов В.З., Конев В.В. О последовательном оценивании параметров дискретных процессов // Автоматика и телемеханика. – 1977. – № 10. – С. 58–64.

DOI: 10.17223/978-5-907442-42-9-2021-11

НAR-RV МОДЕЛЬ И ПРОГНОЗ ВОЛАТИЛЬНОСТИ

Пупков А.В.

Томский государственный университет
andrewpupkov@gmail.com

Введение

Многие динамические процессы описываются стохастическими дифференциальными и стохастическими разностными уравнениями. Как правило, моделирование производится с помощью аппроксимации случайного процесса некоторой моделью определенной структуры. Наиболее известные модели дискретного времени, используемые для таких целей – это модель скользящего среднего (MA(q)), модель авторегрессии (AR(p)), семейство GARCH(p,q) моделей и всевозможные их сочетания. Также существует семейство непрерывных моделей. Наиболее значимым на сегодняшний день среди них является модель броуновского движения или так называемый винеровский процесс, на основе которого строится множество моделей, например, классическое геометрическое броуновское движение, используемое для моделирования финансовых рынков. Во многих случаях процесс моделирования требует достаточно много априорной информации о процессе, например, для аппроксимации броуновским движением некоторого процесса необходимо, чтобы математическое ожидание и приращения этого процесса удовлетворяли гауссовости. В случаях, когда ситуация не позволяет знать распределение процесса, на помощь приходят непараметрические модели ядерного типа. Аппроксимация случайных процессов упомянутыми моделями производится во

многих областях науки и техники, например, в экономике [1–3], физике [4], теории информации и кодирования [5], страховании [6] и т.д.

Для моделирования финансовых рынков, как правило, используются модели для описания поведения основного тренда процесса, а также дополнительная составляющая, описывающая динамику дисперсии процесса или так называемую волатильность. Для описания волатильности используются модели GARCH(p,q) семейства, а также HAR-RV модель реализованной волатильности, которая отличается простотой и интерпретируемостью.

Цель данной работы – представить результаты моделирования последовательности реализованных волатильностей на основе данных динамики индекса S&P500 за 19-летний период с 16.02.2001 по 27.08.2020, используя деление рассматриваемого процесса на режимы.

1. Понятие реализованной волатильности

В данной работе для прогноза волатильности используется модель, предложенная Corsi в 2009 г. в [7]. Автором был рассмотрен диффузионный процесс вида $dp(t) = \mu(t)dt + \sigma(t)dW(t)$, где $p(t)$ – цены актива в момент t , $\mu(t)$ – процесс, описывающий тренд цены актива, $W(t)$ – стандартное броуновское движение, $\sigma(t)$ – процесс, не зависящий от $W(t)$. В качестве описательной характеристики выступает следующая величина

$$IV_t^{(d)} = \int_{t-1d}^t \sigma^2(\omega)d\omega, \quad (1)$$

называемая интегрированной дисперсией, где $1d$ – полный день торгов. Состоятельная оценка (1) имеет вид

$$RV_t = \sqrt{\sum_{j=0}^M r_{t-j\Delta}^2}, \quad r_{t-j\Delta} = \ln\left(\frac{P_{t-j\Delta}}{P_{t-(j+1)\Delta}}\right), \quad (2)$$

где t – день, M – количество внутрисдневных доходностей, $\Delta = 1/M$, P_t – значение индекса в момент времени t . Реализованная волатильность за неделю и за месяц вычисляются соответственно как $RVW_t = \frac{1}{5} \sum_{k=0}^4 RV_k$, $RVW_t = \frac{1}{22} \sum_{k=0}^{21} RV_k$, где среднее рассчитывается по количеству торговых дней за неделю и за месяц соответственно.

2. HAR-RV модель

HAR-RV модель имеет вид $RV_{t+1} = \beta_0 + \beta_1 RV_t + \beta_2 RVW_t + \beta_3 RVM_t + \varepsilon_{t+1}$, где $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^T$ – вектор параметров модели. HAR-RV модель показывает гораздо более качественные результаты при прогнозировании реализованной волатильности, по сравнению со всем семейством GARCH моделей [8].

3. Рассматриваемые данные и вычисление реализованных волатильностей

Рассматриваются значения индекса S&P500 за 19-летний период с 16.02.2001 по 27.08.2020. На рис. 1 представлена последовательность значений индекса S&P500 и последовательность ежедневных логарифмических доходностей индекса.

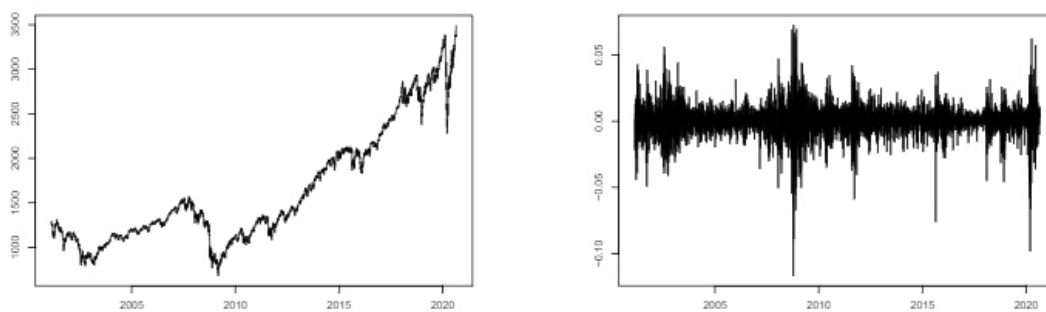


Рис. 1. Последовательность значений индекса S&P500 (слева) и последовательность ежедневных логарифмических доходностей (справа)

Для вычисления последовательности реализованных волатильностей необходимо иметь последовательность внутридневных логарифмических доходностей, часть из которых представлены на рис. 2.

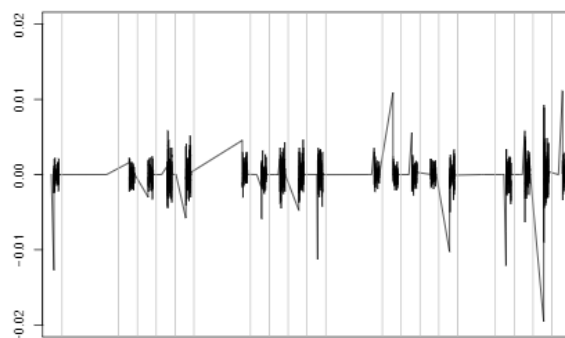


Рис. 2. Последовательность внутридневных доходностей r_t (вертикальные серые линии означают конец торгового дня)

На рис. 3 представлена последовательность логарифмических доходностей (светло-серый график) и соответствующая последовательность реализованных волатильностей, вычисленная по (2) (черный график).

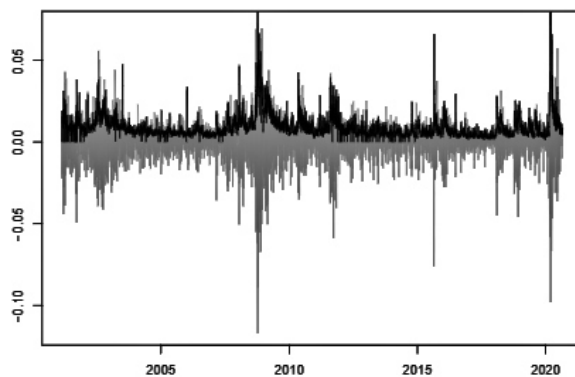


Рис. 3. Последовательность логарифмических доходностей (светло-серый график) и последовательность $\{RV_t\}$ (черный график)

Из рис. 3 видно, что последовательность $\{RV_t\}$ достаточно хорошо описывает последовательность ежедневных логарифмических доходностей индекса.

4. Деление последовательности реализованных волатильностей на режимы

Для деления процесса на режимы использовалась скрытая марковская цепь. Для ее применения необходимо последовательность $\{RV_t\}$ преобразовать к последовательности, которая будет принимать конечное дискретное число значений. В данной работе предлагается подобное преобразование произвести по незамысловатой схеме вида

$$V_t = \begin{cases} HV, & \text{если } RV \geq \alpha, \\ LV, & \text{если } RV < \alpha, \end{cases}$$

где α – некоторое положительное число.

Далее при помощи алгоритма Баума – Велша находится значение эмиссионной матрицы скрытой марковской цепи. На основе полученной эмиссионной матрицы, при помощи алгоритма Витерби, вычисляется наиболее вероятное скрытое состояние, в которой находится скрытая марковская цепь в момент t . Стоит отметить, что значение полученной эмиссионной матрицы, довольно сильно зависит от ее начального значения. На рис. 4 представлены два варианта деления последовательности реализованных волатильностей на режимы в зависимости от различных начальных значений эмиссионной матрицы.

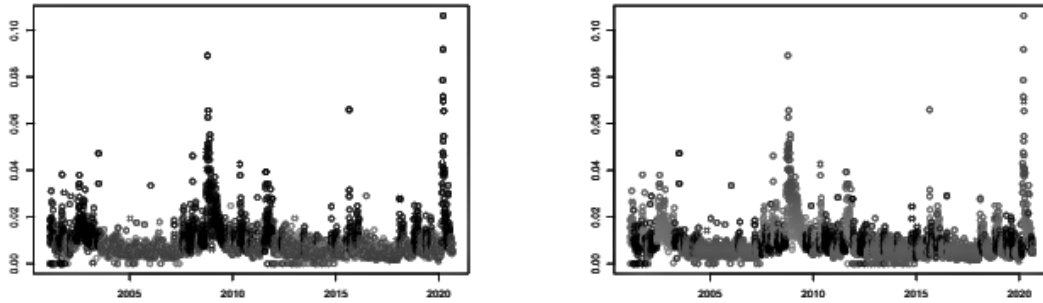


Рис. 4. Последовательность $\{RV_t\}$, разделенная на 3 режима. Приводятся два варианта деления на режимы

5. Оценивание HAR-RV модели на основе данных из разных режимов

Оценивание параметра β HAR-RV модели будем производить для данных из каждого режима, т.е. сначала разделим данные на три кластера, а затем на каждом из этих кластеров оценим параметры HAR-RV модели. Оценивание будем производить при помощи метода наименьших квадратов. Оценка имеет вид $\hat{\beta}_t^i = (A_{t,i}^T A_{t,i})^{-1} A_{t,i}^T y_{t+1,i}$, $i = \overline{1,3}$, где

$$A_{t,i} = \begin{bmatrix} 1 & RV_{t,i} & \frac{1}{5} \sum_{j=0}^4 RV_{t-j,i} & \frac{1}{22} \sum_{j=0}^{21} RV_{t-j,i} \\ 1 & RV_{t-1,i} & \frac{1}{5} \sum_{j=0}^4 RV_{t-j-1,i} & \frac{1}{22} \sum_{j=0}^{21} RV_{t-j-1,i} \\ \dots & \dots & \dots & \dots \\ 1 & RV_{0,i} & \frac{1}{5} \sum_{j=0}^4 RV_{-j,i} & \frac{1}{22} \sum_{j=0}^{21} RV_{-j,i} \end{bmatrix}, \quad y_{t+1,i} = \begin{bmatrix} RV_{t+1,i} \\ RV_{t,i} \\ \dots \\ RV_{1,i} \end{bmatrix},$$

здесь $RV_{t,i}$ – значение реализованной волатильности в день t в i -м кластере.

В следующем разделе рассмотрим прогнозирование последовательности реализованных волатильностей на 1 день вперед.

6. Прогнозирование волатильности при помощи HAR-RV модели

Прогноз реализованной волатильности на 1 день описывается выражением $\overline{RV}_{t+1} = \beta_0^* + \beta_1^* RV_t + \beta_2^* RVW_t + \beta_3^* RVM_t$, где $\beta_0^* = \beta_0^{R(RV_t)}$, $\beta_1^* = \beta_1^{R(RV_t)}$, $\beta_2^* = \frac{1}{5} \sum_{i=0}^4 \beta_2^{R(RV_{t-i})}$,

$\beta_3^* = \frac{1}{22} \sum_{i=0}^{21} \beta_3^{R(RV_{t-i})}$, $R(x)$ – режим, в котором находится наблюдение x .

В рассматриваемой ситуации параметры скрытой марковской цепи оценивались по 3000 первых наблюдений реализованной волатильности. Значения реализованной волатильности разделялись на режимы и внутри выборки оценивания параметров НММ и за ее пределами. Прогноз строился за пределами выборки оценивания параметров HAR-RV.

На рис. 5 представлены прогнозы на 1 день в разные временные интервалы. На рис. 5 (слева) рассмотрена ситуация перехода с высоковолатильного режима на низковолатильный. На рис. 5 (справа) рассмотрена ситуация перехода с низковолатильного на высоковолатильный режим с последующим снижением до низковолатильного. Визуально видно, что прогнозная сила алгоритма ухудшается в ситуации, когда происходит резкое повышение волатильности.

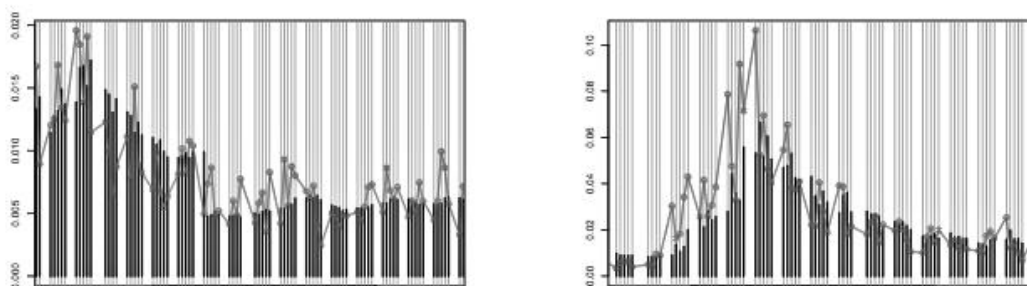


Рис. 5. Прогноз реализованной волатильности на 1 день (серая ломанная линия – реализованная волатильность, черные столбцы – прогноз на 1 день): график слева – с 01.02.2016 до 20.05.2016, график справа – с 12.02.2020 до 20.05.2020

Заключение

В настоящей работе представлен алгоритм прогнозирования реализованной волатильности процесса, описывающего динамику значения индекса S&P500, на основе деления процесса на режимы. Стоит отметить, что рассматриваемая модель ведет себя гораздо лучше в прогнозе реализованных волатильностей, чем все семейство GARCH моделей.

ЛИТЕРАТУРА

1. Ширяев А.Н. Основы стохастической финансовой математики: в 2 т. – Москва: ФАЗИС, 1998.
2. Кошкин Г.М., Пивен И.Г. Непараметрическая идентификация стохастических объектов. – Хабаровск: РАН, Дальневосточное отделение, 2009. – 336 с.
3. Tsay R.S. An Introduction to Analysis of Financial Data with R. – John Wiley & Sons, 2013. – 390 p.
4. Френкель Я.И. Статистическая физика. – Москва Ленинград: АН СССР, 1948. – 760 с.
5. Чечёта С.И. Введение в дискретную теорию информации и кодирования. – М: МЦНМО, 2011. – 224 с.
6. Asmussen S. Albrecher H. Ruin Probabilities. – World Scientific Publishing Co. Pte. Ltd., 2010. – 602 p.
7. Corsi F. A Simple Approximate Long-Memory Model of Realized Volatility // Journal of Financial Econometrics. – 2009. – V. 7. – N. 2. – P. 174 – 196.
8. Азанин А.Д. Сравнение GARCH и HAR-RV моделей для прогноза реализованной волатильности на российском рынке // Прикладная эконометрика. – 2017. – Т. 48. – С. 63 – 84.

IV. ОБРАБОТКА ДАННЫХ, УПРАВЛЕНИЕ И ИССЛЕДОВАНИЕ СТОХАСТИЧЕСКИХ СИСТЕМ

DOI: 10.17223/978-5-907442-42-9-2021-12

ОЦЕНИВАНИЕ МЕТОДОМ МОМЕНТОВ ПАРАМЕТРА РАВНОМЕРНОГО РАСПРЕДЕЛЕНИЯ ДЛИТЕЛЬНОСТИ СЛУЧАЙНОГО НЕПРОДЛЕВАЮЩЕГОСЯ МЕРТВОГО ВРЕМЕНИ В РЕКУРРЕНТНОМ ПОЛУСИНХРОННОМ ПОТОКЕ СОБЫТИЙ

Горцев А.М., Веткина А.В.

*Томский государственный университет
anyavetkina@stud.tsu.ru*

Введение

В настоящее время наиболее точной и адекватной математической моделью реальных информационных потоков в телекоммуникационных системах и сетях являются дважды стохастические потоки событий – это потоки, у которых случайными являются и моменты наступления событий, и интенсивность потока. В связи с этим исследование разновидностей дважды стохастических потоков становится важной актуальной задачей.

Дважды стохастические потоки можно разделить на два класса: первый класс составляют потоки, сопровождающий процесс которых есть непрерывный случайный процесс [1,2]; второй – потоки, сопровождающий процесс которых есть кусочно-постоянный случайный процесс с конечным (произвольным) числом состояний [3–5].

Помимо этого, в зависимости от того, каким образом происходит переход сопровождающего процесса из состояния в состояние, дважды стохастические потоки делятся на три типа: 1) синхронные потоки (потоки, у которых состояние сопровождающего процесса меняется в случайные моменты времени, являющиеся моментами наступления событий) [6–9]; 2) асинхронные потоки (потоки, у которых переход из состояния в состояние сопровождающего процесса происходит в случайные моменты времени и не зависит от моментов наступления событий) [10–13]; 3) полусинхронные потоки (потоки, у которых одна часть состояний сопровождающего процесса меняется в моменты наступления событий потока, другая часть состояний сопровождающего процесса меняется в произвольные моменты времени, не связанные с моментами наступления событий потока) [14–17].

На практике часто приходится иметь дело с потоками, у которых не все события доступны наблюдению. Как правило, причиной ненаблюдаемости служит так называемое мертвое время регистрирующих приборов [18], порождаемое зарегистрированным событием, другие же события, наступившие в этот период, теряются. Регистрирующие приборы, вместе с тем, делятся на два вида: с непродлевающимся мертвым временем и продлевающимся. Кроме того, длительность мертвого времени может быть как детерминированной величиной, одинаковой для всех событий, так и случайной с тем или иным законом распределения [18].

В реальности, наблюдая за системами различного вида, исследователь имеет лишь информацию на выходе (случайную выборку данных), а то, что подается на вход системы, остается для него неизвестным. Именно поэтому важной задачей становится задача оценивания входных параметров потока с помощью использования наблюдаемых данных.

В данной работе рассматривается полусинхронный дважды стохастический поток событий, функционирующий в условиях случайного непродлевающегося мертвого

времени, распределенного по равномерному закону. Выводится явный вид плотности вероятности значений длительности интервала между моментами наступления событий наблюдаемого потока, а также находится явный вид совместной плотности вероятности значений длительностей смежных интервалов. Находятся условия рекуррентности потока в рассматриваемом случае. Для рекуррентного потока методом моментов решается задача оценивания параметра распределения длительности случайного мертвого времени.

1. Математическая модель наблюдаемого потока

Рассматривается стационарный режим функционирования полусинхронного дважды стохастического потока событий, сопровождающий процесс (интенсивность) которого есть кусочно-постоянный стационарный случайный процесс $\lambda(t)$ с двумя состояниями S_1 и S_2 . Будем говорить, что имеет место первое состояние процесса (потока) S_1 , если $\lambda(t) = \lambda_1$ и, наоборот, имеет место второе состояние процесса (потока) S_2 , если $\lambda(t) = \lambda_2$ ($\lambda_1 > \lambda_2 \geq 0$). Если имеет место первое состояние процесса S_1 , то в течение временного интервала, когда $\lambda(t) = \lambda_1$, поступает пуассоновский поток событий с интенсивностью λ_1 . Если имеет место второе состояние процесса S_2 , то в течение временного интервала, когда $\lambda(t) = \lambda_2$, поступает пуассоновский поток событий с интенсивностью λ_2 . Переход из состояния S_1 процесса $\lambda(t)$ в состояние S_2 возможен только в момент наступления события, при этом этот переход осуществляется с вероятностью p (с вероятностью $1 - p$ процесс $\lambda(t)$ остается в состоянии S_1). Переход из состояния S_2 процесса $\lambda(t)$ в состояние S_1 может осуществляться в произвольный момент времени, не связанный с моментом наступления события. При этом длительность пребывания процесса $\lambda(t)$ во втором состоянии есть случайная величина, распределенная по экспоненциальному закону $F(t) = 1 - e^{-\alpha_2 t}$, $t \geq 0$, где α_2 – интенсивность смены состояния S_2 на S_1 . Так как переход из второго состояния в первое не привязан к моменту наступления события во втором состоянии, то поток называется полусинхронным дважды стохастическим потоком событий. В сделанных предположениях $\lambda(t)$ – скрытый марковский процесс ($\lambda(t)$ – принципиально ненаблюдаемый процесс; наблюдаемыми являются только моменты наступления событий потока).

После каждого зарегистрированного события в момент времени t_k наступает период мертвого времени случайной длительности, который порождается этим событием, так что другие события исходного потока, наступившие в течение этого периода мертвого времени, недоступны наблюдению и не вызывают его продления (непродлеваемое мертвое время). Принимается, что случайная длительность мертвого времени распределена по равномерному закону с плотностью вероятности $p(T) = 1/T^*$, где T – значение длительности мертвого времени, $0 \leq T \leq T^*$.

Возможный вариант возникающей ситуации приведен на рис. 1, где S_1 и S_2 – состояния случайного процесса $\lambda(t)$; временная ось $(0, t)$ – ось моментов наступления наблюдаемых событий в моменты времени t_1, t_2, \dots ; временная ось $(0, t^{(1)})$ – ось наступления событий в моменты времени $t^{(1)}, t^{(2)}, \dots$ в первом (S_1) состоянии процесса $\lambda(t)$, на которой также указаны значения длительностей $T^{(1)}, T^{(2)}, \dots$ мертвых времен, порождаемых наблюдаемыми событиями потока; аналогично для временной оси $(0, t^{(2)})$; белыми кружками обозначены наблюдаемые события, черными – ненаблюдаемые, штриховкой – периоды продолжительности мертвого времени; траектория процесса $\lambda(t)$ привязана к временной оси $(0, t^{(1)})$.

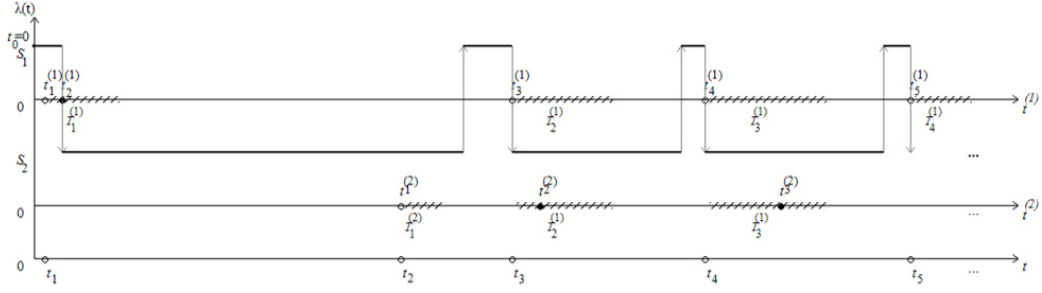


Рис. 1. Формирование наблюдаемого полусинхронного потока событий

Рассмотрим теперь полусинхронный поток событий, функционирующий в условиях непродлевающегося мертвого времени фиксированной длительности $T \geq 0$; если $T = 0$, то мертвое время отсутствует. Из [17] имеем, что плотность вероятности $p(\tau|T)$ значений длительности интервала между соседними событиями в полусинхронном потоке, функционирующем в условиях непродлевающегося мертвого времени фиксированной длительности T имеет вид

$$p(\tau|T) = \begin{cases} 0, & 0 \leq \tau < T, \\ \gamma(T)\lambda_1 e^{-\lambda_1(\tau-T)} + (1-\gamma(T))(\alpha_2 + \lambda_2) e^{-(\alpha_2 + \lambda_2)(\tau-T)}, & \tau \geq T, \end{cases} \quad (1)$$

$$p(\tau_1, \tau_2|T) = 0, \quad \tau_2 \geq 0, \quad 0 \leq \tau_1 < T, \quad \tau_1 \geq 0, \quad 0 \leq \tau_2 < T,$$

$$p(\tau_1, \tau_2|T) = p(\tau_1|T)p(\tau_2|T) + e^{-(\alpha_2 + p\lambda_1)T} \gamma(T) [1 - \gamma(T)] \frac{\lambda_2(1-p)}{\lambda_2 + \alpha_2} \times \\ \times \left[\lambda_1 e^{-\lambda_1(\tau_1-T)} - (\lambda_2 + \alpha_2) e^{-(\lambda_2 + \alpha_2)(\tau_1-T)} \right] \left[\lambda_1 e^{-\lambda_1(\tau_2-T)} - (\lambda_2 + \alpha_2) e^{-(\lambda_2 + \alpha_2)(\tau_2-T)} \right], \quad (2) \\ \tau_1 \geq T, \quad \tau_2 \geq T,$$

где $\gamma(T) = \frac{\alpha_2(\lambda_1 - \lambda_2 - \alpha_2 - \lambda_1 p)}{(\lambda_1 p + \alpha_2)(\lambda_1 - \lambda_2 - \alpha_2)} \left[1 - \frac{p(\lambda_1 - \lambda_2)}{(1-p)\lambda_2 - (\alpha_2 + \lambda_2) e^{(\lambda_1 p + \alpha_2)T}} \right]$, $\lambda_1 - \lambda_2 - \alpha_2 \neq 0$; τ_1 ,

τ_2 – смежные интервалы между наблюдаемыми событиями.

Рассмотрим частные случаи, при выполнении которых полусинхронный поток событий, функционирующий в условиях мертвого времени, становится рекуррентным потоком:

- 1) $p = 0$; тогда $\gamma(T) = 1$ и совместная плотность (2) факторизуется $p(\tau_1, \tau_2|T) = p(\tau_1|T)p(\tau_2|T)$, $\tau_1 \geq T$, $\tau_2 \geq T$, а плотность вероятности (1) принимает вид $p(\tau|T) = \lambda_1 e^{-\lambda_1(\tau-T)}$, $\tau \geq T$; в этом случае имеет место простейший поток с интенсивностью λ_1 .
- 2) $\alpha_2 = 0$; тогда $\gamma(T) = 0$, $p(\tau_1, \tau_2|T) = p(\tau_1|T)p(\tau_2|T)$, $\tau_1 \geq T$, $\tau_2 \geq T$, $p(\tau|T) = \lambda_2 e^{-\lambda_2(\tau-T)}$, $\tau \geq T$; в этом случае имеет место простейший поток с интенсивностью λ_2 .
- 3) $\lambda_1(1-p) - \lambda_2 - \alpha_2 = 0$; тогда $\gamma(T) = 0$, $p(\tau_1, \tau_2|T) = p(\tau_1|T)p(\tau_2|T)$, $\tau_1 \geq T$, $\tau_2 \geq T$, $p(\tau|T) = (\lambda_2 + \alpha_2) e^{-(\lambda_2 + \alpha_2)(\tau-T)}$, $\tau \geq T$; в этом случае имеет место простейший поток с интенсивностью $\lambda_2 + \alpha_2$.
- 4) $\lambda_2 = 0$; в данной ситуации имеет место

$$p(\tau_1, \tau_2 | T) = p(\tau_1 | T) p(\tau_2 | T), \quad \tau_2 \geq 0, 0 \leq \tau_1 < T, \tau_1 \geq 0, 0 \leq \tau_2 < T,$$

$$\gamma(T) = \frac{\alpha_2(\lambda_1 - \alpha_2 - \lambda_1 p)}{(\lambda_1 p + \alpha_2)(\lambda_1 - \alpha_2)} \left[1 + \frac{p \lambda_1}{\alpha_2 e^{(\lambda_1 p + \alpha_2) T}} \right], \quad (3)$$

$$p(\tau | T) = \gamma(T) \lambda_1 e^{-\lambda_1(\tau - T)} + (1 - \gamma(T)) (\alpha_2 + \lambda_2) e^{-(\alpha_2 + \lambda_2)(\tau - T)}, \quad \tau \geq T, \lambda_1 - \alpha_2 \neq 0.$$

Формулы (3) характеризуют альтернирующий рекуррентный полусинхронный поток (отсутствуют события во втором состоянии).

Рассмотрим четвертый случай альтернирующего рекуррентного полусинхронного потока событий, когда $\lambda_2 = 0$. На рис. 2 приведен вариант формирования такого наблюдаемого потока при фиксированном $T > 0$.

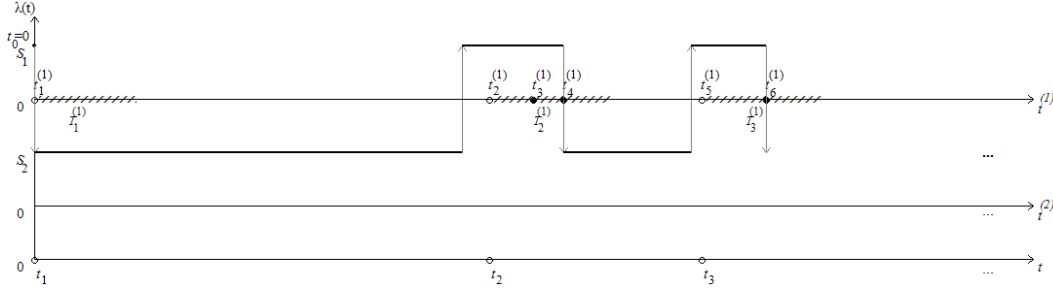


Рис. 2. Формирование наблюдаемого альтернирующего рекуррентного полусинхронного потока событий

2. Уравнение моментов для оценивания параметра T^*

Предварительно отметим, что условия рекуррентности потока 1) – 4), полученные для случая детерминированного мертвого времени, для ситуации случайного непродлевающегося мертвого времени, очевидно, не изменятся, т.к. они являются следствием значений параметров потока или их соотношений. Условия 1) – 3) рекуррентности потока приводят к тому, что исходный исследуемый поток вырождается в простейший, поэтому рассмотрим четвертое условия рекуррентности.

Для оценки неизвестного параметра T^* равномерного распределения длительности случайного непродлевающегося мертвого времени в рекуррентном полусинхронном потоке событий, когда $\lambda_2 = 0$, используется метод моментов [19]. Метод моментов заключается в приравнивании теоретических и эмпирических моментов. Для этого находится теоретический момент – математическое ожидание длительности интервала между соседними событиями в наблюдаемом потоке $M(\tau | T^*)$, а затем значение оценки \hat{T}^* вычисляется численно из уравнения моментов $M(\tau | T^*) = C$, где C – значение оценки математического ожидания $M(\tau | T^*)$, $C = \frac{1}{n} \sum_{k=1}^n \tau_k$, $\tau_k = t_{k+1} - t_k$. Величины τ_k являются выходными данными имитационной модели наблюдаемого альтернирующего рекуррентного полусинхронного потока событий.

Для нахождения теоретического момента имеем формулу $M(\tau | T^*) = \int_0^{\infty} p(\tau) d\tau$.

Здесь плотность вероятности $p(\tau) = \int_{(T)} p(\tau, T) dT = \int_{(T)} p(T) p(\tau | T) dT$, где (T) – область изменения значений случайной величины T ; $p(T) = 1/T^*$, $0 \leq T \leq T^*$, по условию задачи.

Так как область значений случайной величины мертвого времени представляет собой объединение двух областей, когда $0 \leq \tau < T^*$ и когда $\tau \geq T^*$, то выражение для плотности $p(\tau)$ примет следующий вид:

$$p(\tau) = \begin{cases} p_1(\tau) = \int_0^{\tau} p(T)p(\tau|T)dT, & 0 \leq \tau < T^*, \\ p_2(\tau) = \int_0^{T^*} p(T)p(\tau|T)dT, & \tau \geq T^*, \end{cases} \quad (4)$$

где плотность $p(\tau|T)$ определена в (3).

Подставляя выражение (3) в (4) и учитывая, что $p(T) = 1/T^*$, находим

$$p_1(\tau) = \frac{1}{T^*} \left\{ 1 - e^{-\lambda_1 \tau} - e^{-\alpha_2 \tau} + e^{-(\lambda_1 p + \alpha_2) \tau} \right\}, \quad 0 \leq \tau < T^*, \quad (5)$$

$$p_2(\tau) = \frac{1}{T^*} \left\{ -e^{-\lambda_1 \tau} \left[-1 + C_1 e^{\lambda_1 T^*} + C_2 e^{(\lambda_1 - \lambda_1 p - \alpha_2) T^*} \right] + e^{-\alpha_2 \tau} \left[-1 + C_1 e^{-\lambda_1 p T^*} + C_2 e^{\alpha_2 T^*} \right] \right\}, \quad \tau \geq T^*, \quad (6)$$

где $C_1 = \frac{\alpha_2(\lambda_1 - \alpha_2 - \lambda_1 p)}{(\lambda_1 p + \alpha_2)(\lambda_1 - \alpha_2)}$, $C_2 = \frac{\lambda_1^2 p}{(\lambda_1 p + \alpha_2)(\lambda_1 - \alpha_2)}$.

Отметим, что в граничной точке $\tau = T^*$ имеет место равенство двух плотностей $p_1(T^*) = p_2(T^*)$ и неравенство их производных $p_1'(T^*) \neq p_2'(T^*)$, т.е. функция $p(\tau)$, задаваемая формулой (4), есть непрерывная функция и в точке $\tau = T^*$ имеет излом.

По определению математического ожидания случайной величины τ – длительности интервала между двумя соседними событиями наблюдаемого потока – с учетом формулы (4), имеем $M(\tau|T^*) = \int_0^{T^*} \tau p_1(\tau) d\tau + \int_{T^*}^{\infty} \tau p_2(\tau) d\tau$. Подставляя сюда плотности вероятности $p_1(\tau)$, $p_2(\tau)$, определенные в (5), (6), находим

$$M(\tau|T^*) = \frac{T^*}{2} + \frac{1}{\lambda_1} + \frac{\lambda_1 p}{\alpha_2(\lambda_1 p + \alpha_2)} + \frac{p(\lambda_1(1-p)p + \alpha_2)}{T^* \alpha_2(\lambda_1 p + \alpha_2)^2} \left(1 - e^{-(\lambda_1 p + \alpha_2) T^*} \right), \quad T^* \geq 0. \quad (7)$$

Аналитически можно показать, что $M'(\tau|T^*) > 0$ при $T^* > 0$; $\lim_{T^* \rightarrow 0} M'(\tau|T^*) > 0$ при $T^* \rightarrow 0$. Это означает, что функция $M(\tau|T^*)$ является возрастающей функцией T^* .

Теперь для нахождения значения оценки \hat{T}^* необходимо решить уравнение моментов $M(\tau|T^*) = C$. Данное уравнение будем решать численно методом простой итерации [20].

Отметим, что уравнение моментов $M(\tau|T^*) = C$ имеет единственное решение, т.к. можно показать, что математическое ожидание $M(\tau|T^*)$ – возрастающая функция переменной T^* , C – постоянная величина, характеризующая конкретную реализацию потока.

3. Численные результаты

С целью установления стационарного режима и определения свойств найденной оценки проведены статистические эксперименты. Для параметров потока $\lambda_1 = 2$, $\lambda_2 = 0$, $\alpha_2 = 0,2$, $p = 0,6$ и параметров точности $\varepsilon = 0.0001$, $\Delta T^* = 0.001$ (параметры метода простой итерации) получено 100 реализаций ($N = 100$) имитационной модели

потока и, соответственно, получено 100 решений уравнения моментов, для двух значений параметра $T^* = 1; 3$ и для каждого значения времени моделирования $T_m = 150, 300, \dots, 3000$ ед. времени. Далее на основании полученных данных вычислялись выборочное среднее искомой оценки $\hat{M}(T^*) = \frac{1}{N} \sum_{i=1}^N \hat{T}_i^*$ и ее выборочная вариация $\text{Var}(\hat{T}^*) = \frac{1}{N} \sum_{i=1}^N (\hat{T}_i^* - T^*)^2$, где T^* – известное из имитационной модели значение параметра, \hat{T}_i^* – решение уравнения моментов для i -й реализации потока.

В табл. 1 приведены результаты эксперимента для $\hat{M}(T^*)$. В первой строке таблицы указано время моделирования T_m (время наблюдения за потоком) ($T_m = 150, 300, \dots, 3000$ ед. времени); во второй и третьей строках указано выборочное среднее $\hat{M}(T^*)$ для $T^* = 1$ и $T^* = 3$ соответственно.

Таблица 1

Численные результаты статистического эксперимента для $\hat{M}(T^*)$

T_m		150	300	450	600	750	900	1050	
$\hat{M}(T^*)$	$T^* = 1$	1.31445	1.03263	1.02797	1.08673	1.02456	1.07423	0.99706	
	$T^* = 3$	3.35504	2.91497	2.89639	3.00945	2.96404	2.9861	3.04309	
T_m		1200	1350	1500	1650	1800	1950	2100	2250
$\hat{M}(T^*)$	$T^* = 1$	1.0058	1.02834	0.98195	1.0389	0.99737	1.01769	1.00434	1.01259
	$T^* = 3$	2.9291	2.9892	2.98173	3.00373	2.95898	2.98324	3.04984	2.98432

Для наглядности на рис. 3, 4 приведены графики зависимости $\hat{M}(T^*)$ от значения времени моделирования T_m для $T^* = 1$ и $T^* = 3$, построенные по данным табл. 1.

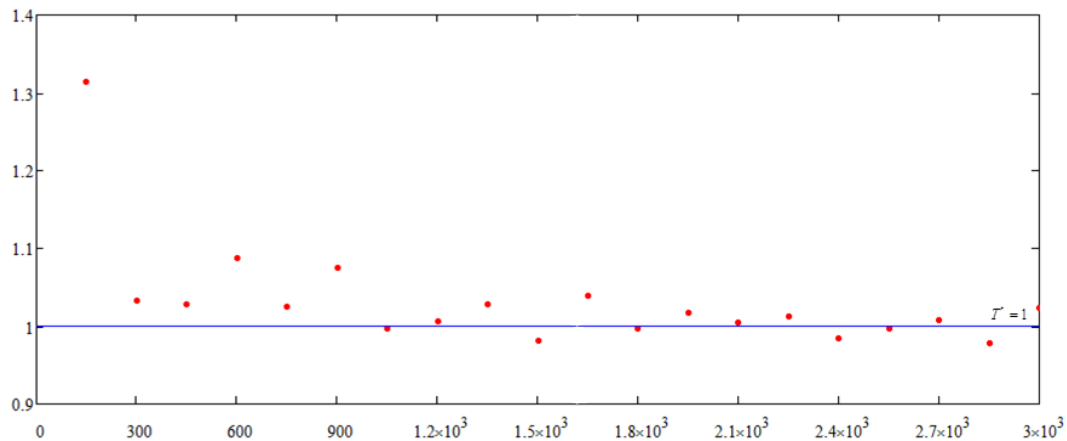


Рис. 3. График зависимости $\hat{M}(T^*)$ от T_m при $T^* = 1$

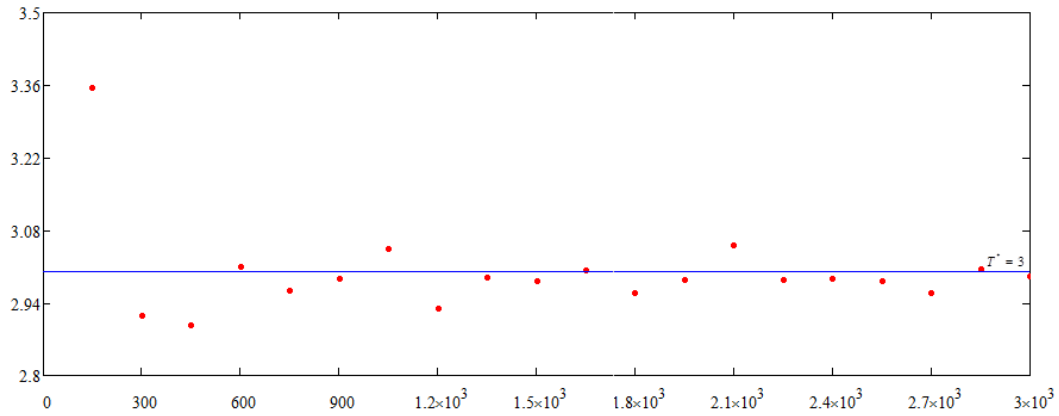


Рис. 4. График зависимости $\hat{M}(\hat{T}^*)$ от T_m при $T^* = 3$

Из анализа результатов табл. 1 и графиков зависимости $\hat{M}(\hat{T}^*)$ от значения T_m следует, что стационарный режим функционирования потока устанавливается при $T_m \geq 1500$ ед. времени, т.к. выборочное среднее $\hat{M}(\hat{T}^*)$ стремится к постоянному значению.

В табл. 2 приведены результаты для $\text{Var}(\hat{T}^*)$. Структура табл. 2 аналогична структуре табл. 1.

Таблица 2

Численные результаты статистического эксперимента для $\text{Var}(\hat{T}^*)$

T_m		150	300	450	600	750	900	1050
$\text{Var}(\hat{T}^*)$	$T^* = 1$	1.89788	0.45199	0.43296	0.26849	0.19075	0.20681	0.11928
	$T^* = 3$	2.17312	1.51272	0.75498	0.71807	0.62688	0.44294	0.47239
T_m	1200	1350	1500	1650	1800	1950	2100	2250
$\text{Var}(\hat{T}^*)$	0.10999	0.11771	0.08557	0.09088	0.06747	0.07263	0.06841	0.07537
	0.28876	0.34313	0.23482	0.2983	0.27287	0.26733	0.23022	0.19273

На рис. 4 и 5 приведены графики зависимости выборочной вариации $\text{Var}(\hat{T}^*)$ от времени моделирования T_m для $T^* = 1$ и $T^* = 3$, построенные по данным табл. 2.

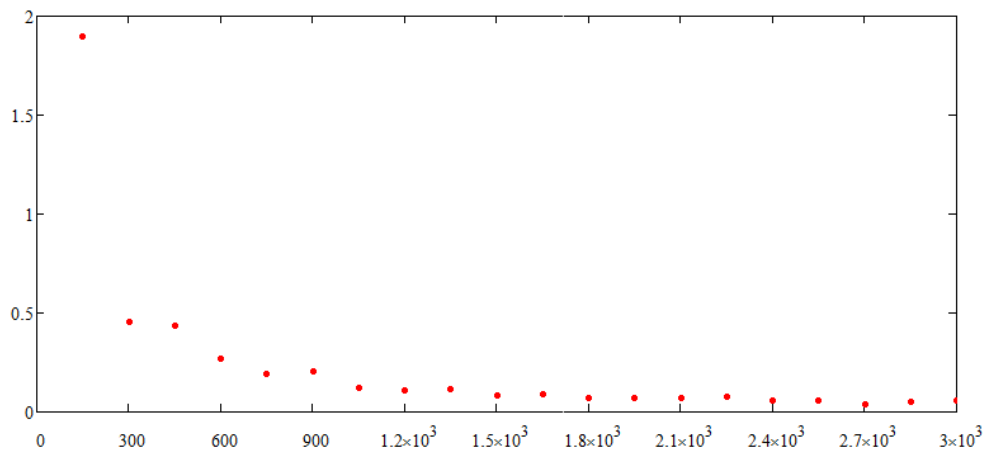


Рис. 5. График зависимости $\text{Var}(\hat{T}^*)$ от T_m при $T^* = 1$

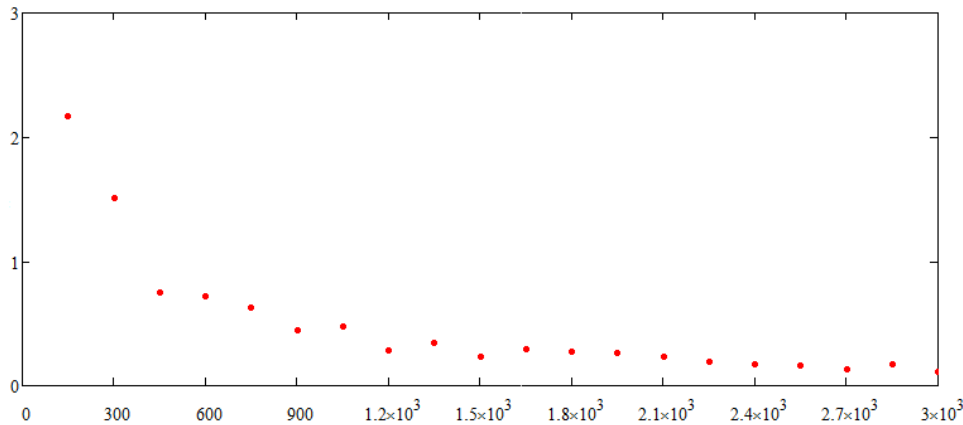


Рис. 6. График зависимости $\text{Var}(\hat{T}^*)$ от T_m при $r^* = 3$

Результаты эксперимента указывают на высокое качество построенной оценки, т.к. выборочная вариация (разброс значений случайной величины \hat{T}^*) стремится к числу, достаточно близкому к нулю, т.е. методика оценивания удовлетворительна и полученную оценку можно принимать за истинную с достаточно малой погрешностью. Также заметим, что выборочная вариация устанавливается возле своего стационарного значения при времени моделирования $T_m \geq 1500$ ед. времени. Таким образом, можно считать, что при $T_m \geq 1500$ достигается нужная для практики точность.

Подчеркнем, что выборочная вариация при $T^* = 3$ больше, чем при $T^* = 1$. Последнее является естественным, т.к. при больших T^* происходит большая потеря событий исходного потока, что влечет за собой ухудшение качества оценивания при одинаковых T_m .

Кроме того, получена состоятельная оценка параметра распределения длительности мертвого времени T^* , т.к. выполняются условия теоремы о состоятельности оценок [19]: длительности интервалов между моментами наступления событий в потоке являются независимыми случайными величинами; теоретическое среднее существует; решение уравнения моментов существует и единственно.

Заключение

В данной работе рассмотрен полусинхронный дважды стохастический поток событий с непродлевающимся случайным мертвым временем. Найдены условия рекуррентности потока и исследован альтернирующий рекуррентный полусинхронный дважды стохастический поток событий.

Аналитически получены формулы (5), (6), определяющие плотность вероятности значений длительности интервала между соседними событиями в наблюдаемом рекуррентном потоке при случайном мертвом времени; выведена формула (7) для математического ожидания длительности интервала между соседними событиями.

Методом моментов найдена оценка параметра равномерного распределения длительности случайного мертвого времени, полученная оценка исследована на качество. Приведенные результаты численных расчетов указывают на приемлемое качество оценивания.

ЛИТЕРАТУРА

1. Cox D.R. The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables // Proceedings of the Cambridge Philosophical Society. – 1955. – V. 51(3). – No. 3. – P. 433–441.

2. Kingman Y.F. C. On doubly stochastic Poisson process // Proceedings of the Cambridge Philosophical Society. – 1964. – V. 60. – No. 4. – P. 923–930.
3. Башарин Г.П., Кокотушкин В.А., Наумов В. А. О методе эквивалентных замен расчета фрагментов сетей связи. Ч.1 // Изв. АНССР. Техн. Кибернетика. – 1979. – №6. – С. 92–99.
4. Neuts M.F. A versatile Markovian point process // Journal of Applied Probability. – 1979. – V. 16. – P. 764–779.
5. Lucantoni D.M. New results on the single server queue with a batch markovian arrival process // Communication in Statistics Stochastic Models. – 1991. – V. 7. – P. 1–46.
6. Горцев А.М., Нежелская Л.А. Оценивание длительности мертвого времени и параметров синхронного альтернирующего потока событий // Вестник Томского государственного университета. – 2003. – № S6. – С. 232–239.
7. Gortsev A.M., Nezhel'skaya L.A. Estimation of parameters of synchronously alternating Poisson stream of events by the moment method // Telecommunications and Radio Engineering. – 1996. – V. 50. – No. 1. – P. 56–63.
8. Горцев А.М., Нежелская Л.А. Оценивание параметров синхронного дважды стохастического потока событий методом моментов // Вестник Томского государственного университета. – 2002. – № S1-1. – С. 24–29.
9. Nezhel'skaya L.A. Probability density function for modulated MAP event flows with unextendible dead time // Communication in Computer and Information Science. – 2015. – V. 564. – P. 141–151.
10. Горцев А.М., Нежелская Л.А. Оценивание параметров асинхронного потока с иницированием лишних событий методом моментов // Вестник Томского государственного университета. – 2006. – № S18. – С. 267–273.
11. Леонова М.А., Нежелская Л.А. Вероятность ошибки при оценивании состояний обобщенного асинхронного потока событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2012. – № 2 (19). – С. 88–101.
12. Горцев А.М., Зуевич В.Л. Оптимальная оценка состояний асинхронного дважды стохастического потока событий с произвольным числом состояний // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2010. – № 2 (11). С. 44–65.
13. Горцев А.М., Леонова М.А., Нежелская Л.А. Сравнение МП- и ММ-оценок длительности мертвого времени в обобщенном асинхронном потоке событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2013. – № 4 (25). – С. 32–42.
14. Калягин А.А., Нежелская Л.А. Сравнение МП- и ММ-оценок длительности мертвого времени в обобщенном полусинхронном потоке событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2015. – № 3 (32). – С. 23–32.
15. Горцев А.М., Калягин А.А., Нежелская Л.А. Оценка максимального правдоподобия длительности мертвого времени в обобщенном полусинхронном потоке // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2015. – № 1 (30). – С. 27–37.
16. Нежелская Л.А. Оптимальное оценивание состояний полусинхронного потока событий в условиях его частичной наблюдаемости // Вестник Томского государственного университета. – 2000. – № 269. – С. 95–98.
17. Горцев А.М., Калягин А.А., Нежелская Л.А. Совместная плотность вероятностей длительности интервалов обобщенного полусинхронного потока событий при непродлеваемом мертвом времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2014. – № 2 (27). – С. 19–29.
18. Апанасович В.В., Коляда А.А., Чернявский А.Ф. Статистический анализ случайных потоков в физическом эксперименте. – Минск: Университетское, 1988. – 256 с.
19. Малинковский Ю.В. Теория вероятностей и математическая статистика (часть 2. Математическая статистика). – Гомель: УО «ГГУ им. Ф. Скорины», 2004. – 146 с.
20. Амосов А.А., Дубинский Ю.А., Копченова Н.В. Вычислительные методы для инженеров: учеб. пособие. – М.: Высшая школа, 1994. – 544 с.

DOI: 10.17223/978-5-907442-42-9-2021-13

ИССЛЕДОВАНИЕ ЭЛЕКТРОМАГНИТНОГО РАССЕЯНИЯ НА ТОНКИХ СООСНЫХ 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 на структурах, состоящих как из идеально проводящих, так и диэлектрических тонких цилиндров конечной длины. Этот интерес обусловлен необходимостью решения таких практически важных проблем, как проблемы радиолокацион-

ной заметности, идентификации объектов, оценки рассеяния диэлектрическими или металлическими цилиндрическими деталями различных геометрически сложных тел и др.

Под тонким цилиндром обычно понимается цилиндрическое тело, чаще кругового сечения, поперечные размеры которого много меньше его длины и длины падающей волны. В [1] предложен численный метод решения задачи электромагнитного рассеяния на структурах, состоящих из одного тонкого диэлектрического цилиндра и одного тонкого идеально проводящего цилиндра. В [2,3] этот метод применен к анализу рассеяния электромагнитной волны на структурах, состоящих из параллельных идеально проводящего и диэлектрического цилиндров, а в [4,5] — к анализу рассеяния электромагнитной волны на структуре, состоящей из ортогональных цилиндров.

Данная работа продолжает цикл работ авторов, посвященных электромагнитному рассеянию на структурах, состоящих из одного тонкого диэлектрического цилиндра и одного идеально проводящего цилиндра. В ней рассмотрено рассеяние на структуре, в которой осевые линии цилиндров лежат на одной прямой линии (структуре из соосных цилиндров).

1. Постановка задачи и ее решение

Рассматриваемая структура в декартовой системе координат *Oxyz* показана на рис. 1.

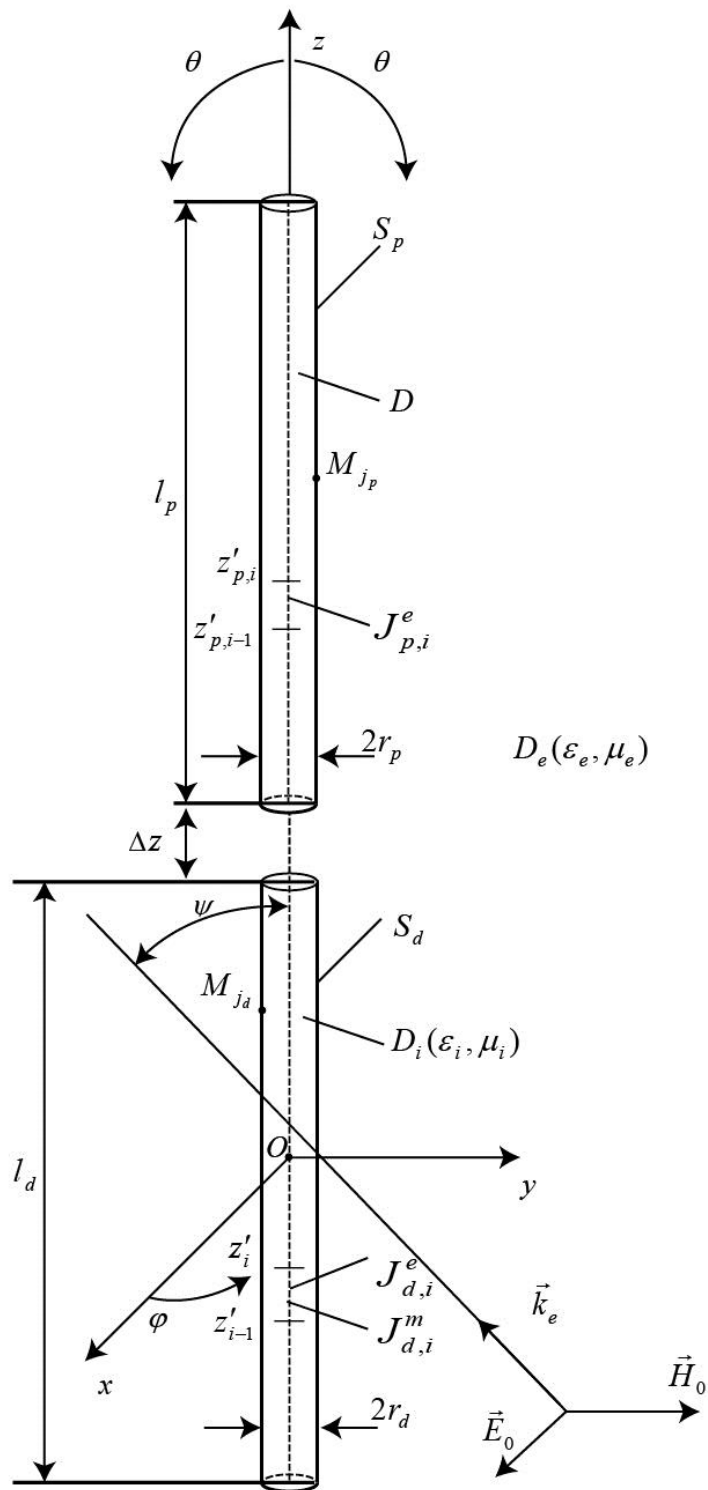


Рис. 1. Геометрия задачи

Осевые линии цилиндров ориентированы вдоль оси z , центр O системы координат расположен посередине осевой линии диэлектрического цилиндра. Диэлектрический цилиндр имеет длину l_d , радиус r_d и характеризуется электродинамическими параметрами ε_i, μ_i . Идеально проводящий цилиндр имеет длину l_p и радиус r_p . Для цилиндров выполняются условия тонкого цилиндра: $2r_d \ll \lambda$, $r_d \ll l_d$; $2r_p \ll \lambda$, $r_p \ll l_p$, где λ – длина падающей волны. Структура размещена в однородной среде D_e с электродинамическими параметрами ε_e, μ_e и возбуждается стационарным полем $\{\vec{E}_0, \vec{H}_0\}$ (зависимость от времени выбрана в виде $e^{-i\omega t}$). Требуется найти рассеянное поле $\{\vec{E}_e, \vec{H}_e\}$ в среде D_e .

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

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

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

граничным условиям

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

на поверхности S_d диэлектрического цилиндра и граничным условиям

$$[\vec{n}, \vec{E}_e] = -[\vec{n}, \vec{E}_0] \quad (4)$$

на поверхности S_p идеально проводящего цилиндра.

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

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

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; $R = \sqrt{x^2 + y^2 + z^2}$; $[\vec{a}, \vec{b}]$ – 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 электромагнитного рассеяния на структурах, содержащих один тонкий диэлектрический и один тонкий идеально проводящий цилиндр, предложен и подробно описан в работах [1,3]. Основные идеи этого метода следующие. Поле $\{\vec{E}_e, \vec{H}_e\}$ во внешней среде D_e представлено в виде суммы полей неизвестных вспомогательных токов, непрерывно распределенных вдоль осей цилиндров. Поле $\{\vec{E}_i, \vec{H}_i\}$ внутри диэлектрического цилиндра представлено в виде суммы полей пар элементарных электрических диполей с неизвестными дипольными моментами, дискретным образом расположенных на вспомогательной поверхности, которая охватывает диэлектрический цилиндр и также имеет 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 дипольные моменты.

Для этого первоначально осевая линия диэлектрического цилиндра разбивается на N_d одинаковых 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 $[z'_{i-1}, z'_i]$ и $[z'_{p,i-1}, z'_{p,i}]$, соответственно. Токи на этих

участках $J_{d,i}^e$, $J_{d,i}^m$ и $J_{p,i}^e$ называются «элементами токов». Далее 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 $\{\vec{E}_e, \vec{H}_e\}$ 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:

$$\begin{aligned} E_{e,\theta}(M) &= \sqrt{\frac{\mu_e}{\varepsilon_e}} H_{e,\varphi}(M) = \frac{e^{ik_e R}}{R} D_\theta(\theta) + O(R^{-2}), \\ E_{e,\varphi}(M) &= -\sqrt{\frac{\mu_e}{\varepsilon_e}} H_{e,\theta}(M) = \frac{e^{ik_e R}}{R} D_\varphi(\theta) + O(R^{-2}). \end{aligned} \quad (6)$$

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

$$\begin{aligned} D_\theta(\theta) &= -\frac{i\omega\mu_e}{4\pi} \sin\theta \left[\sum_{i=1}^{N_d} J_{d,i}^e \int_{z'_{i-1}}^{z'_i} e^{-ik_e z' \cos\theta} dz' + \sum_{i=1}^{N_p} J_{p,i}^e \int_{z'_{p,i-1}}^{z'_{p,i}} e^{-ik_e z' \cos\theta} dz'_p \right], \\ D_\varphi(\theta) &= \frac{ik_e}{4\pi} \sin\theta \sum_{i=1}^{N_d} J_{d,i}^m \int_{z'_{i-1}}^{z'_i} e^{-ik_e z' \cos\theta} dz'. \end{aligned} \quad (7)$$

В выражениях (7) угол θ отсчитывается от положительного направления оси z ; 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

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

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; $E_{e,z}^{j_p}, E_{0,z}^{j_p}$ – ориентированные вдоль оси z 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 $E_{e,z}^{j_p}$ использованы выражения (6) и (7) работы [1].

2. Численные результаты

Изложенные выше соотношения реализованы в виде программы для расчета компонент рассеянного поля. Входными величинами программы являются геометрические параметры структуры: длины цилиндров l_d и l_p , радиусы цилиндров r_d и r_p , относительные диэлектрическая и магнитная проницаемости диэлектрического цилиндра $\varepsilon'_i = \varepsilon_i / \varepsilon_e$ и $\mu'_i = \mu_i / \mu_e$, расстояние между цилиндрами Δz , а также параметры метода: числа разбиений осевых линий цилиндров N_d и N_p , количества точек коллокации L_d и L_p на поверхностях цилиндров и др. Подробная схема основных блоков программы пред-

ставлена в [4]. Программа была использована для определения области применимости метода, а также для исследования зависимости сечений обратного рассеяния структуры от угла падения плоской волны при различных значениях относительной диэлектрической проницаемости диэлектрического цилиндра.

Рис. 2 иллюстрирует область применимости метода. По оси абсцисс отложены значения радиусов цилиндров $k_e r_d = k_e r_p = k_e r$, по оси ординат – относительные (по отношению к возбуждающему полю) значения невязки граничных условий (выражение (8) в [2]). Кривая 1 относится к случаю, когда относительная диэлектрическая проницаемость ϵ'_i диэлектрического цилиндра равна 4, а кривая 2 – к случаю, когда она равна 20. При получении этих результатов длины цилиндров l_d и l_p выбраны равными 0.75λ , а расстояние Δz между цилиндрами выбрано равным 0.48λ , где λ – длина волны возбуждающего поля; числа разбиений осевых токов N_d и N_p выбраны одинаковыми и равными 30; количества точек коллокации на каждой из поверхностей цилиндров выбраны равными 120: $L_d = L_p = 120$.

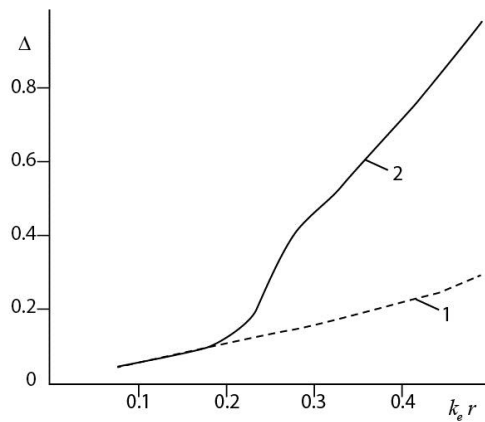


Рис. 2. Зависимость нормы невязки граничных условий от радиусов цилиндров при различных значениях относительной диэлектрической проницаемости цилиндра. Кривая 1 – при $\epsilon'_i = 4$, кривая 2 – при $\epsilon'_i = 20$.

Как показывают кривые, приведенные на рис. 2, область применимости метода сужается с ростом величины относительной диэлектрической проницаемости диэлектрического цилиндра. Этот результат совпадает с аналогичным результатом, полученным для параллельных цилиндров (см. рис. 5 в [2]). Из сравнения результатов, представленных на рис. 5 в [2], и результатов, представленных на рис. 2, следует, что в широком диапазоне значений ϵ'_i диэлектрического цилиндра предложенный нами в [1,3] метод решения задач электромагнитного рассеяния на структурах, состоящих из одного тонкого диэлектрического цилиндра и одного тонкого идеально проводящего цилиндра уверенно можно применять, если радиусы цилиндров $k_e r_d, k_e r_p \leq 0.1$.

На рис. 3 показаны полученные в результате выполненных расчетов зависимости сечений обратного рассеяния исследуемой структуры от угла ψ падения плоской волны при различных значениях относительной диэлектрической проницаемости диэлектрического цилиндра. Предполагается, что плоская волна падает на структуру таким образом, что векторы \vec{k}_e и \vec{E}_0 лежат в плоскости xz , угол ψ – это угол между направлением падения волны, характеризуемым волновым вектором \vec{k}_e , и осью z . Угол ψ отложен по оси абсцисс, а по оси ординат отложено сечение обратного рассеяния, нормированное на квадрат длин волн. Кривая 1 относится к случаю, когда относительная диэлектри-

ческая проницаемость диэлектрического цилиндра ε'_i равна 4, кривая 2 – к случаю, когда она равна 10, и кривая 3 – к случаю, когда $\varepsilon'_i = 20$. Во всех случаях предполагалось, что безразмерные длины цилиндров $k_e l_d$ и $k_e l_p$ равны 4.71, радиусы $k_e r_d$ и $k_e r_p$ равны 0.1, а расстояние между цилиндрами $k_e \Delta z = 3$. Числа разбиений осевых линий цилиндров выбраны равными 30, а числа точек коллокаций на поверхности каждого из цилиндров – 120.

Приведенные на рис. 3 кривые показывают, что значения сечений обратного рассеяния (при одном и том же значении угла ψ) существенно зависят от значения относительной диэлектрической проницаемости ε'_i диэлектрического цилиндра. Эта зависимость, вообще говоря, имеет сложный характер, но интересно отметить, что при углах падения ψ , больших 100° , просматривается тенденция увеличения сечения обратного рассеяния с ростом значения относительной диэлектрической проницаемости ε'_i диэлектрического цилиндра.

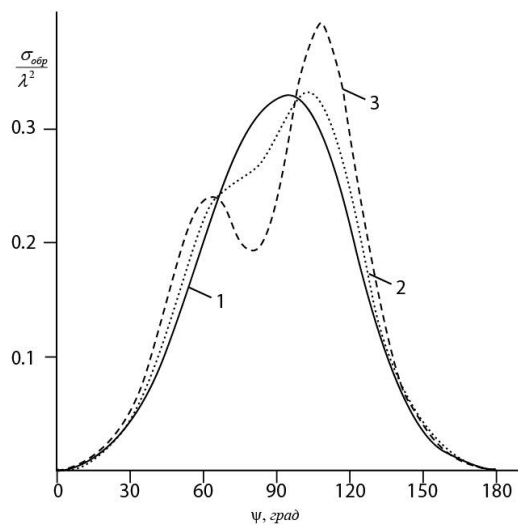


Рис. 3. Зависимости сечений обратного рассеяния от угла ψ падения плоской волны. Кривая 1 соответствует $\varepsilon'_i = 4$, кривая 2 – $\varepsilon'_i = 10$, кривая 3 – $\varepsilon'_i = 20$.

Заключение

Таким образом, в данной работе представлены результаты исследования электромагнитного рассеяния на тонких соосных идеально проводящем и диэлектрическом цилиндрах. Кратко описан метод решения поставленной задачи. Определена область применимости этого метода. Для одной из конкретных структур исследована зависимость сечений рассеяния от угла падения плоской волны при различных значениях относительной диэлектрической проницаемости диэлектрического цилиндра.

ЛИТЕРАТУРА

1. *Дмитренко А.Г., Балашова О.М.* Численный метод решения задачи электромагнитного рассеяния на тонких параллельных идеально проводящем и диэлектрическом цилиндрах // Материалы VI Международной молодежной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем». Томск, 24-26 мая 2018 г. – С. 55–61.
2. *Дмитренко А.Г.б Балашова О.М.* Программное обеспечение для решения задачи электромагнитного рассеяния на тонких параллельных идеально проводящем и диэлектрическом цилиндрах // Материалы VII Международной молодежной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем». Томск, 23-25 мая 2019 г. – С. 107–118.

3. Дмитренко А.Г., Балашова О.М. Моделирование электромагнитного рассеяния на тонких параллельных идеально проводящем и диэлектрическом цилиндрах // Вестник ТГУ. Управление, вычислительная техника и информатика. – 2020. – № 51. – С. 35–44.

4. Дмитренко А.Г., Балашова О.М. Алгоритм и программа расчета электромагнитного рассеяния на тонких ортогональных идеально проводящем и диэлектрическом цилиндрах // Материалы Международной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем». Томск, 28-30 мая 2020 г. – С. 90-98.

5. Дмитренко А.Г., Балашова О.М. Моделирование электромагнитного рассеяния на тонких ортогональных идеально проводящем и диэлектрическом цилиндрах // Материалы тринадцатой международной конференции «Новые информационные технологии в исследовании сложных структур». Томск, 7-9 сентября 2020 г. – С. 7–8.

DOI: 10.17223/978-5-907442-42-9-2021-14

ПЛОТНОСТЬ РАСПРЕДЕЛЕНИЯ ВЕРОЯТНОСТЕЙ В ОБОБЩЁННОМ МАР-ПОТОКЕ СОБЫТИЙ

Кеба А.В., Нежелская Л.А.

Томский государственный университет
mir.na.mig7@mail.ru, ludne@mail.ru

Введение

В настоящее время математические модели теории массового обслуживания (ТМО) широко используются для описания и анализа реальных физических, технических и иных процессов и систем. Интенсивное развитие компьютерных технологий и информационных сетей связи послужило толчком к формированию и развитию ключевых сфер приложений аппарата ТМО: телекоммуникационных, вычислительно-информационных, компьютерных, спутниковых сетей связи и т.п. Так, если рассмотреть входящий поток событий реальных систем, то можно выявить, что характеризующая их интенсивность подвержена случайным изменениям, моменты времени регистрации событий, в большинстве, носят случайный характер. Данные факторы становятся причиной рассмотрения в качестве математических моделей информационных потоков сообщений в реальных сетях и системах дважды стохастических потоков событий [1], которые подразделяются на два класса: 1) интенсивность потоков есть непрерывный случайный процесс [2,3]; 2) интенсивность потоков есть кусочно-постоянный случайный процесс с конечным числом состояний [4–6]. Отметим, что к потокам второго класса относится обобщённый МАР-поток событий с произвольным (конечным) числом состояний, который представляет собой адекватную математическую модель реальных потоков случайных событий.

В данной работе осуществляется исследование обобщённого МАР-потока событий с произвольным числом состояний при его полной наблюдаемости. Находится явный вид плотности вероятности [7–9] длительности интервала между моментами наступления соседних событий. Данная статья является непосредственным развитием работ [10–13].

1. Постановка задачи

Исследуется обобщённый МАР-поток событий с произвольным числом состояний (далее – поток), функционирующий в установившемся режиме (время функционирования потока стремится к бесконечности). Сопровождающий случайный процесс $\lambda(t)$ изучаемого потока представляет собой кусочно-постоянный принципиально ненаблюдаемый случайный процесс с n состояниями: S_1, S_2, \dots, S_n . Полагается, что при $\lambda(t) = \lambda_i$ имеет место i -е состояние (S_i), $i = \overline{1, n}$ процесса $\lambda(t)$. При этом $\lambda_1 > \lambda_2 > \dots > \lambda_n > 0$. Наблюдаемыми являются моменты времени наступления событий потока t_1, t_2, \dots .

Функция распределения случайной величины – длительности пребывания процесса $\lambda(t)$ в состоянии S_i – является экспоненциальной: $F_i(t) = 1 - e^{-\lambda_i t}$, $t \geq 0$, $i = \overline{1, n}$. В момент окончания состояния S_i процесс $\lambda(t)$ переходит из состояния S_i в состояние S_j с вероятностью $P_1(\lambda_j | \lambda_i)$ с наступлением события потока или с вероятностью $P_0(\lambda_j | \lambda_i)$ без наступления события потока, $i, j = \overline{1, n}$. Отметим, что для введённых вероятностей справедливо $\sum_{j=1}^n P_0(\lambda_j | \lambda_i) + \sum_{j=1}^n P_1(\lambda_j | \lambda_i) = 1$, $i = \overline{1, n}$.

Замечание 1. Введение вероятности $P_0(\lambda_i | \lambda_i) \neq 0$, $i = \overline{1, n}$, перехода процесса $\lambda(t)$ из состояния S_i в состояние S_i без наступления события приводит к обобщению классического МАР-потока с произвольным числом состояний.

Утверждение. Для обобщённого МАР-потока событий с n состояниями процесс $\lambda(t)$ является скрытым марковским процессом.

Блочная матрица инфинитезимальных характеристик процесса $\lambda(t)$ имеет вид $\mathbf{D} = \|\mathbf{D}_0 | \mathbf{D}_1\|$, где

$$\mathbf{D}_0 = \begin{vmatrix} -\lambda_1(1 - P_0(\lambda_1 | \lambda_1)) & \lambda_1 P_0(\lambda_2 | \lambda_1) & \dots & \lambda_1 P_0(\lambda_n | \lambda_1) \\ \lambda_2 P_0(\lambda_1 | \lambda_2) & -\lambda_2(1 - P_0(\lambda_2 | \lambda_2)) & \dots & \lambda_2 P_0(\lambda_n | \lambda_2) \\ \dots & \dots & \dots & \dots \\ \lambda_n P_0(\lambda_1 | \lambda_n) & \lambda_n P_0(\lambda_2 | \lambda_n) & \dots & -\lambda_n(1 - P_0(\lambda_n | \lambda_n)) \end{vmatrix}_{n \times n},$$

$$\mathbf{D}_1 = \begin{vmatrix} \lambda_1 P_1(\lambda_1 | \lambda_1) & \lambda_1 P_1(\lambda_2 | \lambda_1) & \dots & \lambda_1 P_1(\lambda_n | \lambda_1) \\ \lambda_2 P_1(\lambda_1 | \lambda_2) & \lambda_2 P_1(\lambda_2 | \lambda_2) & \dots & \lambda_2 P_1(\lambda_n | \lambda_2) \\ \dots & \dots & \dots & \dots \\ \lambda_n P_1(\lambda_1 | \lambda_n) & \lambda_n P_1(\lambda_2 | \lambda_n) & \dots & \lambda_n P_1(\lambda_n | \lambda_n) \end{vmatrix}_{n \times n}.$$

В качестве иллюстрации на рис. 1 приведена одна из реализаций процесса $\lambda(t)$ и наблюдаемого потока, где $\lambda_i(t)$ – значение процесса $\lambda(t)$ в состоянии S_i , $i = \overline{1, n}$; t_1, t_2, \dots – моменты времени наступления событий в наблюдаемом потоке.

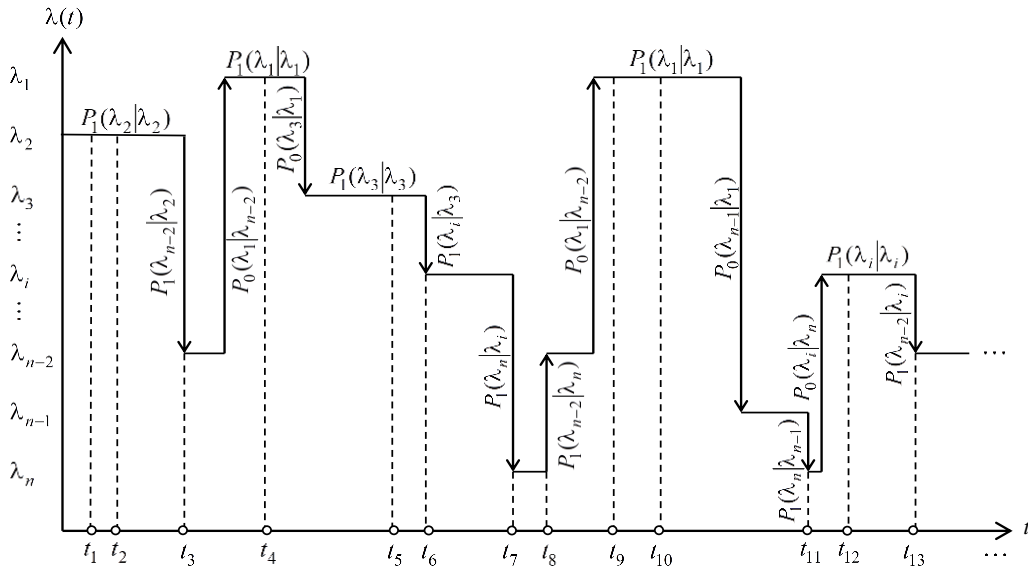


Рис. 1. Реализация обобщённого МАР-потока событий с произвольным числом состояний

Обозначим $\tau_k = t_{k+1} - t_k$, $\tau_k \geq 0$ – значение длительности k -го интервала между соседними событиями наблюдаемого потока t_k и t_{k+1} , $k = 1, 2, \dots$. Для плотности вероятности значений τ_k вследствие функционирования потока в стационарном режиме справедливо $p(\tau_k) = p(\tau)$, $\tau \geq 0$, при любом $k \geq 1$, что позволяет без ограничения общности положить момент наступления события t_k равным нулю или, что то же самое, момент наступления события есть $\tau = 0$.

Задача заключается в нахождении явного вида плотности распределения вероятностей длительности интервала между моментами наступления соседних событий в обобщённом МАР-потоке событий с двумя состояниями.

2. Вид плотности вероятности длительности интервала между соседними событиями в обобщённом МАР-потоке событий

Последовательность t_1, t_2, \dots образует вложенную по моментам времени наступления событий цепь Маркова $\{\lambda(t_k)\}$, поэтому плотность $p(\tau)$ имеет вид

$$p(\tau) = \sum_{i=1}^2 \pi_i(0) \sum_{j=1}^2 \tilde{p}_{ij}(\tau), \quad \tau \geq 0, \quad (1)$$

где $\pi_i(0)$ – условная стационарная вероятность того, что процесс $\lambda(\tau)$ в момент времени $\tau = 0$ находится в состоянии S_i при условии, что в момент времени $\tau = 0$ событие потока наступило, $i = 1, 2$; $\pi_1(0) + \pi_2(0) = 1$; пусть $p_{ij}(\tau)$ – условная вероятность того, что на интервале $(0, \tau)$ нет событий потока и в момент времени τ значение процесса $\lambda(\tau) = \lambda_j$ при условии, что в момент времени $\tau = 0$ значение процесса $\lambda(0) = \lambda_i$, и событие потока в момент времени $\tau = 0$ наступило, $i, j = 1, 2$; $\tilde{p}_{ij}(\tau)$ – соответствующая вероятности $p_{ij}(\tau)$ плотность вероятности, $i, j = 1, 2$.

Для введённых величин справедливы следующие леммы.

Лемма 1. Условные вероятности $p_{ij}(\tau)$, $i, j = 1, 2$, в обобщённом МАР-потоке событий имеют вид

$$\begin{aligned} p_{11}(\tau) &= \frac{1}{z_1 - z_2} \left\{ \left[\lambda_2 (1 - P_0(\lambda_2 | \lambda_2)) - z_1 \right] e^{-z_1 \tau} - \left[\lambda_2 (1 - P_0(\lambda_2 | \lambda_2)) - z_2 \right] e^{-z_2 \tau} \right\}, \\ p_{12}(\tau) &= \frac{\lambda_1 P_0(\lambda_2 | \lambda_1)}{z_2 - z_1} \left\{ e^{-z_1 \tau} - e^{-z_2 \tau} \right\}, \\ p_{22}(\tau) &= \frac{1}{z_1 - z_2} \left\{ \left[\lambda_1 (1 - P_0(\lambda_1 | \lambda_1)) - z_1 \right] e^{-z_1 \tau} - \left[\lambda_1 (1 - P_0(\lambda_1 | \lambda_1)) - z_2 \right] e^{-z_2 \tau} \right\}, \\ p_{21}(\tau) &= \frac{\lambda_2 P_0(\lambda_1 | \lambda_2)}{z_2 - z_1} \left\{ e^{-z_1 \tau} - e^{-z_2 \tau} \right\}, \\ z_{1,2} &= \frac{\lambda_1 (1 - P_0(\lambda_1 | \lambda_1)) + \lambda_2 (1 - P_0(\lambda_2 | \lambda_2)) \mp \sqrt{D}}{2}, \quad 0 < z_1 < z_2, \end{aligned} \quad (2)$$

$$D = \left[\lambda_1 (1 - P_0(\lambda_1 | \lambda_1)) - \lambda_2 (1 - P_0(\lambda_2 | \lambda_2)) \right]^2 + 4\lambda_1 \lambda_2 P_0(\lambda_2 | \lambda_1) P_0(\lambda_1 | \lambda_2).$$

Лемма 2. Плотности вероятностей $\tilde{p}_{ij}(\tau)$, $i, j = 1, 2$, в обобщённом МАР-потоке событий определяются формулами

$$\begin{aligned}
\tilde{p}_{11}(\tau) &= \lambda_1 P_1(\lambda_1 | \lambda_1) p_{11}(\tau) + \lambda_2 P_1(\lambda_1 | \lambda_2) p_{12}(\tau), \\
\tilde{p}_{12}(\tau) &= \lambda_1 P_1(\lambda_2 | \lambda_1) p_{11}(\tau) + \lambda_2 P_1(\lambda_2 | \lambda_2) p_{12}(\tau), \\
\tilde{p}_{21}(\tau) &= \lambda_1 P_1(\lambda_1 | \lambda_1) p_{21}(\tau) + \lambda_2 P_1(\lambda_1 | \lambda_2) p_{22}(\tau), \\
\tilde{p}_{22}(\tau) &= \lambda_1 P_1(\lambda_2 | \lambda_1) p_{21}(\tau) + \lambda_2 P_1(\lambda_2 | \lambda_2) p_{22}(\tau),
\end{aligned} \tag{3}$$

где $p_{ij}(\tau)$, $i, j = 1, 2$, определены в (2).

Лемма 3. Условные финальные вероятности $\pi_i(0)$, $i = 1, 2$, в обобщённом МАР-потоке событий задаются выражениями

$$\begin{aligned}
\pi_1(0) &= \frac{A}{A+B}, \quad \pi_2(0) = \frac{B}{A+B}, \quad \pi_1(0) + \pi_2(0) = 1, \\
A &= P_1(\lambda_1 | \lambda_2)(1 - P_0(\lambda_1 | \lambda_1)) + P_0(\lambda_1 | \lambda_2)P_1(\lambda_1 | \lambda_1), \\
B &= P_1(\lambda_2 | \lambda_1)(1 - P_0(\lambda_2 | \lambda_2)) + P_0(\lambda_2 | \lambda_1)P_1(\lambda_2 | \lambda_2).
\end{aligned} \tag{4}$$

Леммы 2 и 3 позволяют сформулировать следующую теорему.

Теорема. В обобщённом МАР-потоке плотность вероятности длительности интервала между соседними событиями имеет вид

$$\begin{aligned}
p(\tau) &= \gamma z_1 e^{-z_1 \tau} + (1 - \gamma) z_2 e^{-z_2 \tau}, \quad \tau \geq 0, \\
\gamma &= \frac{z_2 - \lambda_1 \pi_1(0) [P_1(\lambda_1 | \lambda_1) + P_1(\lambda_2 | \lambda_1)] - \lambda_2 \pi_2(0) [P_1(\lambda_1 | \lambda_2) + P_1(\lambda_2 | \lambda_2)]}{z_2 - z_1},
\end{aligned} \tag{5}$$

где z_i определены в (2), $\pi_i(0)$ – в (4), $i = 1, 2$.

Доказательство. Подставляя в (1) сначала (3), а затем явные выражения (4) для $\pi_i(0)$, $i = 1, 2$, проделывая необходимые преобразования, приходим к (5). Теорема доказана.

Замечание 2. Полагая в (5) вероятности $P_0(\lambda_j | \lambda_i) = 0$, $i, j = \overline{1, 2}$, $P_1(\lambda_2 | \lambda_1) = p$, $P_1(\lambda_1 | \lambda_2) = q$, приходим к виду плотности для синхронного потока [9], являющегося частным случаем рассматриваемого обобщённого МАР-потока событий:

$$\begin{aligned}
p(\tau) &= \gamma \lambda_1 e^{-\lambda_1 \tau} + (1 - \gamma) \lambda_2 e^{-\lambda_2 \tau}, \quad \tau \geq 0, \\
\gamma &= \frac{q}{p + q}, \quad \lambda_1 > \lambda_2, \quad 0 < p \leq 1, \quad 0 < q \leq 1.
\end{aligned}$$

Заключение

В данной статье получен явный вид (5) плотности вероятности $p(\tau)$, $\tau \geq 0$, длительности интервала между наступлениями соседних событий. Полученный результат можно использовать для решения задачи оценивания неизвестных параметров обобщённого МАР-потока событий с двумя состояниями.

ЛИТЕРАТУРА

1. Grandell J. Doubly stochastic Poisson processes. – Berlin Heidelberg: Springer-Verlag, 1976. – 240 p.
2. Cox D.R., Miller H.D. The theory of stochastic processes. – New York: Wiley, 1965. – 398 p.
3. Kingman J.F.C. On doubly stochastic Poisson process // Proceedings of the Cambridge Philosophical Society. – 1964. – V. 60. – Is. 4. – P. 923–930.
4. Башарин Г.П., Кокотушкин В.А., Наумов В.А. О методе эквивалентных замен расчёта фрагментов сетей связи. Ч. 1. // Изв. АН СССР. Техн. кибернетика. – 1979. – №6. – С. 92–99.
5. Башарин Г.П., Кокотушкин В.А., Наумов В.А. О методе эквивалентных замен расчёта фрагментов сетей связи Ч. 2. // Изв. АН СССР. Техн. кибернетика. – 1980. – № 1. – С. 55–61.
6. Neuts M.F. A versatile Markov point process // Journal of Applied Probability. – 1979. – V. 16. – P. 764–779.

7. Нежелская Л.А. Совместная плотность вероятностей длительности интервалов модулированного МАР-потока событий и условия рекуррентности потока // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2015. – № 1 (30). С. 57–67.

8. *Nezhel'skaya L.* Probability density function for modulated MAP event flows with unextendable dead time // Communications in Computer and Information Science. – 2015. – V. 564. – P. 141–151.

9. Горцев А.М., Нежелская Л.А. Оценивание параметров синхронного дважды стохастического потока событий методом моментов // Вестник Томского государственного университета. – 2002. – № S1-1. – С. 24–29.

10. Кеба А.В., Нежелская Л.А. Статистические эксперименты на имитационной модели обобщенного МАР-потока событий с произвольным числом состояний // Труды Томского государственного университета. Серия физико-математическая. – 2018. – Т. 302. – С. 157–164.

11. Кеба А.В., Нежелская Л.А. Апостериорные вероятности состояний обобщенного МАР-потока событий с произвольным числом состояний // Труды Томского государственного университета. Серия физико-математическая. – 2019. – Т. 304. – С. 23–32.

12. Кеба А.В., Нежелская Л.А. Статистические эксперименты на имитационной модели обобщенного МАР-потока событий с произвольным числом состояний в условиях непродлевающего мертвого времени // Новые информационные технологии в исследовании сложных структур : материалы Тринадцатой Международной конференции, 7–9 сентября 2020 г. – Томск, 2020. – С. 85–86.

13. *Nezhel'skaya L.A., Keba A.V.* Optimal state estimation of a generalized MAP event flow with an arbitrary number of states observability // Automation and Remote Control. – 2021. – V. 82. – No. 5. – P. 798–811.

DOI: 10.17223/978-5-907442-42-9-2021-15

СТАТИСТИЧЕСКИЕ ЭКСПЕРИМЕНТЫ НА ИМИТАЦИОННОЙ МОДЕЛИ ОБОБЩЕННОГО ПОЛУСИНХРОННОГО ПОТОКА СОБЫТИЙ С ПРОДЛЕВАЮЩИМСЯ МЕРТВЫМ ВРЕМЕНЕМ ФИКСИРОВАННОЙ ДЛИТЕЛЬНОСТИ

Нежелская Л.А., Степаненко И.Д.

Томский государственный университет

ludne@mail.ru, 97step@mail.ru

Введение

Системы массового обслуживания (СМО) являются широко распространенными математическими моделями реальных физических, технических, экономических, информационных систем и сетей. СМО включают: 1) множество входящих потоков событий различных типов; 2) множество обслуживающих приборов; 3) множество бункеров (накопителей, очередей); 4) дисциплины обслуживания [1].

В последние десятилетия в связи с интенсивным развитием компьютерной техники и информационных технологий возникла важная сфера приложений теории массового обслуживания – проектирование и создание информационно-вычислительных сетей, спутниковых сетей связи, телекоммуникационных сетей и т.п. Для реализации перечисленных задач классические математические модели входящих потоков событий (пуассоновские потоки) являются в той или иной степени непригодными для описания реальных информационных потоков сообщений [2]. На практике параметры, определяющие входящий поток событий, неизвестны или известны только частично, они изменяются со временем, при этом изменения часто носят случайный характер, что привело к рассмотрению дважды стохастических потоков событий [3–5]. Двойная стохастика характеризуется случайностью моментов времени наступления событий в потоке и представлением интенсивности потока как случайного процесса.

В данной работе исследуется первый элемент, определяющий СМО, в частности, обобщенный полусинхронный поток событий [6,7] при условии неполной наблюдаемости потока. Т.к. решение различных задач, в том числе оценка состояний и параметров дважды стохастического потока, может быть произведено только по наблюдаемым событиям, то наличие периода ненаблюдаемости (далее – мертвого времени) приводит к невозможности зарегистрировать все события потока и, как следствие, к ухудшению качества оценивания [8]. На практике это может быть обусловлено работой приборов, регистрирующих входящий поток. При регистрации прибором события возникает

мертвое время, порожденное зарегистрированным событием. Другие же события исходного потока, наступившие в течение периода мертвого времени, недоступны наблюдению.

Существует две группы приборов, регистрирующих события, для которых возможны ситуации: 1) непродлевающееся мертвое время регистрирующих приборов; 2) продлевающееся мертвое время регистрирующих приборов [9]. Принцип работы первой группы заключается в том, что события, наступившие в течение периода мертвого времени, недоступны наблюдению и не вызывают продление этого периода (непродлевающееся мертвое время); работа же 2-й группы обусловлена тем, что при наступлении события в период ненаблюдаемости потока это событие вызывает продление периода мертвого времени (продлевающееся мертвое время). Мертвое время может быть как фиксированным, так и случайным.

В настоящей работе рассматривается обобщенный полусинхронный поток с продлевающимся мертвым временем фиксированной длительности T (далее – поток). Построена имитационная модель данного потока, проведен ряд статистических экспериментов, подтверждающих работоспособность построенной модели.

1. Постановка задачи

Рассматривается дважды стохастический обобщенный полусинхронный поток событий, сопровождающий процесс которого есть кусочно-постоянный принципиально ненаблюдаемый случайный процесс $\lambda(t)$ с двумя состояниями: если $\lambda(t) = \lambda_1$, будем говорить, что имеет место первое состояние S_1 процесса $\lambda(t)$ (потока), если $\lambda(t) = \lambda_2$, то – второе состояние S_2 процесса $\lambda(t)$ (потока), где $\lambda_1 > \lambda_2 \geq 0$. В течение временного интервала, когда $\lambda(t) = \lambda_i$, имеет место пуассоновский поток событий с параметром λ_i , $i = 1, 2$. Переход из первого состояния процесса $\lambda(t)$ во второе возможен только в момент наступления события, при этом переход осуществляется с вероятностью p ($0 < p \leq 1$); с вероятностью $1 - p$ процесс $\lambda(t)$ остается в первом состоянии. Тогда длительность пребывания процесса $\lambda(t)$ в первом состоянии есть случайная величина с экспоненциальной функцией распределения $F_1(t) = 1 - e^{-p\lambda_1 t}$, $t \geq 0$ [6]. При этом длительность пребывания процесса $\lambda(t)$ во втором состоянии распределена по экспоненциальному закону $F_2(t) = 1 - e^{-\lambda_2 t}$, $t \geq 0$. Переход из второго состояния процесса $\lambda(t)$ в первое состояние может осуществляться в произвольный момент времени. При переходе процесса $\lambda(t)$ из второго состояния в первое инициируется с вероятностью δ ($0 \leq \delta \leq 1$) дополнительное событие в первом состоянии (т.е. сначала осуществляется переход, а затем инициируется дополнительное событие). Очевидно, что в сделанных предположениях $\lambda(t)$ – марковский процесс.

Каждое зарегистрированное в потоке событие порождает период ненаблюдаемости фиксированной длительности T (мертвое время), другие события, произошедшие в этот период, недоступны наблюдению (теряются). Хотя события и не наблюдаются в течение мертвого времени, каждое из них вызывает продление периода ненаблюдаемости на ту же величину T . Следующее событие, наступившее после окончания последнего периода ненаблюдаемости, снова регистрируется и порождает новый период мертвого времени длительности T и т.д.. Таким образом, общий период ненаблюдаемости является случайной величиной.

На рис. 1 представлен пример одной из возможных реализаций процесса $\lambda(t)$ и наблюдаемого потока, где 1 и 2 соответствуют значениям случайного процесса $\lambda(t)$ в состояниях S_i , $i = 1, 2$; черными кружками отмечены ненаблюдаемые события потока; ξ

– значения периодов ненаблюдаемости потока; t_1, t_2, \dots – моменты наступления событий в наблюдаемом потоке.

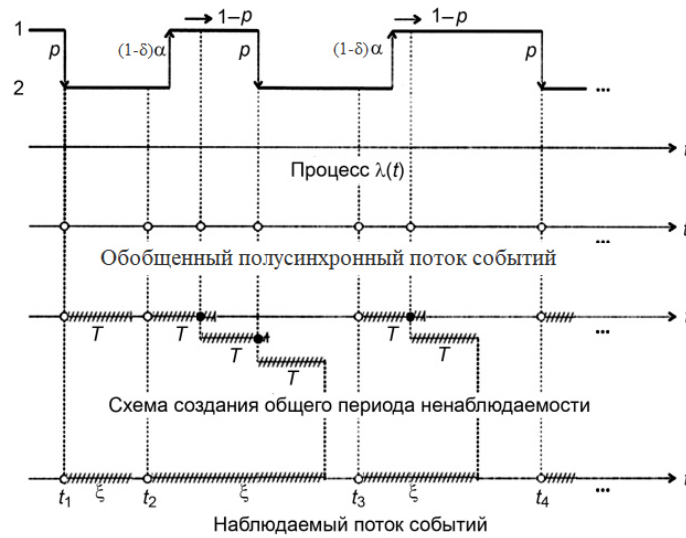


Рис. 1. Реализация обобщенного полусинхронного потока с продлевающимся мертвым временем фиксированной длительности T

Задача состоит в построении имитационной модели обобщенного полусинхронного потока с продлевающимся мертвым временем фиксированной длительности и проверке работоспособности модели путём проведения ряда статистических экспериментов.

2. Имитационное моделирование методом обратных функций

Одним из эффективных методов исследования, при котором строится модель рассматриваемой системы, является имитационное моделирование [10]. Имитационная модель описывает с достаточной степенью точности изучаемую систему и на данной основе проводятся опыты в течение задаваемого периода времени моделирования с целью получения информации об исходной системе. Смысловая нагрузка метода имитационного моделирования заключается в разработке (на базе ЭВМ) программ, реализующих имитацию различных поведений изучаемой системы, её свойств и характеристик в необходимом объёме. Программы позволяют по заданным значениям параметров и состоянию системы в данный момент времени получить характеристики системы, необходимые для дальнейшего применения в решении практических задач. Проведение статистических опытов позволяет исследовать исходную систему и выявить зависимость получаемых статистических характеристик от исходных параметров.

Имитационная модель исследуемого обобщенного полусинхронного потока событий с двумя состояниями в условиях продлевающегося мёртвого времени построена с использованием метода обратных функций. Пусть $t > 0$ – значение длительности интервала между событиями в i -м состоянии процесса $\lambda(t)$. Случайная величина t имеет функцию распределения $F_i(t) = 1 - e^{-\lambda_i t}$, $i = 1, 2$. В соответствии с методом обратных функций обозначим $F_i = x$, тогда равенство вида $t = -\frac{1}{\lambda_i} \ln(1 - x)$ является основой для

моделирования значения длительности интервалов между событиями потока в i -м состоянии, где x – равномерно распределенная на интервале $(0, 1)$ случайная величина, λ_i – параметр пуассоновского потока в i -м состоянии процесса $\lambda(t)$, $i = 1, 2$. Аналогично моделируется значение длительности пребывания процесса $\lambda(t)$ во втором состоянии.

В результате построена имитационная модель обобщенного полусинхронного потока событий с двумя состояниями в условиях продлевающегося мёртвого времени в виде программного кода на языке программирования C# в среде Visual Studio 2019 и поставлен ряд статистических экспериментов с целью установления работоспособности имитационной модели.

3. Результаты статистических экспериментов

В каждом эксперименте производится расчет четырех характеристик потока по соответствующим формулам; при этом используемое ниже обозначение T_m есть время моделирования реализаций потока.

1. Выборочное среднее периода ненаблюдаемости: $\hat{M}(\xi) = \frac{1}{N} \sum_{k=1}^N \hat{\xi}^{(k)}$,

$\hat{\xi}^{(k)} = \frac{1}{i^{(k)}} \sum_{j=1}^{i^{(k)}} \xi_j^{(k)}$, где k – номер реализации; $i^{(k)}$ – количество наблюдаемых событий в реализации (количество периодов ненаблюдаемости); $\xi_j^{(k)}$ – значения периодов ненаблюдаемости после наблюдаемых событий в реализации; N – количество реализаций.

2. Выборочное среднее длительности интервала между наблюдаемыми событиями: $\hat{M}(\tau) = \frac{1}{N} \sum_{k=1}^N \hat{\tau}^{(k)}$, $\hat{\tau}^{(k)} = \frac{1}{n^{(k)}} \sum_{j=1}^{n^{(k)}} \tau_j^{(k)}$, где k – номер реализации; $n^{(k)}$ – количество интервалов между наблюдаемыми событиями; $\tau_j^{(k)} = t_{j+1}^{(k)} - t_j^{(k)}$ – значения длительностей интервалов между соседними наблюдаемыми событиями в k -й реализации; N – количество реализаций.

3. Выборочная дисперсия периода ненаблюдаемости: $\hat{D}(\xi) = \frac{1}{N-1} \sum_{k=1}^N (\hat{\xi}^{(k)} - \hat{M}(\xi))^2$.

4. Процентное отношение числа наблюдаемых событий ко всем событиям потока: $P = \frac{100}{N} \sum_{k=1}^N \frac{i^{(k)}}{i^{(k)} + j^{(k)}}$, где $j^{(k)}$ – количество ненаблюдаемых событий в k -й реализации.

Эксперимент 1. Определение стационарного режима.

Целью данного эксперимента является определение стационарного режима для заданных параметров потока. Для эксперимента 1 фиксируются следующие параметры:

- $p = 0.5$ – вероятность перехода процесса $\lambda(t)$ из 1-го состояния во 2-е состояние;
- $T = 1$ – значение длительности мертвого времени;
- $\alpha = 0.5$ – интенсивность перехода процесса $\lambda(t)$ из 2-го состояния в 1-е состояние;
- $\delta = 0.5$ – вероятность инициирования дополнительного события в момент перехода процесса $\lambda(t)$ из 2-го состояния в 1-е состояние в 1-м состоянии.

Для эксперимента 1 используются следующие наборы параметров пуассоновского потока: 1) $\lambda_1 = 1$, $\lambda_2 = 0.5$; 2) $\lambda_1 = 5$, $\lambda_2 = 1$.

Таблица 1
Результаты эксперимента 1
для первого набора параметров

T_m	$\hat{M}(\xi)$	$\hat{M}(\tau)$	$\hat{D}(\xi)$	P
100	1.6971	2.7480	0.0217	39.380
200	1.6486	2.7072	0.0141	39.754
300	1.6595	2.7315	0.0092	39.369
400	1.6515	2.7182	0.0054	39.683
500	1.6389	2.7135	0.0042	39.976
...
1500	1.6486	2.7148	0.0014	39.644
1600	1.6456	2.7025	0.0014	39.972
1700	1.6436	2.7121	0.0014	39.828
1800	1.6422	2.7093	0.0012	39.921
1900	1.6389	2.7024	0.0011	39.924

Таблица 2
Результаты эксперимента 1
для второго набора параметров

T_m	$\hat{M}(\xi)$	$\hat{M}(\tau)$	$\hat{D}(\xi)$	P
100	2.6371	3.3622	0.1226	16.049
200	2.6355	3.3271	0.0631	16.0057
300	2.6535	3.3321	0.0449	16.057
400	2.6323	3.3118	0.0275	16.259
500	2.6586	3.3363	0.0231	16.002
...
1500	2.6483	3.3122	0.0079	16.144
1600	2.6339	3.2977	0.0062	16.148
1700	2.6432	3.3072	0.0085	16.111
1800	2.6537	3.3172	0.0067	16.086
1900	2.6296	3.2919	0.0046	16.218

На рис. 2–5 представлены результаты эксперимента 1 для первого набора параметров: $\lambda_1 = 1$, $\lambda_2 = 0.5$.

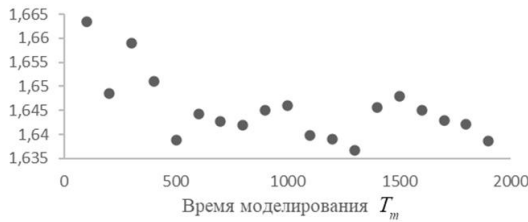


Рис. 2. Оценка значения периода ненаблюдаемости

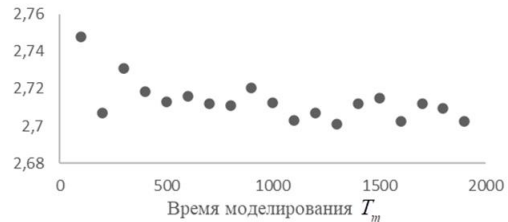


Рис. 3. Оценка значения длительности интервала между соседними наблюдаемыми событиями

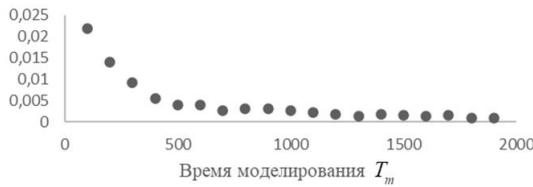


Рис. 3. Оценка значения выборочной дисперсии периода ненаблюдаемости

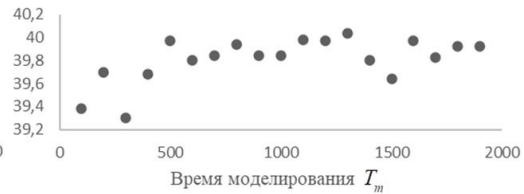


Рис. 4. Процент отношения наблюдаемых событий ко всем событиям, произошедшим в потоке

На рис. 6–9 представлены результаты эксперимента 1 для второго набора параметров: $\lambda_1 = 5$, $\lambda_2 = 1$.

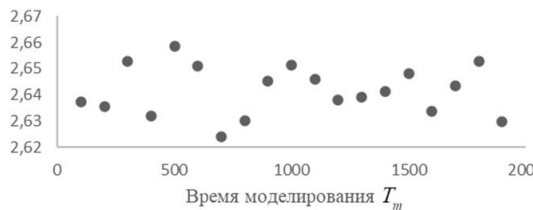


Рис. 6. Оценка значения периода ненаблюдаемости

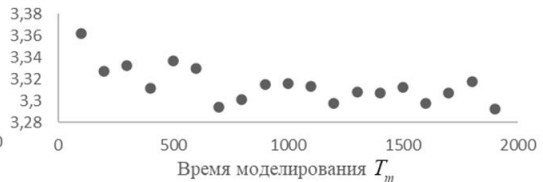


Рис. 7. Оценка значения длительности интервала между соседними наблюдаемыми событиями

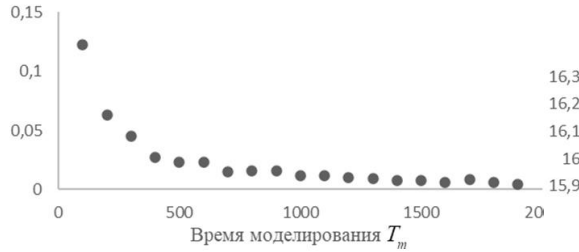


Рис. 8. Оценка значения выборочной дисперсии периода ненаблюдаемости

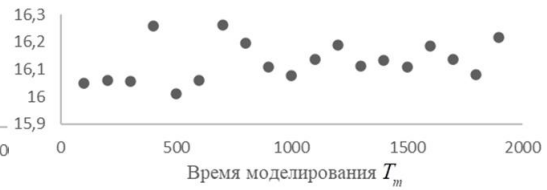


Рис. 9. Процент отношения наблюдаемых событий ко всем событиям, произошедшим в потоке

Вывод. Для данных наборов параметров стационарный режим достигается для значений времени моделирования $T_m \geq 1300$. Значения выборочной дисперсии периода ненаблюдаемости для первого набора параметров (рис. 4) меньше приблизительно в 5 раз по отношению ко второму набору параметров (рис. 8). Это показывает, что вычисляемые характеристики при фиксированных значениях времени моделирования зависят от задаваемых параметров потока.

Эксперимент 2. Изменение вероятности перехода процесса $\lambda(t)$ из 1-го состояния во 2-е состояние.

Цель данного эксперимента заключается в проверке утверждения: «чем больше вероятность перехода p , тем меньше значение периода ненаблюдаемости».

Для эксперимента 2 фиксируются следующие параметры:

- $T_m = 1300$ — время моделирования;
- $T = 1$ — значение длительности мертвого времени;
- $\alpha = 0.5$ — интенсивность перехода процесса $\lambda(t)$ из 2-го состояния в 1-е состояние;
- $\delta = 0.5$ — вероятность инициирования дополнительного события в момент перехода процесса $\lambda(t)$ из 2-го состояния в 1-е состояние в 1-м состоянии.

Для эксперимента 2 используются следующие наборы параметров потока: 1) $\lambda_1 = 1, \lambda_2 = 0.5$; 2) $\lambda_1 = 5, \lambda_2 = 1$.

Таблица 3

Результаты эксперимента 2 для первого набора параметров

p	$\hat{M}(\xi)$	$\hat{M}(\tau)$	$\hat{D}(\xi)$	P
0.1	1.677	2.722	0.0013	38.228
0.2	1.646	2.718	0.0016	39.553
0.3	1.624	2.712	0.0014	40.542
0.4	1.601	2.712	0.0015	41.562
0.5	1.595	2.718	0.0016	41.956
0.6	1.579	2.711	0.0012	42.609
0.7	1.570	2.712	0.0012	43.040
0.8	1.567	2.721	0.0014	43.318
0.9	1.554	2.716	0.0010	43.850

Таблица 4

Результаты эксперимента 2 для второго набора параметров

p	$\hat{M}(\xi)$	$\hat{M}(\tau)$	$\hat{D}(\xi)$	P
0.1	4.485	5.129	0.0473	6.219
0.2	3.376	4.026	0.0163	9.890
0.3	2.984	3.645	0.0100	12.490
0.4	2.772	3.435	0.0089	14.540
0.5	2.627	3.292	0.0083	16.170
0.6	2.564	3.229	0.0066	17.270
0.7	2.497	3.158	0.0049	18.400
0.8	2.444	3.117	0.0058	19.280
0.9	2.431	3.102	0.0049	19.800

На рис. 10–13 представлены результаты эксперимента 2 для первого набора параметров: $\lambda_1 = 1, \lambda_2 = 0.5$.

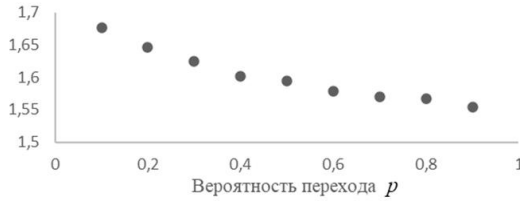


Рис. 10. Оценка значения периода ненаблюдаемости

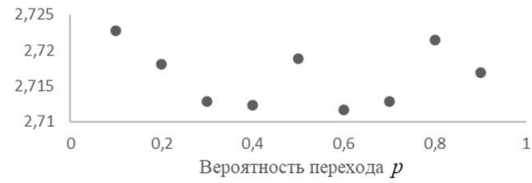


Рис. 11. Оценка значения длительности интервала между соседними наблюдаемыми событиями

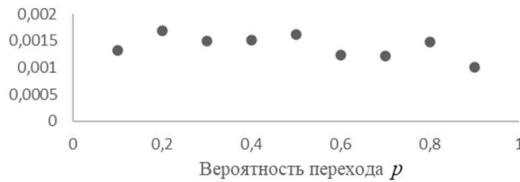


Рис. 12. Оценка значения выборочной дисперсии периода ненаблюдаемости

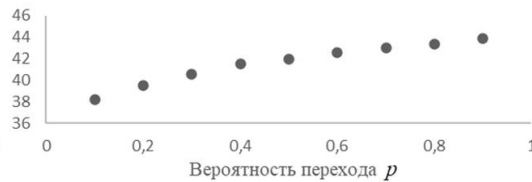


Рис. 13. Процент отношения наблюдаемых событий ко всем событиям, произошедшим в потоке

На рис. 14–17 представлены результаты эксперимента 2 для второго набора параметров: $\lambda_1 = 5$, $\lambda_2 = 1$.

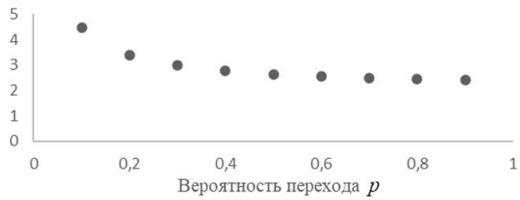


Рис. 14. Оценка значения периода ненаблюдаемости



Рис. 15. Оценка значения длительности интервала между соседними наблюдаемыми событиями

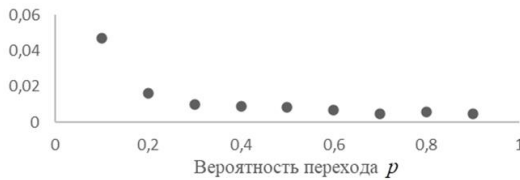


Рис. 16. Оценка значения выборочной дисперсии периода ненаблюдаемости

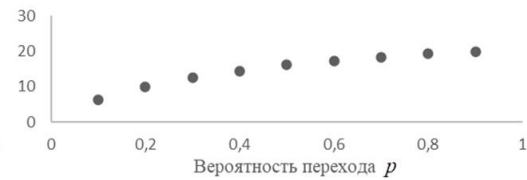


Рис. 17. Процент отношения наблюдаемых событий ко всем событиям, произошедшим в потоке

Вывод. В данном эксперименте для второго набора параметров получаемые характеристики оказываются более стабильными по сравнению с характеристиками для первого набора параметров. Оба набора удовлетворяют поставленной цели эксперимента, что демонстрируют результаты, представленные на рис. 10–17. Такое поведение, в первую очередь, связано с определением исследуемого потока. Т.к. $\lambda_1 > \lambda_2 \geq 0$, то очевидно, что с увеличением вероятности p поток больше времени будет находиться во втором состоянии, вследствие чего уменьшается общее количество событий, что в значительной степени приводит к увеличению процента наблюдаемых событий.

Заключение

В данной работе рассмотрен обобщенный полусинхронный поток событий с продлевающимся мертвым временем фиксированной длительности. В результате построена имитационная модель исследуемого потока в виде программного кода на языке программирования C# в среде Visual Studio 2019.

Анализ полученных результатов статистических экспериментов позволяет утверждать, что полученная работоспособная модель, не противоречащая соответствующим входным данным.

ЛИТЕРАТУРА

1. Горцев А.М., Назаров А.А., Тертугов А.Ф. Управление и адаптация в системах массового обслуживания. – Томск: Издательство Томского университета, 1978. – 208 с.
2. Вишневецкий В.М., Дудин А.Н., Клименок В.И. Стохастические системы с коррелированными потоками. Теория и применение в телекоммуникационных сетях. – М.: Техносфера, 2018. – 564 с.
3. Башарин Г.П., Кокотушкин В.А., Наумов В.А. О методе эквивалентных замен расчёта фрагментов сетей связи. Ч. 1 // Изв. АН СССР. Техн. кибернетика. – 1979. – № 6. – С. 92–99.
4. Башарин Г.П., Кокотушкин В.А., Наумов В.А. О методе эквивалентных замен расчёта фрагментов сетей связи. Ч. 2 // Изв. АН СССР. Техн. кибернетика. – 1980. – № 1. – С. 55–61.
5. Neuts M.F. A versatile Markov point process // Journal of Applied Probability. – 1979. – V. 16. – P. 764–779.
6. Нежелская Л.А. Оценка состояний дважды стохастических потоков событий. – Томск: Издательство Томского государственного университета, 2020. – 210 с.
7. Горцев А.М., Калягин А.А., Нежелская Л.А. Оптимальная оценка состояний обобщенного полусинхронного потока событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2010. – № 2.
8. Nezhel'skaya L. Probability density function for modulated MAP event flows with unextendable dead time // Communications in Computer and Information Science. – 2015. – V. 564. – P. 141–151.
9. Апанасович В.В., Коляда А.А., Чернявский А.Ф. Статистический анализ случайных потоков в физическом эксперименте. – Минск: Университетское, 1988. – 256 с.
10. Лифшиц А.Л., Мальц Э.А. Статистическое моделирование систем массового обслуживания. – М.: Сов. радио, 1978. – 248 с.

DOI: 10.17223/978-5-907442-42-9-2021-16

СТАТИСТИЧЕСКИЕ ЭКСПЕРИМЕНТЫ ПО ОЦЕНИВАНИЮ ПАРАМЕТРА РАСПРЕДЕЛЕНИЯ ДЛИТЕЛЬНОСТИ СЛУЧАЙНОГО МЁРТВОГО ВРЕМЕНИ В АЛЬТЕРНИРУЮЩЕМ АСИНХРОННОМ ПОТОКЕ СОБЫТИЙ

Горцев А.М., Шманкеева А.В.
Томский государственный университет
shmankeeva@yandex.ru

Введение

Распространёнными математическими моделями физических явлений и процессов являются потоки событий. В частности, такие модели применяются при исследовании информационных потоков сообщений в телекоммуникационных системах, в спутниковых сетях связи и т.п. [1] В связи с интенсивным развитием компьютерных сетей модель простейшего потока событий перестала быть адекватной реальным информационным потокам. Требования практики послужили стимулом к рассмотрению дважды стохастических потоков [2,3] в качестве математической модели реальных потоков событий в компьютерных сетях. В большинстве публикаций авторы рассматривают математические модели потоков событий, когда события потока доступны наблюдению. Однако на практике возникают ситуации, когда наступившее событие влечёт за собой ненаблюдаемость последующих событий. Причиной ненаблюдаемости, как правило, выступает мёртвое время регистрирующих приборов [4], в течение которого другие события, наступившие в этот период, теряются. Регистрирующие приборы при этом делятся

на два вида: с непродлевающимся мёртвым временем и продлевающимся. При этом длительность мёртвого времени может быть как детерминированной величиной, так и случайной. Задачи по оценке параметров и состояний потока событий в условиях мёртвого времени фиксированной длительности рассматривались в [5–11].

Однако достаточно открытым остаётся вопрос изучения потоков событий, когда мёртвое время является случайной величиной с тем или иным законом распределения. Здесь отметим работу [12], в которой решается задача оценки параметра распределения непродлевающегося случайного мёртвого времени в пуассоновском потоке и работе [13], в которой находятся формулы для начальных моментов общего периода ненаблюдательности в пуассоновском потоке событий при продлеваемомся случайном мёртвом времени.

1. Математическая модель наблюдаемого потока

Рассматривается альтернирующий асинхронный дважды стохастический поток событий, сопровождающий процесс (интенсивность) которого есть кусочно-постоянный стационарный случайный процесс $\lambda(t)$ с двумя состояниями S_1 и S_2 . Будем говорить, что имеет место первое состояние процесса (потока) S_1 , если $\lambda(t) = \lambda$, и, наоборот, имеет место второе состояние процесса (потока) S_2 , если $\lambda(t) = 0$. Если имеет место первое состояние процесса S_1 (второе состояние процесса S_2), то в течение временного интервала, когда $\lambda(t) = \lambda$ (когда $\lambda(t) = 0$), поступает пуассоновский поток событий с интенсивностью λ (события потока не наступают). Переход из состояния S_1 (из состояния S_2) процесса $\lambda(t)$ в состояние S_2 (в состояние S_1) может осуществляться в произвольный момент времени, не связанный с моментом наступления события потока (свойство асинхронных потоков). При этом длительность пребывания процесса $\lambda(t)$ в состоянии S_i , $i=1,2$, есть случайная величина, распределенная по экспоненциальному закону $F_i(t) = 1 - e^{-\alpha_i t}$, $t \geq 0$, где α_1 (α_2) – интенсивность смены состояния S_1 на S_2 (S_2 на S_1). В сделанных предположениях $\lambda(t)$ – скрытый марковский процесс ($\lambda(t)$ – принципиально ненаблюдаемый процесс).

После каждого зарегистрированного события в момент времени t_k наступает период мёртвого времени случайной длительности, который порождается этим событием, так что другие события исходного потока, наступившие в течение этого периода мёртвого времени недоступны наблюдению и не вызывают его продления (непродлеваемое мёртвое время). Принимается, что случайная длительность мёртвого времени распределена по равномерному закону с плотностью вероятности $p(T) = 1/T^*$, где T – значение длительности мёртвого времени, $0 < T \leq T^*$.

На рис. 1 приведён вариант возникающей ситуации, где S_1, S_2 – состояния случайного процесса $\lambda(t)$; временная ось $(0, T)$ – ось наступления наблюдаемых событий в моменты времени t_1, t_2, \dots ; временная ось $(0, t^{(1)})$ – ось наступления наблюдаемых событий в моменты времени $t_1^{(1)}, t_2^{(1)}, \dots$ в первом (S_1) состоянии процесса $\lambda(t)$, на которой также указаны значения длительностей $T_1^{(1)}, T_2^{(1)}, \dots$ мёртвых времён, порождённых наблюдаемыми событиями потока; белыми кружками обозначены наблюдаемые события, чёрными – ненаблюдаемые, штриховкой – значения периодов мёртвого времени; траектория процесса $\lambda(t)$ привязана к осям $(0, t^{(1)})$ и $(0, t^{(2)})$.

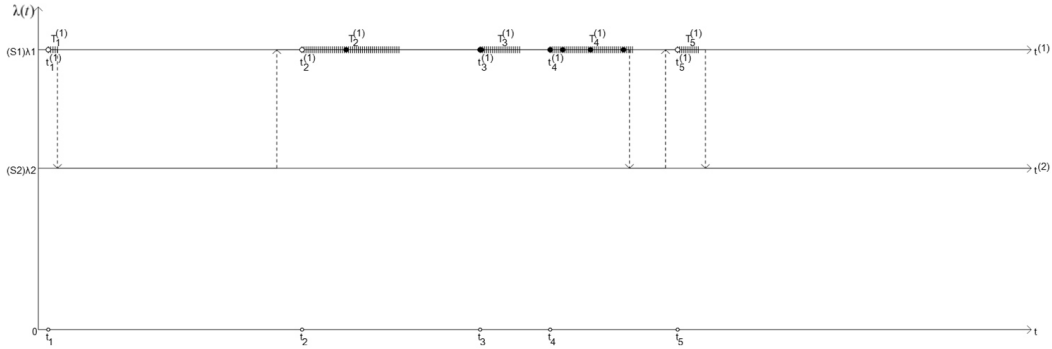


Рис. 1. Альтернирующий асинхронный поток событий при непродлеваемом случайном мёртвом времени

Целью работы является разработка алгоритма оценивания параметра T^* длительности непродлеваемого случайного мёртвого времени в альтернирующем асинхронном потоке событий.

2. Уравнение моментов для оценивание параметра T^*

Оценивание параметра T^* происходит на основе выборки $\{\tau_1, \tau_2, \dots, \tau_k\}$, $\tau_k = t_{k+1} - t_k$, $k \geq 1$, где t_k – моменты наступления событий наблюдаемого потока, функционирующего в стационарном режиме. Для оценки параметра используется метод моментов [14]. Найдём теоретический момент – математическое ожидание случайной величины τ (длительности интервала между двумя соседними событиями наблюдаемого потока) $M(\tau | T^*)$. Построим уравнение моментов $M(\tau | T^*) = C_1$, где $C_1 = \frac{1}{n} \sum_{k=1}^n (t_{k+1} - t_k)$ – оценка теоретического среднего $M(\tau | T^*)$. Из этого уравнения находится оценка \hat{T}^* .

Для нахождения теоретического среднего $M(\tau | T^*)$ воспользуемся формулой $M(\tau | T^*) = \int_0^{\infty} p(\tau) d\tau$. В этой формуле плотность $p(\tau)$ определяется в виде:

$$p(\tau) = \int_{(T)} p(\tau, T) d\tau = \int_{(T)} p(T) p(\tau | T) dT, \text{ где } (T) \text{ — область изменения величины } T.$$

Для случая, когда мёртвое время является детерминированной величиной длительности T , воспользуемся формулой из [15]:

$$p(\tau | T) = \begin{cases} 0, & 0 \leq \tau < T, \\ \gamma(T) z_1 e^{-z_1(\tau-T)} + (1-\gamma(T)) z_2 e^{-z_2(\tau-T)}, & \tau \geq T, \end{cases} \quad (1)$$

где $z_{1,2} = \frac{1}{2} \left[(\lambda_1 + \lambda_2 + \alpha_1 + \alpha_2) \mp \sqrt{(\lambda_1 - \lambda_2 + \alpha_1 - \alpha_2)^2 + 4\alpha_1\alpha_2} \right]$, $z_1 < z_2$, $z_1 > 0$,

$$\gamma(T) = \frac{z_2 - \lambda_1}{(z_2 - z_1)(\alpha_1 + \alpha_2)} \left\{ z_2 - \lambda_2 + \frac{\alpha_2(\lambda_1 - \lambda_2)(z_1 - \lambda_1)e^{-(\alpha_1 + \alpha_2)T}}{\lambda_1\alpha_2 + \lambda_2\alpha_1 + \lambda_1\lambda_2[1 - e^{-(\alpha_1 + \alpha_2)T}]} \right\}. \text{ Тогда плотность } p(\tau)$$

будет находиться по формуле

$$p(\tau) = \begin{cases} p_1(\tau) = \int_0^{\tau} p(T)p(\tau|T)dT, & 0 \leq \tau < T^*, \\ p_2(\tau) = \int_0^{T^*} p(T)p(\tau|T)dT, & \tau \geq T^*. \end{cases} \quad (2)$$

Т.к. во втором состоянии события не наступают, то положим в (1) $\lambda_1 = \lambda$, $\lambda_2 = 0$. Подставляя выражение (1) в (2) и учитывая, что $p(T) = 1/T^*$, получаем

$$p_1(\tau) = \frac{1}{T^*} \left(1 + e^{-(\alpha_1 + \alpha_2)\tau} - e^{-z_1\tau} - e^{-z_2\tau} \right), \quad 0 \leq \tau < T^*, \quad (3)$$

$$p_2(\tau) = \frac{1}{T^* (z_2 - z_1)(\alpha_1 + \alpha_2)} \left(e^{-z_1\tau} \left[z_2 (z_2 - \lambda) e^{z_1 T^*} - (z_2 - z_1)(\alpha_1 + \alpha_2) \right] - e^{-z_1\tau} \times \right. \\ \times e^{(\lambda - z_2)T^*} z_1 (z_1 - \lambda) + e^{-z_2\tau} \left[-z_1 (z_1 - \lambda) e^{z_2 T^*} - (z_2 - z_1)(\alpha_1 + \alpha_2) \right] + e^{-z_2\tau} \times \\ \left. \times e^{(\lambda - z_1)T^*} z_2 (z_2 - \lambda) \right), \quad \tau \geq T^*, \quad (4)$$

$$\text{где } z_{1,2} = \frac{1}{2} \left[(\lambda + \alpha_1 + \alpha_2) \mp \sqrt{(\lambda + \alpha_1 - \alpha_2)^2 + 4\alpha_1\alpha_2} \right].$$

Выполнение двух условий $p_1(\tau = T^*) = p_2(\tau = T^*)$ и $p_1'(\tau = T^*) \neq p_2'(\tau = T^*)$ показывает, что функция (2) непрерывна и имеет излом в точке $\tau = T^*$.

Учитывая (2), получим формулу для математического ожидания длительности интервала между соседними событиями наблюдаемого потока $M(\tau|T^*)$:

$$M(\tau|T^*) = \int_0^{T^*} \tau p_1(\tau) d\tau + \int_{T^*}^{\infty} \tau p_2(\tau) d\tau. \text{ Подставляя (3) и (4) в последнюю формулу, получим}$$

$$M(\tau|T^*) = \frac{T^*}{2} + \frac{\lambda\alpha_1 + (\alpha_1 + \alpha_2)^2}{\lambda\alpha_2(\alpha_1 + \alpha_2)} - \frac{\alpha_1}{T^*\alpha_2(\alpha_1 + \alpha_2)^2} + \frac{\alpha_1}{T^*\alpha_2(\alpha_1 + \alpha_2)^2} e^{-(\alpha_1 + \alpha_2)T^*}, \quad T^* \geq 0. \quad (5)$$

Аналитически доказано, что математическое ожидание длительности интервала между соседними событиями $M(\tau|T^*)$ является возрастающей функцией переменной T^* , отсюда следует, что уравнение моментов $M(\tau|T^*) = C_1$ имеет единственное решение. Т.к. решение уравнения моментов единственно, то получение численного решения особых трудностей не представляет [16]. Для установления качества получаемой оценки параметра T^* проведены статистические эксперименты с применением имитационного моделирования.

3. Численные результаты

Численные результаты получены с использованием имитационной модели исследуемого потока при заданных значениях параметров альтернирующего асинхронного потока $\lambda = 1.5$, $\alpha_1 = 0.1$, $\alpha_2 = 0.1$ с заданной точностью $\varepsilon = 0.0001$ поиска решения уравнения моментов методом простой итерации [16] и заданным шагом $\Delta T^* = 0.001$. Получено $N = 100$ реализаций потока и, следовательно, всего 100 решений уравнения моментов для $T^* = 1, 2, 3, 4, 5$ при фиксированном значении времени моделирования $T_m = 1000$. На основании этих данных вычислены выборочное среднее искомой оценки

$$\hat{M}(\hat{T}^*) = \frac{1}{N} \sum_{i=1}^N \hat{T}_i^* \text{ и её выборочная вариация } \text{Var}(\hat{T}^*) = \frac{1}{N} \sum_{i=1}^N (\hat{T}_i^* - T^*)^2, \text{ где } \hat{T}_i^* - \text{реше-}$$

ние уравнение моментов для i -й реализации потока, T^* – значение оцениваемого параметра, известное из имитационной модели потока.

В табл. 1 приведены численные результаты для $\hat{M}(\hat{T}^*)$, где указано выборочное среднее $\hat{M}(\hat{T}^*)$ для случаев $\lambda = 1$ и $\lambda = 5$ при $T^* = 1, 2, \dots, 5$.

Таблица 1

**Оценка математического ожидания длительности интервала между соседними событиями
при $\lambda = 1$, $\alpha_1 = 0.1$, $\alpha_2 = 0.1$, $T^* = 1, 2, \dots, 5$, $T_m = 1000$**

T^*		1	2	3	4	5
$\hat{M}(\hat{T}^*)$	$\lambda = 1$	0.89	1.87	2.91	3.93	4.95
	$\lambda = 5$	0.97	1.94	2.93	3.95	4.97

На рис. 2 и 3 приведены графики зависимости $\hat{M}(\hat{T}^*)$ от T^* для $\lambda = 1$ и $\lambda = 5$, построенные по данным табл. 1.

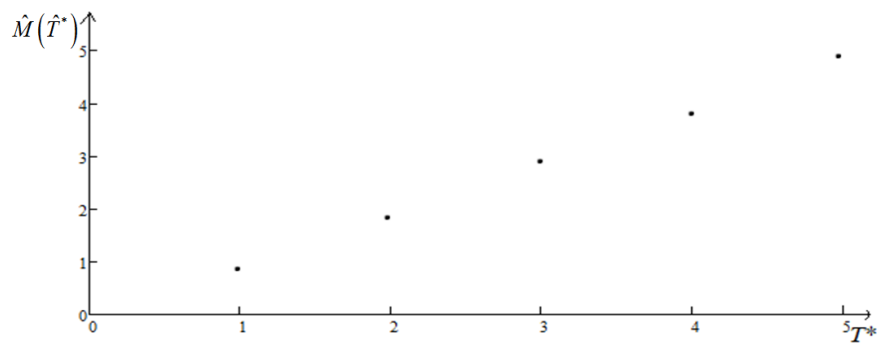


Рис. 2. График зависимости $\hat{M}(\hat{T}^*)$ от T^* для $\lambda = 1$

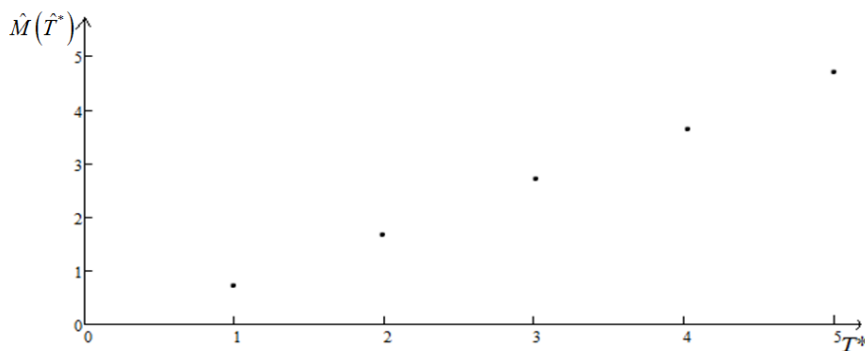


Рис. 3. График зависимости $\hat{M}(\hat{T}^*)$ от T^* для $\lambda = 5$

По данным табл. 1 и рис. 2 и 3 можно сделать следующий вывод: оценка \hat{T}^* – смещённая, т.е. $\hat{M}(\hat{T}^*) < T^*$. Причиной смещения оценки \hat{T}^* относительно известного из имитационной модели T^* служит то, что значение теоретического среднего случайной величины T есть $T^*/2$; в силу этого значения T сосредоточены слева от T^* .

В табл. 2 приведены численные результаты для $\text{Var}(\hat{T}^*)$, где указана выборочная вариация $\text{Var}(\hat{T}^*)$ для случаев $\lambda = 1$ и $\lambda = 5$ при $T^* = 1, 2, \dots, 5$.

Таблица 2

Значение выборочной вариации
при $\lambda = 1$, $\alpha_1 = 0.1$, $\alpha_2 = 0.1$, $T^* = 1, 2, \dots, 5$, $T_m = 1000$

T^*		1	2	3	4	5
$\text{Var}(\hat{T}^*)$	$\lambda = 1$	$7.52 \cdot 10^{-3}$	0.0011	0.0016	0.0019	0.0024
	$\lambda = 5$	$4.658 \cdot 10^{-3}$	0.0080	0.0013	0.0016	0.0020

На рис. 4 и 5 приведены графики зависимости $\text{Var}(\hat{T}^*)$ от T^* для $\lambda = 1$ и $\lambda = 5$, построенные по данным табл. 2.

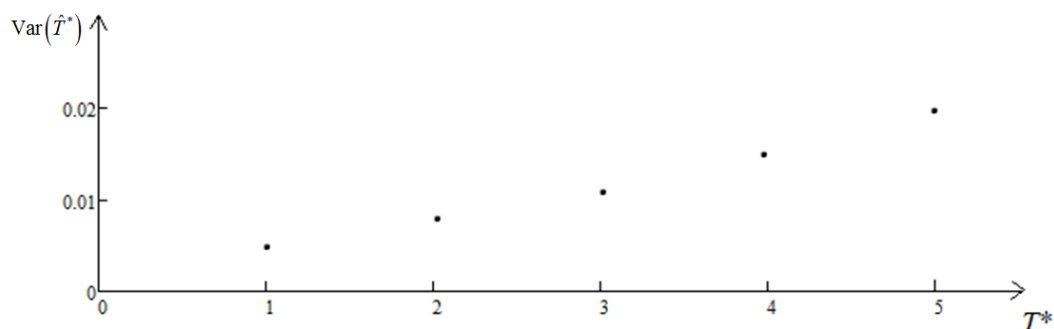


Рис. 4. График зависимости $\text{Var}(\hat{T}^*)$ от T^* для $\lambda = 1$

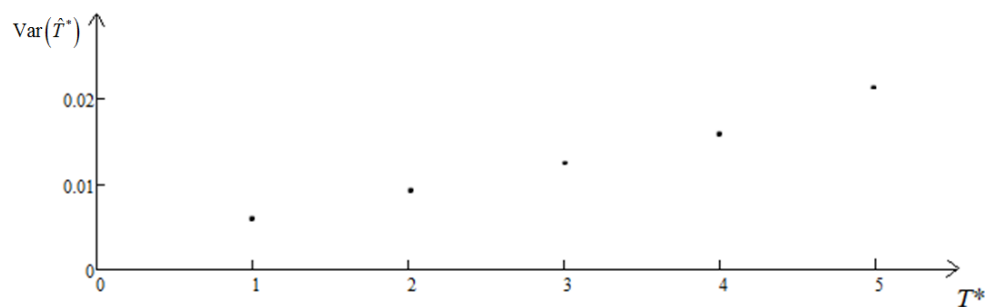


Рис. 5. График зависимости $\text{Var}(\hat{T}^*)$ от T^* для $\lambda = 5$

По данным табл. 2 и рис. 4 и 5 можно сделать следующий вывод: оценка \hat{T}^* – смещённая, т.е. значения \hat{T}^* не стремятся к параметру T^* , но при этом выборочная вариация $\text{Var}(\hat{T}^*)$ стремится к числу, близкому к 0, а, следовательно, можно сказать, что метод оценивания достаточно удовлетворительный и полученную оценку можно принимать за истинную с небольшой погрешностью.

Отметим, что выборочная вариация для $\lambda = 5$ больше, чем для $\lambda = 1$. Причиной этому является то, что при увеличении T^* увеличивается период ненаблюдаемости потока, следовательно, происходит увеличение потерь событий исходного потока, что влечёт за собой ухудшение качества оценивания при одинаковых параметрах.

Заключение

Таким образом, в данной статье рассмотрен альтернирующий асинхронный поток событий с непродлевающимся случайным мёртвым временем. Аналитически получены формулы (3) и (4) для плотности вероятности значений длительности интервала между соседними событиями. Выведено математическое ожидание (5) длительности интервала между соседними событиями. Доказано, что математическое ожидание является возрастающей функцией, и поэтому уравнение моментов имеет единственное решение.

Решение уравнения моментов найдено с использованием численного метода простой итерации, которое является оценкой параметра равномерного распределения длительности случайного мёртвого времени. Проведены численные эксперименты, которые указывают на адекватность и хорошее качество найденной оценки в смысле введённого критерия (выборочной вариации оценки).

ЛИТЕРАТУРА

1. Дудин А.Н., Клименок В.И. Системы массового обслуживания с коррелированными потоками. – Минск: Изд-во БГУ, 2000. – 175 с.
2. Башарин Г.П., Кокотушкин В.А., Наумов В.А. О методе эквивалентных замен расчёта фрагментов сетей связи. Ч. 1 // Изв. АН СССР. Техн. кибернетика. – 1979. – № 6. – С. 92–99.
3. Neuts M.F. A versatile Markovian point process // Journal of Applied Probability. – 1979. – V. 16. – P. 764–779.
4. Апанасович В.В., Коляда А.А., Чернявский А.Ф. Статистический анализ случайных потоков в физическом эксперименте. – Минск: Университетское, 1988. – 256 с.
5. Горцев А.М., Калягин А.А., Нежелская Л.А. Оценка максимального правдоподобия длительности мёртвого времени в обобщённом полусинхронном потоке // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2015. – № 1 (30). – С. 27–37.
6. Горцев А.М., Леонова М.А., Нежелская Л.А. Совместная плотность вероятностей длительности интервалов обобщённого асинхронного потока событий при непродлеваемом мёртвом времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2012. – № 4 (21). – С. 14–25.
7. Горцев А.М., Соловьёв А.А. Совместная плотность вероятностей длительности интервалов потока физических событий при непродлеваемом мёртвом времени // Известия высших учебных заведений. Физика. – 2014. – Т. 57. – № 7. – С. 103–111.
8. Горцев А.М., Василевская Т.П., Нежелская Л.А. Оценивание длительности мёртвого времени и параметров синхронного альтернирующего потока с проявлением либо не проявлением событий // Вестник Томского государственного университета. – 2004. – № S9-2. – С. 129–138.
9. Нежелская Л.А. Оптимальное оценивание состояний полусинхронного потока событий в условиях его частичной наблюдаемости // Вестник Томского государственного университета. – 2000. – № 269. – С. 95–98.
10. Леонова М.А., Нежелская Л.А. Вероятность ошибки при оценивании состояний обобщённого асинхронного потока событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2012. – № 2 (19). – С. 88–101.
11. Горцев А.М., Колягин А.А., Нежелская Л.А. Совместная плотность вероятностей длительности интервалов обобщённого полусинхронного потока событий при непродлеваемом мёртвом времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2014. – № 2 (27). – С. 19–29.
12. Горцев А.М., Загородняя М.Е. Оценивание параметра непродлевающегося мёртвого времени случайной длительности в пуассоновском потоке событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2017. – № 40. – С. 32–40.
13. Глухова Е.В., Тертугов А.Ф. Оценка интенсивности пуассоновского потока событий при наличии продлевающегося мёртвого времени // Известия высших учебных заведений. Физика. – 1995. – Т. 38. – № 3. – С. 22–31.
14. Малинковский Ю.В. Теория вероятностей и математическая статистика (Часть 2. Математическая статистика). – Гомель: УО «ГТУ им. Ф. Скорины», 2004. – 146 с.
15. Васильева Л.А. Оценивание длительности мёртвого времени и параметров асинхронного дважды стохастического потока событий методом моментов // Автореферат диссертации на соискание учёной степени кандидата технических наук. 2005. – 47 с.
16. Амосов А.А., Дубинский Ю.А., Конченко Н.В. Вычислительные методы для инженеров. – М.: Высшая школа, 1994. – 544 с.

V. ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ

DOI: 10.17223/978-5-907442-42-9-2021-17

МОДЕЛИРОВАНИЕ ЭФФЕКТА МИКРОМОБИЛЬНОСТИ В БЕСПРОВОДНЫХ ЛОКАЛЬНЫХ СЕТЯХ ТЕРАГЕРЦОВОГО ДИАПАЗОНА*

Голос Е.С.¹, Острикова Д.Ю.¹, Молчанов Д.А.², Гайдамака Ю.В.¹

¹Российский университет дружбы народов; ²Университет Тампере
goloselizaveta@gmail.com, d.ostrikova@gmail.com, dmitri.moltcanov@tuni.fi, gaydamaka-yuv@rudn.ru

Введение

Несмотря на то, что запуск первых коммерческих сетей 5G в мире состоялся только в апреле 2019 г., ведущие страны и компании-разработчики технологий 5G уже объявили о планах по разработке технологий мобильной связи шестого поколения – 6G, действующей в терагерцовом диапазоне частот [1]. Ожидается, что будут реализованы совершенно новые приложения и услуги. Однако чтобы удовлетворить требования этих приложений, 6G необходимо радикально повысить скорость доступа сотовых технологий [2]. Помимо высокой скорости доступа, БС ТГц диапазона частот должны будут обеспечивать возможность обслуживать гораздо больше устройств, чем нынешние сети и сети 5G [3]. Однако использование диапазона чрезвычайно высоких частот (ТНЧ) приводит к высоким потерям, и, более того, каналы ТГц частот будут подвержены динамической блокировке и микромобильности [4].

Для того, чтобы предотвратить серьезные потери при распространении, будут использованы колоссальные антенные решетки с тысячами элементов как на стороне базовой станции, так и на стороне пользовательского устройства. Эти решетки будут способны создавать диаграммы направленности излучения антенны с шириной луча всего несколько градусов или даже меньше [5].

В данной работе были проанализированы такие характеристики, как вероятность отсутствия связи и эргодическая спектральная эффективность для различных схем переключения: схема моментального переключения и схема поиска луча с учетом микромобильности в зависимости от числа базовых станций и интенсивности блокировки человеческими телами. Была рассмотрена стандартизованная операция мультисвязности 3GPP, когда ПУ одновременно поддерживают связь с несколькими БС в ТГц [6]. В этом исследовании мы фиксируем характеристики распространения в ТГц диапазоне и диаграммы направленности антенны, а также различные конструкции механизмов поиска луча и установления соединения.

1. Математическая модель

В данной работе расположение базовых станций ТГц диапазона частот соответствует точечному пуассоновскому процессу (PPP) в Re^2 с плотностью λ_A . Высота базовой станции (БС) постоянна и равна h_A . Рассматривается одно пользовательское устройство (ПУ) в случайно выбранной точке. Следует отметить, что целевое ПУ остается неподвижным на протяжении всей своей работы. Высота ПУ постоянна и равна h_{ij}

Люди в пешеходной зоне вокруг ПУ действуют как потенциальные блокираторы с плотностью λ_B в Re^2 . Данные блокираторы моделируются как цилиндры с радиусом r_B . Предполагаем, что рост людей постоянный и равен h_B , причем $h_B > h_{ij}$. Для того, чтобы оценить динамику блокировки ТГц LoS, считаем, что люди движутся в соответствии с

* Публикация выполнена при поддержке Программы стратегического академического лидерства РУДН и при финансовой поддержке РФФИ в рамках научных проектов № 19-07-00933 и № 20 07 01064.

моделью случайного направления (RDM). Согласно этому, пользователь сначала случайным образом определяет направление движения равномерно в $(0, 2\pi)$, а затем движется в выбранном направлении с постоянной скоростью v м/с в течение экспоненциально распределенного времени с параметром μ .

Будем считать, что в состоянии блокировки LoS связь невозможна, т.о., значение SINR на ПУ в неблокированном состоянии LoS может быть записано как [**Ошибка! Источник ссылки не найден.**]

$$S(x) = P_T G_A G_U \left[\frac{x^{-\xi_T} e^{-Kx}}{N_0 + M_T} \right].$$

Таблица 1

Данные для численного эксперимента

Обозначение	Значение	Описание
P_T	2	излучаемая мощность передающей антенны
G_A	$10^{0.1 \cdot 17.58}$	коэффициент усиления передающей антенны
G_U	$10^{0.1 \cdot 5.57}$	коэффициент усиления принимающей антенны
ξ_T	2.1	коэффициент затухания сигнала
N_0	$10^{0.1 \cdot (-84)}$	шум
M_T	$10^{0.1 \cdot 3}$	граница затухания сигнала
λ_A	0.001	плотность расположения бс
h_A	4	высота бс
h_B	1.7	высота блокатора
h_U	1.5	высота пу
r_B	0.4	радиус блокатора

Чтобы оценить влияние мультисвязности на производительность ПУ, мы предполагаем, что выбранное ПУ может поддерживать N соединений с соседними ТГц БС. Рассмотрим два способа построения соединений.

1.1. Схема с мгновенным переключением

Данный метод заключается в том, что процедура поиска луча выполняется в каждом кадре, отражающем сотовую структуру системы связи ТГц диапазона. Это означает, что время переключения является мгновенным. При установлении сессии ПУ выбирает свою ближайшую БС. ПУ меняет свою точку доступа всякий раз, когда оно входит в состояние отключения с текущей БС. Как в моменты установления сессии, так и в моменты повторного подключения выбирается ближайшая неблокируемая БС из ближайшего N . В данной работе N равняется двум, т.е. ПУ связано с двумя ближайшими базовыми станциями и испытывает перебои в работе только тогда, когда обе они находятся в состояниях блокировки.

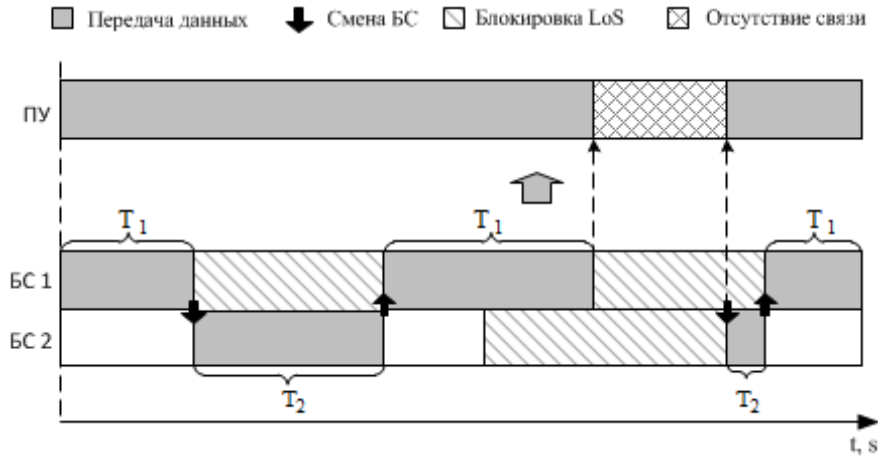


Рис. 1. Схема с мгновенным переключением

Плотность распределения вероятностей до i -й БС равна $f_i(x) = \frac{2(\pi\lambda_A)^i}{(i-1)!} x^{2i-1} e^{-\pi\lambda_A x^2}$, $x > 0$, $i = \overline{1, N}$. Таким образом, емкость ПУ, связанного с i -й БС, равна $E[C_i] = \int_0^{\infty} \log_2(1 + S(x)) f_i(x) dx$. Вероятность блокировки i -й БС ТГц диапазона равна

$$p_{N,i} = 1 - \int_0^{\infty} \frac{2(\pi\lambda_A)^i}{(i-1)!} x^{2i-1} e^{-\pi\lambda_A x^2} e^{-2\pi f_B \lambda_B \frac{h_B - h_U}{h_A - h_U}} dx.$$

В данной работе считаем, что блокировки LoS на БС ТГц независимы друг от друга, вероятность блокировки для N ближайших БС может быть вычислена как $p_N = \prod_{i=1}^N p_{N,i}$. Доля времени, в течение которой сессия остается активной, равна $p_{\tau_i} = (1 - p_{N,i}) \prod_{k=1}^{i-1} p_{N,k}$. Таким образом, средняя спектральная эффективность считается как $C = \sum_{i=1}^N T_i E[C_i]$.

1.2. Схема с поиском луча

Теперь рассмотрим схему с процедурой поиска луча, отражающую стиль проектирования WLAN в системе связи ТГц. В данной системе перебои могут быть вызваны как блокировкой, так и микромобильностью. Для представления процесса микромобильности в ПУ мы используем модель, представленную в [Ошибка! Источник ссылки не найден.]. Согласно ей, микромобильность моделируется как комбинация случайных процессов перемещения по осям $x(t)$ и $y(t)$ вместе со случайными процессами вращения по этим плоскостям, $\varphi(t)$ и $\theta(t)$.

С практической точки зрения, выравнивание лучей требуется, когда происходит их расхождение. Чтобы учесть этот принцип, необходимо расширить подход с идеальным поиском луча, принимая во внимание ненулевое время переключения T_B на другую БС путем введения дополнительной компоненты. T_B рассчитывается по формуле $T_B = (N_U + N_T) \delta$, где N_U и N_T – количество конфигураций антенных решеток, соответственно на ПУ и ТГц БС, δ – время однократного переключения решетки. Следует обратить внимание, что на практике T_B зависит от многих факторов, включая используемые

алгоритмы поиска луча, размеры антенных решеток и время, необходимое для оценки расположения лучей. При переключении ПУ с одной БС на другую, схема подключения будет выглядеть следующим образом: учитывается интервал подключения T_L и интервал отключения T_{NL} . Отметим, что T_{NL} интерпретируется как время, в течение которого нет ни одной БС в незаблокированном состоянии. В свою очередь, T_L начинается, когда ПУ связывается с ближайшей незаблокированной БС в первый раз после T_{NL} и заканчивается, когда нет БС в незаблокированном состоянии. Стоит отметить, что поскольку $T_L \gg T_B$, то предполагается, что процедура поиска луча не может быть прервана блокировкой LoS.

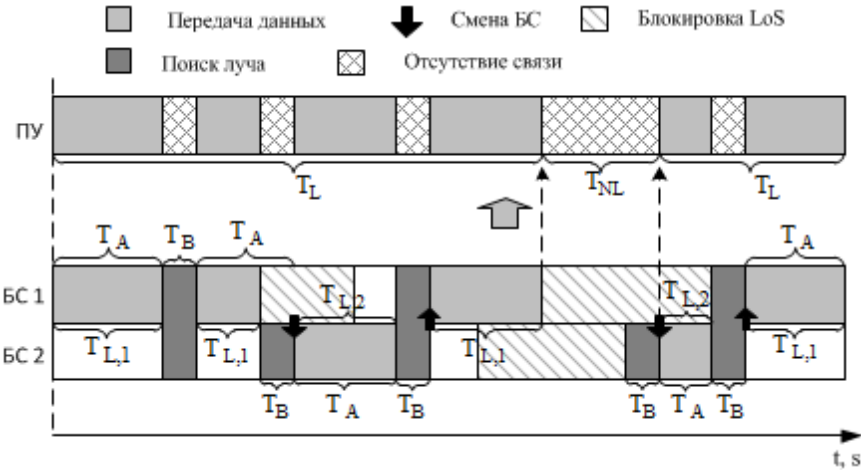


Рис. 2. Схема с поиском луча при учете микроподвижности

Рис. 2 иллюстрирует рассматриваемую схему для случая $N = 2$ БС, описывая возможные события и состояния, определенные моделью. Блокировка LoS и микроподвижность образуют периоды простоя, которые заканчиваются процедурой поиска луча. Таким образом, период простоя состоит из периода, когда все N БС находятся в заблокированном состоянии, T_{NL} , включая время, необходимое для поиска луча, T_B . В качестве альтернативы он может состоять только из процедуры поиска луча, если есть хотя бы одна БС в незаблокированном состоянии. Переключение БС происходит только в результате процедуры поиска луча, которая устанавливает новое активное соединение с ближайшей БС в незаблокированном состоянии LoS.

Детальной рассмотрим интервал подключения T_L , определяемый как временной интервал, когда из всех N доступных БС всегда есть БС в незаблокированном состоянии, с которыми ПУ может связываться. Теперь мы можем представить модель процесса переключения между БС с использованием поглощающей цепи Маркова, показанной на рис. 3, где номер состояния i представляет ближайшую БС, в то время как $j = N + 1, 2N$ состояний обозначают процедуру поиска луча из-за блокировки, а $j = 2N + 1, 3N$ состояний представляют поиск луча из-за несовпадения луча. Состояние поглощения А обозначает конец периода подключения и начало интервала отключения, когда все БС заблокированы. Также учитывается состояние 0, в которое мы попадаем в случае возникновения микроподвижности.

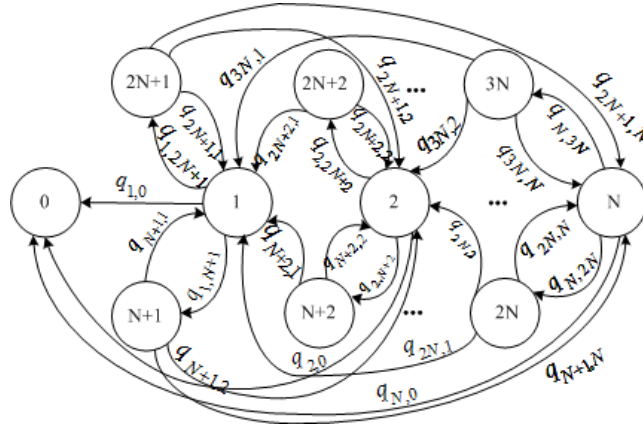


Рис. 3. Поглощающая цепь Маркова с учетом микромобильности

В соответствии с [5], [7], время между последовательными событиями блокировки распределено приблизительно экспоненциально. Таким образом, процесс событий блокировки с i -й БС является однородным пуассоновским с интенсивностью $\mu_{B,i}$:

$$\mu_{B,i} = \int_0^{\infty} f_i(x) \frac{2r_B \lambda_B v (x[h_B - h_U] + r_B [h_A - h_U])}{h_A - h_U} dx.$$

Это означает, что плотность распределения вероятностей и функция распределения неблокированного периода LoS, когда он связан с i -й БС, представлены как $f_{T_{L,i}}(t) = \mu_{B,i} e^{-\mu_{B,i} t}$ и $F_{T_{L,i}}(t) = 1 - e^{-\mu_{B,i} t}$.

Пусть $F_{B,i}(t)$ – функция распределения заблокированного периода с i -й БС. Как показано в [9], распределение заблокированного интервала совпадает с периодом занятости в системе M/GI/inf,

$$F_{B,i}(t) = 1 - \left(\int_0^t [1 - F_{B,i}(t-z)] de^{-\mu_{B,i} F_T(z)} + [1 - F_T(t)] \left[1 - \int_0^t [1 - F_{B,i}(t-z)] e^{-\mu_{B,i} F_T(z)} \mu_{B,i} dz \right] \right),$$

где $F_T(t)$ – это функция распределения времени, необходимого одному блокатору для пересечения зоны блокировки LoS, выраженная ступенчатой функцией Хевисайда

$$F_T(t) = H\left(t - \frac{2r_B}{v}\right).$$

Предполагая независимость процесса блокировки на базовых станциях, функция распределения периода блокировки T_{NL} может быть выражена в виде

$$F_{T_{NL,i}}(t) = 1 - [1 - F_{B,i}(t)] \prod_{j=1, j \neq i}^N [1 - F_{B_j^*}(t)],$$

где B_i и B_j^* – случайные величины, обозначающие периоды блокировки и остаточной блокировки j -й БС, соответственно, в то время как $F_{B,i}(x)$ и $F_{B_j^*}(x)$ являются соответствующими ФР. $F_{B_j^*}(x)$ можно рассчитать следующим образом:

$$F_{B_j^*}(t) = \int_0^{\infty} \frac{F_{B,i}(t+\tau) - F_{B,i}(\tau)}{\tau [1 - F_{B,i}(\tau)]} d\tau, \quad j = \overline{1, N}, \quad j \neq i.$$

Далее переходим к расчету среднего времени блокировки LoS:

$$E[T_{NL}] = T_B + \sum_{i=1}^N p_{\tau_i} \int_0^{\infty} t f_{T_{NL,i}}(t) dt.$$

Вновь вернемся к интервалу T_L , определяемому как временной интервал, когда из всех N доступных БС всегда есть БС в неблокированном состоянии, с которыми ПУ может связываться. Ранее описанная марковская цепь, характеризуется матрицей переходных вероятностей $Q = [q_{ij}]$, $i, j = \overline{0, 3N}$. Элементы $Q = [q_{ij}]$ могут быть выражены как

$$q_{i, N+i} = (1 - p_N) \int_0^{\infty} \int_y^{\infty} f_{T_{L,i}}(y) f_{T_A}(x) dx dy, \quad i = \overline{1, N},$$

$$q_{i, 2N+i} = (1 - p_N) \int_0^{\infty} \int_y^{\infty} f_{T_A}(y) f_{T_{L,i}}(x) dx dy, \quad i = \overline{1, N},$$

$$q_{N+i, i} = \prod_{k=1}^{i-1} p_{N, k}, \quad i = \overline{1, N},$$

$$q_{N+i, j} = (1 - p_{N, j}) \prod_{k=1, k \neq i}^{i-1} p_{N, k}, \quad i, j = \overline{1, N}, \quad i \neq j,$$

$$q_{i, 0} = p_N, \quad i = \overline{1, N},$$

$q_{0,0} = 1$, $q_{i,j} = 0$ в остальных случаях.

Напомним, что среднее количество посещений поглощающей цепью Маркова переходного состояния определяется элементами фундаментальной матрицы $D = (I - U)^{-1}$, где $U = [u_{ij} = q_{ij}]$, $i, j = \overline{1, 3N}$ – подматрица, описывающая процесс до выхода из множества переходных состояний. Элемент (i, j) матрицы D – это ожидаемое число раз, когда цепочка находится в состоянии j , учитывая, что цепочка началась в состоянии i . С использованием фундаментальной матрицы, среднее число шагов от состояния i до поглощения задается соответствующими элементами вектора $\vec{\tau}$ с компонентами $\tau_i = \sum_{j=1}^{3N} \pi_j d_{ji}$, $i = \overline{1, 3N}$.

Используя полученный результат, можем найти среднюю продолжительность поиска луча в неблокированном состоянии: $E[B] = T_B \sum_{j=N+1}^{3N} \tau_{N+j}$. Данное выражение при-

водит к средней продолжительности интервала связи: $E[T_L] = E[B] + \sum_{i=1}^N \tau_i T_{L,i}$.

Теперь средняя продолжительность непрерывной связи с i -й БС может быть найдена из минимума двух случайных величин как

$T_{L,i} = \int_0^{\infty} t (f_{T_{L,i}}(t) [1 - F_A(t)] + f_{T_A}(t) [1 - f_{T_{L,i}}(t)]) dt$. Это выражение дает нам возможность

определить долю времени, в течение которого ПУ связано с i -й БС. Используя это наблюдение, средняя спектральная эффективность в течении времени T_L может быть

найдена как $E[C_{T_L}] = \frac{1}{E[T_L]} \sum_{i=1}^N \tau_i T_{L,i} E[C_i]$.

Таким образом, вероятность отсутствия связи может быть представлена как $p_0 = \frac{E[T_{NL}] + E[B]}{E[T_{NL}] + E[T_L]}$. Также можно определить эргодическую спектральную эффектив-

ность $E[C] = \frac{E[T_L] E[C_{T_L}]}{E[T_{NL}] + E[T_L]}$.

2. Численные результаты

Для проведения численного исследования используем данные, представленные в табл. 1. На рис. 4 изображены графики зависимости вероятности отсутствия связи и спектральной эффективности в ТГц диапазоне частот от плотности блокаторов при степени мультисвязности $N = 1$ и $N = 3$.

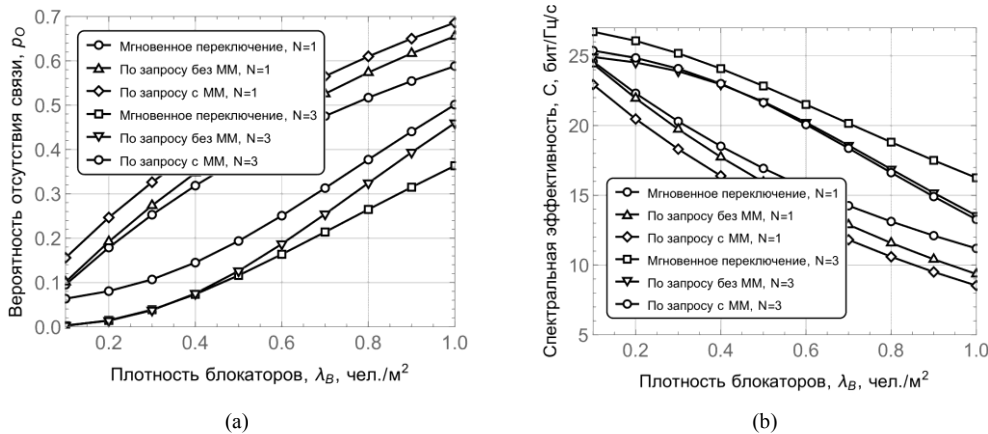


Рис. 4 Вероятность отсутствия связи (a) и спектральная эффективность (b) в зависимости от плотности блокаторов

Как можно заметить из рисунка, с увеличением плотности блокаторов вероятность отсутствия связи также возрастает, а спектральная эффективность, наоборот, убывает. Также из графика видно, что показатели системы при переключении по запросу с учетом микромобильности становятся еще хуже, чем при переключении по запросу без учета микромобильности по сравнению со схемой мгновенного переключения. Аналогичную ситуацию можно заметить и на графике спектральной эффективности. Также стоит отметить, что эффективность работы схемы напрямую зависит от числа БС. С увеличением БС показатели системы улучшаются, т.е. вероятность отсутствия связи становится значительно ниже, чем при тех же значениях плотности блокаторов, но с меньшим числом БС, а спектральная эффективность – выше.

Заключение

В работе была рассмотрена модель обслуживания пользовательского устройства базовыми станциями, работающими в ТГц диапазоне частот в условиях динамической блокировки и микромобильности. Результаты численного эксперимента демонстрируют, что присутствие микромобильности значительно ухудшает исследуемые характеристики обслуживания. Однако использование большего числа ТГц БС по принципу множественных подключений позволяет кардинально улучшить показатели функционирования системы. В дальнейшем планируется рассмотреть влияние динамической блокировки и микромобильности на другие показатели эффективности системы, например, угловую скорость вращений.

ЛИТЕРАТУРА

1. Dang S., Amin O., Shihada B., Alouini M.-S. What should 6g be? // Nature Electronics. – 2020. – V. 3. – No. 1. – P. 20–29.
2. Rappaport T.S., Xing Y., Kanhere O., Ju S., Madanayake A., Mandal S., Alkhateeb A., Trichopoulos G.C. Wireless communications and applications above 100 ghz: Opportunities and challenges for 6g and beyond // IEEE Access. – 2019. – V. 7. – P. 78729–78757.
3. Medvedeva E., Gorbunova Y., Gaidamaka Y., Samouylov K. A discrete queueing model for performance analysis of scheduling schemes in multi-user mimo systems // IEEE International Congress on Ultra Modern Telecommunications and Control Systems and Work-shops (ICUMT). – 2019. – V. 2019. – P. 1–5.

4. Bilgin B.A., Ramezani H., Akan O.B. Human blockage model for indoor terahertz band communication // IEEE International Conference on Communications Workshops (ICC Workshops). – 2019. P. 1–6.

5. Gerasimenko M., Moltchanov D., Gapeyenko M., Andreev S., Koucheryavy Y. Capacity of multiconnectivity mmWave systems with dynamic blockage and directional antennas // IEEE Transactions on Vehicular Technology. 2019. – 68 (4). – P. 3534–3549.

6. 3GPP, “NR; Multi-connectivity; stage 2 (Release 16),” 3GPP TS 37.340 V16.0.0, December 2019.

7. Begishev V., Moltchanov D., Sopin E., Samuylov A., Andreev S., Koucheryavy Y., Samouylov K. Quantifying the impact of guard capacity on session continuity in 3GPP New Radio systems // IEEE Transactions on Vehicular Technology. – 2019. – V. 68. – No. 12. – P. 12345–12359.

8. Petrov V., Moltchanov D., Koucheryavy Y., Jornet J.M. Capacity and outage of terahertz communications with user micromobility and beam misalignment // IEEE Transactions on Vehicular Technology, 2020. – V. 69. – No. 6. – P. 6822–6827.

9. Gapeyenko M., Samuylov A., Gerasimenko M., Moltchanov D., Singh S., Akdeniz M.R., Aryafar E., Himayat N., Andreev S., Koucheryavy Y. On the temporal effects of mobile blockers in urban millimeter wave cellular scenarios // IEEE Transactions on Vehicular Technology. – 2017. – Available online.

DOI: 10.17223/978-5-907442-42-9-2021-18

О КОНЦЕПЦИИ P2P ПРИ ПЕРЕДАЧЕ ДАННЫХ В СЕТИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

Джегюеде А.М.А.Э., Салпагаров С.И.

Российский университет дружбы народов
1042205202@rudn.ru, salpagarov-si@rudn.ru

Введение

Одноранговая сеть peer-to-peer (P2P) – хорошо известный вид распределенной сети, которая в последние 20 лет широко используется для многих целей. Цели варьируются от протоколов обмена файлами [1] до протоколов потоковой передачи [2] и обмен финансовыми транзакциями в блокчейн-приложениях [3]. Помимо P2P, существует еще одна сетевая архитектура – клиент-серверная (CS). CS из-за его централизованной топологии подвержен сбоям, когда количество пользовательских запросов к центральному серверу превышает определенный порог [4], вследствие чего может наблюдаться неспособность клиентов общаться друг с другом. Более того, реализация архитектуры CS обходится слишком дорого из-за стоимости серверов и их обслуживания. P2P представляет собой децентрализованную распределенную сеть, в которой каждый узел имеет одинаковые привилегии и одинаковый набор функций. Благодаря недорогой реализации эта архитектура пользуется большим успехом в различных областях, таких как телевидение [2], системы обмена файлами [1], VoIP сервисы [5] и блокчейн-приложения.

1. P2P топологии

Как правило, топология описывает взаимосвязь между узлами в сети [2]. Она может быть физической или логической. В архитектуре P2P чаще всего используются топологии ячеистые и древовидные. В древовидной топологии существуют два класса узлов – родительские (корневые) и дочерние узлы. Для древовидной топологии характерно наличие единой точки отказа в лице корневого узла. Механизм быстрой замены был реализован в рамках реализации системы MAPSTER – одноранговая система обмена данными [6]. В MAPSTER была реализована Super Peer топология. В ней различались SP-узлы (суперузлы), ASP-узлы (запасные суперузлы) и NP-узлы (нормальные узлы). Сеть разделена на домены (кластеры). В каждом домене SP-узел имеет главную роль маршрутизации и управления. ASP-узлы являются резервными, выбираются из ряда наиболее устойчивых по ряду параметров (мощности процессора, доступной пропускной способности и т.д.) среди NP-узлов и готовы в установленном порядке заменить SP-узлы. В [7] была представлена гибридная топология под названием UStream. Она предполагает три слоя, состоящих из Root-узлов, Parent-узлы и Children-узлов со-

ответственно. Каждый корневой узел образует вместе с нижестоящими родительскими узлами и дочерними узлами отдельное поддерево, узлы одного слоя формируют ячеистую структуру. UStream была опробована в [2] в рамках разработки протокола передачи данных потокового вещания в слабых одноранговых сетях. Функция мониторинга сети была поручена отдельно стоящему видеосерверу.

2. Поиск данных и алгоритмы маршрутизации в P2P сетях

Поиск и передача данных остаются базовыми задачами при проектировании одноранговых сетей. Решение этих задач привело к появлению DHT и P2PDB систем. Распределенные хэш-таблицы – это класс алгоритмов или механизмов, позволяющих в некоторых P2P сетях образовать оверлейную сеть, а также распределить файлы по узлам. DHT имеет ряд реализаций, например, Kademlia, Pastry, Tapestry, Chord и CAN. «Kademlia» опирается на расстояние между идентификаторами узлов для формирования логической структуры сети. Расстояние вычисляется по формуле: $d = d(\text{IDNODE}_y, \text{IDNODE}_z) = \text{ID}_y \oplus \text{ID}_z$. Знания об общей структуре сети поддерживаются за счет заполнения списков, соответствующих определенным интервалам расстояний. Списки имеют следующий формат: $\langle \text{IP}, \text{UDP}_{\text{port}}, \text{NODE_ID} \rangle$. На функциональном уровне «Kademlia» реализует четыре RPC (Remote Procedure Call): PING, STORE, FIND_NODE и FIND_VALUE. Chord (Хорд) является распределенным поисковым протоколом [8] и отвечает за распределение хэш-ключей по узлам. Хорд использует консистентное хеширование [9], реализованное функцией SHA-1 для генерации ключей и позволяющее балансировать нагрузку (равномерное распределение ключей). Распределение ключей K происходит по кругу так, что каждый ключ K переходит к самому ближайшему узлу с NODE_ID, большим или равным K. Алгоритм поиска в самом его примитивном виде реализует передачу сообщений запроса по цепочке, пока не найдется узел, содержащий искомый ключ. Такой поиск слишком затратный по времени в особо больших сетях. Ускорение поиска привело к изобретению алгоритма «Scalable Key Location» (SKL) [8]. SKL допускает использование таблицы маршрутизации, содержащей информацию об узлах по мере их удаленности.

3. Методы обеспечения качества обслуживания

Обеспечение качества обслуживания остается важной задачей в сетях с ограниченными параметрами. Управление очередями – один из методов обеспечения качества обслуживания. Оно опирается на наличие трафика различного типа и предъявляемые качественные требования к ним. Критерии оценки качества обслуживания сети: односторонняя и двусторонняя задержка пакетов, вариация задержек пакетов, потери пакетов. Уровень использования пропускной способности – один из важных факторов, влияющих на качество обслуживания. В [2] представлен механизм контроля перегрузки, называемый STS (Steady traffic scheduler). Данный механизм комбинирует технику «взвешенного справедливого обслуживания» с алгоритмом «текущего ведра». Функционально STS можно разделить на три модуля: модуль классификатора, модуль планировщика и модуль «текущего ведра маркеров». Классификатор принимает на вход поступающий поток пакетов, и, основываясь на заданных критериях, распределяет их по буферам (очередям). Каждая очередь наделена определенным приоритетом. Модуль планировщика, основываясь на приоритетах очередей, выделяет определенную временную рамку каждой очереди для передачи её содержания в следующий модуль. Последний модуль реализует непосредственно алгоритм «текущего ведра маркеров».

4. Техники управления очередями

Использование очередей является прерогативой сетей с коммутацией пакетов. Несмотря на их пользу, могут возникнуть ряд проблем в отсутствии их надлежащего управления. К их числу можно отнести переполнение буфера с последующей потерей пакетов и неопределенную временную задержку. В теории очередей хорошо изучена зависимость средней длины очереди и среднего времени ожидания пакета в очереди от характеристик входного потока (μ – среднее время обслуживания пакета, λ – средняя интенсивность поступления пакетов). Под коэффициентом загрузки обслуживающего устройства будем понимать ρ , такое, что $\rho = \lambda/\mu$.

Зависимость ω среднего времени ожидания в очереди от ρ показывает резкое увеличение задержки при переходе некоторого порогового значения коэффициента загрузки. Пульсации характеризуют трафики с переменной битовой скоростью (VBR), когда вслед за периодами низкой скорости передачи данных возникает большая интенсивность передачи данных. Коэффициент пульсации можно вычислить как отношение пиковой скорости передачи данных (PIR) на небольшом периоде к средней скорости передачи на большом отрезке времени (SIR): коэффициент пульсации = PIR/SIR.

Анализ показывает, что чем сильнее пульсирует трафик, тем быстрее формируется очередь при относительно маленьких коэффициентах загрузки. Таким образом, для снижения временных задержек необходимо снижать не только коэффициент загрузки, но и степень пульсации трафика. Классификация трафика исходит из предположения о наличии в сети трафика, генерируемого разнообразными приложениями, предъявляющими разные требования к задержкам, вариации задержек, потери пакетов. Дифференцированное обслуживание различных потоков трафика позволяет достигнуть необходимого качества обслуживания. Оно подразумевает наличие различных очередей. Существует три техники обслуживания очередей: приоритетное обслуживание, взвешенное обслуживание и комбинированные алгоритмы обслуживания.

Приоритетное обслуживание заключается в том, чтобы обслуживать очереди в порядке от более приоритетной очереди к менее приоритетной. Переход к менее приоритетной очереди происходит только после опустошения обслуживаемой очереди. Приоритет наделяется классам на основе разных признаков. Признаками для классификации могут служить идентификаторы приложения источника потока, адрес отправителя, адрес назначения или любое другое соображение. Назначение приоритетов относится к политике администрирования сети. Размер буферной памяти для хранения каждой очереди выбирается с учетом интенсивности λ и коэффициента пульсации данного потока. Взвешенное обслуживание очередей разработано с целью предоставить каждому классу потока определенный процент пропускной способности. К каждому классу прикрепляют вес, соответствующий определенному проценту пропускной способности. Очереди обслуживаются последовательно и циклически. Распределение времени просмотра цикла между очередями сохраняет пропорции, установленные весовыми коэффициентами, закрепленными за каждой очередью. Выбор слишком большого периода обработки очереди может отрицательно сказаться на временной задержке пакетов в очередях, т.к. алгоритм повторно обращается к определенной очереди, только через время T . В данном случае на формирование задержек также оказывает влияние и относительный коэффициент загрузки $\rho_{\text{отн.}}$: $\rho_{\text{отн.}} = \lambda_i/C_i$, где C_i – выделенная пропускная способность в соответствие с весом очереди, λ_i – средняя интенсивность потока i -й очереди.

Заключение

В рамках данной обзорной статьи были рассмотрены основные аспекты Р2Р сетей с ограниченными параметрами. Обзор существующих топологий показал наличие ячеистой и древовидной топологии. Выбор определенной топологии зависит от цели, задачи

и предназначения проектируемой сети. Тем не менее, Super-Peer топология предоставляет большее удобство в плане администрирования. Поиск данных в одноранговых сетях основан в большинстве случаев на распределенных хэш-таблицах. Рассмотрение разных реализаций, а именно – Kademia и Chord, выявило их разные подходы к логическому структурированию сети, но в обоих случаях механизм поиска основан на определении расстояния между ключами и идентификаторами узлов. Рассмотрение методов обеспечения качества было вызвано ограниченными параметрами сети. Акцент был сделан конкретно на техники управления очередями. Приоритетное обслуживание очередей показало свои пределы при высокой интенсивности потока с наивысшим приоритетом, в то время как взвешенное обслуживание показало более приемлемое решение в распределении пропускной способности.

ЛИТЕРАТУРА

1. J.J., Karlsen L., Birkeland S. Peer-to-peer networking with BitTorrent. – 2005.
2. Ojo O.E., Oluwatope A.O., Ajadi S.O. A reliable peer-to-peer streaming protocol in low-capacity networks // Peer-to-Peer Networking and Applications. – Oct. 2020. – V. 14. – No. 2. – P. 559–584. – doi: 10.1007/s12083-020-01002-4.
3. Delgado-Segura S., Pérez-Solà C., Herrera-Joancomartí J., Navarro-Arribas G., Borrell J. Cryptocurrency Networks: A New P2P Paradigm // Mobile Information Systems. – 2018. – V. 2018. – doi: 10.1155/2018/2159082.
4. Ramzan N., Park H., Izquierdo E. Video streaming over P2P networks: Challenges and opportunities // Signal Processing: Image Communication. – May 2012. – V. 27. – No. 5. – P. 401–411. – doi: 10.1016/j.image.2012.02.004.
5. Fiedler J., Kupka T., Magedanz T., Kleis M. Reliable VoIP Services Using a Peer-to-Peer Intranet. – 2006. – doi: 10.1109/ism.2006.132.
6. R.C., B.S. Mapster: a peer-to-Peer data sharing environment // South Afr. Comput. J. – 2007. – V. 39. – P. 35–46.
7. O.O., O.A., O.O. UStream: Ultra-Metric Spanning Overlay Topology for Peer-to-Peer Streaming Systems // IEEE International Symposium on Multimedia (ISM), 2015. – P. 601–604. – doi: 10.1109/ISM.2015.82.
8. Stoica I. et al. Chord: a scalable peer-to-peer lookup protocol for Internet applications // IEEE/ACM Transactions on Networking. – 2003. – V. 11. – No. 1. – P. 17–32. – doi: 10.1109/TNET.2002.808407.
9. Karger D., Lehman E., Leighton T., Panigrahy R., Levine M., Lewin D. Consistent hashing and random trees. – 1997. – doi: 10.1145/258533.258660.

DOI: 10.17223/978-5-907442-42-9-2021-19

К АНАЛИЗУ НЕПРЕРЫВНОСТИ ПОЛЬЗОВАТЕЛЬСКОЙ СЕССИИ В УСЛОВИЯХ МИКРОМОБИЛЬНОСТИ И БЛОКИРОВКИ ПРЯМОЙ ВИДИМОСТИ*

Конюхов Р.И.¹, Бесчастный В.А.¹, Молчанов Д.А.², Самуйлов К.Е.¹

¹Российский университет дружбы народов, ²Университет Тампере

konyukhov.roman@mail.ru, beschastnyy-va@rudn.ru, dmitri.moltcanov@tuni.fi, samuylov-ke@rudn.ru

Введение

Интересы исследовательского сообщества в наши дни направлены на изучение новой технологии радиointерфейса для систем 6G [1]. По общему мнению, этот новый интерфейс будет работать в нижней части терагерцового диапазона частот (0.3 – 3 ТГц) [2]. Используя массивные антенные решетки как на стороне базовой станции (БС), так и пользовательского оборудования (ПУ) для компенсации затухания радиосигнала, в ТГц системах будет применяться небольшая ширина диаграммы направленности по уровню половинной мощности (НРВМ), приближенная к долям градуса [3].

В качестве побочного эффекта такая высокая направленность позволяет эффективно подавлять помехи между соседними сотами [4]. Однако это также ставит перед разработчиками систем ряд трудноразрешимых задач. В частности, в дополнение к большей чувствительности к блокировкам прямой видимости по сравнению с диапазоном

* Публикация выполнена при поддержке Программы стратегического академического лидерства РУДН и при финансовой поддержке РФФИ в рамках научных проектов № 19 07 00933 и № 20 07 01064.

миллиметровых волн, высокая направленность может привести к частым сбоям в работе, вызванными микроомобильностью ПУ, т.е. быстрыми изменениями положения рук пользователя по вертикальной и поперечной осям, а также небольшим смещениям [5]. При возникновении таких сбоев необходимо применять процедуру поиска луча, чтобы восстановить активное соединение. Производительность на уровне канала ТГц при наличии микроомобильности ПУ была исследована в [6], где авторы продемонстрировали компромиссы между долей времени в условиях отсутствия связи и достижимой пропускной способностью канала. В зависимости от типа эффекта, приводящего к потере сигнала, блокировки или микроомобильности, пользовательские приложения могут находиться в состоянии простоя в течение периодов разной продолжительности, влияющие на общую непрерывность сессии.

Одним из способов избежать блокировки в системах миллиметрового и терагерцового диапазона является использование стандартизованных 3GPP функциональных возможностей мультисвязности [7]. В соответствии с ним ПУ может поддерживать два или более одновременных соединения с соседними БС и использовать их, когда потеряна связь с ближайшей БС. Мультисвязность изучалась на системном уровне в контексте систем 5G mmWave, например, в [8]. Использование мультисвязности для ТГц систем исследовалось в [9], где авторы продемонстрировали спектральную эффективность и пользу от множественных соединений при блокировках прямой видимости. Однако этот механизм также можно использовать и для борьбы с эффектами микроомобильности.

Известные на данный момент исследования мультисвязности как в миллиметровом, так и в терагерцовом диапазоне частот, были сосредоточены на усредненных значениях показателей производительности, таких как вероятность отсутствия связи или средняя спектральная эффективность. Однако на практике приложениям, использующим сеть, часто требуется непрерывное подключение, и решить эту проблему значительно сложнее, поскольку необходимо постоянно отслеживать состояние устройства. В данной работе исследуется непрерывность пользовательской сессии, для чего выводится распределение плотности интервалов времени непрерывного соединения. При этом в явном виде учитываются конкретные условия распространения, плотность развертывания ТГц БС, конфигурация антенны и процедуры поиска луча, а также степень мультисвязности. Затем, охарактеризовав интервалы периода простоя приложения, исследуется вероятность успешного завершения сессии.

Основными результатами нашего исследования являются:

1. Математическая модель для оценки времени непрерывного соединения ТГц системах в условиях микроомобильности и блокировки прямой видимости при наличии мультисвязности;
2. Численный эксперимент показал, что вероятность завершения сессии линейно увеличивается с увеличением степени мультисвязности до 5, на которую в основном влияет микроомобильность.

1. Системная модель

Модель системы проиллюстрирована на рис. 1.

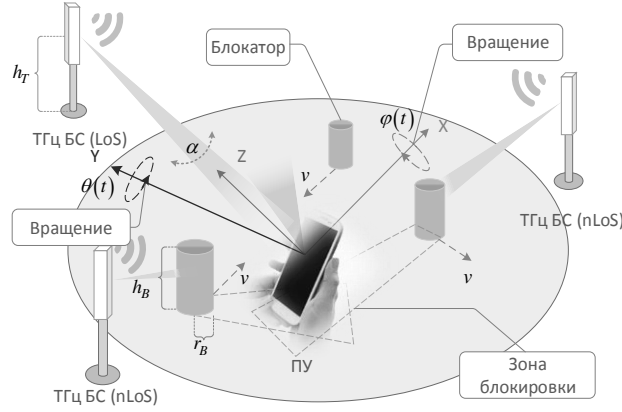


Рис. 1. Иллюстрация модели рассматриваемой системы

Мы предполагаем, что положения базовых станций ТГц диапазона следуют Пуассоновскому точечному процессу в R^2 с плотностью λ_A . Высота ТГц БС – h_T . Местоположение интересующего нас ПУ является случайным в области БС, что означает, что плотность распределения вероятностей расстояния до ближайшей i -й ТГц БС [Ошибка! Источник ссылки не найден.] можно записать как

$$f_i(x) = \frac{2(\pi\lambda_A)^i}{(i-1)!} x^{2i-1} e^{-\pi\lambda_A x^2}, \quad x > 0, \quad i = 1, \dots$$

Предполагается, что линия прямой видимости может быть перекрыта из-за динамической блокировки человеческим телом. Потенциальные блокаторы моделируются в виде цилиндров с радиусом r_B и высотой h_B . Плотность блокаторов составляет λ_B бл./м², и предполагается, что они движутся согласно модели случайного направления со скоростью v м/с. Высота ПУ предполагается равной h_U , $h_U \leq h_B$. Согласно [11], время между последовательными событиями блокировки при модели случайного направления аппроксимируется экспоненциальным распределением. Таким образом, процесс блокировки i -й БС является пуассоновским со средней интенсивностью $\mu_{B,i}$, представленной формулой

$$\mu_{B,i} = \int_0^\infty f_i(x) \frac{(x[h_B - h_U] + r_B[h_T - h_U])}{(2r_B\lambda_B v)^{-1}(h_T - h_U)} dx.$$

Поскольку известно, что эффект блокировки человеческим ТГц диапазоне приводит к огромным потерям сигнала [12], мы предполагаем, что в заблокированном состоянии линии прямой видимости связь невозможна. Значение SINR на UE в незаблокированном состоянии линии прямой видимости может быть записано как

$$S(x) = P_T G_T G_U \left[\frac{x^{-\xi_T} e^{-Kx}}{N_0} \right],$$

где ξ_T – коэффициент затухания сигнала; N_0 – мощность шума; P_T – излучаемая мощность; G_T и G_U – коэффициенты усиления антенны на БС и ПУ, K – коэффициент поглощения.

Следуя [6], мы описываем процесс микромобильности как броуновское движение со случайным процессом смещения по x - и y -плоскостям и вращениями по вертикальной и поперечной осям, $\varphi(t)$ и $\theta(t)$. Смещения по плоскости z , т.е. по направлению к базовой станции и от неё, и вращения по продольной оси не учитываются, поскольку их вклад в эффект расхождения лучей достаточно мал [6]. Это обеспечивает плотность распределения вероятностей времени до рассогласования луча, T_A , в виде минимума из двух случайных величин с плотностями распределения вероятностей, соответствующую

щими временам первого прохождения броуновской частицы, т.е. $S_{T_A}(t) = f_0(t)[1 - F_\phi(t)] + f_\phi(t)[1 - F_0(t)]$, где компоненты представлены формулой

$$f_{(\cdot)}(t) = \frac{M_{\phi 0}}{\sqrt{4\pi D_{(\cdot)} t^3}} \exp\left(-\frac{M_{\phi 0}^2}{4D_{(\cdot)} t}\right), \text{ здесь } D_{(\cdot)} - \text{ постоянная диффузии в соответствующей}$$

плоскости. При этом $M_{\phi 0} = (102\pi/360)(1/N_T + 1/N_U)$, где N_T, N_U – количество антенных элементов на БС и ПУ соответственно.

В работе рассматриваются квадратные плоские антенные решетки как на ПУ, так и на ТГц БС. Аналогично [13], используется модель диаграммы направленности в виде конуса с шириной луча, соответствующей ширине по уровню половинной мощности (англ. HPBW). Следуя [14], мы аппроксимируем HPBW как $102/N_{(\cdot)}$.

Мы предполагаем, что в случае блокировки линии прямой видимости, или расхождения лучей, базовые станции ТГц диапазона инициируют итеративный алгоритм поиска луча, которые используются, например, в системах IEEE 802.11ad/ay [15]. Длительность процедуры поиска луча $T_B = (N_U + N_T)\sigma$, где σ – время одного переключения.

Чтобы оценить влияние эффекта мультисвязности на непрерывность сессии, предполагается, что целевое ПУ поддерживает $N = 1, 2, \dots$ связей с ближайшими ТГц БС. Поскольку показатель отношения сигнал-шум (англ. SNR) обратно пропорционален расстоянию между ТГц БС и ПУ, то ближайшая БС обеспечивает наилучшее качество связи с точки зрения усредненного по времени SNR. Предполагается, что во время процедуры поиска луча выбирается БС с наибольшим значением SNR на данный момент, т.е. выбирается ближайшая БС в условиях линии прямой видимости. Также предполагается, что поиск луча запускается каждый раз, когда активное соединение потеряно в результате микроподключности или блокировки прямой видимости.

В отличие от других исследований эффекта мультисвязности, где авторы в первую очередь концентрируются на средних показателях производительности, таких как вероятность отсутствия связи или средняя спектральная эффективность, мы характеризуем свойства процесса обслуживания сессии на временном интервале. В частности, сначала приводится плотность распределения времени непрерывного подключения, а затем, с использованием полученных выражений, находится вероятность успешного завершения сессии.

2. Анализ рабочих характеристик системы

Интервалы подключения и отключения в системе организуют альтернирующий полумарковский процесс. Подход основан на применении теории полумарковских процессов [16], позволяющей характеризовать процесс восстановления матрицей интервальных переходных вероятностей $\Phi(s)$, состоящей из условных переходных вероятностей за время t . На рис. 2 представлена диаграмма вероятностей переходов с 5 непересекающимися множествами состояний: (I) активная связь с БС, $j = \overline{1, N}$; (II) поиск луча в результате блокировки, $j = \overline{N+1, 2N}$; (III) поиск луча из-за микроподключности, $j = \overline{2N+1, 3N}$, (IV) условия отключения с непрерывным подключением $j = \overline{3N+1, 4N}$ и (V) нулевое состояние для отключения. Состояния $i = \overline{1, N}$ соответствуют периодам непрерывной связи с i -й БС в условиях блокировки. Состояния $j = N+i, i, \dots, 2N$ представляют процедуру поиска луча в результате блокировки после установления связи с i -й БС.

Применяя одностороннее преобразование Лапласа $f^*(s) = \int_0^{\infty} e^{-sx} f(x) dx$, получаем

искомую матрицу в виде $\Phi^*(s) = [I - U \circ F^*(s)]^{-1} \Psi^*(s)$, где I – единичная матрица, U – матрица переходных вероятностей, F – матрица распределений условных вероятностей, Ψ – вспомогательная диагональная матрица.

Элементы $U = [u_{ij}]$, $i, j = \overline{0, 4N}$, можно записать в виде:

$$u_{i, N+i} = 1 - \int_0^{\infty} \int_0^{\infty} f_{T_A}(y) F_{T_{L,i}}(x) dx dy, \quad i = \overline{1, N},$$

$$u_{i, 2N+i} = \int_0^{\infty} \int_0^{\infty} f_{T_A}(y) F_{T_{L,i}}(x) dx dy, \quad i = \overline{1, N},$$

$$u_{N+i, j} = (1 - F_I(T_B)) \prod_{k=1}^{j-1} p_{N,k}, \quad i, j = \overline{1, N},$$

$$u_{2N+i, i} = (1 - F_I(T_B)) \prod_{k=1}^{i-1} p_{N,k}, \quad i = \overline{1, N},$$

$$u_{N+i, 3N+i} = (1 - F_I(T_B)) \prod_{k=1, k \neq i}^N p_{N,k}, \quad i = \overline{1, N},$$

$$u_{N+i, 0} = F_I(T_B), \quad i = \overline{1, 2N},$$

$$u_{3N+i, 2N+i} = F_{NL}(T_N; i) \prod_{k=1}^{i-1} p_{N,k}, \quad i = \overline{1, N},$$

$$u_{3N+i, 0} = 1 - F_{NL}(T_N; i), \quad i = \overline{1, N},$$

$$u_{i, j} = 0, \quad \text{в других случаях.}$$

Элементы $F(t) = [f_{ij}(t)]$, $i, j = \overline{0, 4N}$, состоящие из переходных вероятностей за время t , могут быть записаны как:

$$f_{i, N+i}(t) = f_{T_{L,i}}(t) (1 - F_{T_A}(t)), \quad i = \overline{1, N},$$

$$f_{i, 2N+i}(t) = f_{T_A}(t) (1 - F_{T_{L,i}}(t)), \quad i = \overline{1, N},$$

$$f_{N+i, j}(t) = \delta(t - T_B), \quad i = \overline{1, N},$$

$$f_{2N+i, i}(t) = \delta(t - T_B), \quad i = \overline{1, N},$$

$$f_{N+i, 3N+i}(t) = \delta(t - T_B), \quad i = \overline{1, N},$$

$$f_{N+i, 0}(t) = \delta(t - T_N), \quad i = \overline{1, 2N},$$

$$f_{3N+i, 2N+i}(t) = f_{NL}(t; i) F_{NL}(T_N; i), \quad i = \overline{1, N},$$

$$f_{3N+i, 0}(t) = f_{NL}(t + T_N; i) (1 - F_{NL}(T_N; i)), \quad i = \overline{1, N},$$

$$f_{i, j} = 0, \quad \text{в других случаях,}$$

где f_{T_L} – распределение времени до блокировки прямой видимости, f_{T_A} – распределение времени до микромобильности, f_{NL} – длительность блокировки прямой видимости, $\delta(t)$ – дельта-функция Дирака.

Наконец, мы определяем плотность распределения вероятностей времени непрерывного подключения и вероятность успешного завершения сеанса следующим образом: $f_U(t) = 1 - \sum_{i=1}^{4N} \pi_i \phi_{i0}(t)$, $p_U = \int_0^{T_C} f_U(t) dt$, где T_C – продолжительность сессии.

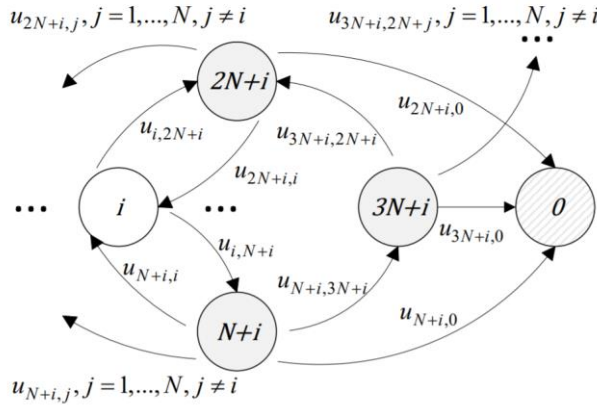


Рис. 2. Поглощающая полумарковская модель

3. Численные результаты

В этом разделе представлены результаты численного эксперимента.

На рис. 3.а показано влияние степени мультисвязности на вероятность успешного завершения сессии. Ожидаемо, увеличение значения N приводит к более высокой вероятности успешного завершения сессии. Однако, начиная с $N=5$ эффект от использования дополнительных ТГц БС резко сокращается. На рис. 3б охарактеризована рассматриваемая метрика как функция продолжительности сессии, T_C , для четырех рассматриваемых значений T_N и при $N=1$. Здесь $T_N=5$ мс и 5 с соответствуют крайним случаям высокочувствительных и нечувствительных приложений, в то время как два оставшихся значения представляют собой промежуточные случаи. Как можно заметить, существует резкая разница между вероятностью успешного завершения сеанса для $T_N=5$ мс и 5 с. В первом случае, если случается микроподвижность или блокировка, это сразу же приводит к прерыванию сессии (т.к. у нас продолжительность поиска луча занимает некоторое время, порядка 8 мс, а средняя продолжительность блокировки составляет порядка секунды), в то время как в случае $T_N=5$ с приложения могут переносить как блокировку, так и нарушения микроподвижности, которые как правило длятся меньше 5 секунд. При этом стоит отметить, что нет большой разницы между промежуточными значениями в 50 и 500 мс. Это означает, что, адаптируя будущие приложения к технологии доступа ТГц диапазона, необходимо гарантировать, что приложения могут выдерживать время простоя в масштабе длительности процедуры поиска луча.

Теперь рассмотрим влияние интенсивности блокаторов на вероятность успешного завершения сессии, проиллюстрированного рис. 2 для двух степеней множественного подключения $N=1$ и $N=3$ и двух длительностей сессии, $T_C=10$ с и $T_C=30$ с. Как видно, положительный эффект мультисвязности сохраняется на всем рассматриваемом диапазоне значений, особенно, когда продолжительность сессии велика.

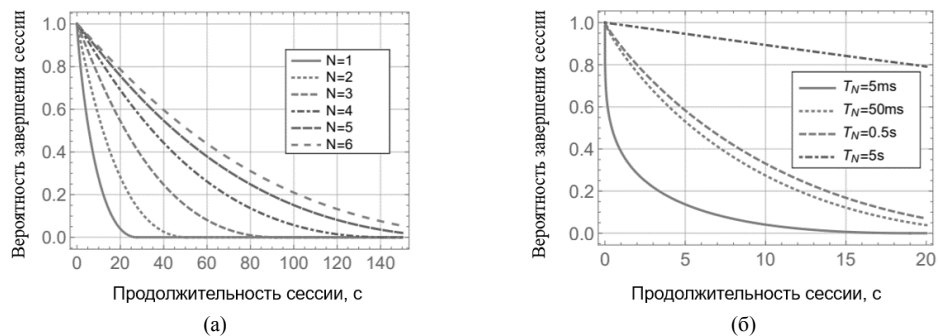


Рис. 3. Вероятность успешного завершения сессии в зависимости от степени мультисвязности и допустимого периода простоя

Наконец, рассмотрим влияние плотности развертывания ТГц БС на вероятность успешного завершения сессии, показанного на рис.4б, для двух длительностей сессии и двух степеней мультисвязности. Как можно заметить, увеличение плотности развертывания ТГц БС приводит к небольшому увеличению вероятности завершения сессии, т.е. рано или поздно, такое увеличение перестанет давать нам желаемый результат, эффект влияния пропадает. Обоснование этого состоит в том, что использование функции мультисвязности в ТГц системах по своей сути требует очень плотного развертывания, но при этом расстояние до ближайших базовых станций при таких плотностях изменяется лишь незначительно.

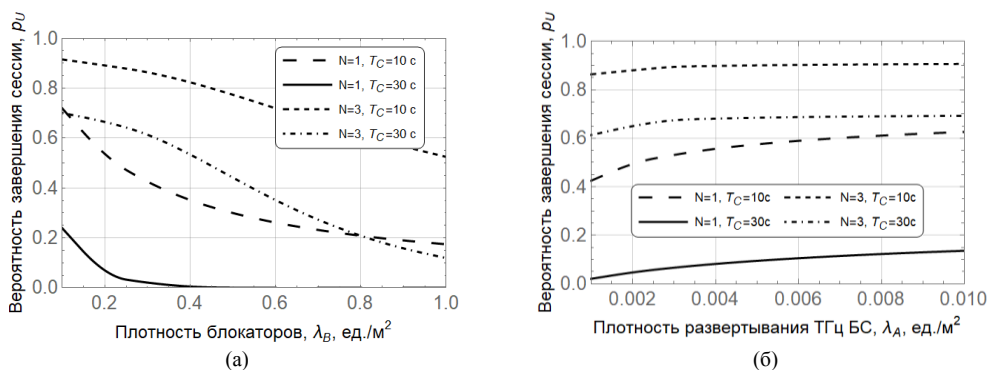


Рис. 4. Вероятность успешного завершения сессии в зависимости от плотности блокаторов и плотности развертывания БС

Заключение

В данной работе исследованы временные характеристики процесса подключения в плотных развертываниях ТГц диапазона в условиях динамической микромобильности пользователей и условиях блокировки линии прямой видимости. Для этой цели используется полумарковский процесс, характеризующий процесс подключения и интервалы простоя в виде матрицы интервальных переходных вероятностей. Проведен численный анализ зависимости вероятности успешного завершения сессии от различных параметров системы. В частности, в работе показано, что выигрыш от использования мультисвязности наблюдается до 5 одновременно поддерживаемых каналов. Этот выигрыш в значительной степени зависит от ожидаемой продолжительности сессии и сохраняется во всем рассматриваемом диапазоне плотностей блокаторов. На вероятность успешного завершения сессии приложений, которые очень чувствительны к продолжительности простоя, в основном влияет микромобильность. На практике, чтобы улучшить искомую метрику, необходимо убедиться, что приложение может выдерживать про-

стои, вызванные временем поиска луча, которое составляет порядка нескольких миллисекунд.

ЛИТЕРАТУРА

1. David K., Berndt H. 6G vision and requirements: Is there any need for beyond 5G? // IEEE Vehicular Technology Magazine. – 2018. – V. 13. – No. 3. – P. 72–80.
2. Petrov V., Kurner T., Hosako I. IEEE 802.15. 3d: First standardization efforts for sub-terahertz band communications toward 6G // IEEE Communications Magazine. – 2020. – V. 58. – No. 11. – P. 28–33.
3. Tousei Y., Afshari E. A high-power and scalable 2-D phased array for terahertz cmos integrated systems // IEEE Journal of Solid-State Circuits. – 2014. – V. 50. – No. 2. – P. 597–609.
4. Petrov V., Komarov M., Moltchanov D., Jornet J.M., Koucheryavy Y. Interference and sinr in millimeter wave and terahertz communication systems with blocking and directional antennas // IEEE Transactions on Wireless Communications. – 2017. – V. 16. – No. 3. – P. 1791–1808.
5. Petrov V., Moltchanov D., Koucheryavy Y., Jornet J.M. The effect of small-scale mobility on terahertz band communications // Proceedings of the 5th ACM International Conference on Nanoscale Computing and Communication. – 2018. – P. 1–2.
6. Petrov V., Moltchanov D., Koucheryavy Y., Jornet J.M. Capacity and outage of terahertz communications with user micro-mobility and beam misalignment // IEEE Transactions on Vehicular Technology. – 2020.
7. 3GPP, “NR; Multi-connectivity; stage 2 (Release 16),” 3GPP TS 37.340V16.0.0, December 2019.
8. Begishev V., Sopin E., Moltchanov D., Pirmagomedov R., Samuylov A., Andreev S., Koucheryavy Y., Samouylov K. Performance Analysis of Multi-Band Microwave and Millimeter-Wave Operation in 5G NR Systems // IEEE Transactions on Wireless Communications. – 2021.
9. Shafie A., Yang N., Han C. Multi-connectivity for indoor terahertz communication with self and dynamic blockage // ICC 2020-2020 IEEE International Conference on Communications (ICC). – 2020. – P. 1–7.
10. Moltchanov D. Distance distributions in random networks // Elsevier Ad Hoc Networks. – August 2012. – V. 10. – P. 1146–1166.
11. Begishev V., Moltchanov D., Sopin E., Samuylov A., Andreev S., Koucheryavy Y., Samouylov K. Quantifying the impact of guard capacity on session continuity in 3GPP new radio systems // IEEE Transactions on Vehicular Tech. – 2019. – V. 68. – No. 12. – P. 12345–12359.
12. Kurner T., Fricke A., Rey S., Le Bars P., Mounir A., Kleine-Ostmann T. Measurements and modeling of basic propagation characteristics for intra-device communications at 60 GHz and 300 GHz // Journal of Infrared, Millimeter, and Terahertz. – 2015. – V. 36. – No. 2. – P. 144–158.
13. Mokrov E., Ponomarenko-Timofeev A., Gudkova I., Masek P., Hosek J., Andreev S., Koucheryavy Y., Gaidamaka Y. Modeling transmit power reduction for a typical cell with licensed shared access capabilities // IEEE Vehicular Technology Magazine. – 2018. – V. 67. – No. 6. – P. 5505–5509.
14. Constantine A.B. et al. Antenna theory: analysis and design. – Microstrip Antennas, John Wiley & Sons, 2005.
15. IEEE, “Telecommunications and information exchange between systems local and metropolitan area networks – specific requirements. Part 11. Amendment 3,” IEEE standard for information technology, 2012.
16. Cinlar E. Markov renewal theory // Advances in Applied Probability. – 1969. – V. 1. – No. 2. – P. 123–187.

DOI: 10.17223/978-5-907442-42-9-2021-20

МОДЕЛЬ ПРИОРИТЕТНОГО ДОСТУПА ТРАФИКА URLLC С АДАПТИВНЫМ ИЗМЕНЕНИЕМ СКОРОСТИ ТРАФИКА eMBB В СЕТИ 5G: ЗАДАЧА ОПТИМИЗАЦИИ

Куцазли А.И., Агеева А.С., Харин П.А.

Российский университет дружбы народов
aikushch@yandex.ru, anastasia.ageeva.it@gmail.com, gruzavjeg@mail.ru

Введение

На сегодняшний день в цифровом обществе с каждым днём растёт роль телекоммуникационных услуг. Мы уже не можем представить себе жизнь без скоростного интернета, улучшенной сотовой сети. В современных быстро развивающихся телекоммуникационных технологиях услуги мобильной связи широко проникают во все слои жизни общества, тем самым беспроводная сеть 5-го поколения покрывает более широкий диапазон использования. Услуги 5G поставляются с различными вариантами использования, такими как расширенная мобильная широкополосная связь (eMBB), сверхнадежная связь с низкой задержкой (URLLC) и массовая связь машинного типа (mMTC). Если же рассматривать каждую услугу по отдельности, то к отличительной особенности URLLC являются требования к задержке, требования же к eMBB это вы-

сокая скорость передачи данных, а mMTC это возможность поддержки высокой плотности соединений [1,2].

Стандарты сотовой сети растут вместе с требованиями промышленного сектора и обычных пользователей. Практически в каждом секторе бизнеса существует повышенный спрос на новые или усовершенствованные приложения, которые повышают производительность труда или улучшают качество обслуживания клиентов. Сочетание всех возможностей устройств с более быстрой, высокой пропускной способностью и высокоинтеллектуальными сетями будет способствовать широкому экспериментированию и внедрению передовых мультимедийных приложений, которые способствуют и будут способствовать увеличению трафика и поддержки обеспечения работы новых сценариев использования в будущем.

В работе рассматривается модель совместного обслуживания трафика URLLC и eMBB в рамках одной базовой станции. Трафик первого типа считается приоритетным в силу своих высоких требований к задержке, а трафик второго типа имеет высокую скорость. Вследствие приоритета первый тип трафика преобладает над вторым типом трафика. В ситуации отсутствия свободных ресурсов для обслуживания URLLC трафика, выделенные ресурсы для eMBB трафика будут уменьшаться вплоть до полного изъятия.

1. Постановка задачи

Совместная работа услуг URLLC и eMBB на одном радио-ресурсе приводит к возникающей при пользовании проблеме планирования, которую достаточно сложно решить ввиду требований к QoS и расположенного в основе компромисса между правилами и условиями работы этих услуг. Следует отметить, что в отличие от трафика URLLC, требования к eMBB менее восприимчивы с точки зрения задержки, и, следовательно, обслуживание может быть отложено в пользу обслуживания более приоритетного URLLC трафика.

Рассмотрим задачу исследования совместного обслуживания трафика URLLC и eMBB в рамках одной базовой станции. Выделяемые ресурсы представлены в виде ресурсных блоков PRB (Physical Resource Block). Для обслуживания eMBB трафика выделяются слоты, которые в последствии будут называться ресурсными блоками, для URLLC выделяются мини-слоты, которые будут именоваться как ресурсные единицы.

Рассматривается одна сота с одной базовой станцией внутри, в области которой пользователи пытаются установить соединение. В ситуации отсутствия свободных ресурсов для обслуживания URLLC трафика, выделенные ресурсы для eMBB трафика будут уменьшаться вплоть до полного изъятия [3].

Емкость системы представлена в виде N ресурсных блоков, которые используются для обслуживания eMBB трафика. В свою очередь, каждый ресурсный блок состоит из b_1 ресурсных единиц, которые используются для обслуживания URLLC трафика. Соответственно общее количество ресурсных единиц равно $C = N \cdot b_1$.

Предположим, что поступающая на обслуживание сессия URLLC занимает одну ресурсную единицу и общее число активных сессий URLLC в некоторый момент времени составляет n . Сессии типа eMBB поступают на обслуживание на максимальной скорости, занимая целый ресурсный блок. При поступлении сессии URLLC скорость обслуживания сессии eMBB может быть уменьшена ввиду приоритета сверхнадёжного трафика. Для обозначения числа активных сессий eMBB с учетом их текущей скорости используется вектор $\vec{m} = (m_1, \dots, m_K)$, в то время как вектор скоростей обслуживания сессий eMBB $\vec{b} = (b_1, \dots, b_K)$, при этом $b_1 > b_2 > \dots > b_K$.

Состояния системы представлены следующим образом: $(m_1, \dots, m_K, n) = (\vec{m}, n) = \vec{x}$.

Сессии обоих типов трафика поступают в соответствии с пуассоновским законом и их интенсивности равны λ_m и λ_u . Следует отметить, что индекс m соответствует трафику eMBB, а u – URLLC.

Предполагается, что время обслуживания сессий URLLC и eMBB распределено экспоненциально. Для трафика URLLC интенсивность равна μ_u , а для eMBB – равна μ_m и не зависит от скорости обслуживания. Соответственно, нагрузка на систему будет иметь вид: $\rho_m = \frac{\lambda_m}{\mu_m}$ и $\rho_u = \frac{\lambda_u}{\mu_u}$ [4].

Схема модели со снижением и прерыванием обслуживания показана на рис. 1.

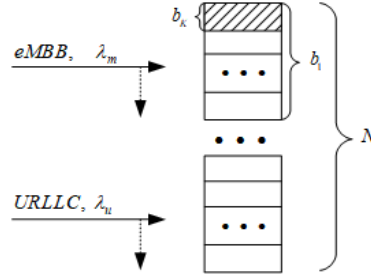


Рис. 1. Схема модели со снижением скорости и прерыванием обслуживания трафика eMBB

Используя системную модель, опишем процесс совместного обслуживания трафика eMBB и URLLC. Если C – максимальная нагрузка системы, то пространство состояний выглядит следующим образом:

$$N = \left\{ \begin{array}{l} (\bar{m}, n) = (m_1, \dots, m_k, n) : m_k \geq 0, \sum_{k=1}^K m_k \cdot b_k \leq C, \\ 0 \leq n \leq C, \sum_{k=1}^K m_k \cdot b_k + n \leq C, k = \overline{1, K}. \end{array} \right\}$$

Опишем возможные ситуации для обоих видов трафика. Так при возникновении события с интенсивностью λ_m , т.е. поступает запрос на обслуживание eMBB трафика:

- Если в системе есть хотя бы один свободный ресурсный блок, то новая сессия eMBB принимается к обслуживанию.
- Если в системе нет свободных ресурсных блоков, то новая сессия eMBB блокируется.

Возможные ситуации при возникновении события с интенсивностью λ_u , т.е. поступает запрос на обслуживание URLLC трафика:

- Если в системе есть хотя бы одна свободная ресурсная единица, то новая сессия URLLC принимается к обслуживанию.
- Если в системе нет свободной ресурсной единицы и в системе обслуживается хотя бы одна eMBB сессия не на минимальной скорости, тогда скорость обслуживания eMBB сессии снижается на одну скорость и URLLC сессия поступает на обслуживание.
- Если в системе нет свободной ресурсной единицы и в системе обслуживается как минимум одна eMBB сессия на минимальной скорости, то обслуживание eMBB сессии прерывается и URLLC сессия поступает на обслуживание.
- Если в системе нет свободной ресурсной единицы и нет активной eMBB сессии, то новая сессия URLLC блокируется.

Возможные ситуации при возникновении события с интенсивностью μ_m , т.е. обслуживание eMBB трафика окончено:

- Обслуживание сессии eMBB трафика прекращается и происходит освобождение ресурсов.

Возможные ситуации при возникновении события с интенсивностью μ_u , т.е. обслуживание URLLC трафика окончено:

- Если в системе не обслуживаются сессии eMBB или обслуживаются только на максимальной скорости, тогда обслуживание URLLC сессии прекращается и происходит освобождение ресурсов.
- Если в системе обслуживается как минимум одна eMBB сессия не на максимальной скорости, тогда, когда обслуживание URLLC сессии прекращается, происходит освобождение ресурсов и скорость активной сессии eMBB восстанавливается на единицу.

Диаграмма интенсивности переходов из центрального состояния представлена на рис. 2.

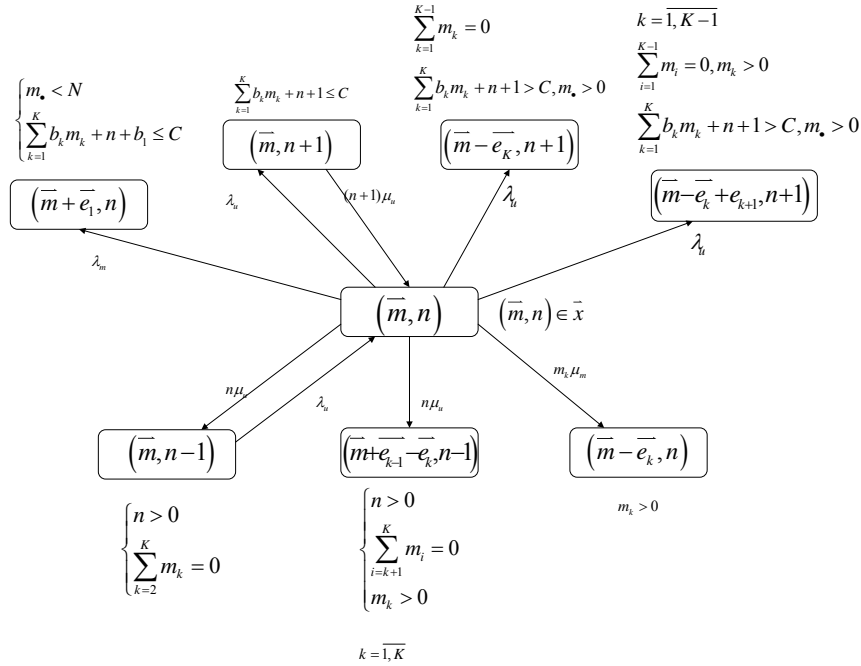


Рис. 2. Диаграмма интенсивности переходов

2. Вероятностно-временные характеристики

Рассматриваются следующие вероятностно-временными характеристики: среднее количество eMBB и URLLC сессий, вероятность блокировки запросов eMBB и URLLC сессий, средняя скорость сессий eMBB.

- Среднее количество сессий eMBB (\bar{m}_k) и URLLC (\bar{n}):

$$\bar{m}_k = \sum_{m_k=1}^N m_k \cdot (\sum p(\bar{m}, n)), \quad \bar{n} = \sum_{n=1}^C \left(n \cdot \sum_{\bar{m} \in \mathcal{N}} p(\bar{m}, n) \right).$$

- Вероятность блокировки запросов eMBB (B_m) и URLLC (B_u):

$$B_m = \sum_{(\bar{m}, n) \in M} p(\bar{m}, n),$$

где $M = (\bar{m}, n) \in \mathcal{N} : \sum_{i=1}^K m_i \cdot b_i + n + b_1 > C$, $B_u = p(\bar{0}, C)$.

- Средняя скорость сессий eMBB (\bar{b}):

$$\bar{b} = \frac{1}{\sum_{(\bar{m}, n) \in \mathcal{N}_0} p(\bar{m}, n)} \cdot \sum_{k=1}^K \sum_{(\bar{m}, n) \in \mathcal{N}_k} b_k \cdot p(\bar{m}, n),$$

где $\mathcal{N}_k = \{(\bar{m}, n) \in \mathcal{N} : \sum_{i=1}^{k-1} m_i = 0, m_k > 0, k = \overline{1, K}\}$.

Пространство состояний \mathcal{N} разбивается на k непересекающихся множеств, образуя множества \mathcal{N}_k , т.е. $\mathcal{N} = \bigcup_{k=0}^K \mathcal{N}_k$.

При расчете средней скорости eMBB заявок стоит рассматривать только те состояния, в которых присутствует как минимум одна активная eMBB сессия. Следовательно, не учитываются состояния, в которых отсутствуют eMBB заявки ($\mathcal{N} / \mathcal{N}_0$).

3. Задача оптимизации

Чтобы облегчить эффективную поддержку качества обслуживания (QoS) в беспроводных сетях следующего поколения, важно смоделировать беспроводной канал с точки зрения таких показателей QoS на уровне соединения, как скорость передачи данных, задержка и вероятность нарушения задержки. В то же время нельзя забывать и об уровне предоставления услуги (SLA – Service Level Agreement). К ключевым понятиям можно отнести: качество, доступность, обязанности. Эти дисциплины согласовываются между поставщиком и пользователем используемой услуги [5].

В связи с этим необходимо провести максимизацию средней скорости трафика eMBB в описанной выше модели. Получается, что параметр b^* , описываемый как значение, ниже которого не должна опускаться скорость заявок, не должен быть ниже начального b_K . Также важно учесть тот факт, что средняя скорость зависит и от начальных скоростных значений, т.е. $W = \{w_1, \dots, w_B\}$ и от количества K . По определению имеем начальное условие: $b_1 > b_2 > \dots > b_K$. Следовательно, описать задачу оптимизации средней скорости можно следующим образом:

$$\begin{aligned} & \bar{b}(K, b_1, \dots, b_K) \rightarrow \max \\ & \begin{cases} b_k \in W \{w_1, \dots, w_B\}, k = \overline{1, K}, \\ b_1 > b_2 > \dots > b_K, \\ b_K \geq b^*. \end{cases} \end{aligned}$$

Заключение

В работе рассмотрена модель совместного обслуживания двух видов трафика со снижением скорости и прерыванием обслуживания трафика eMBB. Представлены условия для распределения вероятностей в матричном виде, выведены формулы вероятностно-временных характеристик. Сформулирована задача максимизации средней скорости.

ЛИТЕРАТУРА

1. *Ishan Budhiraja, Sudhanshu Tyagi, Sudeep Tanwar, Neeraj Kumar, Joel J.P.C. Rodrigues* Tactile Internet for Smart Communities in 5G: An Insight for NOMA-based Solutions. – 2019.
2. 3GPP, 'Technical Specification Group Radio Access Network; Study on Scenarios and Requirements for Next Generation Access Technologies;', TR 38.913 Release 14, V.14.3.0, 2017.
3. *Makeeva E., Polyakov N., Kharin P., Gudkova I.* Probability Model for Performance Analysis of Joint URLLC and eMBB Transmission in 5G Networks. // Internet of Things, Smart Spaces, and Next Generation Networks and Systems. NEW2AN 2019, ruSMART 2019. Lecture Notes in Computer Science. – 2019. – V. 11660. – Springer, Cham. https://doi.org/10.1007/978-3-030-30859-9_55.
4. *Баширин Г.П.* Лекции по математической теории телеграфика. – М.: РУДН, 2004.
5. *Yan Chen, Toni Farley, Nong Ye* QoS Requirements of Network Applications on the Internet // Information Knowledge Systems Management 4. – 2004. – P. 55–76.

VI. ТЕОРИЯ МАССОВОГО ОБСЛУЖИВАНИЯ И ЕЕ ПРИЛОЖЕНИЯ

DOI: 10.17223/978-5-907442-42-9-2021-21

ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК РАБОТЫ ГИБРИДНОЙ ДВУХУРОВНЕВОЙ МОДЕЛИ ОБРАБОТКИ ЗАПРОСОВ

Воркожоков А.А., Лапатин И.Л.
Томский государственный университет
vorkozhokovandrey@gmail.com, ilapatin@mail.ru

Введение

Одним из вариантов систем массового обслуживания являются многоуровневые системы, состоящие из нескольких СМО. В частности, двухуровневые системы с повторными вызовами были впервые рассмотрены С.М. Krishna и Y.H.Lee [1].

Системы массового обслуживания с повторными вызовами [2,3] представляют собой математические модели телекоммуникационных систем: компьютерных и телефонных сетей, систем передачи данных, радио, телевидения, мобильной связи и других. В нашем случае заявки обслуживаются по алгоритму CSMA (Carrier-sense multiple access), который используется для обмена данными в компьютерных сетях. Когда станция готова к передаче данных, она прослушивает канал. Если канал оказывается свободным, происходит передача кадра. Если же канал занят, то станция ждет в течение случайного интервала времени, а затем прослушивает линию снова. Наш входящий поток представляет собой суммарный поток всех обращений к каналу. При этом мы рассматриваем дополнительную фазу, которая может быть интерпретирована как фаза форматирования пакета для передачи по каналу. Данная фаза является некоторым фильтром, который контролирует количество запросов, поступающих в канал. Поэтому правильное построение фазы фильтрации позволит наиболее эффективно использовать пропускную способность канала. Мы будем исследовать количество обслуженных запросов узлом связи, т.е. выходящий поток нашего узла. Исследование характеристик выходящего потока при условии большой задержки на орбите [4] необходимо для исследования сетевых структур, в которых выходящий поток одного узла является входящим для другого.

1. Описание математической модели

В данной работе исследуется двухуровневая система массового обслуживания, которая на первом уровне содержит конечную очередь и неограниченную орбиту на втором уровне.

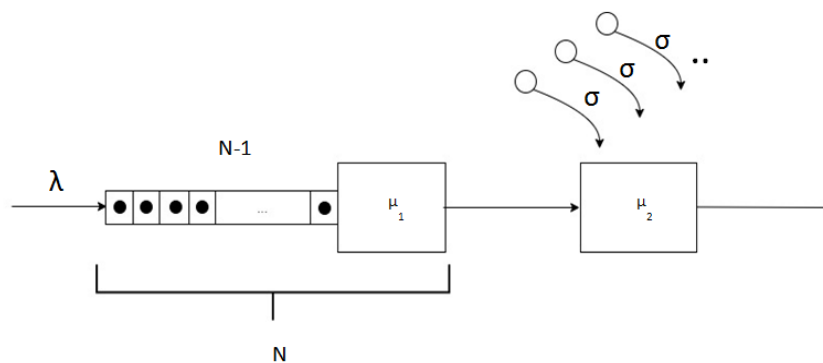


Рис. 1. Модель системы

На вход первого уровня приходит простейший поток с параметром λ . Заявка начинает обслуживаться в течение времени, распределённого экспоненциально с параметром μ_1 , если в очереди перед ней нет заявок и прибор свободен. Если прибор занят и очередь не заполнена, то заявка встаёт в конец очереди. Если на первой фазе уже находится N заявок, то вновь пришедшая заявка терается.

После обслуживания на первой фазе на вход второго уровня приходит поток заявок. Поступая на вторую фазу и обнаруживая прибор свободным, заявка занимает его, а прибор начинает обслуживание в течение времени, распределённого экспоненциально с параметром μ_2 . Если же заявка, поступая в систему, обнаруживает прибор занятым, она мгновенно уходит на орбиту и осуществляет там случайную задержку в течение экспоненциально распределённого времени с параметром σ .

После обслуживания на двух уровнях заявка попадет в выходящий поток. Исследование характеристик именно выходящего потока является особенностью данной работы.

Опишем систему с помощью случайных процессов: $k(t)$ – число заявок на первой фазе в момент времени t , $i(t)$ – число заявок на орбите в момент времени t , $l(t)$ – состояния прибора на второй фазе в момент времени t (0 – прибор свободен, 1 – прибор занят), $m(t)$ – число обслуженных заявок в момент времени t .

Теперь запишем распределение вероятностей для многомерного случайного процесса:

$$P\{k(t) = k, i(t) = i, l(t) = l, m(t) = m\} = P_l(k, i, m, t).$$

Составим равенства, используя теорему о полной вероятности:

$$\begin{aligned} P_0(k, i, m, t + \Delta t) &= P_0(k, i, m, t)(1 - \lambda\Delta t)(1 - \mu_1\Delta t)(1 - i\sigma\Delta t) + \\ &+ P_1(k, i, m - 1, t)\mu_2\Delta t + P_0(k - 1, i, m, t)\lambda\Delta t + o(\Delta t), \\ P_1(k, i, m, t + \Delta t) &= -P_1(k, i, m, t)(\lambda + \mu_1 + \mu_2)(1 - \lambda\Delta t)(1 - \mu_1\Delta t)(1 - \mu_2\Delta t) + \\ &+ P_0(k + 1, i, m, t)\mu_1\Delta t + P_1(k + 1, i - 1, m, t)\mu_1\Delta t + P_0(k, i + 1, m, t)(i + 1)\sigma\Delta t + \\ &+ P_1(k - 1, i, m, t)\lambda\Delta t + o(\Delta t). \end{aligned} \quad (1)$$

Далее, для того, чтобы описать переходы системы из одного состояния в другое, нам необходимо составить систему дифференциальных уравнений Колмогорова. Чтобы это сделать нужно в системе (1) представить $\Delta t \rightarrow 0$.

Составим систему дифференциальных уравнений Колмогорова:

$$\begin{aligned} \frac{\partial P_0(k, i, m, t)}{\partial t} &= -P_0(k, i, m, t)(\lambda + \mu_1 + i\sigma) + P_1(k, i, m - 1, t)\mu_2 + P_0(k - 1, i, m, t)\lambda, \\ \frac{\partial P_1(k, i, m, t)}{\partial t} &= -P_1(k, i, m, t)(\lambda + \mu_1 + \mu_2) + P_0(k + 1, i, m, t)\mu_1 + \\ &+ P_1(k + 1, i - 1, m, t)\mu_1 + P_0(k, i + 1, m, t)(i + 1)\sigma + P_1(k - 1, i, m, t)\lambda. \end{aligned} \quad (2)$$

Введём частичные характеристические функции:

$H_l(k, u_1, u_2, t) = \sum_{i=0}^{\infty} \sum_{m=0}^{\infty} e^{ju_1 i} \cdot e^{ju_2 m} \cdot P_l(k, i, m, t)$. Подставим их в систему уравнений (2):

$$\begin{aligned} \frac{\partial H_0(k, u_1, u_2, t)}{\partial t} &= \frac{\partial H_0(k, u_1, u_2, t)}{\partial u_1} j\sigma - H_0(k, u_1, u_2, t)\lambda - H_0(k, u_1, u_2, t)\mu_1 + \\ &+ H_1(k, u_1, u_2, t)\mu_2 e^{ju_2} + H_0(k - 1, u_1, u_2, t)\lambda, \\ \frac{\partial H_1(k, u_1, u_2, t)}{\partial t} &= -H_1(k, u_1, u_2, t)(\lambda + \mu_1 + \mu_2) + H_0(k + 1, u_1, u_2, t)\mu_1 + \\ &+ H_1(k + 1, u_1, u_2, t)\mu_1 e^{ju_1} - \frac{\partial H_0(k, u_1, u_2, t)}{\partial u_1} j\sigma e^{-ju_1} + H_1(k - 1, u_1, u_2, t)\lambda. \end{aligned} \quad (3)$$

Теперь представим систему в матрично-векторном виде. Так как процесс $k(t)$ ограничен, мы можем утверждать, что параметр k принимает значения от 0 до N . Введём вектор характеристических функций размерностью $N+1$, где вместо параметра k запишем его значения:

$$\mathbf{H}_l(u_1, u_2, t) = \{H_l(0, u_1, u_2, t), H_l(1, u_1, u_2, t), \dots, H_l(N, u_1, u_2, t)\}, \quad (4)$$

и матрицы размерностью $(N+1) \times (N+1)$:

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}, \quad \mathbf{B}^T = \begin{pmatrix} 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \lambda & 0 & \dots & 0 & 0 \\ 0 & \lambda + \mu_1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda + \mu_1 & 0 \\ 0 & 0 & \dots & 0 & \mu_1 \end{pmatrix}. \quad (5)$$

Подставим (4) и (5) в систему (3):

$$\begin{aligned} \frac{\partial \mathbf{H}_0(u_1, u_2, t)}{\partial t} &= \frac{\partial \mathbf{H}_0(u_1, u_2, t)}{\partial u_1} j\sigma - \mathbf{H}_0(u_1, u_2, t) [\lambda \mathbf{B}^T - \mathbf{A}] + \mathbf{H}_1(u_1, u_2, t) \mu_2 e^{ju_2}, \\ \frac{\partial \mathbf{H}_1(u_1, u_2, t)}{\partial t} &= \mathbf{H}_1(u_1, u_2, t) [\lambda \mathbf{B}^T - \mathbf{A} - \mu_2 \mathbf{I} + \mu_1 e^{ju_1} \mathbf{B}] + \\ &+ \mu_1 \mathbf{H}_0(u_1, u_2, t) \mathbf{B} - \frac{\partial \mathbf{H}_0(u_1, u_2, t)}{\partial u_1} j\sigma e^{-ju_1}. \end{aligned} \quad (6)$$

2. Асимптотический анализ

Для исследования характеристик выходящего потока, возьмём условие большого времени задержки заявки на орбите.

Сделаем замену:

$$\sigma = \varepsilon, \quad u_1 = \varepsilon w, \quad \mathbf{H}_l(u_1, u_2, t) = \mathbf{F}_l(w, u_2, t, \varepsilon). \quad (7)$$

Подставим (7) в систему (6):

$$\begin{aligned} \frac{\partial \mathbf{F}_0(w, u_2, t, \varepsilon)}{\partial t} &= \frac{\partial \mathbf{F}_0(w, u_2, t, \varepsilon)}{\partial w} j - \mathbf{F}_0(w, u_2, t, \varepsilon) [\lambda \mathbf{B}^T - \mathbf{A}] + \mathbf{F}_1(w, u_2, t, \varepsilon) \mu_2 e^{ju_2}, \\ \frac{\partial \mathbf{F}_1(w, u_2, t, \varepsilon)}{\partial t} &= \mathbf{F}_1(w, u_2, t, \varepsilon) [\lambda \mathbf{B}^T - \mathbf{A} - \mu_2 \mathbf{E} + \mu_1 e^{j\varepsilon w} \mathbf{B}] + \\ &+ \mu_1 \mathbf{F}_0(w, u_2, t, \varepsilon) \mathbf{B} - \frac{\partial \mathbf{F}_0(w, u_2, t, \varepsilon)}{\partial w} j e^{j\varepsilon w}. \end{aligned}$$

Теорема. Асимптотическое приближение двумерной характеристической функции числа обслуженных заявок входящего потока за некоторое время t имеет вид

$$\begin{aligned} \lim_{\sigma \rightarrow 0} M e^{ju_2 m} &= \lim_{\varepsilon \rightarrow 0} \sum_{l=0}^1 \mathbf{H}_l(0, u_2, t) \mathbf{E} = \lim_{\varepsilon \rightarrow 0} \sum_{l=0}^1 \mathbf{F}_l(0, u_2, t, \varepsilon) \mathbf{E} = \\ &= \sum_{l=0}^1 \mathbf{F}_l(u_2, t) \mathbf{E} = \mathbf{R} e^{\mathbf{J}(u_2)t} \mathbf{E} \mathbf{E}, \end{aligned} \quad (8)$$

где $\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2\}$ – стационарное распределение вероятностей, κ – среднее число заявок на орбите [5], \mathbf{E} – вектор размерностью $N+1$, $\mathbf{E}\mathbf{E}$ – вектор размерностью $(2N+2) \times (2N+2)$, $\mathbf{J}(u_2) = \begin{bmatrix} \kappa \mathbf{I} - \lambda \mathbf{B}^T + \mathbf{A} & \mu_1 \mathbf{B} - \kappa \mathbf{I} \\ \mu_2 e^{ju_2} \mathbf{I} & \lambda \mathbf{B}^T - \mathbf{A} - \mu_2 \mathbf{I} + \mu_1 \mathbf{B} \end{bmatrix}$.

Чтобы перейти от характеристических функций обратно к распределению, нам необходимо воспользоваться обратным преобразованием Фурье:

$$P\{m(t) = m\} = P(m, t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-ju_2 m} \mathbf{R} e^{J(u_2) t} \mathbf{E} \mathbf{E} du_2. \quad (9)$$

Заключение

В ходе работы была рассмотрена гибридная система с очередью на первой фазе и орбитой на второй. Для описания модели была составлена система дифференциальных уравнений Колмогорова (2). С использованием метода частичных характеристических функций и метода асимптотического анализа была получена формула (8), являющаяся асимптотическим приближением характеристической функции числа событий, наступивших в выходящем потоке. С применением к полученной формуле (8) обратного преобразования Фурье было получено распределение вероятностей числа событий, наступивших в выходящем потоке за время t (9).

ЛИТЕРАТУРА

1. Krishna C.M., Lee Y.H. A study of two-phase service // Operations Research Letters. – 1990. – V. 9. – P. 91–97.
2. Kendall D.G. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain. – The Annals of Mathematical Statistics, 1953. – P. 338–354.
3. Artalejo J.R., Gomez-Corral A. Retrial Queueing Systems: A Computational Approach. – Springer, 2008. – 309 p.
4. Nawel Gharbi, Claude Dutheillet An algorithmic approach for analysis of finite-source retrial systems with unreliable servers // Computers & Mathematics with Applications. – 2011. – V. 62, Iss. 6. – P. 2535–2546.
5. Назаров А.А., Анисимова А.А. Асимптотический анализ первого порядка двухфазной СМО с конечной очередью и орбитой в условии большой задержки заявок на орбите. // Молодежная научная школа по прикладной теории вероятностей и телекоммуникационным технологиям (АРТСТ–2017): материалы молодежной научной школы. Россия, Москва, 23–27 октября 2017 г.; под общ. ред. К.Е. Самуйлова, Е.А. Кучерявого, А.Н. Дудина. – Москва : РУДН, 2017. – С. 175–177.

DOI: 10.17223/978-5-907442-42-9-2021-22

ИССЛЕДОВАНИЕ НАДЕЖНОСТИ ДУБЛИРОВАННОЙ СИСТЕМЫ С ПОЛНЫМ ВОССТАНОВЛЕНИЕМ ПРИ ПРОИЗВОЛЬНЫХ ДЛИТЕЛЬНОСТЯХ ЖИЗНИ И РЕМОНТА ЕЁ ЭЛЕМЕНТОВ*

Иванова Н.М.^{1,2}, Нибасумба Э.¹

¹ Российский университет дружбы народов

² Институт проблем управления им. В.А. Трапезникова РАН
nm_ivanova@bk.ru, ema.patiri2015@yandex.ru

Введение

Теория регенерирующих процессов была предложена Смитом в 1955 г. [1] в качестве нового метода исследования случайных процессов. Эта теория нашла множество обобщений (см., например, [2,3]) ввиду широкого круга применений к прикладным задачам. Исследование систем с произвольными распределениями времен жизни и ремонта их компонент важно как с теоретической точки зрения, так и прикладной. В [4] для функций распределения длительностей безотказной работы и ремонта были введены модифицированные преобразования Лапласа-Стилтьеса. На основе этого преобразования в [5] с помощью теории разложимых полугенерирующих процессов [6] были найдены аналитические выражения для основных характеристик надежности данной системы при сценарии частичного ремонта. Настоящая работа продолжает ис-

* Публикация выполнена при поддержке Программы стратегического академического лидерства РУДН и при финансовой поддержке РФФИ в рамках научного проекта № 20-01-00575А (Иванова Н.М., проведение аналитических расчетов).

следование надежности дублированной системы холодного резервирования, но при сценарии полного восстановления. Вычисляются такие характеристики надежности, как функция надежности и среднее время жизни системы, а также нестационарные и стационарные вероятности состояний при произвольных распределениях длительностей жизни и восстановления.

1. Постановка задачи и обозначения

Рассмотрим систему холодного резервирования, состоящую из двух компонент и одной ремонтной единицы. Каждая из компонент может находиться в двух состояниях: рабочем и отказовом, при этом вся система отказывает, когда отказывают обе её компоненты.

Обозначим через A_i , B_i и C_i ($i = 1, 2, \dots$) последовательность независимых одинаково распределенных (н.о.р.) случайных величин (с.в.) времени безотказной работы (в.б.р.), частичного и полного ремонта элементов и системы соответственно. Эти с.в. имеют функции распределения (ф.р.) $A(t) = P\{A_i \leq t\}$, $B(t) = P\{B_i \leq t\}$ и $C(t) = P\{C_i \leq t\}$. Предположим, что мгновенные отказы и восстановления невозможны и их средние значения конечны:

$$A(0) = B(0) = C(0) = 0,$$

$$a = \int_0^{\infty} (1 - A(x)) dx < \infty, \quad b = \int_0^{\infty} (1 - B(x)) dx < \infty, \quad c = \int_0^{\infty} (1 - C(x)) dx < \infty.$$

Введем следующие обозначения.

1. Преобразования Лапласа – Стильтеса (ПЛС) ф.р. времен жизни и ремонта

$$\tilde{a}(s) = \int_0^{\infty} e^{-sx} dA(x), \quad \tilde{b}(s) = \int_0^{\infty} e^{-sx} dB(x), \quad \tilde{c}(s) = \int_0^{\infty} e^{-sx} dC(x).$$

2. Модифицированные преобразования Лапласа – Стильтеса (МПЛС)

$$\tilde{a}_B(s) = \int_0^{\infty} e^{-sx} B(x) dA(x), \quad \tilde{b}_A(s) = \int_0^{\infty} e^{-sx} A(x) dB(x).$$

3. Соответствующие усеченные средние

$$a_B = -\tilde{a}'_B(0) = \int_0^{\infty} xB(x) dA(x), \quad b_A = -\tilde{b}'_A(0) = \int_0^{\infty} xA(x) dB(x).$$

4. Вероятности $P\{B \leq A\}$ и $P\{B > A\}$ связаны с преобразованиями через следующие соотношения

$$\tilde{a}_B(0) = \int_0^{\infty} B(x) dA(x) = P\{B \leq A\} \equiv p, \quad \tilde{b}_A(0) = \int_0^{\infty} A(x) dB(x) = P\{B > A\} \equiv q = 1 - p.$$

Рассмотрим случайный процесс $J = \{J(t), t \geq 0\}$, где $J(t)$ – количество неисправных компонент в момент времени t . Множество всех возможных состояний системы обозначим как $E = \{i = 0, 1, 2\}$, где i – число отказавших компонент. Исследование процесса J проводится с помощью теории разложимых полу-регенерирующих процессов (РПРП). Обозначим через F и $F(t) = P\{F \leq t\}$ время до полного отказа системы и соответствующую ф.р. Таким образом, в работе вычисляются функция надежности $R(t) = P\{F > t\} = 1 - F(t)$, нестационарные вероятности состояний системы $\pi_j(t) = P\{J(t) = j\}$, $j = 0, 1, 2$, а также стационарные $\pi_j = \lim_{t \rightarrow \infty} \pi_j(t) = \lim_{t \rightarrow \infty} P\{J(t) = j\}$, $j = 0, 1, 2$.

2. Функция надежности и среднее время жизни системы

Процесс J , траектория которого представлена на рис 1, является регенерирующим (р.п.) с задержкой. Пусть W – это время до отказа системы из состояния 1, F – это время до полного отказа системы, G – это длина цикла регенерации. Вычислим соответствующие функции распределения.

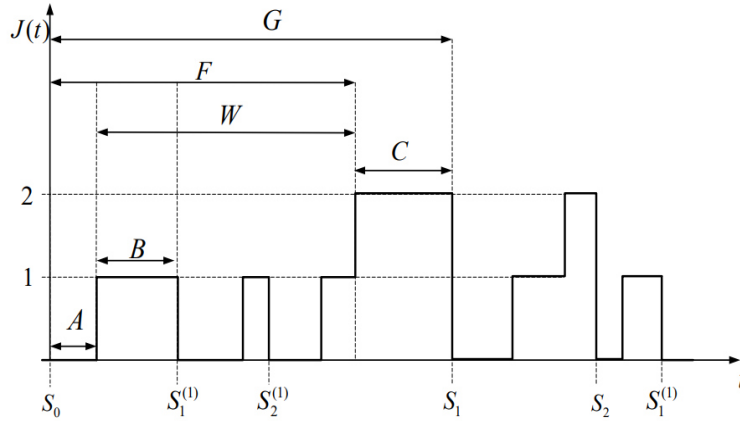


Рис. 1. Траектория процесса J

Из рис. 1 видно, что W удовлетворяет следующему стохастическому уравнению

$$W = \begin{cases} A + W, & B < A, \\ A, & B \geq A, \end{cases}$$

тогда как F и G имеют вид соответственно $F = A + W$ и $G = C + F$.

Лемма 1. ПЛС плотности распределений времени между отказами $\tilde{w}(s) = E[e^{-sW}]$ и времени до полного отказа системы $\tilde{f}(s) = E[e^{-sF}]$ имеют вид

$$\tilde{w}(s) = \frac{\tilde{a}(s) - \tilde{a}_B(s)}{1 - \tilde{a}_B(s)}, \quad \tilde{f}(s) = \tilde{a}(s) \frac{\tilde{a}(s) - \tilde{a}_B(s)}{1 - \tilde{a}_B(s)}. \quad (1)$$

Лемма 2. Производящая функция времени между двумя последовательными посещениями состояния 0 после отказа всей системы $\tilde{g}(s) = E[e^{-sG}]$ определяется как

$$\tilde{g}(s) = \tilde{a}(s)\tilde{c}(s) \frac{\tilde{a}(s) - \tilde{a}_B(s)}{1 - \tilde{a}_B(s)}. \quad (2)$$

После вычисления распределения времени до отказа системы легко перейти к функции надежности $R(t) = P\{F > t\}$ с помощью второго выражения из (1).

Теорема 1. ПЛ функции надежности $\tilde{R}(s) = \frac{1}{s} - \tilde{F}(s)$ системы холодного дублирования при произвольных длительностях отказов и ремонта имеет вид

$$\tilde{R}(s) = \frac{(1 - \tilde{a}(s))(1 + \tilde{a}(s) - \tilde{a}_B(s))}{s(1 - \tilde{a}_B(s))}.$$

Следствие 1. Среднее время безотказной работы системы $E[F] = \lim_{s \rightarrow 0} \tilde{R}(0)$ и среднее время периода регенерации $E[G] = -\frac{d}{ds} \tilde{g}(s)|_{s=0}$ принимают вид соответственно

$$E[F] = a + \frac{a}{q}, \quad E[G] = a + c + \frac{a}{q}.$$

3. Исследование процесса на периоде регенерации

Переходим к исследованию поведения процесса на отдельном периоде регенерации. Моменты регенерации $S_0 = 0$, $S_1 = S_0 + G_1$, ..., $S_n = S_{n-1} + G_n$, ... являются моментами возвращения процесса в состояние 0 после полного восстановления системы. Здесь G_n ($n = 1, 2, \dots$) – последовательность н.о.р. с.в. интервалов времени между двумя последовательными возвратами системы в состояние 0 после полного отказа.

Вероятности состояний процесса J в любой момент времени t $\pi_j(t) = P\{J(t) = j\}$, $j = 0, 1, 2$, могут быть выражены в терминах их же распределения на отдельном периоде регенерации $\pi_j^{(1)}(t) = P\{J(t) = j, t < G\}$, $j = 0, 1, 2$, а также функции восстановления

$H(t) = \sum P\left\{\left[\sum_{1 \leq i \leq n} G_i\right] \leq t\right\} = \sum_{n \geq 1} G^{*n}(t)$, где $G(t) = P\{G_i \leq t\}$ – ф.р. с.в. G_i ($i = 1, 2, \dots$). Тогда

$$\pi_j(t) = \pi_j^{(1)}(t) + \int_0^t \pi_j^{(1)}(t-u) dH(u). \quad (3)$$

Согласно теории восстановления, ПЛС функции восстановления $H(t)$ определяется с помощью (2) как $\tilde{h}(s) = \frac{\tilde{g}(s)}{1 - \tilde{g}(s)}$, где $\tilde{h}(s) = \int_0^\infty e^{-st} dH(t)$. Переходя к ПЛ для (3) с учетом $\tilde{h}(s)$, получаем

$$\tilde{\pi}_j(s) = (1 + \tilde{h}(s)) \tilde{\pi}_j^{(1)}(s) = \frac{1 - \tilde{a}_B(s)}{1 - \tilde{a}(s) + (\tilde{a}(s) - \tilde{a}_B(s))(1 - \tilde{a}(s)\tilde{c}(s))} \tilde{\pi}_j^{(1)}(s), \quad j = 0, 1, 2. \quad (4)$$

Далее необходимо вычислить вероятности на отдельном периоде регенерации $\pi_j^{(1)}(t)$.

4. Нестационарные вероятности состояний системы

Согласно теории РПП, если поведение процесса на основном периоде регенерации достаточно сложно и удаётся найти так называемые вложенные моменты регенерации, то распределение процесса на основном периоде регенерации определяется по его распределению на вложенных периодах регенерации и вложенной функции восстановления.

Вероятности состояний процесса J на отдельном периоде регенерации вычисляются следующим образом:

$$\pi_j^{(1)}(t) = (1 - \delta_{j2}) \pi_j^{(F)}(t) 1_{\{t < F\}} + \delta_{j2} \pi_j^{(C)}(t) 1_{\{F < t < G\}}. \quad (5)$$

Событие $\{J(t) = 2, t > G\}$ возникает тогда и только тогда, когда $\{F \leq t < F + C\}$, таким образом, при $j = 2$ уравнение (5) принимает следующий вид:

$$\pi_2^{(1)}(t) = P\{F \leq t < F + C\} = \int_0^t dF(u)(1 - C(t-u)).$$

Лемма 3. ПЛ вероятности $\tilde{\pi}_2^{(1)}(s)$ на периоде регенерации вычисляется как

$$\tilde{\pi}_2^{(1)}(s) = \tilde{f}(s) \cdot \frac{1 - \tilde{c}(s)}{s} = \frac{\tilde{a}(s)(\tilde{a}(s) - \tilde{a}_B(s))}{1 - \tilde{a}_B(s)} \cdot \frac{1 - \tilde{c}(s)}{s}. \quad (6)$$

Процесс J является р.п. с задержкой, поэтому для вычисления распределения вероятностей для состояний $j = 0, 1$ представим период F в виде двух интервалов A и W . Тогда

$$\pi_0^{(F)}(t) = P\{J(t) = 0, t < F\} = P\{t < A\} + \int_0^t dA(u)\pi_0^{(W)}(t-u),$$

$$\pi_1^{(F)}(t) = P\{J(t) = 1, t < F\} = \int_0^t dA(u)\pi_0^{(W)}(t-u).$$

В терминах ПЛС соответствующих функций получаем

$$\tilde{\pi}_0^{(F)}(s) = \frac{1 - \tilde{a}(s)}{s} + \tilde{a}(s)\tilde{\pi}_0^{(W)}(s), \quad \tilde{\pi}_1^{(F)}(s) = \tilde{a}(s)\tilde{\pi}_0^{(W)}(s). \quad (7)$$

При наличии вложенных моментов регенерации $S_k^{(1)}$ распределение на периоде W вероятностей первого уровня $\pi_j^{(W)}(t)$ аналогично предыдущему и выражается через распределения вероятностей на вложенных периодах регенерации второго уровня $\pi_j^{(2)}(t)$ и вложенную функцию восстановления $H^{(W)}(t)$ в виде

$$\pi_j^{(W)}(t) = \pi_j^{(2)}(t) + \int_0^t dH^{(W)}(u)\pi_0^{(2)}(t-u), \quad j = 0, 1, \quad (8)$$

где $\pi_j^{(W)}(t) = P\{J(t) = j, t < G^{(1)}\}$, $j = 0, 1$. В рассматриваемом случае в качестве вложенных моментов регенерации $S_k^{(1)}$ рассмотрим случайное число $\nu = \min\{n : A_n < B_n\}$ моментов: $S_1^{(1)} = A_1 1_{\{A_1 > B_1\}}$, $S_2^{(1)} = S_1^{(1)} + A_2 1_{\{A_1 > B_1, A_2 > B_2\}}$, ..., до тех пор, пока событие $\{A_n < B_n\}$ не произойдет впервые. Таким образом, распределение $G^{(1)}(t) = A(t)$, а вложенная функция восстановления $H^{(W)}(t)$ удовлетворяет уравнению

$$H^{(W)}(t) + W(t) = A(t) + \int_0^t dH^{(W)}(u)A(t-u). \quad (9)$$

Лемма 4. Переходя к ПЛ в (8) и (9) по определению, получаем

$$\tilde{\pi}_j^{(W)}(s) = (1 + \tilde{h}^{(W)}(s))\tilde{\pi}_j^{(2)}(s) = \frac{1}{1 - \tilde{a}_B(s)}\tilde{\pi}_j^{(2)}(s). \quad (10)$$

Вычислим вероятности на втором уровне регенерации для состояний 0, 1. Из рис. 1 можно определить следующие события:

1. Событие $\{J(t) = 0, t < G^{(1)}\}$ имеет место тогда и только тогда, когда $\{B < t < A\}$, откуда

$$\pi_0^{(2)}(t) = P\{B < t < A\} = B(t)(1 - A(t)).$$
2. Событие $\{J(t) = 1, t < G^{(1)}\}$ имеет место тогда и только тогда, когда $\{t < B < A\}$ или $\{t < A < B\}$, откуда

$$\pi_1^{(2)}(t) = P\{t < B < A\} + P\{t < A < B\} = \int_t^\infty (1 - A(u))dB(u) + \int_t^\infty (1 - B(u))dA(u).$$

Лемма 5. ПЛ вероятностей на втором уровне регенерации имеет вид

$$\tilde{\pi}_0^{(2)}(s) = \frac{1}{s} [\tilde{b}(s) - (\tilde{a}_B(s) + \tilde{b}_A(s))], \quad \tilde{\pi}_1^{(2)}(s) = \frac{1}{s} [1 - (\tilde{a}(s) + \tilde{b}(s)) + \tilde{a}_B(s) + \tilde{b}_A(s)]. \quad (11)$$

Таким образом, собирая полученные результаты (6)–(7), (10)–(11) и подставив их в определение (4), получаем следующую теорему.

Теорема 2. ПЛС нестационарных вероятностей состояний системы холодного дублирования при сценарии полного ремонта системы и произвольных распределениях времени жизни и ремонта принимает вид:

$$\begin{aligned}\tilde{\pi}_0(s) &= \frac{1}{s} \cdot \frac{1 - \tilde{a}_B(s) - \tilde{a}(s)(1 - \tilde{b}(s) + \tilde{b}_A(s))}{1 - \tilde{a}(s) + (\tilde{a}(s) - \tilde{a}_B(s))(1 - \tilde{a}(s)\tilde{c}(s))}, \\ \tilde{\pi}_1(s) &= \frac{1}{s} \cdot \frac{\tilde{a}(s)(1 - (\tilde{a}(s) + \tilde{b}(s) + \tilde{a}_B(s) + \tilde{b}_A(s)))}{1 - \tilde{a}(s) + (\tilde{a}(s) - \tilde{a}_B(s))(1 - \tilde{a}(s)\tilde{c}(s))}, \\ \tilde{\pi}_2(s) &= \frac{1}{s} \cdot \frac{\tilde{a}(s)(1 - \tilde{c}(s))(\tilde{a}(s) - \tilde{a}_B(s))}{1 - \tilde{a}(s) + (\tilde{a}(s) - \tilde{a}_B(s))(1 - \tilde{a}(s)\tilde{c}(s))}.\end{aligned}\tag{12}$$

5. Стационарные вероятности

Стационарные вероятности состояний процесса можно получить из (12) с помощью предельного перехода: $\pi_j = \lim_{t \rightarrow 0} \pi_j(t) = \lim_{s \rightarrow 0} s\tilde{\pi}_j(s)$.

Теорема 3. Стационарные вероятности состояний системы холодного дублирования при сценарии полного ремонта системы и произвольных распределениях времени жизни и ремонта принимает вид

$$\pi_0 = \frac{aq - (a_B + b_A) - b}{a + q(a + c)}, \quad \pi_1 = \frac{a + b + (a_B + b_A)}{a + q(a + c)}, \quad \pi_2 = \frac{cq}{a + q(a + c)}.\tag{13}$$

Замечание 1. Рассмотрим Марковскую модель $A(t) = 1 - e^{-\alpha t}$, $B(t) = 1 - e^{-\beta t}$, $C(t) = 1 - e^{-\gamma t}$, где α, β, γ – интенсивности в.б.р., частичного и полного восстановления элементов и системы. Тогда стационарные вероятности состояний системы примут вид

$$\pi_0 = \frac{(\alpha + \beta)\gamma}{\alpha^2 + 2\alpha\gamma + \beta\gamma}, \quad \pi_1 = \frac{\alpha\gamma}{\alpha^2 + 2\alpha\gamma + \beta\gamma}, \quad \pi_2 = \frac{\alpha^2}{\alpha^2 + 2\alpha\gamma + \beta\gamma}.$$

Эти же вероятности можно получить путем замены ПЛ на экспоненциальное распределение длительностей жизни и ремонта в (13).

Заключение

В данной работе была исследована надежность дублированной системы холодного резервирования с полным восстановлением. С помощью модифицированных преобразований Лапласа – Стильтеса и теории разложимых полурегенерирующих процессов были найдены основные характеристики надежности системы. Для произвольных распределений в.б.р., частичного и полного ремонта вычислены функция надежности и среднее время жизни системы, а также ее нестационарные и стационарные вероятности состояний системы.

ЛИТЕРАТУРА

1. *Smith W.L.* Regenerative stochastic processes. // Proc. R. Soc. – 1955. – Ser. A. – 232. – DOI:10.1098/rspa.1955.0198
2. *Cinlar E.* On semi-Markov processes on arbitrary space // Proc. Camb. Philos., Math. Proc. Camb. Philos. Soc. – 1969. – 66. – P. 381–392.
3. *Nummelin E.* Uniform and ratio-limit theorems for Markov-renewal and semi-regenerative processes on a general state space // Ann. inst. Henri Poincaré B. – 1978. – 14. – P. 119–143.
4. *Rykov V.* On Reliability of a Double Redundant Renewable System. // Springer Nature Switzerland AG 2020. M. Gribaudo et al. (Eds.): ASMTA 2019, LNCS 12023. – P. 1–9. – https://doi.org/10.1007/978-3-030-62885-7_3
5. *Rykov V, Efrosinin D, Stepanova N, Sztrik J.* On Reliability of a Double Redundant Renewable System with a Generally Distributed Life and Repair Times. Mathematics. – 2020. – 8(2):278. – <https://doi.org/10.3390/math8020278>
6. *Rykov V.V.* Decomposable Semi-Regenerative Processes and Their Applications // Lap Lambert Academic Publishing: Berlin, Germany. – 2011. –P. 75.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РЕСУРСНОЙ СМО, ФУНКЦИОНИРУЮЩЕЙ В СЛУЧАЙНОЙ СРЕДЕ

Кришталева Н.А., Моисеев А.Н., Полин Е.П.

Томский государственный университет

krishtalevnik@gmail.com, moiseev.tsu@gmail.com, polin_evgeny@mail.ru

Введение

В настоящий момент большой интерес в теории массового обслуживания представляют модели, функционирующие в случайной среде [1,2], т.к. они позволяют описывать динамически изменяющиеся системы. Кроме того, часто требуется моделировать такие процессы, где кроме временных затрат на обслуживание встречаются некоторые дополнительные, например, занимаемая оперативная или дисковая память, занимаемый ресурс радиоканала связи, потребности в дополнительном оборудовании или ресурсах. Такие системы массового обслуживания (СМО) называют ресурсными [3,4].

При исследовании ресурсных систем, функционирующих в случайной среде, и других сложных моделей СМО, получить точные аналитические результаты оказывается не всегда возможным, поэтому одним из основных методов исследования таких моделей является имитационное моделирование. Кроме того, нередко в теории массового обслуживания используются асимптотические методы, которые позволяют получить аппроксимации вероятностных характеристик исследуемых процессов. В этом случае имитационное моделирование позволяет подтвердить достоверность аппроксимаций, оценить их погрешность и область применимости.

В настоящей работе представлена архитектура разработанного приложения для имитационного моделирования ресурсной системы массового обслуживания М/М/∞, функционирующей в случайной среде.

1. Математическая модель ресурсной СМО М/М/∞, функционирующей в случайной среде

Рассмотрим систему массового обслуживания с неограниченным числом приборов и неограниченным количеством некоторого ресурса. Функционирование системы зависит от состояния случайной среды. Случайная среда задана цепью Маркова с непрерывным временем, пространством состояний $s \in \{1, \dots, S\}$ и инфинитезимальной матрицей Q . На вход поступает пуассоновский поток заявок с интенсивностью λ_s , зависящей от состояния среды. Время обслуживания заявки на каждом приборе подчинено экспоненциальному распределению с параметром μ_s , также зависящим от состояния среды. Кроме того, каждая поступающая заявка формирует запрос на получение ресурса случайного объема $v_s \geq 0$ с функцией распределения вероятностей $G_s(y) = P\{v_s < y\}$, которая зависит от состояния среды.

Пусть $s(t)$ – состояние среды в момент времени t , $s \in \{1, \dots, S\}$, q_{ks} – интенсивность перехода среды из состояния k в состояние s ($k \neq s$), $i(t)$ – число заявок в системе в момент времени t , $v(t)$ – суммарный объем занятого ресурса, $p_s(i, v, t) = P\{i(t) = i, v(t) < v, s(t) = s\}$ – вероятность того, что в момент времени t в системе находится i заявок, суммарный объем занятого ресурса меньше v , а среда находится в состоянии s .

При смене состояния среды меняются: параметр μ_s экспоненциального распределения времени обслуживания заявок (все заявки обслуживаются с новым временем обслуживания), интенсивность входящего потока λ_s , функция распределения объема за-

нимаемого ресурса (занятые ресурсы перераспределяются заново по закону $G_s(y) = P\{v_s < y\}$ с новым значением s).

Когда заявка завершает обслуживание, она покидает систему и освобождает ресурс, который она заняла во время захвата (под захватом понимается момент прихода заявки или смены состояния среды, при котором произойдет перераспределение ресурса).

Целью исследования является получение стационарного распределения суммарного объема ресурса $p(v, t)$, занимаемого всеми заявками.

2. Разработка приложения для имитационного моделирования системы

Для имитационного моделирования подобных систем наиболее подходящим является дискретно-событийный подход [4]. На рис. 1 представлена объектная модель разрабатываемого приложения.

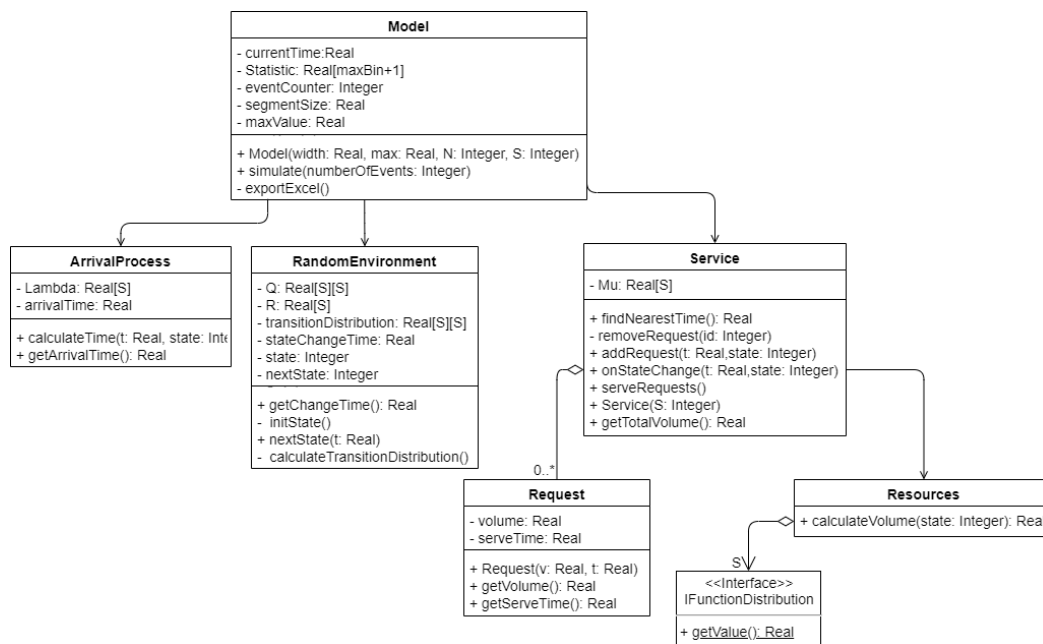


Рис. 1. Объектная модель СМО

Класс RandomEnvironment представляет собой реализацию случайной среды, ее функционирование определяется цепью Маркова с непрерывным временем и задается инфинитезимальной матрицей Q . Метод calculateTransitionDistribution(...) на основе матрицы Q вычисляет вероятности переходов. Метод initState(...) находит вектор r стационарного распределения состояний среды из системы уравнений

$$\begin{cases} rQ = 0, \\ re = 1, \end{cases}$$

и на основе этого вектора определяет начальное состояние случайной среды. Метод nextState(...) генерирует время смены состояния среды, а также определяет следующее состояние.

Класс Request представляет собой реализацию заявки. Имеет атрибут volume – объем занимаемого ресурса, и атрибут serveTime – время завершения обслуживания.

Класс Resources содержит массив функций распределения объема для каждого состояния. Операция calculateVolume(...) используется для генерации случайной величини

ны объема по соответствующей текущему состоянию случайной среды функции распределения.

Класс Service представляет собой реализацию пула серверов, отвечает за хранение массива заявок и моделирование процессов их обслуживания. Метод addRequest(...) принимает текущее время и состояние среды, генерирует для заявки время завершения обслуживания, распределенное по экспоненциальному закону с параметром, зависящим от состояния среды, а также вызывает метод calculateVolume(...) у класса ResourceSystem для генерации объема ресурса для этой же заявки, затем добавляет заявку в массив. Операция findNearestTime(...) находит среди всех заявок в пуле такую, которая имеет ближайшее к текущему время завершения обслуживания, и возвращает это время как результат. Операция onStateChange(...) должна вызываться классом Model, когда происходит смена состояния среды. В этом случае необходимо для каждой заявки переопределить время завершения обслуживания и заново перераспределить ресурс в соответствии с текущим состоянием среды. Операция serveRequests() удаляет из массива заявку с наименьшим временем завершения обслуживания. Операция getTotalVolume(...) считает суммарный объем занимаемого ресурса и возвращает его как результат.

Класс ArrivalProcess представляет собой пуассоновский поток поступающих заявок с интенсивностью зависящей от состояния случайной среды. Операция calculateTime(...) генерирует время прихода следующей заявки в систему.

Класс Model представляет систему в целом и отвечает за обработку событий, сбор статистики. Метод simulate(...) принимает количество событий, которые будут обработаны во время симуляции; сравнивает время наступления различных событий и запускает соответствующий сценарий, добавляет в статистику текущее значение суммарного объема занимаемого ресурса, после вызывает приватный метод exportToExcel(...) для экспорта результата моделирования в файл формата MS Excel. Диаграмма последовательностей для метода simulate(...) представлена на рис. 2.

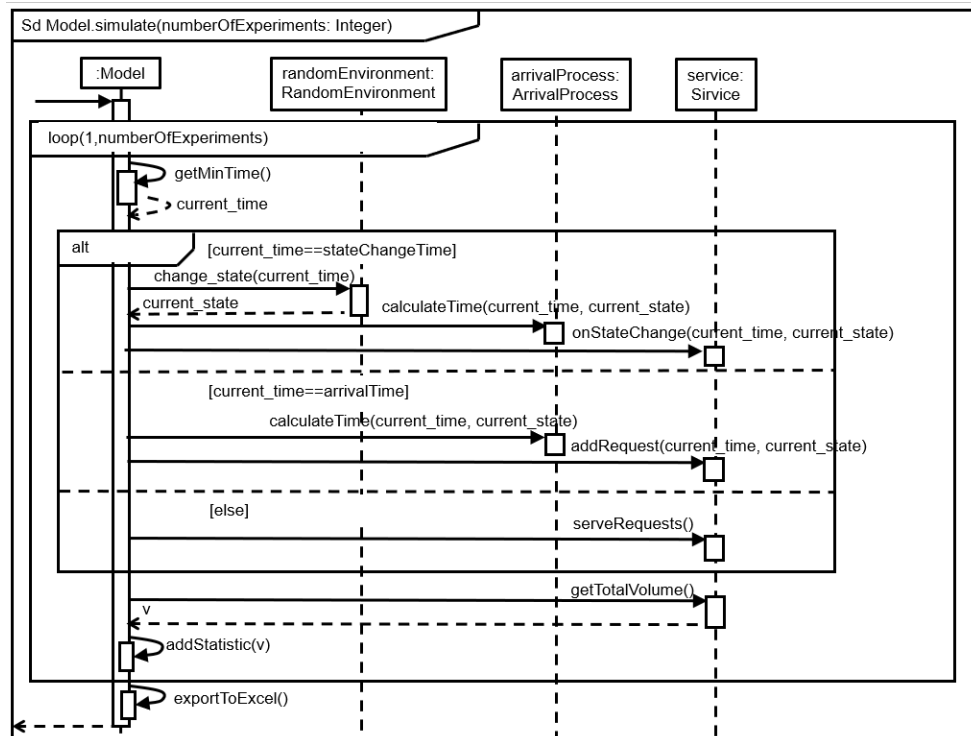


Рис. 2. Метод simulate(...) класса Model

Заключение

В результате проделанной работы было спроектировано и реализовано приложение для имитационного моделирования ресурсной системы массового обслуживания $M/M/\infty$, функционирование которой зависит от состояния случайной среды. Приложение успешно используется для научных исследований специалистами по теории массового обслуживания Томского государственного университета.

ЛИТЕРАТУРА

1. Назаров А.А., Баймеева Г.В. Исследование системы $M/M/\infty$ в полумарковской случайной среде // Известия вузов. Физика. – 2015. – Т. 58. – № 11/2. – С. 198–203.
2. Полин Е.П., Моисеева С.П., Рожкова С.В. Асимптотический анализ неоднородной системы массового обслуживания $M/M/\infty$ в марковской случайной среде // Вестн. Том. гос. ун-та. УВТиИ. – 2019. – № 47. – С. 75–83.
3. Чернышова Е.Н., Галлейская А.А., Лисовская Е.Ю. Суммарный объем занятого ресурса в системе обработки информации с реализацией хранения резервной копии сообщений // Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем: материалы Всероссийской конференции с международным участием, Москва, 13-17 апреля 2020 г. – М.: Изд-во РУДН, 2020. – С. 59–61.
4. Лисовская Е.Ю., Моисеев А.Н., Моисеева С.П., Пагано М. Моделирование процессов обработки данных физических экспериментов в виде немарковской многоресурсной системы массового обслуживания // Известия вузов. Физика. – 2018. – Т. 61. – № 12. – С. 39–46.

DOI: 10.17223/978-5-907442-42-9-2021-24

ИМИТАЦИОННАЯ МОДЕЛЬ ТАНДЕМНОЙ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ С ПОВТОРНЫМИ ВЫЗОВАМИ

Морозова М.А., Благинин А.Л.

Томский государственный университет

morozova_mariya_a@mail.ru, alex-b.l@yandex.ru

Введение

В рамках дипломной работы мы проводим исследование числа заявок на орбите тандемной системы с повторными вызовами и двумя последовательно обслуживающими приборами методом асимптотического анализа. Для того, чтобы определить область применимости полученного решения, было решено построить имитационную модель исследуемой системы. Метод имитационного моделирования [1] представляет из себя компьютерный эксперимент [2], который осуществляется на базе программного продукта, воссоздающего структуру и принципы работы исследуемой системы. Данный подход позволяет получить статистический материал, необходимый для построения эмпирического распределения вероятностей, которое можно сравнить с результатами, полученными аналитически.

Целью преддипломной практики является реализация программы имитационной модели тандемной RQ-системы с двумя последовательно обслуживающими приборами для нахождения эмпирического распределения вероятностей числа заявок на орбите.

Для достижения цели были поставлены следующие задачи:

- определить методику проведения имитационного моделирования; спроектировать архитектуру имитационной модели исследуемой системы;
- определить набор инструментов для реализации;
- реализовать спроектированную имитационную модель; оценить точность результатов имитационного моделирования.

В первом разделе содержится описание математической модели и постановка задачи, во втором сказано о методике проведения имитационного моделирования, в третьем – основные структурные элементы программы имитационной модели, в четвертом говорится о реализации имитационной модели, в пятом – о точности результатов имитационного моделирования.

1. Описание математической модели и постановка задачи

Прежде чем перейти к проектированию, опишем исходную модель (рис. 1). Мы исследуем систему с повторными вызовами и двумя последовательно обслуживающими приборами, на вход которой поступает простейший поток заявок с интенсивностью λ . Если заявка входящего потока обнаруживает первый прибор свободным, то она встает на прибор и обслуживается в течении экспоненциально распределенного времени с параметром μ_1 , после чего обращается ко второму прибору. Если второй прибор свободен, заявка встает на него и обслуживается в течении экспоненциально распределенного времени с параметром μ_2 . Если заявка входящего потока застает первый прибор занятым, она мгновенно отправляется на орбиту, где после случайной экспоненциально-распределенной задержки с параметром σ снова пытается встать на обслуживание на первый прибор. Если заявка после обслуживания на первом приборе застает второй прибор занятым, она мгновенно отправляется на орбиту, где после случайной экспоненциально-распределенной задержки с параметром σ снова пытается встать на обслуживание на первый прибор.

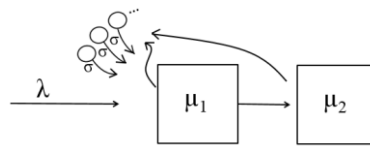


Рис. 1. Тандемная RQ-система

Ставится задача нахождения распределения вероятностей числа заявок на орбите.

2. Методика проведения имитационного моделирования

В основу метода имитационного моделирования было решено положить дискретно-событийный подход [3], который активно применяется для моделей массового обслуживания. В процессе функционирования в системе происходят события, которые переводят ее в разные состояния, и дискретно-событийный подход позволяет обойти моделирование интервалов времени, в которые состояние системы остается неизменным. Это способствует ускорению получения результатов, что немаловажно для проведения исследований.

К событиям в исследуемой системе относятся:

- 1) поступление заявки из входящего потока;
- 2) завершение обслуживания заявки на первом приборе,
- 3) завершение обслуживания заявки на втором приборе;
- 4) обращение заявки с орбиты к первому прибору.

Таким образом, в процессе моделирования модельное время постоянно корректируется, сдвигаясь после каждого события к моменту следующего, т.е., если обозначить T_1 – момент времени поступления заявки из входящего потока, T_2 – момент времени завершения на первом приборе, T_3 – момент времени завершения на втором приборе, T_4 – момент времени обращения заявки с орбиты к первому прибору, T – модельное время, то

$$T = \min(T_1, T_2, T_3, T_4). \quad (1)$$

Каждое из перечисленных событий предполагает выполнение определенных действий:

1. Поступление заявки из входящего потока: система пытается отправить заявку на обслуживание первым прибором. Если он свободен, то забирает данную заявку и определяет время события T_2 ; если прибор занят, то он возвращает предложенную заявку, которую система направляет на орбиту, определяющую событие T_4 для этой заявки.

2. Завершение обслуживания заявки на первом приборе: система пытается отправить заявку на обслуживание вторым прибором. Если он свободен, то забирает данную заявку и определяет время события T_3 ; если прибор занят, то он возвращает предложенную заявку, которую система направляет на орбиту, определяющую событие T_4 для этой заявки.
 3. Завершение обслуживания заявки на втором приборе: заявка покидает систему.
 4. Обращение заявки с орбиты к первому прибору: аналогично п.1.
- Завершение процесса моделирования происходит тогда, когда в системе произойдет заданное число событий.

3. Основные структурные элементы программы имитационной модели

Проектирование имитационной модели осуществлено на основе объектно-ориентированного подхода [4] с помощью языка моделирования UML [5]. В данной главе описаны основные структурные элементы программы, изображенные на диаграмме классов (рис. 2).

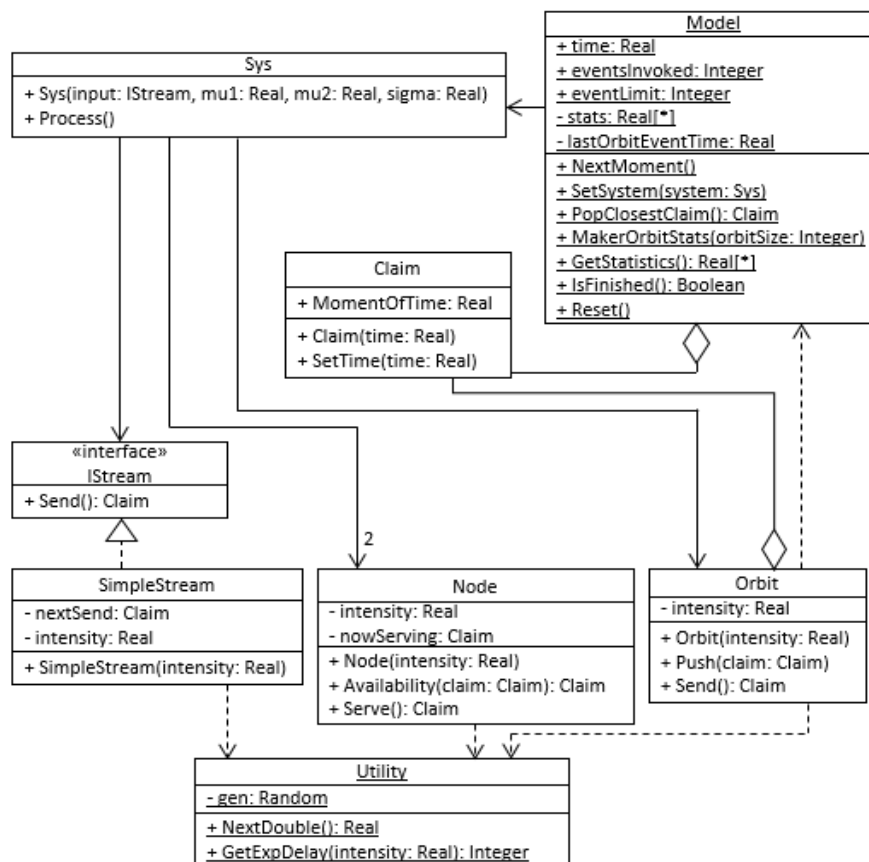


Рис. 2. Объектно-ориентированная модель базовых элементов автономного модуля для исследуемой тандемной RQ-системы

Начнем рассмотрение архитектуры с класса заявки (Claim). Он имеет единственный атрибут (MomentOfTime), который содержит момент времени, в который с этой заявкой должно произойти одно из событий T_1, T_2, T_3, T_4 .

Далее о реализации входящего потока. Поскольку в дальнейших исследованиях мы планируем рассматривать и другие (помимо простейшего) входящие потоки, было решено реализовать интерфейс IStream с методом Send(), который занимается созданием следующей заявки, которая поступит в систему, что характерно для любого потока.

У простейшего потока (SimpleStream) параметром конструктора является его интенсивность (intensity), на базе которой создается заявка, которая должна следующей поступить в систему (nextSend). Аналогичные генерации следующей заявки происходят в методе Send() после каждого фактического поступления заявки из потока.

У класса прибора (Node) есть два атрибута: интенсивность обслуживания (intensity) и обслуживаемая заявка (nowServing). Метод Availability(...) получает в качестве параметра заявку, пытающуюся попасть на прибор. Если прибор занят, заявка возвращается, а если свободен, то заявка встает на прибор и обслуживается (метод Serve()). По завершению обслуживания метод Serve() возвращает обслуженную заявку, а атрибут nowServing обнуляется.

В классе орбиты (Orbit) есть два атрибута: интенсивность повторных вызовов (intensity), которая является параметром конструктора орбиты, и коллекция заявок, находящихся на орбите. В методе Push(...) реализовано поступление заявки на орбиту, а в методе Send(), соответственно, уход с нее.

Связующим для вышеперечисленных классов является класс системы (Sys), который содержит в себе входной поток событий (input), два последовательно работающих прибора (Node1, Node2) и орбиту (Orbit), а также логику работы, исследуемой СМО (метод Process()).

Статический класс Utility предназначен для моделирования экспоненциального распределения. За отслеживание изменений в системе и сбор статистики отвечает статический класс Model. Свойство time – это условное «время», которое прошло с начала запуска моделирования или, другими словами, текущее время моделирования. Условным мы называем его по причине того, что оно не соответствует реальному, поскольку мы используем преимущества дискретно-событийного подхода, о котором говорили ранее при выводе (1).

С помощью метода PopClosestClaim() модель определяет, какое событие должно произойти следующим (заявка с минимальным значением поля MomentOfTime), а в методе NextMoment() производится переход к нему (согласно (1)) и происходит запуск основного процесса системы.

Далее, у класса Model есть свойства eventsInvoked и eventLimit. Первое является счетчиком для уже прошедших событий, второе содержит значение числа событий, при достижении которого моделирование завершается. Массив stats используется для накопления статистических данных о числе заявок на орбите в процессе моделирования.

Метод GetStatistics() вызывается после завершения моделирования и формирует массив, представляющий из себя искомое распределение вероятностей числа заявок на орбите.

В результате была спроектирована архитектура автономного модуля, она объединяет в себе все классы, о которых сказано в данном и предыдущем разделах.

4. Реализация имитационной модели

Имитационная модель реализована как десктопное приложение с пользовательским интерфейсом Windows Forms.

Для разработки был выбран язык C# в силу его объектной ориентированности, высокой популярности, автоматической «сборки мусора». Платформа реализации – Microsoft .NET Framework (версия 4.6.1) с огромной базой библиотек и шаблонов.

Среда разработки (интегрированная) – Visual Studio Community 2017.

На рис. 3 показано главное окно программы, где пользователь может задать параметры системы: интенсивность экспоненциального обслуживания приборов (Mu1 и Mu2), интенсивность экспоненциального обращения заявок с орбиты (Sigma), интенсивность входящего потока. Также пользователю предлагается задать число событий, при достижении которого процесс моделирования завершится (Event Limit).



Рис. 3. Интерфейс приложения

При нажатии кнопки Start запускается процесс моделирования и показывается соответствующее окно (рис. 4). Процесс можно прервать, нажав кнопку Cancel.

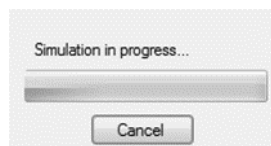


Рис. 4. Окно, свидетельствующее о процессе моделирования

Результатом работы программы является график распределения вероятностей числа заявок на орбите, а также в левом нижнем углу дополнительно посчитано математическое ожидание (Expected value) и дисперсия (Dispersion) для полученного распределения (рис. 5).

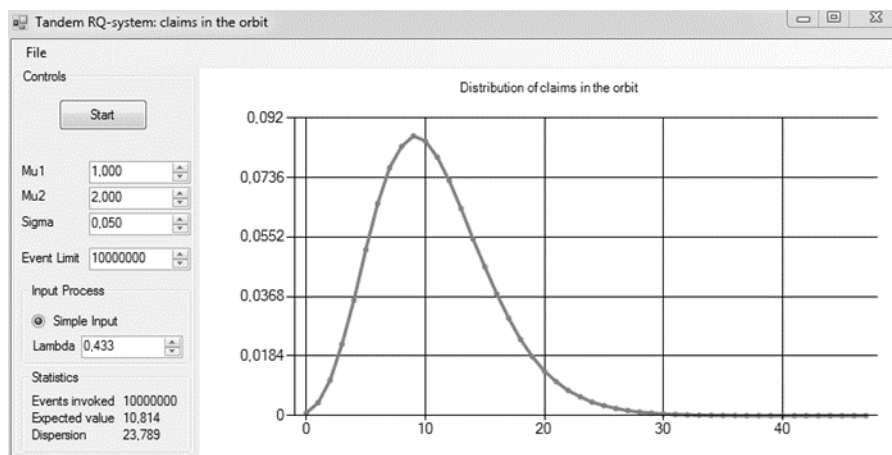


Рис. 5. Пример работы программы

Также в программе предусмотрено сохранение полученного распределения в файл формата .csv. Такие файлы удобно использовать в среде Mathcad для работы с полученными результатами. Чтобы сохранить распределение, нужно в верхней части главного окна нажать на кнопку File, где будет предложено сохранить файл.

5. Точность результатов имитационного моделирования

Прежде чем использовать имитационную модель, реализованную в рамках преддипломной практики, и говорить об области применимости полученных аналитически асимптотических результатов, следует сказать о надежности результатов имитационного моделирования, на базе которых мы оценим решение, полученное методом асимптотического анализа.

Как известно из закона больших чисел, чем больше повторений эксперимента, тем меньше отклонение эмпирического распределения вероятностей от «эталонного» результата, т.е. напрашивается вывод о том, что результатам имитационного моделирования можно доверять при объеме выборки, стремящемся к бесконечности. Однако на практике мы не можем работать с бесконечными выборками, поэтому встал вопрос о том, каким лучше задавать число моделируемых событий (Event Limit).

Чтобы количественно определять погрешность получаемых распределений, было решено применять расстояние Колмогорова, описываемое формулой

$$d = \max_{0 \leq i < \infty} \left| \sum_{v=0}^i (P1_i - P2_i) \right|,$$

где $P1_i$ и $P2_i$ – эмпирические вероятности числа заявок на орбите, полученные в результате работы имитационной модели.

Экспериментально было выявлено, что выборки объемом около $10^6 \times M$ (где M – среднее число заявок на орбите) гарантируют стабильность получаемых распределений: расстояние Колмогорова для них не превышает значения 0.005.

Стабильность работы имитационной модели свидетельствует о ее точности и, как следствие, возможности применимости ее для оценки асимптотических результатов.

Заключение

В результате работы была спроектирована и реализована имитационная модель тандемной системы с повторными вызовами и двумя последовательно обслуживающими приборами для нахождения распределения вероятностей числа заявок на орбите. Программа представляет себя десктопное приложение с пользовательским интерфейсом Windows Forms, написанное на языке программирования C#. В основу моделирования был положен дискретно-событийный подход, который позволяет значительно ускорить время сбора статистических данных, необходимых для получения искомого распределения.

Точность результатов моделирования определена по критерию расстояния Колмогорова, которое не превышает значения 0.005 при объеме выборки, превышающем математическое ожидание в 10^6 раз.

ЛИТЕРАТУРА

1. Лоу А., Кельтон В. Имитационное моделирование. 3-е изд. – СПб.: Питер, 2004. – 848 с.
2. Бахвалов Л.А. Моделирование систем: учеб. пособие для вузов. – М.: Изд-во Московского гос. горного ун-та, 2006. – 295 с.
3. Robinson S. Simulation: the practice of model development and use. – Hoboken: Wiley, 2004. – 336 p.
4. Буч Г., Максимчук Р.А., Энгл М.У. и др. Объектно-ориентированный анализ и проектирование с примерами приложений. пер. Д. Клошин. 3-е изд. – М.: Вильямс, 2010. – 720 с.
5. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. – СПб.: Питер, 2007. – 544 с.

ГИБРИДНАЯ СИСТЕМА МАССОВОГО ОБСЛУЖИВАНИЯ С ПОВТОРНЫМИ ВЫЗОВАМИ

Петрива Н.В., Пауль С.В.

Томский государственный университет
nata0338@yandex.ru, paulsv82@mail.ru

Введение

Теория массового обслуживания представляет собой теоретические основы эффективного конструирования и эксплуатации систем массового обслуживания (СМО). В основу теории массового обслуживания легла теория потоков однородных событий, разработанная советским математиком А.Я. Хинчиным [1]. СМО могут быть двух видов: с ожиданием или с потерями. В первом случае заявка, пришедшая в момент, когда нужный прибор занят, остается ждать момента обслуживания. Во втором случае она «покидает систему» и не требует внимания СМО. В середине 20-го века большую роль стали играть телекоммуникационные системы, для которых, в отличие от классических СМО, характерна ситуация, когда заявка, заставшая обслуживающий прибор занятым, не встает в очередь, а уходит на орбиту, откуда через некоторые промежутки времени предпринимает попытки вновь обратиться за обслуживанием. Такие модели описываются в виде СМО с повторными вызовами.

СМО с повторными вызовами – математическая модель, применяемая для анализа и оптимизации различных телекоммуникационных систем. Характерная черта данной модели заключается в наличии повторных обращений заявок к прибору после неудачной попытки обслуживания спустя некоторое случайное время. Такие ситуации могут быть вызваны не только отсутствием свободных серверов в моменты поступления заявок в систему, но техническими причинами.

В данной работе исследуется тандемная СМО, содержащая бункер с конечным числом мест для ожидания перед первой фазой и орбиту на второй фазе. Из-за различий дисциплин ожидания система была названа гибридной. Данная система массового обслуживания может применяться, например, при моделировании удаленного доступа к приложению/файлу/БД, в котором абоненты сначала авторизуются, после чего получают обновленные данные.

1. Математическая модель и постановка задачи

Рассмотрим систему массового обслуживания (рис. 1), на вход которой поступает простейший поток заявок с интенсивностью λ .

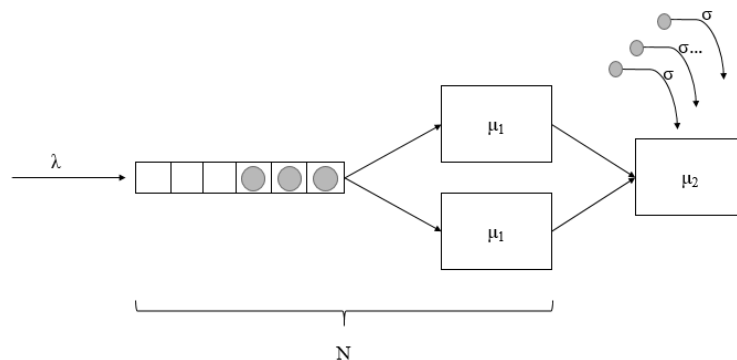


Рис. 1. Гибридная тандемная система массового обслуживания

Каждая заявка последовательно проходит две фазы обслуживания. Первая фаза системы – это система с двумя обслуживающими приборами и одним конечным бункером, в котором $N-2$ места для ожидания. Таким образом, на первой фазе одновременно может находиться не более N заявок. Вторая фаза – это система массового обслуживания с повторными вызовами и бесконечной орбитой. Если при поступлении заявки на первую фазу она обнаруживает, что хотя бы один из приборов свободен, то поступившая заявка занимает его и обслуживается время, распределенное по экспоненциальному закону с интенсивностью μ_1 . В противном случае, если оба прибора заняты, заявка попадает в бункер, где ожидает своей очереди обслуживания.

После обслуживания на первой фазе заявка переходит на вторую фазу, при этом, если она застаёт прибор на второй фазе свободным, она занимает его для обслуживания в течение случайного времени, распределенного по экспоненциальному закону с параметром μ_2 . В противном случае поступившая на вторую фазу заявка переходит на орбиту, где осуществляет случайную задержку, продолжительность которой имеет экспоненциальное распределение с параметром σ , после которой заявка вновь обращается к прибору с повторной попыткой его занять. Если прибор свободен, то заявка занимает его на случайное время обслуживания, если же он занят, то заявка мгновенно возвращается на орбиту для реализации следующей задержки случайной продолжительности с параметром σ . После обслуживания на второй фазе заявка покидает систему.

Найдем распределения вероятностей числа заявок на второй фазе методом асимптотического анализа [2] в условии большой задержки заявок на орбите на второй фазе [3].

1.1. Система уравнений Колмогорова

Обозначим: процесс $i_1(t)$ – число заявок на первой фазе, $i_2(t)$ – число заявок орбите, $k(t)$ определяет состояние прибора второй фазе следующим образом: 0 – прибор свободен, 1 – прибор занят.

Рассмотрим трехмерный Марковский процесс $\{i_1(t), i_2(t), k(t)\}$, для распределения вероятностей $P_k(i_1, i_2, t) = P\{i_1(t) = i_1, i_2(t) = i_2, k(t) = k\}$ которого составим систему уравнений Колмогорова в стационарном режиме:

$$i_1 = 0 :$$

$$\begin{aligned} & -(\lambda + i_2\sigma)P_0(0, i_2) + \mu_2P_1(0, i_2) = 0, \\ & -(\lambda + \mu_2)P_1(0, i_2) + \mu_1P_1(1, i_2 - 1) + \mu_1P_0(1, i_2) + (i_2 + 1)\sigma P_0(0, i_2 + 1) = 0, \end{aligned}$$

$$i_1 = 1 :$$

$$\begin{aligned} & -(\lambda + \mu_1 + i_2\sigma)P_0(1, i_2) + \mu_2P_1(1, i_2) + \lambda P_0(0, i_2) = 0, \\ & -(\lambda + \mu_1 + \mu_2)P_1(1, i_2) + \mu_1P_1(2, i_2 - 1) + \mu_1P_0(2, i_2) + \lambda P_1(0, i_2) + (i_2 + 1)\sigma P_0(1, i_2 + 1) = 0, \end{aligned}$$

$$1 < i_1 < N :$$

$$\begin{aligned} & -(\lambda + 2\mu_1 + i_2\sigma)P_0(i_1, i_2) + \mu_2P_1(i_1, i_2) + \lambda P_0(i_1 - 1, i_2) = 0, \\ & -(\lambda + 2\mu_1 + \mu_2)P_1(i_1, i_2) + 2\mu_1P_1(i_1 + 1, i_2 - 1) + 2\mu_1P_0(i_1 + 1, i_2) + \\ & \quad + \lambda P_1(i_1 - 1, i_2) + (i_2 + 1)\sigma P_0(i_1, i_2 + 1) = 0, \end{aligned}$$

$$i_1 = N :$$

$$\begin{aligned} & -(2\mu_1 + i_2\sigma)P_0(N, i_2) + \mu_2P_1(N, i_2) + \lambda P_0(N - 1, i_2) = 0, \\ & -(2\mu_1 + \mu_2)P_1(N, i_2) + \lambda P_1(N - 1, i_2) + (i_2 + 1)\sigma P_0(N, i_2 + 1) = 0. \end{aligned}$$

1.2. Метод частичных характеристических функций

Введем частичные характеристические функции: $H_k(i_1, u) = \sum_{i_2=0}^{\infty} e^{ju_2} P_k(i_1, i_2)$,

$j = \sqrt{-1}$. Получим систему уравнений для частичных характеристических функций:

$$\begin{aligned}
 i_1 = 0: & -\lambda H_0(0, u) + j\sigma \frac{\partial H_0(0, u)}{\partial u} + \mu_2 H_1(0, u) = 0, \\
 & -(\lambda + \mu_2) H_1(0, u) + \mu_1 e^{ju} H_1(1, u) + \mu_1 H_0(1, u) - j\sigma e^{-ju} \frac{\partial H_0(0, u)}{\partial u} = 0, \\
 i_1 = 1: & -(\lambda + \mu_1) H_0(1, u) + j\sigma \frac{\partial H_0(1, u)}{\partial u} + \mu_2 H_1(1, u) + \lambda H_0(0, u) = 0, \\
 & -(\lambda + \mu_1 + \mu_2) H_1(1, u) + \mu_1 e^{ju} H_1(2, u) + \mu_1 H_0(2, u) + \lambda H_1(0, u) - j\sigma e^{-ju} \frac{\partial H_0(1, u)}{\partial u} = 0, \\
 1 < i_1 < N: & -(\lambda + 2\mu_1) H_0(i_1, u) + j\sigma \frac{\partial H_0(i_1, u)}{\partial u} + \mu_2 H_1(i_1, u) + \lambda H_0(i_1 - 1, u) = 0, \\
 & -(\lambda + 2\mu_1 + \mu_2) H_1(i_1, u) + 2\mu_1 e^{ju} H_1(i_1 + 1, u) + 2\mu_1 H_0(i_1 + 1, u) + \lambda H_1(i_1 - 1, u) - \\
 & - j\sigma e^{-ju} \frac{\partial H_0(i_1, u)}{\partial u} = 0, \\
 i_1 = N: & -2\mu_1 H_0(N, u) + j\sigma \frac{\partial H_0(N, u)}{\partial u} + \mu_2 H_1(N, u) + \lambda H_0(N - 1, u) = 0, \\
 & -(2\mu_1 + \mu_2) H_1(N, u) + \lambda H_1(N - 1, u) - j\sigma e^{-ju} \frac{\partial H_0(N, u)}{\partial u} = 0.
 \end{aligned} \tag{1}$$

Представим систему (1) в векторно-матричном виде. Введём вектор: $\mathbf{H}_k(u) = [H_k(0, u) \ H_k(1, u) \ \dots \ H_k(N, u)]$, $k = 0, 1$ и матрицы

$$\mathbf{A} = \begin{pmatrix} \lambda & 0 & 0 & \dots & 0 \\ 0 & \lambda + \mu_1 & 0 & \dots & 0 \\ 0 & 0 & \lambda + 2\mu_1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 2\mu_1 \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 2 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}.$$

После подставим их в систему (1), получим систему уравнений в векторно-матричном виде:

$$\begin{aligned}
 \mathbf{H}_0(u) [\lambda \mathbf{B} - \mathbf{A}] + \mu_2 \mathbf{H}_1(u) + j\sigma \mathbf{H}'_0(u) &= 0, \\
 \mathbf{H}_1(u) [\lambda \mathbf{B} - \mathbf{A} - \mu_2 \mathbf{I} + \mu_1 e^{ju} \mathbf{D}] + \mu_1 \mathbf{H}_0(u) \mathbf{D} - j\sigma e^{-ju} \mathbf{H}'_0(u) &= 0.
 \end{aligned}$$

Просуммируем уравнения, тогда окончательно получим систему уравнений:

$$\begin{aligned}
\mathbf{H}_0(u)[\lambda\mathbf{B} - \mathbf{A}] + \mu_2\mathbf{H}_1(u) + j\sigma\mathbf{H}'_0(u) &= 0, \\
\mathbf{H}_1(u)[\lambda\mathbf{B} - \mathbf{A} - \mu_2\mathbf{I} + \mu_1e^{ju}\mathbf{D}] + \mu_1\mathbf{H}_0(u)\mathbf{D} - j\sigma e^{-ju}\mathbf{H}'_0(u) &= 0, \\
\mu_1\mathbf{H}_1(u)\mathbf{D}\mathbf{e} + j\sigma e^{-ju}\mathbf{H}'_0(u)\mathbf{e} &= 0.
\end{aligned} \tag{2}$$

Данная система является основной в дальнейших исследованиях.

1.3. Распределение вероятностей числа заявок на первой фазе в гибридной тандемной системе

Вектор \mathbf{r} с компонентами i_1 – распределение вероятностей числа заявок на первой фазе, имеет распределение

$$r(i_1) = \left(\frac{1}{2}\right)^{i_1-1} \left(\frac{\lambda}{\mu_1}\right)^{i_1} r(0), \quad i_1 = \overline{1, N}, \quad \sum_{i_1=0}^N r(i_1) = 1.$$

1.4. Асимптотика первого порядка

Систему (2) будем решать методом асимптотического анализа в условии большой задержки заявок на орбите ($\sigma \rightarrow 0$). Выполним в (2) следующие замены: $\sigma = \varepsilon$, $u = \varepsilon w$, $\mathbf{H}_k(u) = \mathbf{F}_k(w, \varepsilon)$, получим

$$\begin{aligned}
\mathbf{F}_0(w, \varepsilon)[\lambda\mathbf{B} - \mathbf{A}] + \mu_2\mathbf{F}_1(w, \varepsilon) + j \frac{\partial \mathbf{F}_0(w, \varepsilon)}{\partial w} &= 0, \\
\mathbf{F}_1(w, \varepsilon)[\lambda\mathbf{B} - \mathbf{A} - \mu_2\mathbf{I} + \mu_1e^{j\varepsilon w}\mathbf{D}] + \mu_1\mathbf{F}_0(w, \varepsilon)\mathbf{D} - je^{-j\varepsilon w} \frac{\partial \mathbf{F}_0(w, \varepsilon)}{\partial w} &= 0, \\
\mu_1\mathbf{F}_1(w, \varepsilon)\mathbf{D}\mathbf{e} + je^{-j\varepsilon w} \frac{\partial \mathbf{F}_0(w, \varepsilon)}{\partial w} \mathbf{e} &= 0.
\end{aligned}$$

Теорема 1. Пусть $i_2(t)$ – число заявок на орбите второй фазы в тандемной гибридной системе, тогда

$\lim_{\sigma \rightarrow 0} M e^{jw\sigma i_2(t)} = e^{jw\kappa_1}$, $\mathbf{r}_0[\lambda\mathbf{B} - \mathbf{A} - \kappa_1\mathbf{I}] + \mu_2\mathbf{r}_1 = 0$, $\mathbf{r}_1[\lambda\mathbf{B} - \mathbf{A} - \mu_2\mathbf{I} + \mu_1\mathbf{D}] + \mathbf{r}_0(\mu_1\mathbf{D} + \kappa_1\mathbf{I}) = 0$, κ_1 является решением уравнения $\mu_1\mathbf{r}_1\mathbf{D}\mathbf{e} - \mathbf{r}_0\kappa_1\mathbf{I}\mathbf{e} = 0$. Здесь \mathbf{r}_k , $k = 0, 1$ – векторы с компонентами r_{i_1k} , $i_1 = \overline{0, N}$, где r_{i_1k} – вероятность, того, что на первой фазе i_1 заявок, а на второй фазе прибор находится в состоянии k .

Таким образом, получили систему уравнений, которая позволит нам найти величину κ_1 .

Асимптотика первого порядка, т.е. доказанная теорема, определяет лишь асимптотическое среднее значение κ_1 числа заявок в системе на второй фазе. Для более детального исследования процесса $i_2(t)$ рассмотрим асимптотику второго порядка.

1.5. Асимптотика второго порядка

В системе (2) положим $\mathbf{H}_k(u) = \mathbf{H}_k^{(2)}(u)e^{ju\frac{\kappa_1}{\sigma}}$, получим

$$\begin{aligned}
\mathbf{H}_0^{(2)}(u)[\lambda\mathbf{B} - \mathbf{A} - \kappa_1\mathbf{I}] + \mu_2\mathbf{H}_1^{(2)}(u) + j\sigma\mathbf{H}_0^{(2)'}(u) &= 0, \\
\mathbf{H}_1^{(2)}(u)[\lambda\mathbf{B} - \mathbf{A} - \mu_2\mathbf{I} + \mu_1e^{ju}\mathbf{D}] + \mathbf{H}_0^{(2)}(u)(\mu_1\mathbf{D} + \kappa_1e^{-ju}\mathbf{I}) - j\sigma e^{-ju}\mathbf{H}_0^{(2)'}(u) &= 0, \\
\mu_1\mathbf{H}_1^{(2)}(u)\mathbf{D}\mathbf{e} - \kappa_1e^{-ju}\mathbf{H}_0^{(2)}(u)\mathbf{e} + j\sigma e^{-ju}\mathbf{H}_0^{(2)'}(u)\mathbf{e} &= 0.
\end{aligned}$$

Далее в системе сделаем следующую замену $\sigma = \varepsilon^2$, $u = \varepsilon w$, $\mathbf{H}_k^{(2)}(u) = \mathbf{F}_k^{(2)}(w, \varepsilon)$ и получим систему вида

$$\mathbf{F}_0^{(2)}(w, \varepsilon)[\lambda\mathbf{B} - \mathbf{A} - \kappa_1\mathbf{I}] + \mu_2\mathbf{F}_1^{(2)}(w, \varepsilon) + j\varepsilon \frac{\partial \mathbf{F}_0^{(2)}(w, \varepsilon)}{\partial w} = 0,$$

$$\mathbf{F}_1^{(2)}(w, \varepsilon) [\lambda \mathbf{B} - \mathbf{A} - \mu_2 \mathbf{I} + \mu_1 e^{jw} \mathbf{D}] + \mathbf{F}_0^{(2)}(w, \varepsilon) (\mu_1 \mathbf{D} + \kappa_1 e^{-jw} \mathbf{I}) - j \varepsilon e^{-jw} \frac{\partial \mathbf{F}_0^{(2)}(w, \varepsilon)}{\partial w} = 0,$$

$$\mu_1 \mathbf{F}_1^{(2)}(w, \varepsilon) \mathbf{D} \mathbf{e} - \kappa_1 e^{-jw} \mathbf{F}_0^{(2)}(w, \varepsilon) \mathbf{e} + j \varepsilon e^{-jw} \frac{\partial \mathbf{F}_0^{(2)}(w, \varepsilon)}{\partial w} \mathbf{e} = 0.$$

Теорема 2. Пусть $i_2(t)$ – число заявок на орбите второй фазе в тандемной гибридной системе, тогда $\lim_{\sigma \rightarrow 0} E \exp \left\{ jw \sqrt{\sigma} \left(i(t) - \frac{\kappa_1}{\sigma} \right) \right\} = \exp \left\{ \frac{(jw)^2}{2} \kappa_2 \right\}$, где κ_2 определяется равенством $\kappa_2 = \frac{\kappa_1 \mathbf{r}_0 \mathbf{e} - \kappa_1 \Phi_0 \mathbf{e} + \mu_1 \mathbf{D} \Phi_1 \mathbf{e}}{\mathbf{r}_0 \mathbf{e} + \kappa_1 \Psi_0 \mathbf{e} - \mu_1 \mathbf{D} \Psi_1 \mathbf{e}}$. Здесь векторы \mathbf{r}_k определены выше, величины $\Phi_0, \Phi_1, \Psi_0, \Psi_1$, определяются системами

$$\begin{aligned} \Phi_0 [\lambda \mathbf{B} - \mathbf{A} - \kappa_1 \mathbf{I}] + \mu_2 \Phi_1 &= 0, \\ \Phi_0 [\mu_1 \mathbf{D} + \kappa_1 \mathbf{I}] + \Phi_1 [\lambda \mathbf{B} - \mathbf{A} - \mu_2 \mathbf{I} + \mu_1 \mathbf{D}] &= \kappa_1 \mathbf{r}_0 \mathbf{I} - \mu_1 \mathbf{r}_1 \mathbf{D}, \\ \Psi_0 [\lambda \mathbf{B} - \mathbf{A} - \kappa_1 \mathbf{I}] + \mu_2 \Psi_1 &= \mathbf{r}_0, \\ \Psi_0 [\mu_1 \mathbf{D} + \kappa_1 \mathbf{I}] + \Psi_1 [\lambda \mathbf{B} - \mathbf{A} - \mu_2 \mathbf{I} + \mu_1 \mathbf{D}] &= -\mathbf{r}_0. \end{aligned}$$

Вторая асимптотика показывает, что асимптотическое распределение вероятностей числа $i_2(t)$ заявок на второй фазе является гауссовским распределением с математическим ожиданием κ_1/σ и дисперсией κ_2/σ .

Заключение

В результате исследования тандемной системы массового обслуживания, которая содержит конечный бункер на первой фазе и орбиту на второй фазе, было найдено распределения вероятностей числа заявок на орбите. Задача решена методом асимптотического анализа в условии большой задержки заявок на орбите.

ЛИТЕРАТУРА

1. Хинчин А.Я. Работы по математической теории массового обслуживания. Под редакцией Б.В. Гнеденко. – М.: 1963. – 236 с.
2. Назаров А.А., Пауль С.В., Лизюра О.Д. Асимптотический анализ RQ-системы с N типами вызываемых заявок в предельном условии большой задержки заявок на орбите // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2019. – № 48. – С. 13-20.
3. Судыко Е.А., Назаров А.А. Допредельные характеристики RQ-системы с конфликтами заявок // Научное творчество молодежи: материалы XIV Всероссийской научно-практической конференции, 15-16 апреля 2010 г. – Томск, 2010. – Ч. 1. – С. 97-100.

DOI: 10.17223/978-5-907442-42-9-2021-26

ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК РАБОТЫ НЕНАДЕЖНОГО УЗЛА ОБРАБОТКИ ЗАПРОСОВ

Решетников О.А., Лапатин И.Л.
Томский государственный университет
oleg.reshetnikov.0411@gmail.com, ilapatin@mail.ru

Введение

В век информационных технологий большой популярностью пользуются телекоммуникационные системы. Зачастую математической моделью таковых являются RQ-системы, т.е. системы с повторными вызовами. Появились они благодаря работам американских ученых R.I. Wilkinson и J.W. Cohen еще в середине XX в., в которых особое внимание уделялось решению практических задач, возникающих в телефонных сетях, и описанию влияния эффекта повторных вызовов на производительность технических

систем. Поскольку данный вопрос представляет реальный практический интерес, имеет смысл изучение, а также и возможное совершенствование уже существующих моделей.

В работе проводится исследование выходящего потока заявок системы М/М/1 с орбитой и прогулками приборов. Следует отметить, что под прогулками прибора подразумевается переключение прибора на обслуживание потока заявок другого типа или приоритета, т.е. не все системное время отводится под один конкретный поток.

Функционирует рассматриваемая система на основе протокола CSMA/CD (Carrier Sense Multiple Access with Collision Detection), в основе которого заложена методика обнаружения коллизий. В момент, когда станция готова к передаче данных, выполняется прослушивание канала. Если он оказывается свободным, происходит передача сигнала. В противном случае, станция ждет в течение случайного интервала времени, а затем прослушивает линию снова. Заявки, которые пытаются обратиться к узлу связи, формируют входящий поток и обращаются к нему строго по одной. Это позволяет избежать коллизий, при которых кадры накладываются друг на друга, в результате чего передача сообщения не может быть осуществлена.

Основным методом для исследования был выбран метод асимптотического анализа, который позволяет исследовать выходящий поток обслуженных заявок при условии большой задержки на орбите и установить вероятностно-временные характеристики выходящего потока.

Исследование характеристик выходящего потока представляет практический интерес, т.к. в результате формируется входящий поток данных уже для других узлов связи.

1. Математическая модель

На свободный и готовый к работе прибор с интенсивностью обслуживания заявок, имеющей экспоненциальное распределение с параметром μ , через входящий поток поступают заявки с интенсивностью λ . Если прибор занят, то заявка попадает на орбиту. На ней заявки задерживаются в течение экспоненциально-распределенного времени с параметром σ , после чего заново пытаются обратиться к прибору. Работающий прибор, занятый обслуживанием заявки, может в случайный момент времени «уйти на прогулку», т.е., к примеру, переключиться на обслуживание другого потока заявок или же выйти из строя на случайный промежуток времени. Интенсивность такого перехода – α_1 . В таком случае заявка уходит обратно на орбиту и через какое-то время снова попытается обратиться к прибору. Поступив на прибор, она будет обслуживаться с самого начала. Как только обслуживающее устройство завершает работу над заявкой, она передается на выход, а прибор в свободном состоянии ожидает обращения случайной заявки с орбиты, если она не пуста, либо ожидает поступления заявки из входящего потока, а также может переключиться на обслуживание другого потока заявок («уйти на прогулку»). Выходящие из обслуживающего устройства заявки формируют выходящий поток $m(t)$. Он характеризуется числом заявок, обслуженных к моменту времени t , т.е. за промежуток времени $[0, t]$. Считаем, что память не ограничена, т.е. все заявки учитываются и не теряются из-за исчерпания памяти.

Основной задачей работы является получение распределения вероятностей числа событий, наступивших в выходящем потоке системы с «прогулками» за время t .

2. Система дифференциальных уравнений Колмогорова

Поскольку исследуемый процесс $m(t)$ сам по себе не является марковским, т.к. зависит еще от состояния прибора, то в рассмотрение добавляется процесс $k(t)$, изменения которого зависят от наличия заявок на орбите, т.е. от значения процесса $i(t)$. Трехмерный процесс $\{k(t), i(t), m(t)\}$ является марковским, т.к., зная текущее состояние его компонент, можно с вероятностной точки определить будущее состояние, что согласуется с понятием марковского процесса.

Функционирование рассматриваемой модели описывается рядом случайных процессов:

1. $k(t)$ – состояние прибора в момент времени $t \in \{0,1,2\}$. 0 – прибор свободен, 1 – прибор занят обслуживанием поступившей заявки, 2 – прибор «ушел на прогулку».
2. $i(t)$ – количество заявок на орбите в момент времени $t \in [0, +\infty)$.
3. $m(t)$ – число обслуженных заявок за время $t \in [0, t]$.

Запишем вероятность того, что прибор в момент времени t находится в состоянии k , на орбите i заявок и к текущему моменту времени обслужено m заявок:

$$P\{k(t) = k, i(t) = i, m(t) = m\} = P_k(i, m, t).$$

Для распределения вероятностей составим систему дифференциальных уравнений Колмогорова:

$$\begin{cases} \frac{\partial P_0(i, m, t)}{\partial t} = -(\lambda + i\sigma + \alpha_1) P_0(i, m, t) + \mu P_1(i, m - 1, t) + \alpha_2 P_2(i, m, t), \\ \frac{\partial P_1(i, m, t)}{\partial t} = -(\lambda + \mu + \alpha_1) P_1(i, m, t) + \lambda P_1(i - 1, m, t) + \\ + (i + 1) \cdot \sigma P_0(i + 1, m, t) + \lambda P_0(i, m, t), \\ \frac{\partial P_2(i, m, t)}{\partial t} = -(\lambda + \alpha_2) P_2(i, m, t) + \alpha_1 P_0(i, m, t) + \alpha_1 P_1(i - 1, m, t) + \lambda P_2(i - 1, m, t). \end{cases} \quad (1)$$

Система не ограничивается тремя приведенными уравнениями, поскольку каждое представленное уравнение является системой бесконечного числа уравнений, т.к. значения i и m могут принимать неограниченное число состояний.

Введем частичные характеристические функции и обозначим $j = \sqrt{-1}$:

$$H_k(u_1, u_2, t) = \sum_{i=0}^{\infty} \sum_{m=0}^{\infty} e^{ju_1 i} e^{ju_2 m} P_k(i, m, t). \quad (2)$$

Применяя замену (2) к (1), получим

$$\begin{cases} \frac{\partial H_0(u_1, u_2, t)}{\partial t} = -(\lambda + \alpha_1) H_0(u_1, u_2, t) + j\sigma \frac{\partial H_0(u_1, u_2, t)}{\partial u_1} + \\ + \mu e^{ju_2} H_1(u_1, u_2, t) + \alpha_2 H_2(u_1, u_2, t), \\ \frac{\partial H_1(u_1, u_2, t)}{\partial t} = -(\lambda + \mu + \alpha_1) H_1(u_1, u_2, t) + \lambda e^{ju_1} H_1(u_1, u_2, t) - \\ - j\sigma e^{-ju_1} \frac{\partial H_0(u_1, u_2, t)}{\partial u_1} + \lambda H_0(u_1, u_2, t), \\ \frac{\partial H_2(u_1, u_2, t)}{\partial t} = -(\lambda + \alpha_2) H_2(u_1, u_2, t) + \alpha_1 H_0(u_1, u_2, t) + \\ + e^{ju_1} [\alpha_1 H_1(u_1, u_2, t) + \lambda H_2(u_1, u_2, t)]. \end{cases} \quad (3)$$

Полученную систему (3) будем решать методом асимптотического анализа при условии большой задержки на орбите, т.е. $\sigma \rightarrow 0$.

3. Метод асимптотического анализа

Введем обозначение $\sigma = \varepsilon$ и проведем замену в системе (3): $u_1 = \varepsilon w$, $H_k(u_1, u_2, t) = F_k(w, u_2, t, \varepsilon)$. Получим систему

$$\left\{ \begin{array}{l} \frac{\partial F_0(w, u_2, t, \varepsilon)}{\partial t} = -(\lambda + \alpha_1) F_0(w, u_2, t, \varepsilon) + \\ \quad + j \frac{\partial F_0(w, u_2, t, \varepsilon)}{\partial w} + \mu e^{ju_2} F_1(w, u_2, t, \varepsilon) + \alpha_2 F_2(w, u_2, t, \varepsilon), \\ \frac{\partial F_1(w, u_2, t, \varepsilon)}{\partial t} = -(\lambda + \mu + \alpha_1) F_1(w, u_2, t, \varepsilon) + \\ \quad + \lambda e^{j\varepsilon w} F_1(w, u_2, t, \varepsilon) - j e^{-j\varepsilon w} \frac{\partial F_0(w, u_2, t, \varepsilon)}{\partial w} + \lambda F_0(w, u_2, t, \varepsilon), \\ \frac{\partial F_2(w, u_2, t, \varepsilon)}{\partial t} = -(\lambda + \alpha_2) F_2(w, u_2, t, \varepsilon) + \alpha_1 F_0(w, u_2, t, \varepsilon) + \\ \quad + e^{j\varepsilon w} [\alpha_1 F_1(w, u_2, t, \varepsilon) + \lambda F_2(w, u_2, t, \varepsilon)]. \end{array} \right.$$

С использованием условия согласованности для многомерных распределений характеристическая функция процесса $m(t)$ выражается следующим образом через введенные функции:

$$\lim_{\sigma \rightarrow 0} M(e^{ju_2 m(t)}) = \lim_{\sigma \rightarrow 0} \sum_{k=0}^2 H_k(0, u_2, t) = \lim_{\varepsilon \rightarrow 0} \sum_{k=0}^2 F_k(0, u_2, t, \varepsilon) = \sum_{k=0}^2 F_k(u_2, t).$$

Теорема. Асимптотическое приближение характеристической функции числа обслуженных заявок входящего потока за время t при условии большой задержки на орбите имеет вид

$$\lim_{\sigma \rightarrow 0} M(e^{ju_2 m(t)}) = \sum_{k=0}^2 F_k(u_2, t) = F(u_2, t) = \mathbf{R} \cdot e^{\mathbf{M}(u_2)t} \cdot \mathbf{E}, \quad (4)$$

где $\mathbf{M}(u_2) = \begin{bmatrix} -(\lambda + \alpha_1 + \kappa) & \kappa + \lambda & \alpha_1 \\ \mu e^{ju_2} & -(\mu + \alpha_1) & \alpha_1 \\ \alpha_2 & 0 & -\alpha_2 \end{bmatrix}$, \mathbf{E} – единичный вектор-столбец соответствующей размерности, $\mathbf{R} = \{R_0, R_1, R_2\}$ – стационарное распределение вероятностей процесса $k(t)$ ($R_0 = \frac{\alpha_2(\mu + \alpha_1)}{(\alpha_1 + \alpha_2)(\lambda + \alpha_1 + \kappa + \mu)}$, $R_1 = \frac{\alpha_2(\lambda + \kappa)}{(\alpha_1 + \alpha_2)(\lambda + \alpha_1 + \kappa + \mu)}$, $R_2 = \frac{\alpha_1}{\alpha_1 + \alpha_2}$) κ – норми-

рованное среднее процесса $i(t)$, $\kappa = \frac{\lambda^2(\alpha_1 + \alpha_2) + \lambda\alpha_1(\mu + \alpha_1 + \alpha_2)}{\mu\alpha_2 - \lambda(\alpha_1 + \alpha_2)}$.

4. Распределение вероятностей числа событий, наступивших в выходящем потоке

Поскольку в ходе работы была получена формула (4), являющаяся асимптотическим приближением характеристической функции числа событий, наступивших в выходящем потоке, перейдем к распределению вероятностей числа событий, наступивших в выходящем потоке за время t . Для этого применим обратное преобразование Фурье к формуле (4):

$$P\{m(t) = m\} = P_m(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-ju_2 m} \mathbf{R} e^{\mathbf{M}(u_2)t} \mathbf{E} du_2. \quad (5)$$

Заключение

В ходе проделанной работы были изучены методы теории массового обслуживания, благодаря которым была построена математическая модель узла связи с орбитой и прогулками прибора. Рассматриваемая система представляет собой трехмерную цепь

Маркова с непрерывным временем. Для описания функционирования модели была составлена система дифференциально-конечноразностных уравнений Колмогорова (1). Как результат, с помощью метода частичных характеристических функций и модификации метода асимптотического анализа была получена формула (4), являющаяся асимптотическим приближением характеристической функции числа событий, наступивших в выходящем потоке системы М/М/1. Применяв к полученной формуле (4) обратное преобразование Фурье, было получено распределение вероятностей числа событий, наступивших в выходящем потоке за время t , формула (5).

DOI: 10.17223/978-5-907442-42-9-2021-27

ЗАДАЧА MARL ДЛЯ СВЕТОФОРА НА ПЕРЕКРЁСТКЕ

Тисленко Т.И.

Сибирский федеральный университет
timtisko@mail.ru

Введение

В Красноярске с каждым годом растет количество желающих стать автомобилистом. По данным агентства «Автостат» на 1 января 2020 г. Красноярский край оказался на 12 месте в топ-20 регионов по объему автомобильного парка в России. С количеством автомобилистов увеличивается и время, которое проводится в пробках. Таким образом, проблема оптимизации сети светофоров с целью уменьшения задержек трафика автомобилей является актуальной и востребованной в современных мегаполисах. Решение проблемы – согласование действий сети светофоров по управлению процессом выбора задержки сигнала. Внедрение адаптивных систем светофоров в городскую транспортную сеть может его уменьшить.

На данный момент существует множество адаптивных систем управления трафиком. Например, модель Split Cycle Offset Optimization Technique (SCOOT) в Великобритании [1]. При ее использовании транспортные средства обнаруживаются по мере приближения к перекрестку задолго до стоп-линии, затем с нескольких перекрестков сигналы об обнаружении передаются в центральную систему, которая моделирует поток транспорта в этом районе и производит адаптацию фаз светофора в соответствии с трафиком, тем самым сводя к минимуму ненужные зеленые фазы. Также к централизованному типу координации можно отнести систему Traffic-responsive Urban Control (TUC) в Греции [1,2]. Использование центральных систем координации в случае большого количества перекрестков ресурсозатратно, поэтому в качестве альтернативы возникли распределенные модели [1]: Sydney Coordinated Adaptive Traffic System (SCATS) в Австралии, Dynamic Programming Algorithm (PRODYN) и Real-time Hierarchical Optimized Distributed Effective System (RHODES). Приведенные системы оперируют количеством транспортных средств, проходящих на определенном отрезке полосы дороги, и поэтому их можно отнести к моделям, основанным на плотности потока.

Как альтернативный способ координации можно рассмотреть модели, основанные на марковских процессах принятия решений, которые могут не оперировать интенсивностью потока. В качестве математической модели сети светофоров в работе рассматривается управляемый марковский процесс с конечным числом действий и состояний [3,4]. Каждый агент (светофор) не располагает ресурсами и решает задачу целесообразности выбора того или иного сигнала. Среда – перекресток с машинами, где на отрезках дорог за 100 м до стоп-линий засекается время. Состояние среды отражает активность фазы светофора. Тогда проблема оптимизации сводится к задаче MARL (Multiagent Reinforcement Learning) [4]. Как правило, обучение с подкреплением используется для одного агента в среде, чтобы максимизировать его долгосрочную награду. Модель среды – марковский процесс принятия решения [5]. В работе исследуется

модель Q -обучения одного агента (светофора) в стационарной среде [6]. Q -обучение сводится к оптимальному сопоставлению агентом действия a состоянию окружающей среды s на основе кумулятивных наград $r(s,a)$. Предложенная математическая модель процесса выбора фазы светофора учитывает текущее расположение светофоров и их загрузки и позволяет сформулировать оптимизационные задачи, целью которых является минимизация задержки трафика автомобилей. Отметим, что структура мультиагентной системы, включающая в себя единственного агента – светофор, обеспечивает наиболее эффективное распараллеливание всей задачи на подзадачи, которые будут решены агентом.

1. Задача MARL для одного светофора

Пусть светофор – это агент, не располагающий ресурсами. В качестве среды будем рассматривать перекресток с машинами, где на отрезках дорог за 100 м до стоп-линий засекается время (рис. 1).



Рис. 1. Визуальная интерпретация модели

Считаем, что состояние среды отражает активность фазы светофора. Обозначим $S = \{s_0 = \text{«активна фаза 0»}, s_1 = \text{«активна фаза 1»}\}$ – пространство состояний агента, а множество решений агента как $A = \{a_0 = \text{«оставить фазу»}, a_1 = \text{«сменить фазу»}\}$. Диаграмма переходов для светофора представлена на рис. 2.

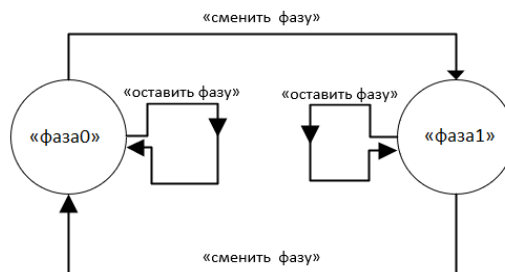


Рис. 2. Диаграмма переходов для задачи управления светофором

Пусть в момент времени t_k активна фаза светофора S_k . Под задержкой на фазе S_k будем понимать суммарное засечённое время всех машин, проходящих через отрезок дороги. Задержка на фазе s после действия a есть функция вознаграждения $r(s, a)$. Будем считать, что функция вознаграждения всецело определяется текущим состоянием, выбранной стратегией и состоянием, в которое перейдет процесс на следующем шаге. Далее предполагаем, что вероятность перехода $p(S_i, A_k; S_j)$ системы из состояния S_i при выборе решения A_k в состояние S_j полностью определяется состоянием, в которое переходит процесс.

Функция суммарных доходов от оптимальной политики в состоянии s в общем виде задается формулой [5]:

$$V^*(s) = \max_{A(\bullet)} \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t), \quad (1)$$

где γ – это коэффициент переоценки, а $A(\bullet)$ – управление.

Известно [4], что уравнение Вальда – Беллмана для управляемого марковского процесса с конечным числом действий и состояний в общем виде имеет вид

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} p(s, a; s') (r(s, a; s') + \gamma V^*(s')). \quad (2)$$

С учётом (1) и (2) задача MARL для одного светофора примет следующий вид.

Задача MARL для одного светофора

Требуется найти такое управление $A(\bullet)$, которое доставляет максимум функции

$$V^*(s) = \max \left\{ p(s, A_0; S_0) (r(s, A_0) + \gamma V^*(S_0)) + p(s, A_0; S_1) (r(s, A_0) + \gamma V^*(S_1)); \right. \\ \left. p(s, A_1; S_0) (r(s, A_1) + \gamma V^*(S_0)) + p(s, A_1; S_1) (r(s, A_1) + \gamma V^*(S_1)) \right\},$$

где функция $V^*(s)$ представима в виде $V^*(s) = \max_{a \in A} Q(s, a) = \max_{a \in A} Q_t(s, a)$, а $Q(s, a)$ есть функция

$$Q(s, a) = \sum_{s' \in S} p(s, a; s') (r(s, a; s') + \gamma V^*(s')) = \sum_{s' \in S} p(s, a; s') \left(r(s, a; s') + \gamma \max_{a' \in A} Q(s', a') \right).$$

Решение задачи MARL для одного светофора базируется на идее Q -обучения, которая заключается в оценке невычислимой правой части [4]:

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha_t(s, a) \left(r(s, a) + \gamma \max_{a' \in A} Q_t(s', a') - Q_t(s, a) \right), \quad (3)$$

где s' – положение процесса на шаге $t+1$, если на шаге t в состоянии s было выбрано действие a ; α – коэффициент скидки.

Если на шаге t итерационный процесс находился в состоянии s и было выбрано действие a , то в (3) полагаем $0 \leq \alpha_t \leq 1$, иначе $\alpha_t = 0$.

Последовательность векторов $Q = \{Q(s, a)\}_{s \in S, a \in A}$ можно записать итеративно в виде $Q_{t+1} = B(Q_t)$, где $B: \square_{\infty}^{|A|+|S|} \rightarrow \square_{\infty}^{|A|+|S|}$ является сжимающим отображением [4]. Действительно,

$$\begin{aligned}
\rho(BQ_1, BQ_2) &= \max_{a \in A, s \in S} \left| \sum_{s' \in S} p(s, a; s') \left(r(s, a; s') + \gamma \max_{a' \in A} Q_1(s', a') \right) - \right. \\
&\quad \left. - \sum_{s' \in S} p(s, a; s') \left(r(s, a; s') + \gamma \max_{a' \in A} Q_2(s', a') \right) \right| = \\
&= \max_{a \in A, s \in S} \left| \sum_{s' \in S} p(s, a; s') \gamma \left(\max_{a' \in A} Q_1(s', a') - \max_{a' \in A} Q_2(s', a') \right) \right| \leq \\
&\leq \gamma \max_{a \in A, s \in S} \sum_{s' \in S} p(s, a; s') \left| \max_{a' \in A} Q_1(s', a') - \max_{a' \in A} Q_2(s', a') \right| \leq \\
&\leq \gamma \max_{s' \in S} \left| \max_{a' \in A} (Q_1(s', a') - Q_2(s', a')) \right| \leq \\
&\leq \gamma \max_{a' \in A, s' \in S} |Q_1(s', a') - Q_2(s', a')| = \gamma \max_{a \in A, s \in S} |Q_1(s, a) - Q_2(s, a)| = \gamma \rho(Q_1, Q_2),
\end{aligned} \tag{4}$$

где $0 < \gamma < 1$, $\rho(\bullet, \bullet)$ – метрика в $\square_{\infty}^{|A|+|S|}$.

Полученная последовательность $\{Q_t\}$ представляет собой приближенные решения уравнения $BQ = Q$. Таким образом, начиная с определенного момента времени вычислять значение функции Q не требуется, и эффективная оценка точности такого приближенного решения рассчитывается по формуле

$$\rho(Q_{t_n}, Q_{t_0}) = \rho(B^n Q, Q) \leq \frac{\gamma^n \rho(BQ, Q)}{1 - \gamma}. \tag{5}$$

Теперь можем сформулировать критерий сходимости процесса $\{Q_t\}$ [4].

Критерий сходимости

Если стратегия $A(\bullet)$ приводит к тому, что каждая пара (s, a) бесконечное число раз встречается, то из условия сжимаемости B при $\sum_{t=0}^{\infty} \alpha_t = \infty$, $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ следует сходимость процесса $\{Q_t\}$ с оценкой (4).

С учётом формул (3)–(5) будем искать оптимальное управление в виде $a_t = \arg \max_{a' \in A} Q_t(s, a')$. Оптимальное управление a_t является решением задачи MARL для одного светофора, т.е. доставляет максимум функции суммарных доходов $V_t^* = \max_{a \in A} Q_t(s, a)$.

2. Вычислительные эксперименты

Для исследования представленной модели была разработана программа имитационного моделирования в системе AnyLogic и проведены серии вычислительных экспериментов. Эксперименты проводились на ПК с процессором Intel® Core™ i7-10510U CPU @ 1.80ГГц и оперативной памятью объемом 8ГБ.

Целью экспериментов было сравнение времени задержки машин в модели системы управления светофоров, длительность фаз p_0 , p_1 которой получена перебором (рис. 3), и управляемой марковским процессом (рис. 4), в системе Anylogic. Имитационное моделирование процесса управления светофором проводилось с учётом следующих условий (рис. 3, 4):

- 1) машины прибывают на перекресток с каждого из трех направлений: carSource, carSource1, carSource2, с интенсивностью 1000 в час,
- 2) коэффициенты скидки α (alfa) и переоценки γ (gamma) подобраны эмпирическим путём,

3) дискретное время period составляет 5 секунд.

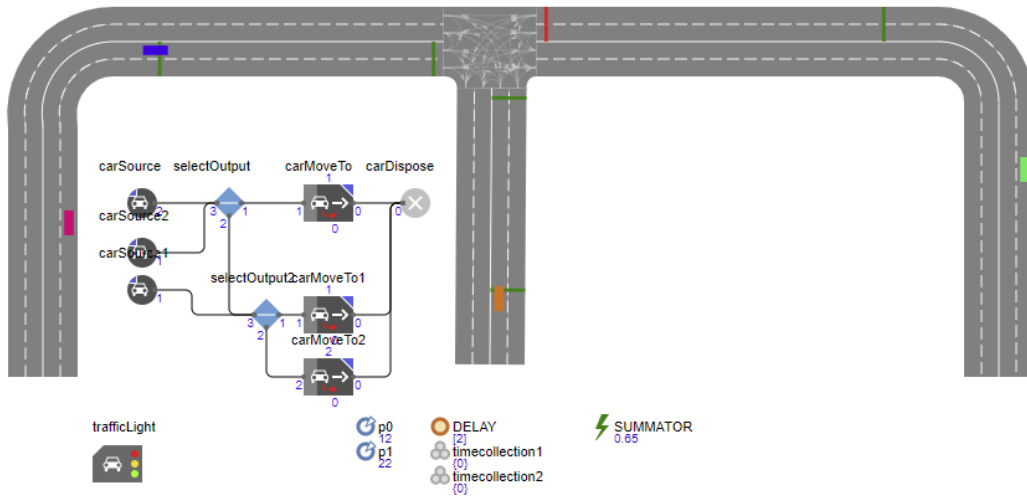


Рис. 3. Модель, длительность фаз которой получена перебором

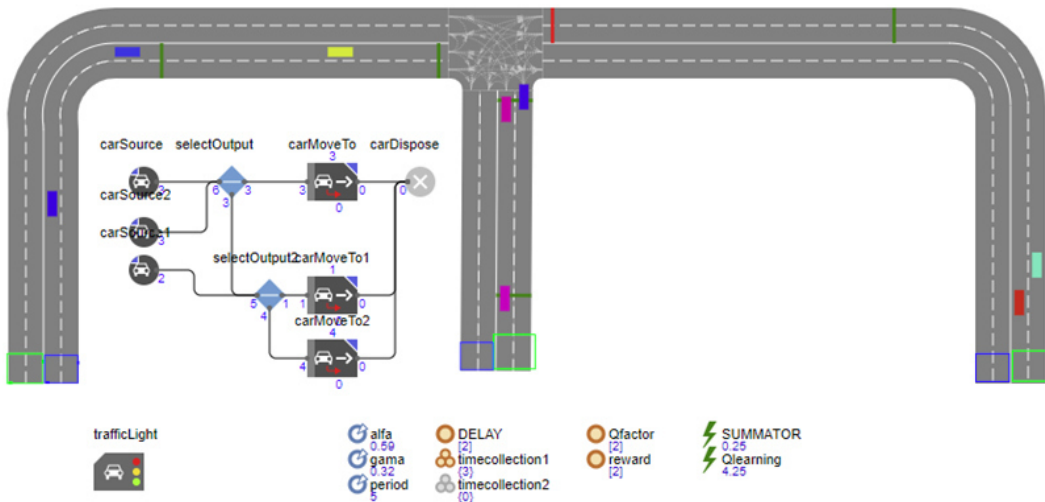


Рис. 4. Модель, управляемая марковским процессом

При имитационном моделировании машины создаются на одной из позиций, отмеченных зеленым цветом, и перемещаются в направлении позиций, отмеченных синим, после чего удаляются (рис. 4). При пересечении полосы, на расстоянии 100 метров до стоп-линии, пары, состоящие из указателей на объект машины и текущего времени модели, добавляются в одну из коллекций `timecollection1`, `timecollection2`. Далее машины удаляются из коллекции при проезде через перекресток. В течении периода времени 1 секунда вызывается событие `SUMMATOR`, прибавляющее суммарное время задержки машин, находящихся в коллекции, к текущей задержке `DELAY`. В течении периода времени period вызывается событие `Qlearning`, реализующее решение задачи MARL. На основе события `Qlearning` принимается решение об остановке или продлении фазы светофора.

Результаты и выводы: модель, управляемая марковским процессом, показала уменьшение суммарной задержки в 1.5 раза по сравнению с системой управления светофором, длительность фаз которой подобрана перебором от 5 секунд до 30 секунд с

шагом 1. Значение вектора всевозможных значений функции $Q(s,a)$ по метрике из $\square_{\infty}^{|A|+|S|}$ сходится (рис. 5).

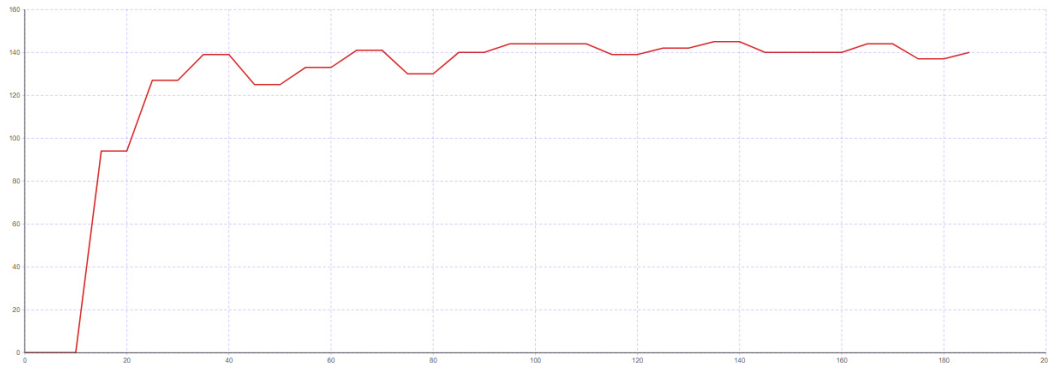


Рис. 5. График сходимости итерационного процесса для модели, управляемой марковским процессом

Заключение

В работе исследована математическая модель процесса выбора фазы светофора, отличающаяся учетом текущего расположения светофора и его загрузки. Сформулирована задача MARL для одного светофора на перекрестке. Решение задачи минимизации задержки трафика рассматривается как итерационный процесс, для которого получен критерий сходимости. Проведено имитационное моделирование в среде Anylogic 8. В дальнейшем планируется увеличить количество агентов и сравнить результаты с существующими моделями.

ЛИТЕРАТУРА

1. *Goel S., Bush S., Gershenson C.* Self-Organization in Traffic Lights: Evolution of Signal Control with Advances in Sensors and Communications ArXiv. – 2017. – abs. 1708.07188.
2. *Manolis D., Diakaki C., Papamichail I. et al.* Simulation investigations of the coordinated traffic-responsive signal control strategy TUC with actuation at the local junction level // Eur. Transp. Res. Rev. – 2018. – 10. – 25.
3. *El-Tantawy S., Abdulhai B., Abdelgawad H.* Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto // IEEE Transactions on Intelligent Transportation Systems. – 2013. V. 14. – No. 3. – P. 1140–1150.
4. *Гасников А.В., Горбунов Э.А., Гуз С.А. и др.* Лекции по случайным процессам: учебное пособие. Под ред. А.В. Гасникова. – «Москва»: МФТИ, 2019. – 285 с.
5. *Майн Х., Осаки С.* Марковские процессы принятия решений. – Главная редакция физико-математической литературы издательства «Наука», 1977. – 176 с.
6. *Sandholm T.W.* Contract Types for Satisficing Task Allocation: I Theoretical Results // AAAI Spring Symposium Series: Satisficing Models. – 1998. – P. 68–75.

ВИ. ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ АСПЕКТЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И DEVOPS-ИНЖЕНЕРИИ

DOI: 10.17223/978-5-907442-42-9-2021-28

РАЗРАБОТКА И АНАЛИЗ ПРОГРАММНОЙ СИМУЛЯЦИИ ЭВОЛЮЦИОНИРУЮЩЕЙ ВОДНОЙ ЭКОСИСТЕМЫ С ПРОЦЕДУРНОЙ ГЕНЕРАЦИЕЙ ЛАНДШАФТА

Березовский Н.В., Буторина Н.Б.

Томский государственный университет
hemenbgmb@gmail.com, nnatta07@mail.ru

Введение

С развитием ЭВМ у человека появилась возможность производить ресурсоемкие вычисления в различных сферах деятельности. Люди стали имитировать математические модели, позволяющие им лучше понимать природу тех или иных явлений, таких как движение тектонических плит, распространение инфекции, процесс эволюции. Например работа Карла Симса “Evolved virtual creatures” [1], представляющая из себя симуляцию и исследование эволюции Дарвина виртуальных блочных существ, которым давались различные задания. Программные гены лучших из них отбирали, комбинировали и подвергали мутации, чтобы повторно пройти тестирование. Некоторые существа вырабатывали стратегии, схожие с теми, что есть в нашем мире. Свое место в этой сфере нашли и симуляции различного рода экосистем, например, знаменитая “Игра в жизнь” Джона Конвея [2], представляющая из себя клеточный автомат с определенным набором правил, поведение которого очень схоже с развитием популяций примитивных организмов.

Оказалось, что на основе клеточных автоматов можно моделировать явления из физики, химии, биологии или строить интересные фрактальные узоры, схожие с теми, которые мы можем встретить в природе. Некоторые же алгоритмы, такие как шум Перлина, направлены именно на генерацию случайных шаблонов. В основном он находит применение в области компьютерной графики, а впервые был использован в киноиндустрии для процедурной генерации текстур. Он полезен при имитации случайных структур из природы, таких как туман, облака, дым, огонь или ландшафт.

1. Постановка задачи

Цель работы: Разработать симуляцию эволюционирующей водной экосистемы, обитателями которой являются водоросли, нехищные и, в противоположность последним, хищные рыбы. В процессе естественного отбора они будут эволюционировать, приобретая те или иные характеристики, позволяющие им лучше приспосабливаться к меняющимся условиям. На популяции рыб будет исследовано воздействие ландшафта, наличие или отсутствие хищников, условия избытка и дефицита пропитания по отдельности и в композиции.

2. Разработка искусственного интеллекта рыб

Существуют разные подходы к разработке искусственного интеллекта, которые отличаются сложностью реализации и применимостью, например, деревья решений, автоматный подход, деревья поведений, системы, основанные на полезности или на машинном обучении [3].

Для реализации искусственного интеллекта рыб был выбран автоматный подход, т.к. для данной задачи он является оптимальным вариантом по сложности и возможностям. Ниже будет описан набор состояний и характеристики, которые определяют их смену. Для начала нужно определить параметры объекта:

- **Размер** – влияет на количество энергии и способность к размножению.
- **Скорость** – быстрота движения рыбы. Влияет на расход энергии.
- **Энергия** – способность рыбы к действиям. Если она кончается, то рыба умирает.
- **Время** до следующего размножения – размножение с определённым периодом.
- **Минимальный необходимый размер для размножения** – данный параметр позволяет ограничить размножение слишком маленьких и слабых особей.
- **Разброс плавания** – величина, на которую объект будет уплывать от текущей позиции.
- **Страх к другим рыбам** – характеризует вероятность, с которой особь будет уплывать от незнакомой ей рыбы.
- **Коэффициент потери энергии** – коэффициент, на который умножаются все затраты и восполнения энергии.

Далее можно определить возможные состояния рыбы. У травоядной особи (рис. 1):

1. Бездействует (Idle) – плавает и анализирует окружение.
2. Ест (Eating) – приближается к растению, чтобы пополнить запас энергии.
3. Размножается (Mating) – приближается к партнеру, чтобы передать гены.
4. Скрывается (Hiding) – уплывает от попавшего в область видимости хищника.

Для хищной рыбы (рис. 2):

1. Бездействует (Idle) – плавает с низким разбросом плавания, сохраняя энергию.
2. Ищет добычу (Searching) – плавает и анализирует окружение.
3. Ест (Eating) – приближается к добыче, чтобы атаковать и съесть.
4. Размножается (Mating) – подплывает к партнеру для передачи генов.

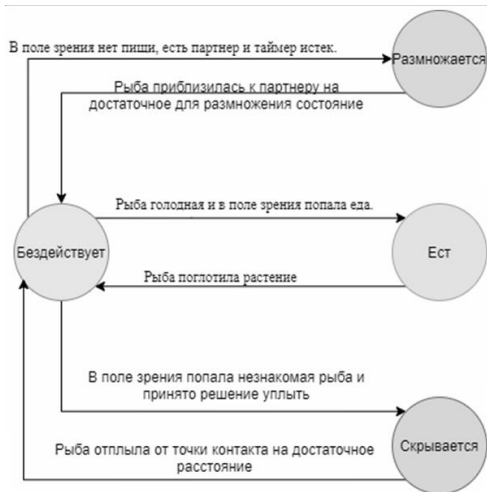


Рис. 1. Переходы состояний травоядной рыбы



Рис. 2. Переходы состояний хищной рыбы

Параметры хищных и нехищных рыб сильно отличаются. У хищных рыб запас энергии сильно больше и они могут долго не есть. Размножение происходит медленнее. Разброс плавания дублируется на состояния поиска пищи и сохранения энергии. По понятным причинам скорость передвижения хищных рыб также больше.

Особь будет анализировать окружение с помощью зрения, которое основано на методе бросания лучей [4]. Для реализации этого метода нужно сгенерировать векторы, расположенные вокруг окружности и направленные наружу. Затем для каждого вектора применяется метод бросания лучей, т.е. проверяется факт пересечения этим вектором

других объектов, а именно – рыб, водорослей или поверхности ландшафта. На основе пересечений, текущего состояния и текущих значений параметров особи и принимается решение о смене состояния.

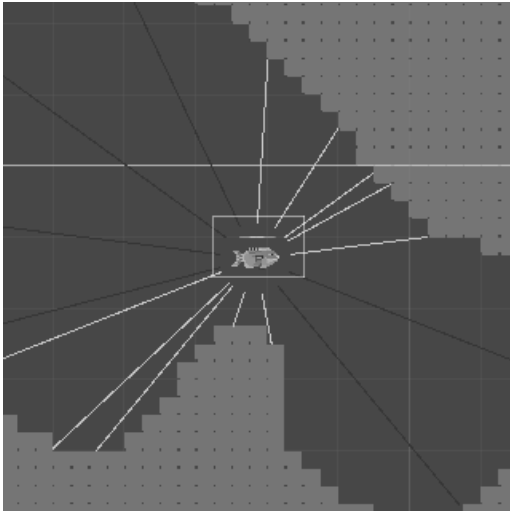


Рис. 3. Бросание лучей в поверхность ландшафта

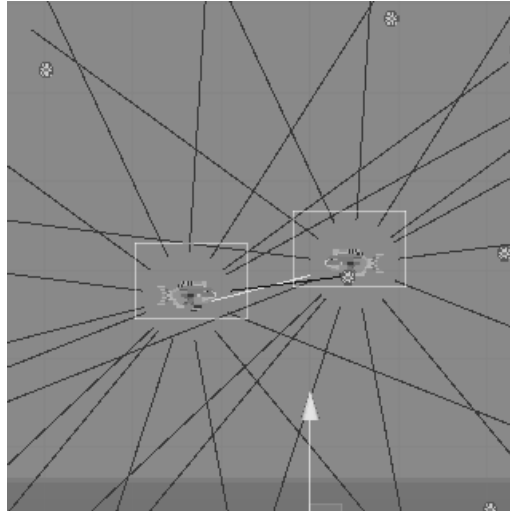


Рис. 4. Бросание лучей для определения объектов

3. Разработка процедурной генерации ландшафта

Случайные числа, сгенерированные компьютером можно разбить на 2 класса – случайные и псевдослучайные. Первая категория использует случайность, основанную на природе, например, числа генерируются на основе датчиков, считывающих космический шум. Вторая использует различные алгоритмы для генерации псевдослучайных чисел, например, линейный конгруэнтный метод. Отличительной особенностью второй категории является использование числа, на основе которого генерируется псевдослучайная последовательность, его, как правило, называют зерном. Свое применение псевдослучайные числа нашли и в генерации различных структур.

Создавать структуры на основе псевдослучайных чисел можно с помощью заранее заготовленных элементов, определяя их расположение с помощью набора правил, или полностью с нуля, используя процедурную генерацию [5].

В данной работе целью является генерация ландшафта, похожего на морское дно, или на подводную систему пещер. Эта задача была разбита на 2 этапа – генерация дна и системы пещер. На первом этапе генерируется пустая матрица, размеры которой соответствуют требуемой длине и высоте ландшафта. Элементы матрицы заполняются в соответствии с данными, сгенерированными шумом Перлина. Вариативность ландшафта достигается за счет использования зерна для шума (рис. 5).

На втором этапе на матрицу накладывается 2-хмерный шум. Размер пещер можно регулировать, накладывая только тот шум, который выше заданного пользователем значения (рис. 6).

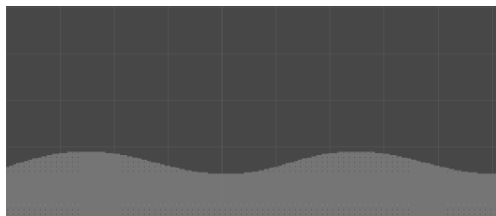


Рис. 5. Ландшафт на первом этапе генерации

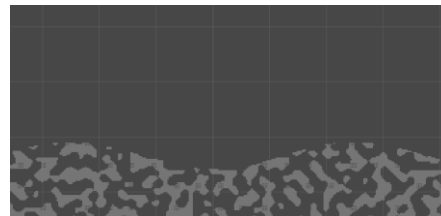


Рис. 6. Ландшафт на втором этапе генерации

4. Введение эволюции рыб

Эволюция популяции рыб будет основана на идеях генетического алгоритма. Идея генетического алгоритма тесно связана с основными положениями теории Дарвина. Для того, чтобы естественный отбор происходил так, как это происходит в природе, необходимо присутствие следующих элементов: наследственность, изменчивость и естественный отбор [6].

В общем виде последовательность для реализации генетического алгоритма выглядит следующим образом [7]:

1. Составляется популяция из n особей.
2. Для каждой особи вычисляется ее приспособленность.
3. Выбираются родители с помощью какого-нибудь способа отбора.
4. Производится потомок, гены которого унаследованы от родителей.
5. Производится мутация потомка.
6. Повторяем 2–5, пока не достигнем нужного результата.

Естественный отбор будет основан на параметрах экосистемы, а именно – количество хищников, ландшафт и наличие пищи. Особи с наилучшими характеристиками будут выживать и передавать свои гены.

Далее определим параметры особей, которые будут передаваться по наследству. Имеет смысл передавать именно те параметры, которые неоднозначно влияют на выживаемость рыбы, т.к. если передавать по наследству, например, скорость передвижения, то она со временем будет расти, позволяя быстрее находить пищу и уплывать от угрозы. Перечислим эти параметры:

1. **Минимальный размер для размножения.** Нельзя однозначно сказать, как будет изменяться этот параметр при естественном отборе, но можно предположить, что он будет расти, т.к. у рыбы будет больше шанс выжить.

2. **Разброс плавания** – покажет, какая стратегия выгоднее – постоянно плавать в поисках пищи или преодолевать небольшие расстояния, экономя энергию.

3. **Коэффициент потери энергии** – влияет как на затраты, так и на восполнения энергии; предполагается, что он будет стремиться принять какое-то значение, близкое или равное оптимальному.

4. **Страх к другим рыбам** – покажет, имеет ли смысл особи не тратить энергию на попытку уплыть от незнакомой ей рыбы. Если хищная рыба не голодная или её жертва не попадает в поле зрения, то это сохранит энергию травоядной особи. Этот параметр существует только у травоядных рыб.

5. **Порог голода** – влияет на частоту потребления пищи рыбой. Параметр не является однозначным, т.к. энергия, получаемая от поедания растения или рыбы – фиксированная, и есть слишком часто, не усваивая полезную энергию и уничтожая большие запасы пищи, может быть бессмысленно. С другой стороны, чаще пополнять шкалу энергии, съедая все запасы растений или рыб, выгодно для выживания.

Особь будет наследовать случайным образом k генов от первого родителя и $n - k$ от второго. После создания потомка он подвергается мутации, т.е. случайно выбранные гены с малой вероятностью изменяются на небольшую величину.

Перед запуском симуляции пользователь определяет начальные условия, влияющие на ландшафт, частоту появления водорослей и начальный размер популяций, при которых он хочет исследовать поведение экосистемы. Далее создаются популяции с различными генами.

5. Языки и среда разработки

В качестве основного языка программирования для разработки был взят C# и использующая его среда разработки Unity, на основе которой разрабатываются графиче-

ские приложения. С помощью Unity было создано приложение для симуляции экосистемы, а именно – классы для генерации ландшафта, классы, реализующие логику поведения и движения объектов рыб. Написан класс для настройки параметров экосистемы и класс для сбора информации о работе приложения. Она записывается в файл, который затем подвергается обработке на другом языке программирования – Python.

6. Анализ полученной экосистемы

Для анализа экосистемы ведется сбор данных о количестве особей в популяциях с определенным интервалом времени.

Для начала посмотрим, как ведет себя экосистема с обычным ландшафтом в отсутствии хищников. По оси y отмечено количество особей популяции, по оси x – время в секундах (рис. 8). Количество особей в популяции имеет периодический характер. Это можно объяснить тем, что популяция травоядных рыб зависит от популяции водорослей. При росте популяции рыб количество особей водных растений падает, что в свою очередь вызывает вымирание популяции рыб. Далее рассмотрим экосистему без хищников с ландшафтом пещер (рис. 9).

Пещеры служат ограничителем для чрезмерного поедания водорослей травоядными рыбами, поэтому здесь график травоядных особей реже пересекает график водорослей. Вымирание потребителей в данном случае наступает не из-за полного истребления популяции водорослей, а из-за невозможности найти достаточное пропитание в стенах пещер.

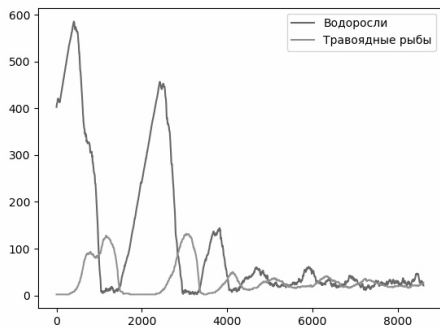


Рис. 8. Графики популяций, обычный ландшафт

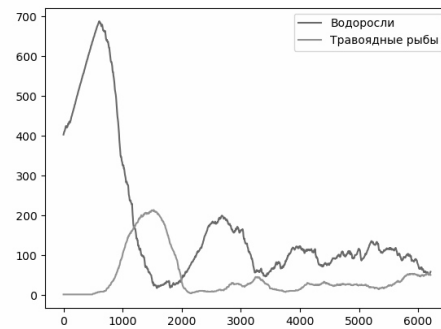


Рис. 9. Графики популяций, пещерный ландшафт

С наличием хищников экосистема с обычным ландшафтом ведёт себя следующим образом (рис. 10). Хищные рыбы контролируют популяцию травоядных рыб, поэтому популяция водорослей держится на отметке 200 – 300 особей. Далее – экосистема с пещерным ландшафтом (рис. 11). В случае пещерного ландшафта над популяцией травоядных идет контроль как со стороны хищников, так и со стороны ландшафта.



Рис. 10. Графики популяций, при обычном ландшафте, с хищниками

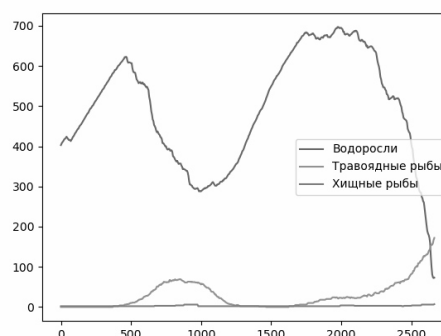


Рис. 11. Графики популяций при пещерном ландшафте, с хищниками

Заключение

Таким образом, была разработана симуляция экосистемы, обитателями которой являются водоросли, хищные и травоядные рыбы. Для живых обитателей был разработан искусственный интеллект на основе конечного автомата и введены элементы генетического алгоритма, позволяющие адаптироваться к меняющимся условиям. Была исследована зависимость популяций и их генов от параметров экосистемы, определяемых пользователем. Как показали эксперименты, при пещерном ландшафте популяции менее склонны к вымиранию. С обычным ландшафтом популяция травоядных рыб чрезмерно возрастает и поглощает всю без исключения пищу, что приводит либо к вымиранию всей популяции, либо к вымиранию большинства особей. Наличие хищников хоть и исправило положение, но при росте популяции хищных рыб с большой вероятностью вымирают травоядные. В дальнейшем планируется подробнее исследовать зависимость генов от параметров экосистемы и улучшить механизмы наследования и мутации, а также оптимизировать производительность программы в целом для больших исследований.

ЛИТЕРАТУРА

1. Sims K. Evolving Virtual Creatures // SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques. – 1994. – P. 15–22.
2. Википедия [Электронный ресурс]: Игра «Жизнь» URL: https://ru.wikipedia.org/wiki/Игра_«Жизнь». (дата обращения 1.05.2021).
3. Хабр [Электронный ресурс]: Как создать игровой ИИ URL: <https://habr.com/ru/company/pixonix/blog/428892/>. (дата обращения 27.04.2021).
4. Википедия [Электронный ресурс]: Бросание лучей URL: https://ru.wikipedia.org/wiki/Ray_casting. (дата обращения 2.05.2021).
5. Hyttinen T. Terrain synthesis using noise. – Tampere, 2017. – 54 p.
6. Shiffman D. The Nature Of Code [Электронный ресурс]: Chapter 9. The Evolution of Code URL: <https://natureofcode.com/book/chapter-9-the-evolution-of-code/>. (дата обращения 29.03.2021)
7. Панченко Т.В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. – 87 с.

СОЗДАНИЕ ВНЕШНИХ ОБРАБОТОК ДЛЯ ПЕРЕНОСА БУХГАЛТЕРСКИХ ПРОВОДОК ИЗ РАЗЛИЧНЫХ КОНФИГУРАЦИЙ НА ПЛАТФОРМЕ 1С: ПРЕДПРИЯТИЕ

Васильева О.В., Морозова А.С.

Томский государственный университет

vasilyeva.olga.95@gmail.com, annamo12@yandex.ru

Введение

Любая организация или предприятие ведет бухгалтерский учет. Большинство крупных организаций использует для этого специальные программы 1С. В данном случае речь пойдет о государственном бюджетном учреждении, которое ведет бухгалтерский учет в программе 1С: Бухгалтерия государственного учреждения 1.0 (1С: БГУ 1.0). Помимо этого, данной организации необходимо вести кадровый учет и производить расчет заработной платы сотрудникам. Для этого используется программа 1С: Зарплата и кадры государственного учреждения 3.1 (1С: ЗКГУ 3.1).

Организации обязаны предоставлять годовую бухгалтерскую отчетность в виде определенного набора документов в налоговую инспекцию. В ее составе также сдается бухгалтерский баланс – свод информации о стоимости имущества и обязательствах организации, представленный в табличной форме.

Далее речь пойдет о заработной плате сотрудников, которая является текущей задолженностью перед работниками. Для отражения операций по расчетам с сотрудниками в бухучете, а также для формирования бухгалтерского баланса, помимо прочего, необходимо учитывать бухгалтерские проводки по заработной плате. Но, поскольку данные проводки формируются в программе 1С: ЗКГУ 3.1, а бухгалтерский баланс – в программе 1С: БГУ 1.0, бухгалтеру необходимо перенести проводки по заработной плате в программу 1С: БГУ 1.0.

Таким образом, для решения данной задачи необходимо создать две внешние обработки для переноса бухгалтерских проводок из программы 1С: ЗКГУ 3.1 в программу 1С: БГУ 1.0.

1. Учет, расчет и начисление заработной платы государственных бюджетных учреждений

Важной частью бухгалтерской отчетности является бухгалтерский баланс. Данный отчет представляет из себя свод информации о стоимости имущества и обязательствах организации, представленный в табличной форме, и содержит сведения, по которым можно судить о финансовом состоянии предприятия, о том, каким имуществом оно обладает и сколько имеет долгов.

Неотъемлемой частью бухгалтерского баланса является заработная плата – денежная форма цены, выплачиваемой за использование труда как экономического ресурса в процессе создания товаров и услуг. Она начисляется и выплачивается сотрудникам ежемесячно.

К системе расчета заработной платы сотрудникам государственных бюджетных учреждений относятся следующие выплаты: заработная плата, компенсационные выплаты, стимулирующие выплаты (премии и надбавки), отпускные, компенсации и пособия.

Каждый месяц производится начисление страховых взносов – обязательные ежемесячные платежи, которые уплачивает любая организация или предприятие, если она привлекла хотя бы одного сотрудника по трудовому или гражданско-правовому договору. Они начисляются на следующие выплаты: заработная плата, премии, отпускные. Уплачиваются в государственные фонды, такие как ПФР, ФСС, ФФОМС.

Также каждый месяц начисляется НДФЛ – налог на доходы физических лиц или подоходный налог, который уплачивается работодателем в налоговую в размере 13% от зарплаты сотрудника.

Помимо НДФЛ начисляются также прочие удержания, такие как профсоюзные взносы, добровольные страховые взносы в счет будущей пенсии, алименты, удержания по исполнительным листам за неуплату кредитов. Они удерживаются работодателем и уплачиваются взыскателю.

Таким образом, можно сказать, что расчет заработной платы – это достаточно сложный и трудоемкий процесс, особенно с учетом немалой численности работников в организации. Поэтому для ее учета, расчета и начисления требуется специальная автоматизированная программа.

2. Специфика ведения учета заработной платы в программах 1С: ЗКГУ и 1С: БГУ

Учет заработной платы ведется в программе 1С: ЗКГУ, а бухгалтерский учет – в 1С: БГУ. Поэтому для отражения операций по расчету зарплаты в бухучете, а также для формирования бухгалтерского баланса, требуется произвести перенос проводок по зарплате в бухгалтерскую программу. Для этого необходимо создать 2 внешние обработки. Первая обработка выгрузит данные из программы 1С: ЗКГУ в текстовый файл, вторая обработка загрузит их из текстового файла в программу 1С: БГУ.

Учет заработной платы ведется в программе 1С: ЗКГУ. Все вышеперечисленные начисления ежемесячно производятся с помощью документа «Начисление зарплаты и взносов» (рис. 1).

← → ☆ Начисление зарплаты и взносов 0000-000042 от 30.09.2020 *

Провести и закрыть Записать Провести Выплатить

Месяц: Сентябрь 2020 Дата: 30.09.2020

Подразделение:

Заполнить Подбор Очистить

Начислено: 14 197 347,89 ? Доначислено: 0,00 Удержано: 3 079 183,52 ? Взнос

Начисления Договоры Пособия Удержания НДФЛ Взносы Корректировки выплаты Доначисления

Добавить ↑ ↓ Найти... Отменить поиск Отмена исправлений Расчеты

N	Сотрудник	Подразд...	Начисление	Результат	Финансир., расходы
1		ЛПСА	Оплата по окладу (по часам)	1 000,00	002(211)
2		ЛПСА	Северная надбавка		002(211)

Рис. 1. Документ «Начисление зарплаты и взносов»

Затем они отражаются в документе «Отражение зарплаты в бухучете» в программе 1С: ЗКГУ по кнопке «Заполнить» (рис. 2).

← → ☆ Отражение зарплаты в бухучете 0000-000029 от 28.10.2020 *

Провести и закрыть Записать Провести

Месяц: Октябрь 2020 Дата: 28.10.2020 Номер: 0000-000029

Подразделение: ИФО: Бюджет

Заполнить

Начисленная зарплата и взносы Начисленный НДФЛ Удержанная зарплата Выплата отпусков за счет оценочных обязательств

Добавить ↑ ↓

N	Сотрудник	Подразделение	Вид операции	Статья финансирования	Статья расходов	Способ отражения	Сумма	ПФР (до превыш.)
1	РемГру...		Договор (рабо...	Субсидии-внештатники	225	6.16.901.200.225.30	1 000,00	1 000,00
2		ЛПСА	Начислено	Субсидии-----	211	1.56.080.200.211.20	1 000,00	1 000,00
3		ЛПСА	Начислено	Субсидии-----	211	1.56.080.200.211.20	1 000,00	1 000,00

Рис. 2. Документ «Отражение зарплаты в бухучете»

Данный документ представляет из себя таблицу со следующими параметрами:

1. Сотрудник.
2. Подразделение, в котором работает сотрудник.
3. Вид операции. Показывает, какое начисление было произведено: зарплата штатным сотрудникам или по договору, отпускные, больничные, страховые взносы и т.д.
4. Статья финансирования: показывает из каких средств организации происходит выплата.
5. Статья расходов или вид затрат. Показывает, на что расходуются средства.
6. Начисленная сумма в рублях.
7. Способ отражения зарплаты в бухучете или шифр затрат. Представляет из себя последовательность чисел, разделенных точкой. Он необходим для того, чтобы определить какие данные следует подставить в проводке по затратному счету. Например, 1.34.005.220.211.20:
 - a) 1-е число обозначает направление научной деятельности;
 - b) 2-е число – код подразделения;
 - c) 3-е число – тема научного исследования;
 - d) 4-е число – статья финансирования;
 - e) 5-е число – вид затрат (статьи расходов);
 - f) 6-е число – категория персонала: 10 – административно-управленческий персонал, 20 – основной, 30 – вспомогательный.

Далее эти сведения переносятся из программы 1С: ЗКГУ в программу 1С: БГУ посредством внешней обработки и хранятся в документе «Операция (бухгалтерская)» (рис. 3), который имеет определенную структуру. Данный документ состоит из шапки, в которой содержатся следующие поля: учреждение, ИФО (источник финансового обеспечения), номер документа, сумма операции. Основная часть документа состоит из таблицы, отражающей бухгалтерские проводки.

№	КФД	(Дт) КПС	(Дт) Субконто 1	(Дт) Субконто 2 / 3 / 4	(Кт) Счет	(Кт) КПС	(Кт) Субконто 1	(Кт) Субконто 2 / 3 / 4	(Дт) Количество	(Дт) Валюта	(Кт) Валюта	Сумма	Содержание	Первичный документ
1	4	01100000000000244	109.81	01100000000000244	302.25	737					RUB		4	
		225	225.30	работы, услуги по содержанию недвижим.										
		Базовый госбюджет												
		РемГруппа												
		Все сотрудники												
		Зарплата внештатн.сопр.												
2	4	01100000000000244	109.81	01100000000000244	303.10	731							8	
		225	225.30	работы, услуги										
		2015		Страховые взносы в ПФР										

Рис. 3. Документ «Операция (бухгалтерская)»

Каждая проводка представляет собой бухгалтерскую операцию и состоит из следующих параметров:

1. Порядковый номер.
2. КФО – код финансового обеспечения (ранее КВД – код вида деятельности).
3. КПС (КБК) (дебет, кредит, далее – Д, К) – классификационный признак счета. Это аналитический код, характеризующий поступления или выбытия, определяет содержание всей операции.
4. Субконто 1 (Д, К) КЭК – коды экономической классификации. Содержит КОСГУ – коды операций сектора государственного управления, отраженные в законодательстве. Показывает вид операции.
5. В зависимости от счета на субконто 2 (Д, К), субконто 3 (Д, К), субконто 4 (Д, К) отражается подразделение, вид затрат, номенклатура (исследование), или контрагент.
6. Сумма операции в рублях.
7. Счет (Д, К) – синтетический счет учета. Состоит из 26 разрядов: 1-17 – КПС, 18 – КФО, 19-21 – код синтетического счета, 22-23 – код субсчета, 24-26 – КЭК.

Таким образом, для создания внешних обработок по переносу проводок необходимо знать механизм учета заработной платы и учитывать все вышеперечисленные коды для реализации.

3. Обзор существующих решений

Для решения такой задачи, как обмен бухгалтерскими проводками между конфигурациями, существует механизм, встроенный разработчиками непосредственно в программу 1С. Этот механизм «Синхронизации», который позволяет настроить обмен данными между базами данных различных конфигураций напрямую или через файл. Он предусматривает передачу результатов расчета заработной платы для автоматического формирования в бухгалтерском учете проводок, отражающих расчеты по оплате труда, начисление налогов и отчисления в фонды.

Также существует второе решение: «Конвертация данных 2.0». Это конфигурация, которая может производить конвертацию данных и обмен данными между типовыми конфигурациями на платформе «1С: Предприятие 8». Обмен данными по технологии «Конвертация данных 2.0» можно разделить на две стадии – стадию подготовки правил обмена и стадию обмена данными.

Каждое из предлагаемых решений выполняет свои функции в полном объеме. Однако более понятным, легким в применении и имеющим низкую вероятность возникновения неполадок и необходимости вызова специалиста решением являются разрабатываемые внешние обработки для переноса проводок из программы 1С: ЗКГУ в программу 1С: БГУ.

4. Проектирование

Для четкого представления системы была разработана модель предметной области (рис. 4):

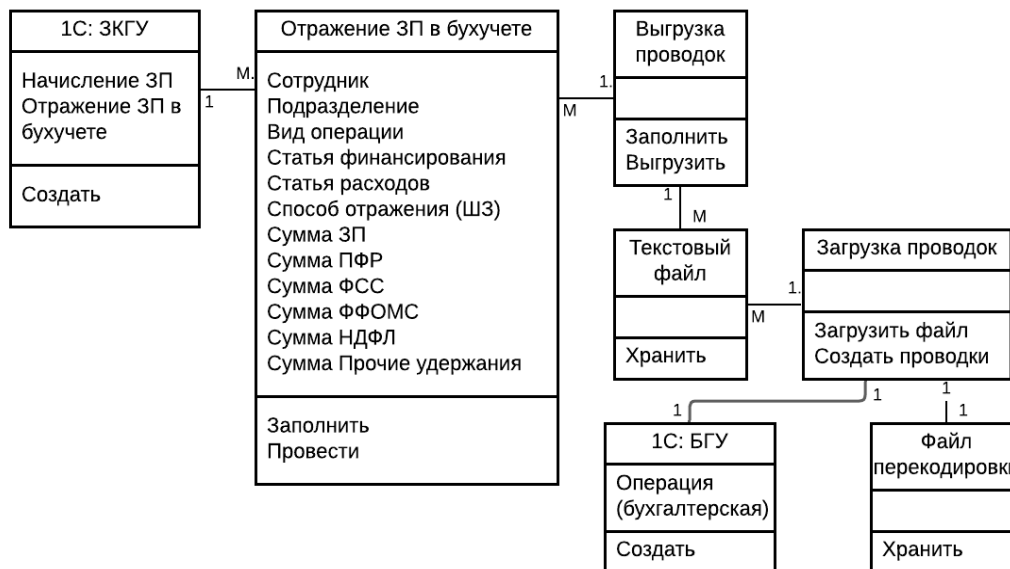


Рис. 4. Модель предметной области

Планирование последовательности этапов работы внешних обработок:

1. Из документа «Отражение зарплаты в бухучете» в программе 1С: ЗКГУ будут выгружаться проводки по заработной плате с помощью внешней обработки (далее – ВО) «Выгрузка проводок» в текстовый файл с расширением .txt (рис. 5). Этот файл будет создан обработкой в указанном каталоге.
2. Далее из текстового файла проводки будут загружены в программу 1С: БГУ с помощью ВО «Загрузка проводок» (рис. 6). Данная обработка произведет сопоставление объектов программы 1С: ЗКГУ и 1С: БГУ и запишет найденные совпадения в файл перекодировки, который создаст. При отсутствии совпадений пользователю будет необходимо вручную заполнить пробелы, которые также запишутся в файл перекодировки. Затем данная обработка сформирует бухгалтерские проводки, которые будут записаны в документ «Операция (бухгалтерская)».

В результате спроектированная модель предметной области будет являться основой для реализации.

5. Реализация

На основе спроектированной модели предметной области можно приступить к реализации. Разработка внешних обработок (ВО) для переноса бухгалтерских проводок производится в программах 1С в режиме Конфигуратора на встроенном языке программирования 1С. ВО «Выгрузка проводок» создается в программе 1С: ЗКГУ, а ВО «Загрузка проводок» – в программе 1С: БГУ.

Первым делом создается документ «внешняя обработка». Далее на вкладке «Форма» конструируется интерфейс будущей обработки: добавляются необходимые поля, кнопки, реквизиты и т.д. Затем на вкладке «Модуль» реализуются необходимые процедуры и функции путем написания программного кода. Завершающим этапом производится отладка внешних обработок (иначе говоря, попытка перенести проводки в бухгалтерскую программу). При необходимости программный код редактируется и производится повторная отладка.

Интерфейс готовой внешней обработки «Выгрузка проводок» выглядит следующим образом (рис. 5):

← → **Выгрузка проводок ЗП: Форма**

Дата проводок: 30.04.2021

Каталог выгрузки: C:\Учебное\ВКР

Заполнить Выгрузить

Добавить ↑ ↓

Категория данных	Подразделение	Вид операции
------------------	---------------	--------------

Рис. 5. ВО «Выгрузка проводок»

Интерфейс внешней обработки «Загрузка проводок» (рис. 6):

Шифр затрат	Счет ЕПСУ	Статья финансирования ЗП	Баланс	Статья расхода ЗП	Подразделение ЗП	Подразделение	Контрагент	Номенклатура	КЭК ФАНО	Н.	
6.05.901.200.211.30	302.11	субсидии	Бюджет	211	Сторожевая охрана	Сторожевая о...		Базовый госбюджет	211.30	вспомогательного п...	1
6.08.901.200.211.30	302.11	субсидии	Бюджет	211	Транспортный учас...	Транспортный ...		Базовый госбюджет	211.30	вспомогательного п...	1
6.09.901.200.211.30	302.11	субсидии	Бюджет	211	ГИПВ	ГИПВ		Базовый госбюджет	211.30	вспомогательного п...	1
6.10.093.220.211.30	302.11	контрактов...	ПД	211	Дворники	Дворники		Ассоль - 20	211.30	вспомогательного п...	1
6.10.901.200.211.30	302.11	субсидии	Бюджет	211	Служба междунаро...			Базовый госбюджет	211.30	вспомогательного п...	0
6.11.020.225.211.10	302.11	целевые	ЦС и БП	211	Дирекция			РНФ в ч. Накл. раск.-20	211.10	административно уп...	0
6.11.020.225.211.30	302.11	целевые	ЦС и БП	211	Транспортный учас...	Транспортный ...		РНФ в ч. Накл. раск.-20	211.30	вспомогательного п...	1
6.11.063.220.211.30	302.11	контрактов...	ПД	211	Эксперт участок	Эксперт участок		ФС-20. 1 этап	211.30	вспомогательного п...	1
6.11.090.220.211.30	302.11	контрактов...	ПД	211	РемГруппа	РемГруппа		РФФИ Физ.л. - 20 Н.Р.	211.30	вспомогательного п...	1
6.11.210.211.10	302.11	контрактов...	ЦС и БП	211	Дирекция						0
6.11.210.211.30	302.11	контрактов...	ЦС и БП	211	Транспортный учас...	Транспортный ...					0
6.11.737.220.211.30	302.11	контрактов...	ПД	211	Эксперт участок	Эксперт участок		АСУ-эксперт, доп.согл №2	211.30	вспомогательного п...	1
6.11.900.200.211.10	302.11	субсидии	Бюджет	211	СУС	СУС		Компенсация выплаты на м...	211.10	административно уп...	1
6.11.900.200.211.20	302.11	субсидии	Бюджет	211	Эксперт участок	Эксперт участок		Компенсация выплаты на м...	211.20	основного персонала	1
6.11.900.200.211.30	302.11	субсидии	Бюджет	211	СУС	СУС		Компенсация выплаты на м...	211.30	вспомогательного п...	1
6.11.901.200.211.30	302.11	субсидии	Бюджет	211	СУС	СУС		Базовый госбюджет	211.30	вспомогательного п...	1
6.11.902.225.211.10	302.11	целевые	ЦС и БП	211	ЛКАС	ЛКАС		РНФ в ч. Накл. раск.-19	211.10	административно уп...	1
6.11.902.225.211.20	302.11	целевые	ЦС и БП	211	ЛРВ	ЛРВ		РНФ в ч. Накл. раск.-19	211.20	основного персонала	1
6.11.902.225.211.30	302.11	целевые	ЦС и БП	211	Издательская служ...	Издательская ...		РНФ в ч. Накл. раск.-19	211.30	вспомогательного п...	1
6.11.915.225.211.30	302.11	целевые	ЦС и БП	211	Издательская служ...	Издательская ...		ФШП № 05.619.21.0012	211.30	вспомогательного п...	1

Создать проводки | Закрыть

Рис. 6. ВО «Загрузка проводок»

Заключение

В результате были разработаны внешние обработки для переноса бухгалтерских проводок из программы 1С: ЗКГУ в программу 1С: БГУ. В перспективе, данные обработки могут стать удобным и легким в применении инструментом для бухгалтера при ежемесячном расчете заработной платы. В случаях, когда нет возможности настроить «Синхронизацию», воспользоваться «Конвертацией 2.0» или в организации отсутствует программист, их применение может стать незаменимым решением.

РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В КИБЕРСПОРТЕ НА ПРИМЕРЕ СЕРИИ ИГР FIFA

Гетц К.Е., Якунина Е.Н.

Томский государственный университет
kirillgoetze19@gmail.com, YE_2010@mail.ru

Введение

В настоящее время игровая индустрия развивается очень быстрыми темпами, и вместе с ней развивается и растёт сообщество киберспортсменов – профессиональных компьютерных игроков, зарабатывающих на успешных выступлениях в специализированных турнирах. Количество профессиональных спортсменов растёт с каждым годом [1], и много игроков-любителей пытается попасть в это сообщество. Людей, пытающихся это сделать, принято считать начинающими киберспортсменами. Однако не всем это даётся с лёгкостью, и без посторонней помощи зачастую не получается перейти с любительского на профессиональный уровень. Далеко не каждый может себе позволить личного тренера или тратить время на просмотр большого количества информации, связанной с игрой, для повышения своего уровня. Эти факторы могут стать решающими и не позволить игрокам, имеющим хороший потенциал, превратить своё хобби в работу.

Для решения этой проблемы предлагается разработать экспертную систему поддержки принятия решений в киберспорте на примере серии игр FIFA. Экспертная система позволит начинающим игрокам более эффективно разрабатывать план подготовки к соревнованиям, вести тренировочный процесс, выбирать стратегии во время соревнований. Такая экспертная система будет моделировать действия спортсменов и просчитывать модели, которые могут быть реализованы. Из всех возможных вариантов действий будет предлагаться конечный набор (настраиваемый пользователем параметр) сценариев для успешного выступления на киберспортивных соревнованиях.

Для достижения этой цели были поставлены следующие задачи:

1. Провести исследование предметной области действий игроков в игре FIFA и составить краткое описание экспертной системы.
2. Изучить и проанализировать принцип работы существующих экспертных систем.
3. Проработать требования к экспертной системе с учётом требований пользователей
4. Разработать ЭС на основе изученных материалов.

1. Исследование предметной области

В игре существует четыре основных составляющих: схема, тактика, задачи игрокам, состав.

В составе присутствуют 11 футболистов. У каждого из них существует собственная карточка, на которой показаны игровые характеристики футболиста – показатели скорости, удара, передач, дриблинга, защиты и физики. Эти параметры влияют на возможности футболиста во время игры. Приобретать карточки футболистов можно на трансферном рынке игры за монеты – игровую валюту.

Игровая схема показывает расположение игроков на поле во время матча. Всего существует 30 схем. В игровом меню предоставляется возможность выбрать любую из представленных вариантов и расположить футболистов на любую из позиций. Однако рекомендуется ставить игроков на позиции, соответствующие тем, которые указаны на их карточках.

В зависимости от амплуа (вратарь, защитник, полузащитник, нападающий) существует определённый список задач. В зависимости от выбранной схемы задачи каждого амплуа могут претерпеть небольшие изменения. Вариации, заданные пользователем, определяют поведение игроков на поле.

В отличие от задач игрокам выбор тактики определяет поведение команды в целом во время игры. Пользователь может настроить 4 разных варианта тактики на игру и менять их во время матча.

Все эти составляющие необходимы для проведения успешного матча – должен быть подобран конкурентоспособный состав, удобная игровая схема, задачи и тактика, соответствующая этой схеме.

2. Принцип работы экспертной системы. Структура ЭС

Типичная статическая ЭС состоит из следующих основных компонентов: решателя (интерпретатора); рабочей памяти называемой также базой данных (БД); базы знаний (БЗ); компонентов приобретения знаний; диалогового компонента [2].

БД (рабочая память) предназначена для хранения исходных и промежуточных данных решаемой в текущий момент задачи. База знаний (БЗ) в ЭС предназначена для хранения долгосрочных данных, описывающих рассматриваемую область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области. Решатель, используя исходные данные из рабочей памяти и знания из БЗ, формирует такую последовательность правил, которые, будучи применёнными к исходным данным, приводят к решению задачи. Компонент приобретения знаний автоматизирует процесс наполнения ЭС знаниями, осуществляемый пользователем-экспертом. Диалоговый компонент ориентирован на организацию дружественного общения с пользователем как в ходе решения задач, так и в процессе приобретения знаний и объяснения результатов работы [3].

3. Модель представления знаний в экспертных системах на примере ЭС GURU

Важнейший элемент ЭС – база знаний. Она состоит из переменных и правил, по которым будет находиться ответ на вопрос, поставленный пользователем.

Для представления данных в системе GURU применяются рабочие переменные. Они аналогичны переменным, используемым в языках программирования. В GURU имеются следующие основные типы переменных: строковые, числовые, логические.

Существует три типа переменных: целевые переменные, внутренние переменные, входные переменные.

Целевая переменная – это переменная, значение которой является результатом сеанса работы с ЭС. Как только целевая переменная получает значение, сеанс работы с ЭС завершается. Входные переменные – переменные, которые ЭС получает на вход от пользователя. После их означивания начинается сеанс работы с ЭС. Внутренние переменные – переменные, которые вычисляются в ходе работы ЭС, они являются вспомогательными и служат для нахождения целевой переменной.

После проведения детального анализа предметной области были выявлены следующие переменные:

Целевая переменная – GAMESETTINGS

Входные переменные:

- MAST – мастерство пользователя (переменная типа int, значения от 1 до 10),
- TIME – время (в часах), которое пользователь тратит на игру в неделю (переменная типа int),
- STYLEG – стиль игры, который предпочитает пользователь (переменная типа string, возможные значения: «АТАКУЮЩИЙ», «ЗАЩИТНЫЙ», «СБАЛАНСИРОВАННЫЙ», «СВЕРХЗАЩИТНЫЙ», «СВЕРХАТАКУЮЩИЙ»),

- BUDGET – количество игровой валюты пользователя (переменная типа int, минимальное значение 100 000)

Внутренние переменные:

- SCHEMEG – игровая схема (переменная типа string),
- TACTIC – набор тактических указаний команде (переменная типа string), TASK – задачи футболистов (переменная типа string),
- SQUAD – состав футболистов (переменная типа string).

Описание правил представляет собой определение посылки (часть IF) и заключения (часть THEN). Посылка представляет собой некоторое условие; если оно верно, то выполняются действия, указанные в заключении.

Пример правила – «Если мастерство игрока больше пяти (по шкале от 1 до 10) и время, потраченное на игру в неделю больше 6 (часов), и выбранный стиль игры пользователем означен как атакующий, то рекомендуемая игровая схема – 4-5-1»

IF: MAST > 5 AND TIME > 6 AND STYLEG = «АТАКУЮЩИЙ»

THEN: SCHEMEG = «4-5-1»

После выявления переменных был составлен граф их означивания, по которым были составлены правила. На самом графе (рис. 1) стрелка выходит из переменной посылки и заходит в переменную заключения.

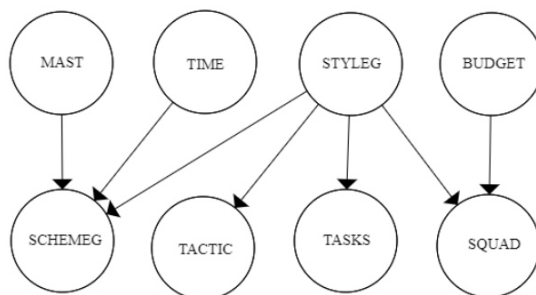


Рис. 1. Граф означивания переменных

На графе означивания переменных можно увидеть, что все правила в ЭС можно разделить на три типа.

В правилах первого типа с помощью переменных MAST, TIME и STYLEG вычисляется переменная SCHEMEG. В правилах второго типа с помощью переменной STYLEG вычисляется переменная TACTIC и переменная TASKS. В правилах третьего типа с помощью переменных STYLEG и BUDGET вычисляется переменная SQUAD.

Также одной из важнейших частей ЭС является механизм вывода. Механизм вывода «знает», как использовать базу знаний так, чтобы можно было получать разумные заключения. Когда экспертной системе задается вопрос, механизм вывода выбирает способ применения правил базы знаний для решения поставленной задачи, т.е., фактически, механизм вывода запускает экспертную систему в работу. Механизм вывода выполняет правила, определяет, когда найдено приемлемое решение, и передает результаты программе интерфейса с пользователем.

При использовании прямого логического вывода сначала рассматривается посылка правила. Если она верна, то правило включается (срабатывает), т.е. выполняются действия, указанные в заключении этого правила. Если посылка неверна, то правило не включается, и рассматривается другое правило. Порядок рассмотрения правил может быть различным; обычно правила рассматриваются в порядке расположения. Рассмотрение правил продолжается, пока не будет установлено значение целевой переменной

(или будет выяснено, что определить эту переменную невозможно). Именно этот вывод был реализован в ходе работы.

4. Разработка приложения

После определения требований к приложению были получены варианты использования, представленные ниже на рис. 2.

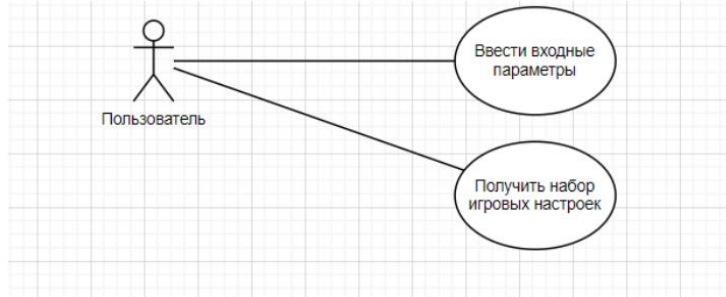


Рис. 2. Диаграмма ВИ

На рис. 3 представлена спроектированная диаграмма классов.

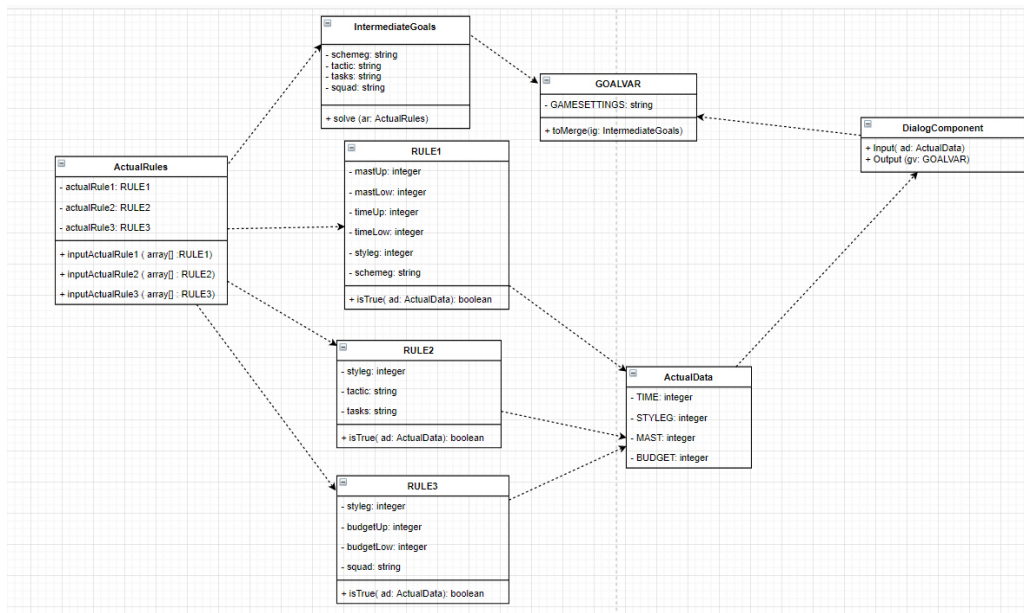


Рис. 3. Диаграмма классов

Поля класса ActualData являются входными переменными, которые вводит пользователь. Они же являются посылками правил, т.к. во всех правилах в качестве посылки используются входные переменные. Классы RULE 1, RULE 2, RULE 3 выбирают необходимые для работы ЭС правила. Класс ActualRules содержит правила, с которыми будет работать ЭС. Класс IntermediateGoals содержит все временные цели. Класс GOALVAR содержит целевую переменную – результатом работы ЭС. Класс DialogComponent осуществляет взаимодействие с пользователем.

Заключение

В ходе работы над экспертной системой была детально рассмотрена предметная область, изучены и выбраны важнейшие компоненты игры, необходимые для правильной работы экспертной системы. Также были проанализированы составляющие стати-

ческих экспертных систем, в частности, на примере экспертной системы GURU была рассмотрена и применена модель представления знаний. На основе вышеперечисленных данных была разработана экспертная система – построены диаграмма ВИ и диаграмма классов, которые необходимы для создания будущего приложения.

ЛИТЕРАТУРА

1. Баленко Е. Технологии и медиа, февраль 2019 [электронный ресурс]. URL: https://www.rbc.ru/technology_and_media/12/02/2019/5c61921d9a79478a13fdd100 (Дата обращения 17.05.2021).
2. WIKIPEDIA [электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Экспертная_система (Дата обращения 17.05.2021).
3. Структура экспертной системы, AIRPORTAL [электронный ресурс]. URL: <http://www.aiportal.ru/articles/expert-systems/structure.html> (Дата обращения 17.05.2021).

DOI: 10.17223/978-5-907442-42-9-2021-31

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ДЛЯ ЭФФЕКТИВНОГО УПРАВЛЕНИЯ РЕСУРСАМИ КОМПАНИИ ПО МЕТОДОЛОГИИ О.К.Р

Литвинова Н.И., Шкуркин А.С.
Томский государственный университет
natgraham1864@gmail.com, shkurkin@mail.ru

Введение

В IT-индустрии очень часто можно встретить с таким выражением, как «управление проектами». Данное выражение подразумевает под собой изучение, планирование и осуществление определенных действий, которые приведут человека, команду или всю компанию к достижению поставленной цели. При изучении этой темы становится понятно, что простого планировщика зачастую недостаточно и многие компании используют популярные на данный момент методологии по управлению проектами.

В настоящее время существует большое количество различных методологий: Scrum, KPI, Agile, Waterfall и др. [1] Несмотря на то, что данных методологий достаточно много – однозначного ответа, какую из них лучше использовать, все равно нет. Также в последние несколько лет в России начали активно внедрять методологию O.K.R – Objectives and Key results [2].

Т.к. Objectives and Key results только набирают популярность в компаниях России и странах СНГ, то найти удобную программу, в основе которой парадигма O.K.R, на русском языке почти невозможно. По этой причине было решено реализовать собственное web-приложение для управления проектами по методологии O.K.R.

1. Выявление требований

Целью работы является проектирование и разработка web-приложения для управления проектами, в основе которой парадигма O.K.R.

Требования к системе должны быть следующие: система должна предоставлять возможность

- создать аккаунт пользователю;
- создать цели и ключевые результаты;
- редактировать цели и ключевые результаты;
- удалять цели и ключевые результаты;
- предлагать цели;
- одобрять цели;
- отклонять цели и указывать причину отклонения в виде комментария;
- фильтровать цели по категориям: предложенные цели, отклоненные цели, одобренные цели и неотправленные цели;

- вносить изменения в данные пользователя.

Помимо вышеуказанных требований, должно быть учтено то, что у приложения должен быть интерфейс на русском языке.

2. Сравнение существующих решений

Многие компании давно пользуются таск-менеджерами – программами, где наглядно видно, какие задачи в работе, какие на проверке, какие уже закончены, а за какие стоит еще взяться.

При изучении предметной области были рассмотрены также аналоги приложений, в основе которых лежит парадигма О.К.Р. Были выявлены плюсы и минусы существующих аналогов, а также отмечены полезные характеристики, которые помогут улучшить создаваемое web-приложение. Также была составлена таблица с наглядным сравнением аналогичных приложений по следующим характеристикам.

Сравнение существующих решений

Характеристика	Приложение			
	Microsoft Excel	Weekdone	Profit	Kazoo
Наглядная зависимость целей	Нет	Есть	Есть	Есть
Удобный интерфейс	Нет	Нет	Есть	Есть
Наличие русского языка	Есть	Есть, но не весь функционал приложения	Нет	Нет
Постановка ключевых результатов	Есть	Есть	Есть	Есть
Наличие пробного периода использования	Есть	Есть	Есть	Нет

В результате проведенного анализа было выявлено, что существующие приложения не предоставляют необходимый функционал для людей, которые впервые столкнулись с методологией. Основным недостатком приложений является то, что в приложениях нет русского языка.

Таким образом, было принято решение при создании приложения уделить внимание особенностям методологии, постараться создать простое и удобное приложение на русском языке.

3. Проектирование приложения

При работе с методологией О.К.Р цели можно декомпозировать не только до отдела, но также до отдельной команды или конкретного сотрудника. В создаваемом web-приложении было решено оставить декомпозицию до отдела.

В разрабатываемом web-приложении было решено придерживаться алгоритма, при котором цели отделов будут оговариваться на ежеквартальной встрече, после чего руководители отделов вносят цели, которые согласованы со всеми сотрудниками отдела, в приложение и отправляют их на проверку директору. Если цель будет отклонена, на следующем собрании отдел может ее доработать и отправить вновь на проверку. Данный процесс отображен на рис. 1.

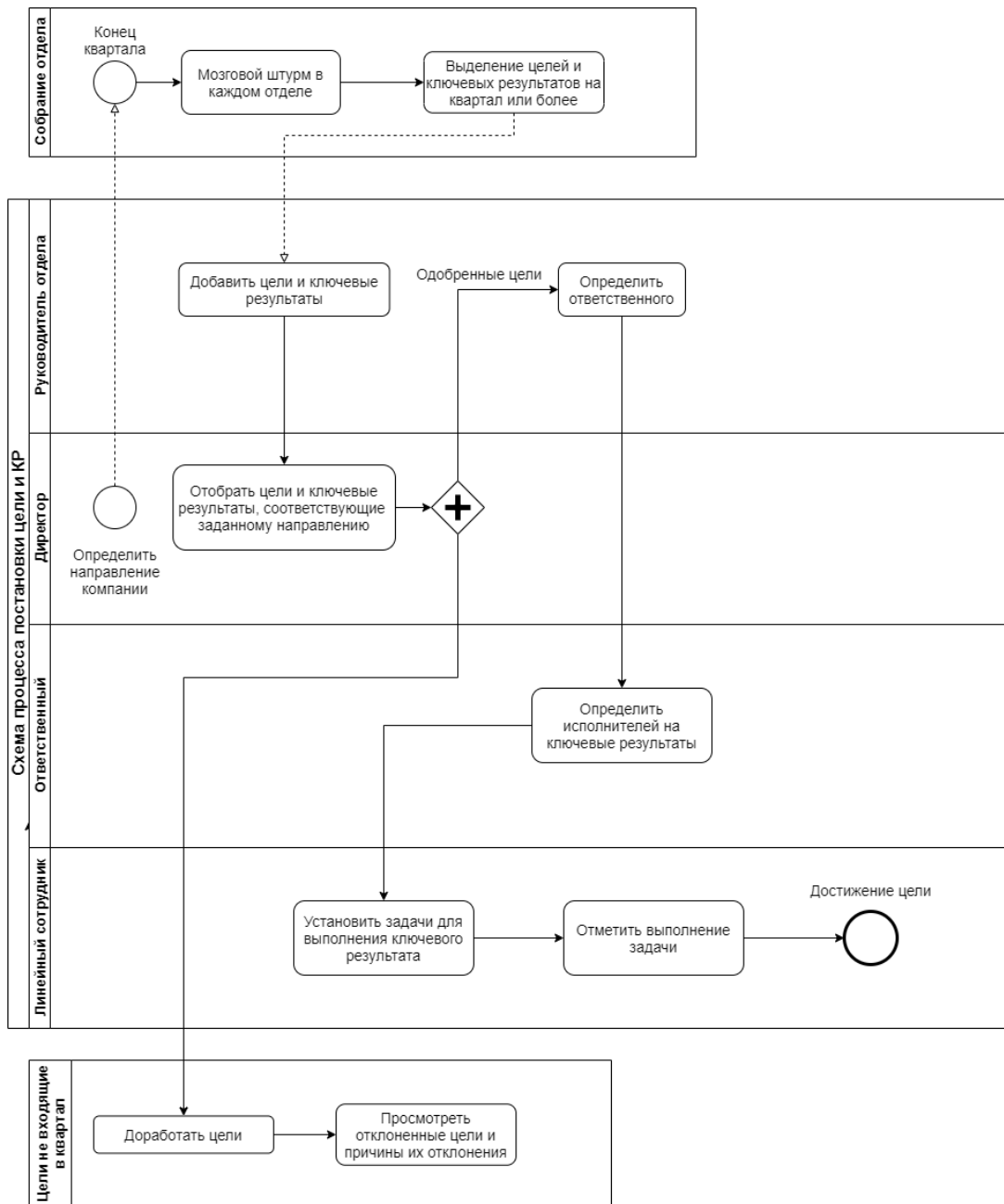


Рис. 1. Процесс постановки целей и ключевых результатов в приложении

При постановке целей сотрудники должны опираться на стратегические цели компании. Они определяют вектор развития компании, то, к чему руководство компании хочет привести свою фирму. Данные цели может установить только руководство компании.

3.1. Модель предметной области

Для того, чтобы приложение работало в соответствии с выделенными требованиями, а также было интуитивно понятным и простым в использовании, было решено постараться минимизировать количество сущностей в системе. Всего получилось четыре сущности, которые отображены на модели предметной области (рис. 2).

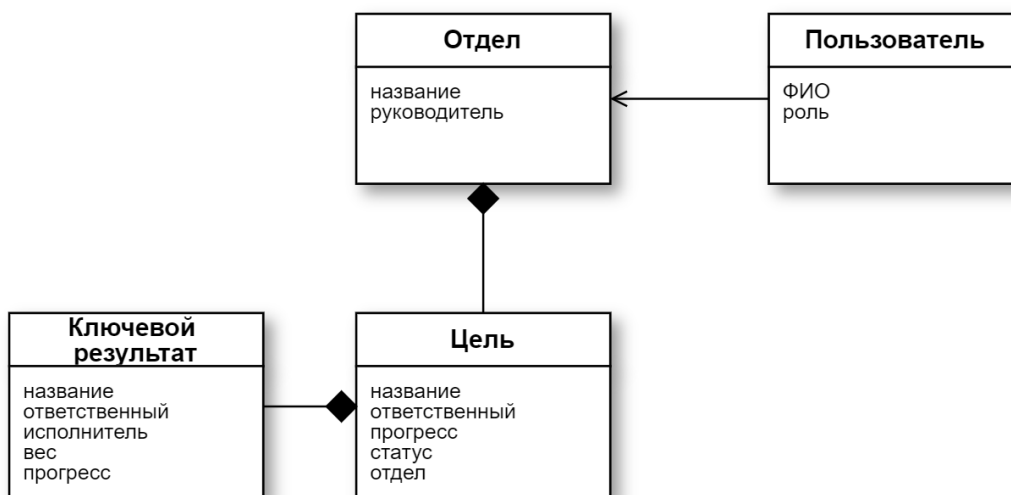


Рис. 2. Модель предметной области

Рассмотрим некоторые из представленных сущностей. Начнем с сущности «Отдел». Данная сущность подразумевает, что в компании есть различные отделы: отдел разработки, отдел маркетинга, отдел тестирования и др.

Далее, одной из важных частей системы является сущность «Цель». Для цели выбирается ответственный, который будет следить за сроками ее выполнения и корректировать в случае необходимости. Также для данной сущности автоматически указывается отдел, к которому она относится и автор – тот, кто внес цель в систему.

Следующая важная часть системы – это сущность «Ключевой результат». Для ключевого результата по требованиям предусмотрена связь с целью. Для ключевого результата автоматически проставляется ответственный, но также указывается и команда исполнителей. Исполнителем может быть команда или один человек.

Также на рис. 1 можно увидеть, что сущность «Ключевой результат» не может существовать без сущности «Цель», так же, как и «Цель» не может существовать без сущности «Отдел».

3.2. Общая архитектура

Для написания кода для серверной части приложения можно использовать почти любой из языков программирования. В данном проекте предпочтение было отдано языку PHP с использованием фреймворка Laravel.

Laravel – популярный бесплатный PHP-фреймворк с открытым исходным кодом. Данный фреймворк предназначен для разработки с использованием архитектурной модели MVC.

Model-View-Controller (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») – схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер, чтобы модификация каждого компонента могла осуществляться независимо [3].

Таким образом, архитектура создаваемой системы представляет концепцию Model-View-Controller или MVC (рис. 3).

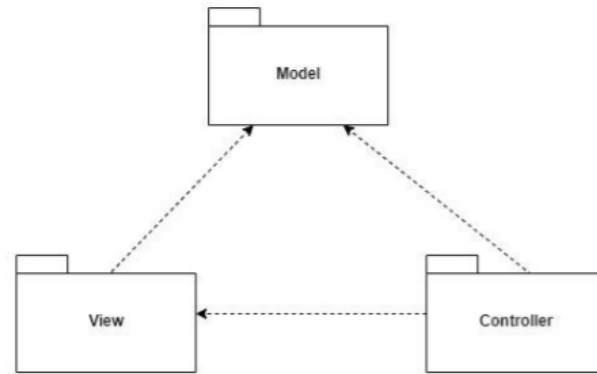


Рис. 3. Диаграмма пакетов web-приложения

Модель – представляет некоторую информацию о домене, реагирует на команды контроллера, изменяя своё состояние. У модели нет визуального интерфейса, она содержит в себе все данные и поведение, не связанные с пользовательским интерфейсом.

Представление – отвечает за отображение данных модели, реагируя на её изменение. Функции представления заключаются только в отображении информации на экране.

Контроллер – получает входные данные от пользователя, оповещая модель о необходимости изменений, указывает представлению на необходимость соответствующего обновления.

4. Описание реализованного web-приложения

В результате проделанной работы получилось web-приложение, некоторые скриншоты которого представлены далее.

При входе в программу пользователь попадает в меню «Цели» на вкладку с идентичным названием, где отображены все цели компании (рис. 4). При клике на одну из целей система раскроет подробную информацию, и пользователь сможет просмотреть ключевые результаты, связанные с целью, а также на сколько процентов выполнены данные ключевые результаты (рис. 5). Если данное действие выполнит сотрудник, который является ответственным за выполнение цели, то он сможет увеличить процент выполнения ключевого результата в случае необходимости.

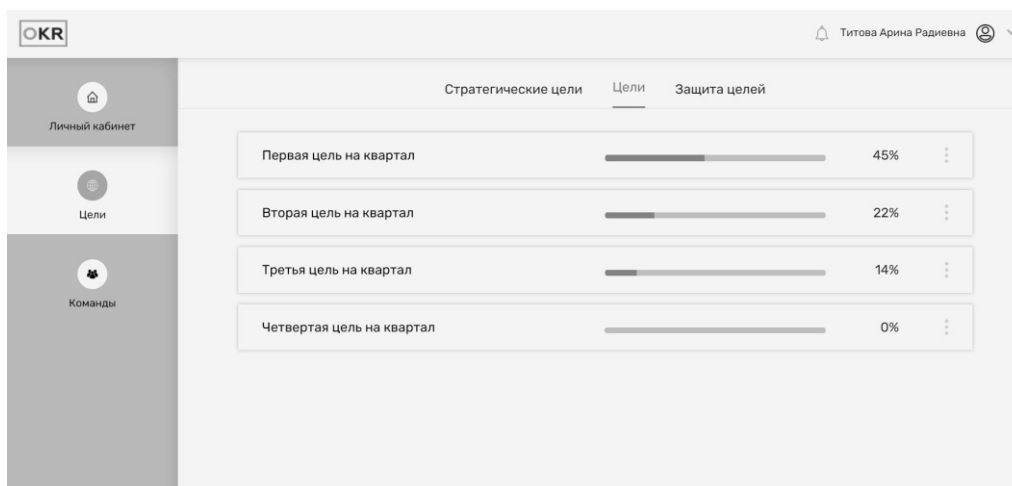


Рис. 4. Отображение целей всей компании

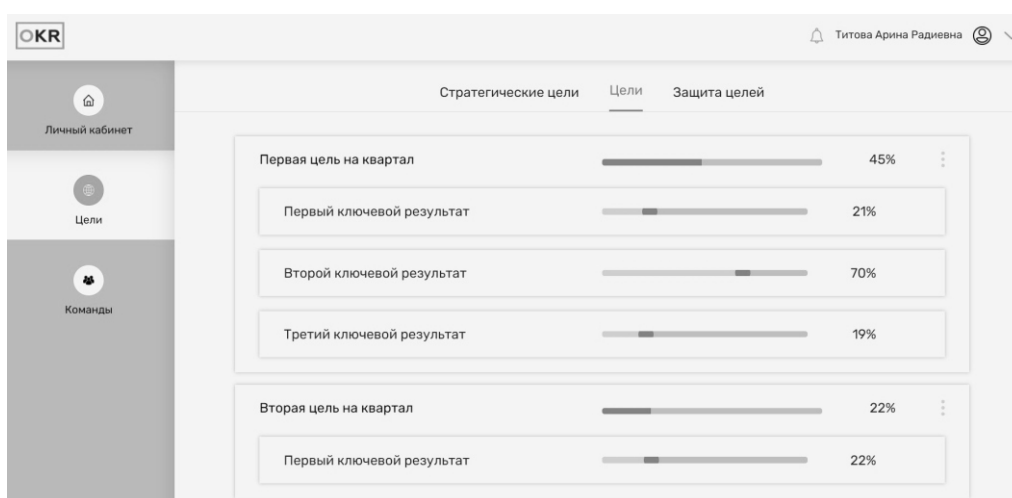


Рис. 5. Отображение целей всей компании с развернутыми ключевыми результатами

Заключение

В рамках проделанной работы была изучена и исследована предметная область, проанализированы существующие решения, были выявлены их преимущества и недостатки, и на этой основе сформированы требования к разработанному web-приложению. Также в результате данной работы было спроектировано и разработано web-приложение, которое позволяет пользователю ставить цели и ключевые результаты, отмечать, насколько продвинулось выполнение ключевых результатов, и следить за выполнением поставленных целей.

ЛИТЕРАТУРА

1. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие / pmservices. – [Б. м], 2014-2021. – URL: <https://www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/> (дата обращения: 29.05.2021).
2. Измеряйте самое важное / RomanSergeevCom. – [Б. м], 2016-2019. – URL: <https://summary.romansergeev.com/izmeryajte-samoe-vazhnoe/> (дата обращения: 29.05.2021).
3. Model-View-Controller / freejournal. – [Б. м], 2020. – URL: <https://ru.freejournal.org/213101/1/model-view-controller.html> (дата обращения: 10.06.2021).

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ПО УПРАВЛЕНИЮ ПРЕДПРИЯТИЕМ С СИСТЕМОЙ ВОЗНАГРАЖДЕНИЙ СОТРУДНИКОВ

Орел И.А., Морозова А.С.

Томский государственный университет
iulianaorel@gmail.com, annamo12@yandex.ru

Введение

Современный ритм развития программного обеспечения требует особого отношения к сотрудникам, т.к. именно от них зависит успешность создаваемых продуктов, их окупаемость, в итоге и успех самой компании.

В свою очередь, качественное программное обеспечение (ПО) невозможно разработать, если не уделять должного внимания внутренним процессам работы, управлению задачами и грамотному онбордингу сотрудника. Адекватное участие в вопросах автоматизации внутренних процессов обеспечивает более высокую удовлетворенность сотрудников работы в компании, что важно в условиях дефицита кадров в IT-отрасли.

Особенное внимание внутренним процессам уделяется в небольших компаниях, т.к. они не могут предложить зарплату мирового уровня, зато могут дать специалистам интересные задачи и грамотно выстроить систему управления проектами и знакомства нового сотрудника с компанией.

Одним из способов мягкого погружения – геймификация процесса. Это глобальный тренд, который захватывает все: бизнес, HR, маркетинг, а также IT и образование. Не только дети, но и огромное количество взрослых людей любят проводить свое свободное время, играя в игры. Однако есть большая разница между игрой и геймифицированной системой, суть которой заключается в целеполагании. Игра призвана развлекать, в то время как геймификация необходима для мотивации и изменения поведения человека.

1. Выявление требований

Целью работы является спроектировать и разработать web-приложение с системой геймифицированных вознаграждений сотрудников.

Требования к системе должны быть следующие: система должна

- предоставлять возможность создать аккаунт пользователю;
- предоставлять возможность авторизация пользователя в системе;
- предоставлять возможность просмотреть, редактировать информацию профиля;
- предоставлять возможность просмотреть, создать, редактировать, удалять задачи;
- позволять фильтровать задачи по категориям;
- предоставлять возможность просмотреть, создать, редактировать, удалять проекты;
- Обучать пользователя работе в системе (при помощи личного помощника);
- предоставлять возможность просмотреть список достижений пользователя;
- предоставлять возможность посмотреть статистику выполнения задач;
- предоставлять возможность получать достижения за определённые действия в системе.

Помимо вышеуказанных требований должно быть учтено то, что у приложения должен быть интерфейс на русском языке, а также простота и удобство интерфейса.

2. Сравнение существующих решений

При изучении предметной области были рассмотрены также аналоги приложений, были отобраны несколько web-приложений, которые часто используются для автома-

тизации процесса управления предприятием и task-менеджером. Проанализировать готовые решения с внедренной геймификацией невозможно, т.к. таких систем нет в открытом доступе. Приложения имеют схожие функции и понятия. Проведем анализ и выявим сильные и слабые стороны существующих решений. Также была составлена таблица с наглядным сравнением аналогичных приложений по следующим характеристикам:

Характеристика	Jira	Trello	Мегатлан	Habitica
Онбординг пользователя	Нет	Частично	Да	Да
Удобство использования	Да, но сложно настроить	Да	Да	Да
Функционал	Избыточен	Не хватает для больших компаний	Настраивается	Не хватает
Работа в команде	Да	Да	Да	Нет
Геймификация	Нет	Нет	Нет	Да
Цена	Высокая для команд >10 человек	Хватает бесплатной версии	Очень высокая	Бесплатно
Работа с несколькими проектами	Да	Неудобно	Да	Нет
Создание подзадач для задачи	Да	Нет	Да	Нет

В результате анализа аналогичных решений можно сделать вывод, что ни один из существующих на рынке отечественных продуктов не реализует совокупность функций по организации взаимодействия, контролю работ и стимулированию заинтересованности в результате работ сотрудников предприятий, осуществляющих проектную деятельность, а внедрение схожих по функционалу продуктов (в том числе техническое сопровождение) является дорогостоящим. Разрабатываемый продукт будет доступен для небольших российских предприятий, преимущественно в сфере ИТ, занимающихся проектной деятельностью, в силу стоимости, а также гибкости предлагаемых решений.

Особое внимание будет уделено работе с мотивацией сотрудников и процессу адаптации пользователя в системе.

3. Выбор игровой механики

Использование элементов геймификации в управлении проектами повышает энтузиазм целевой аудитории. Однако общая стратегия должна быть сбалансированной, и нужно учитывать важный фактор – разные люди по-разному относятся к одному и тому же элементу дизайна.

Например, если вы полностью полагаетесь на конкуренцию, вы можете потерять аудиторию, более склонную к сотрудничеству. Перед выбором игровых механик нужно четко сформулировать цели внедрения геймификации. На их основе и учитывая сбалансированную стратегию, учитывающую разные психотипы людей как игроков, можно создать систему, которая будет работать в долгосрочной перспективе, а не увеличивать вовлеченность исключительно из-за новизны [1].

Цели внедрения геймификации:

1. Поднять настроение и дать сотрудникам больше удовольствия от выполнения рутинных процессов.
2. Мягкое погружение во внутренние процессы компании (онбординг).
3. Дать возможность сотрудникам, которые не являются лидерами в работе, заявить о себе и стать первыми в игре.

На основе целей и анализа существующих решений были выбраны следующие игровые механики: достижение (achievement) поведенческий импульс (behavioral momentum) и личный помощник – персонаж, который помогает сделать первые шаги в

системе и интерактивно объясняет правила поведения. Такой набор игровых механик обеспечивает заинтересованность всех пользователей (вне зависимости от психотипа).

4. Проектирование web-приложения

В результате изучения предметной области и на основе сформированных требований к разрабатываемому web-приложению была спроектирована модель предметной области.

На рис. 1 отображены связи между существующими сущностями web-приложения.

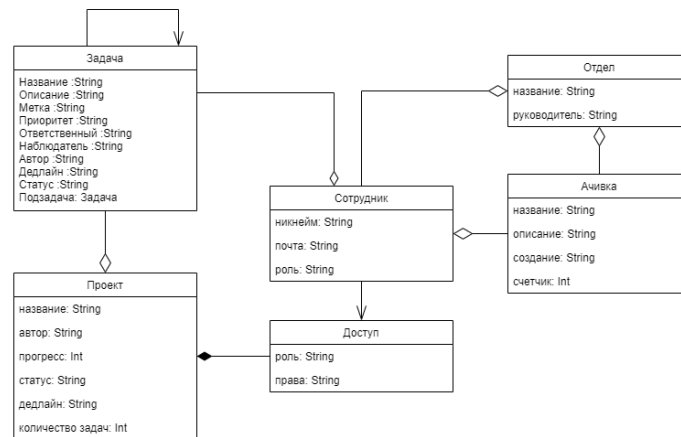


Рис. 1. Модель предметной области

Рассмотрим некоторые из представленных сущностей.

Сущность "Задача" – действие или набор действий, которые пользователь должен выполнять; может разбиваться на подзадачи, о чем свидетельствует вложенность.

"Проект" – набор задач, сгруппированных по определенному направлению работ, определяемых в соответствии с требованиями конкретного предприятия. К проекту может быть выдан доступ – это возможность пользователя видеть проекты и задачи по ним.

"Ачивка" – вознаграждение, которое пользователь получает за совершение определенных действий в процессе работе с task-менеджером. Ачивка может быть уникальной для конкретного отдела и принадлежать сотруднику, который ее получил.

4.1. Общая архитектура

В результате проделанной работы реализовано web-приложение, которое имеет клиент-серверную архитектуру, поэтому реализация разделена на две части – клиентскую и серверную.

Общую архитектуру диктует выбор PHP-фреймворк Laravel, т.к. он предназначен для разработки с использованием архитектурной модели MVC.

Model-View-Controller (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») – схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — чтобы модификация каждого компонента могла осуществляться независимо [2].

Есть одно отличие от стандартной схемы, пакет представление (View) пустой, а отображение происходит через клиентскую часть, а связь с серверной частью идет че-

рез API. Но все же архитектура создаваемой системы представляет концепцию Model-View-Controller или MVC

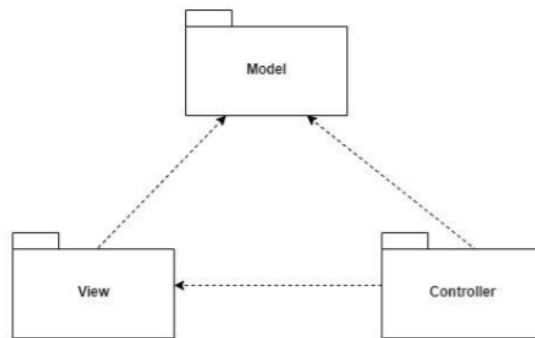


Рис. 2. Диаграмма пакетов концепции Model-View-Controller

4.2. Серверная часть web-приложения

Серверная часть приложения была написана на языке PHP с использованием Laravel. В ходе работы были реализованы следующие модули: `app.php`, `config.php`, `database.php`, `routes.php`, `test.php`.

Модуль `app.php` – это основной модуль, где описано множества объектов с их полями и методами. Модуль `config.php` содержит в себе конфигурацию серверной части, например, настройку подключения к базе данных для приложения и конфигурацию. Модуль `database.php` содержит работу с реляционной базой данных. Модуль `routes.php` содержит все маршруты, связанные с веб-интерфейсом или API web-приложения.

4.3. Клиентская часть web-приложения

Данная часть web-приложения разработана с использованием JavaScript фреймворка Vue.js. Для отображения клиентской части потребовались следующие основные компоненты: `main`, `home`, `advantages`, `projects`, `tasks`.

В данной работе был использован паттерн управления состоянием Vuex. Он служит центральным хранилищем данных для всех компонентов приложения [3].

Код состоит из четырёх логических блоков:

- `state` – в нем хранится массив данных;
- `mutations` – синхронные функции, которые используются для записи данных;
- `actions` – асинхронные функции, которые инициируют обращение к серверному;
- `getters` – используются для организации хранилища в нескольких небольших файлах.

5. Описание реализованного web-приложения

В результате проделанной работы получилось web-приложение, некоторые скриншоты которого представлены ниже на рисунках.

Регистрация происходит по специальной ссылке от администратора или руководителям компании, затем при первом входе в систему пользователь увидит приветствие от личного помощника Иннокентия о том, как работать в task-менеджере (рис. 3).

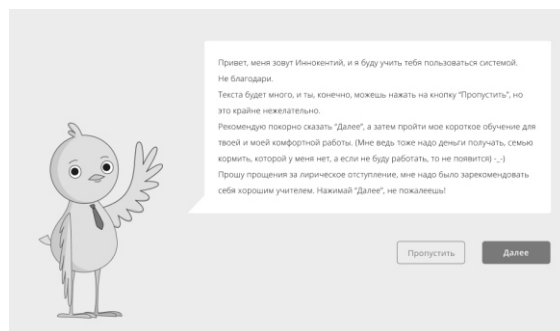


Рис. 3. Отображение приветствия

После авторизации пользователь попадает на экран профиля пользователя, где может посмотреть статистику по текущим задачам, изменить пароль или информацию профиля. Затем пользователь попадает в раздел «Мои задачи», из которого можно перейти обратно в Профиль пользователя или в раздел «Проекты». Если пользователь является руководителем (роль указывается при регистрации), то он может добавить проект.

Также пользователь может добавить задачу и в ней указать название, описание, добавить ссылку, выбрать исполнителя, наблюдателя, привязать задачу к Проекту, указать дату и время дедлайна, планируемое и фактическое время выполнения задачи, ее приоритет и выбрать метку.

При создании первой задачи пользователь получает достижение «Самостоятельный» и эта ачивка в списке становится цветной.

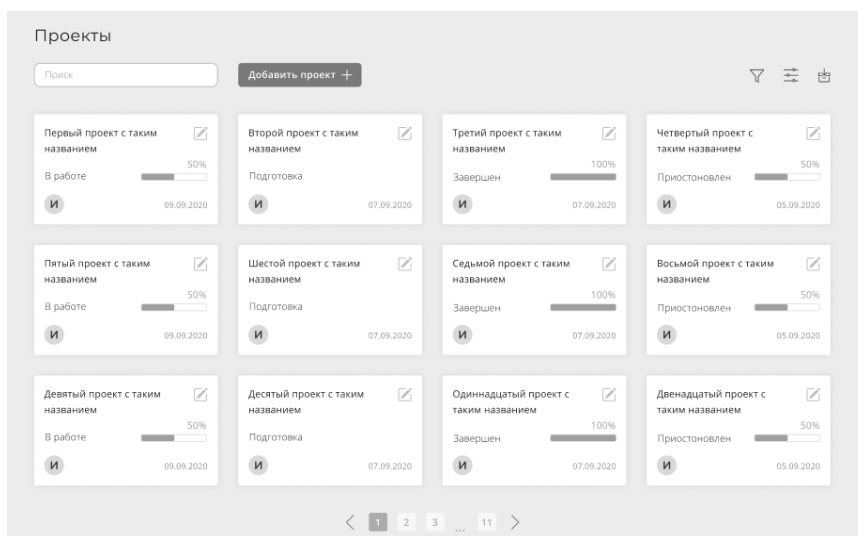


Рис. 4. Отображение проектов пользователь

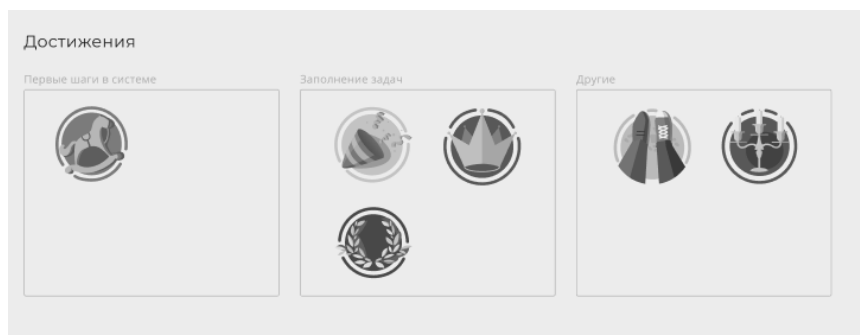


Рис. 5. Отображение списка ачивок

Заключение

В результате данной работы было спроектировано, разработано и протестировано web-приложение, которое гарантирует более высокую удовлетворенность сотрудников работе в компании за счет внедрения системы геймифицированных вознаграждений сотрудников и постепенный процесс ознакомления нового сотрудника с рабочими процессами.

ЛИТЕРАТУРА

- 1.8 психотипов + HEXAD Марчевского / Octalysis Group. – [Б. м.], 2005–2021. – URL: <https://octalysis.ru/psihotipy-bartla/> (дата последнего обращения: 26.05.2021).
2. Model-View-Controller / freejournal. – [Б. м], 2020. – URL: <https://ru.freejournal.org/213101/1/model-view-controller.html> (дата обращения: 10.06.2021).
3. Vue.js documentation [Электронный ресурс] // Vue.js. – Электрон, дан. – [Б.м.], 2020. – URL: <https://ru.vuejs.org/> (дата обращения: 01.06.2021).

DOI: 10.17223/978-5-907442-42-9-2021-33

РАЗРАБОТКА AR-ПРИЛОЖЕНИЯ ДЛЯ ВИЗУАЛИЗАЦИИ ИСТОРИЧЕСКОЙ ЗАСТРОЙКИ ГОРОДА ТОМСКА

Попов Н.С., Шкуркин А.С.

Томский государственный университет
popovns99@mail.ru, shkurkin@mail.ru

Введение

Большинство информации об архитектуре и исторической застройке города люди чаще получают дистанционно (видео на телевидении, в сети интернет), реже это происходит естественным способом. Это связано с цифровизацией общества или отсутствием возможности попасть в место, где находится объект архитектуры. Для решения данной проблемы поможет AR-приложение, в котором человек может посмотреть интересующее его здание исторической застройки.

Технология дополненной реальности (AR) уже достаточно активно используется в повседневной жизни в различных сферах деятельности. Часто для создания дополненной реальности используется смартфон. С помощью AR-приложения дополненная реальность позволит в интересной и непривычной форме взаимодействовать с историческими зданиями города. Приложение с такой функциональностью будет интересно использовать в разных сферах деятельности, будь то расширение кругозора людей, привлечение внимания общественности к проблеме уменьшения количества исторической застройки или изучение архитектуры города дистанционно (например, изучение конкретных зданий исторической застройки для архитекторов).

1. Постановка задачи

Целью работы является разработка приложения с дополненной реальностью для визуализации исторической застройки города Томска. Приложение будет своего рода живой открыткой. Основная идея живой открытки заключается в том, что через приложение на смартфоне пользователь сможет выбрать интересующее его здание и увидеть его 3D-модель.

Задачи исследования:

1. Провести анализ предметной области.
2. Провести обзор аналогичных приложений.
3. Сформировать требования к разрабатываемому приложению.
4. Изучить области знаний для проведения исследования и программной разработки ПО: AR, 3D-моделирование.
5. Изучить программные средства разработки: Unity и ARCore.
6. Спроектировать мобильное AR-приложение.
7. Реализовать AR-приложение.

2. Механика работы AR в Unity

Разберем механику работы дополненной реальности в Unity [1]. Один из первых вопросов, который возникает: “Как смартфон понимает, что его вращают или перемещают?” Для этого в смартфонах установлен гироскоп или система IMU.

Далее, после того, как телефон получает сигнал, что он вращается, идет передача сигнала в приложение, и таким же образом вращается камера в проекте Unity. Схема работы показана на рис. 1.

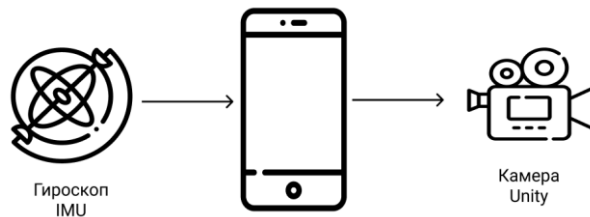


Рис. 1. Схема работы AR в Unity

Пока приложение работает, будет существовать два параллельных мира: реальный, где вы передвигаетесь с телефоном, и цифровой, где вместо телефона есть камера Unity.

Если пользователь располагает какие-то объекты и камера в Unity их видит, то они отображаются на экране вашего смартфона. Объекты отображаются в виде альфа-канала – это такой формат прозрачной картинки, который заполняется цветом только в отдельной области. Например, в реальной жизни вы могли видеть его в виде бегущей строки рекламы в телевизоре. Данный альфа-канал накладывается поверх того, что видит камера. Именно поэтому в AR [2,3] можно видеть объекты сквозь стены.

3. Функциональные требования

Приложение предоставляет возможность увидеть здание исторической застройки в виде 3D-модели, а также позволит взаимодействовать с ней. Для просмотра и взаимодействия с 3D-моделью дома реализовано понятное и удобное приложение, в котором можно без проблем ориентироваться и взаимодействовать с 3D-моделями домов.

Так как приложение направлено на ознакомительную и образовательную деятельность, пользоваться им смогут как студенты, так и обычные люди, которым просто интересно узнать что-то новое.

Таким образом, к функциональным требованиям приложения можно отнести:

- выбрать 3D-модель дома;
- добавить 3D-модель дома по маркеру на виртуальную плоскость;
- перемещать 3D-модель по виртуальной плоскости;
- вращать 3D-модель вокруг своей оси для детального осмотра со всех сторон;
- удалить 3D-модель при нажатии на нее пальцем;
- войти в приложение;
- выйти из приложения.

4. Проектирование архитектуры

На основе сформированных требований к приложению была спроектирована общая диаграмма вариантов использования, изображенная на рис. 2.

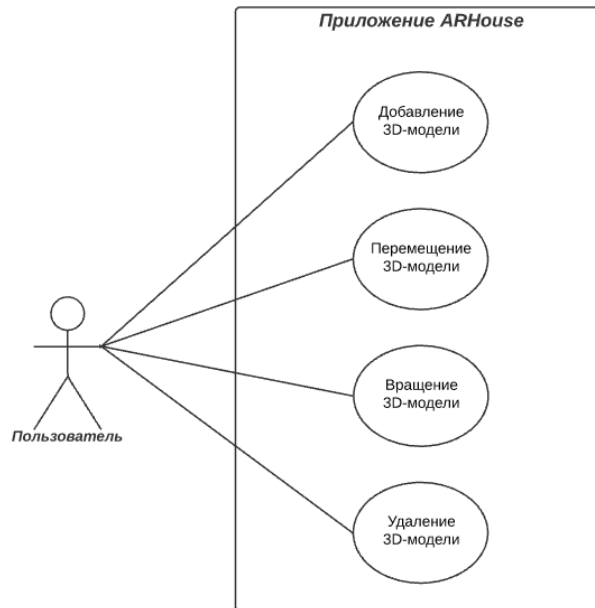


Рис. 2. Общая диаграмма вариантов использования

Операции актера «Пользователь»:

- Перемещение 3D-модели дома – пользователь может переместить 3D-модель дома по виртуальной плоскости, зажав пальцем модель и потянув в нужное место.
- Вращение 3D-модели дома – пользователь может вращать 3D-модель дома. Вращение активируется специальной кнопкой; если горит красный свет, значит, вращать модель нельзя.
- Удаление 3D-модели дома – пользователь может удалить 3D-модель дома. Эта функция также активируется специальной кнопкой. Для того, чтобы удалить 3D-модель дома, необходимо нажать на него пальцем, и дом удалится.

Вариант использования «Добавление 3D-модели» показан на рис. 3.

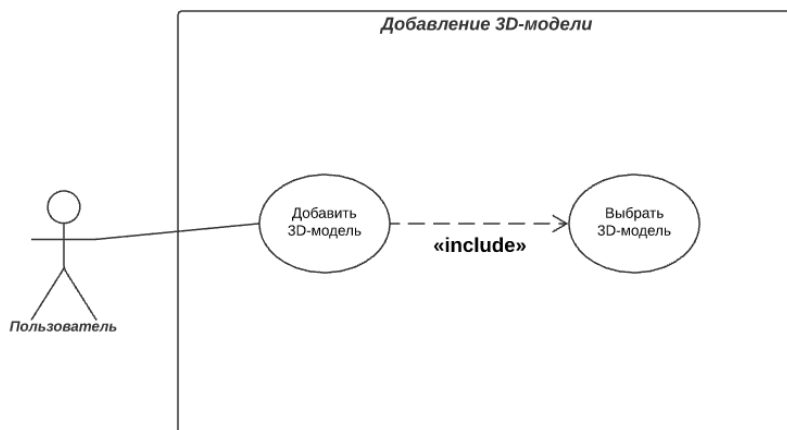


Рис. 3. Вариант использования «Добавление 3D-модели»

Также для приложения была спроектирована диаграмма классов, изображенная на рис. 4.

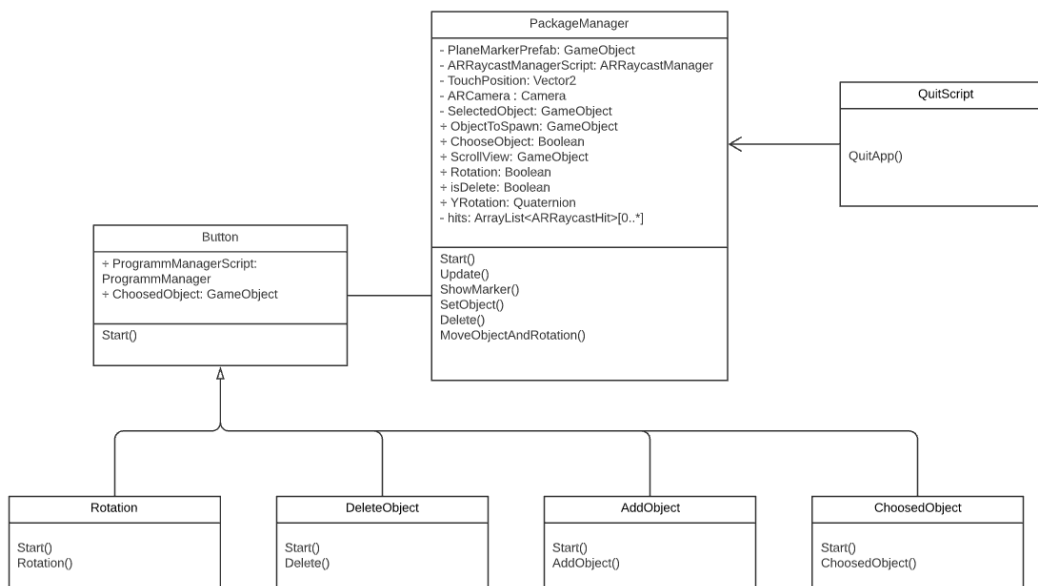


Рис. 4. Диаграмма классов

Рассмотрим подробнее основные классы.

“PackageManager” – содержит в себе основную логику приложения и работу с дополненной реальностью. В классе “PackageManger” имеются следующие атрибуты и операции:

- PlaneMarkerPrefab: GameObject – Контейнер, где будет храниться маркер.
- ARRaycastManagerScript: ARRaycastManager – Скрипт, который был добавлен на AR Session Origin. Из ARRaycastManagerScript берутся лучи Raycast для определения плоскости.
- TouchPosition: Vector2 – Позиция, куда коснулись пальцем. Она будет двумерной, а не трехмерной.
- ARCamera: Camera – Ячейка для камеры дополненной реальности.
- ObjectToSpawn – Объект, который будет устанавливаться по маркеру.
- SelectedObject: GameObject – Переменная, где будет храниться выбранный объект.

- ChooseObject: Boolean – Флаг, для определения выбран объект или нет.
- ScrollView: GameObject – Переменная для хранения кнопки, которая скрывает выпадающий список.
- Rotation: Boolean – Флаг для отслеживания, включена функция поворота 3D-модели дома или выключена.
- isDelete: Boolean – Флаг для отслеживания, включена функция удаления 3D-модели дома или выключена.
- YRotation: Quaternion – Переменная, которая содержит углы по трем осям координат.
- hits:ArrayList<ARRaycastHit>[0..*] – Список, где будут храниться объекты, которые попали на пути луча Raycast.
- Start() – Функция вызывается до обновления первого кадра или физики объекта.
- Update() – Функция нужна для изменении позиции, состояния и поведения объектов в игре прямо перед отрисовкой кадра. Update вызывается перед отрисовкой кадра и перед расчётом анимаций.
- ShowMarker() – Функция показывает положение маркера на виртуальной плоскости.
- SetObject() – Установка 3D-модели дома на виртуальную плоскость.
- Delete() – Удаление 3D-модели дома.
- MoveObjectAndRotation() – Перемещение и поворот 3D-модели дома.

Класс “QuitScript” содержит логику закрытия приложения. В классе “QuitScript” имеется единственная операция QuitApp, которая закрывает приложение.

Button и все наследуемые от него классы содержат логику взаимодействий с кнопками. В классе “Button” имеются следующие атрибуты и операции:

- ProgrammManagerScript: ProgrammManager – Ссылка на скрипт ProgrammManager.
- ChosedObject: GameObject – Флаг, для определения выбран объект или нет.
- Start() – Функция вызывается до обновления первого кадра или физики объекта.

5. Интерфейс приложения

После разработки основной функциональности приложения, а именно – работы с дополненной реальностью и функциональностью взаимодействия с 3D-моделями, был разработан простой и понятный интерфейс. На рис. 5, 6 представлена часть функциональности работы приложения.

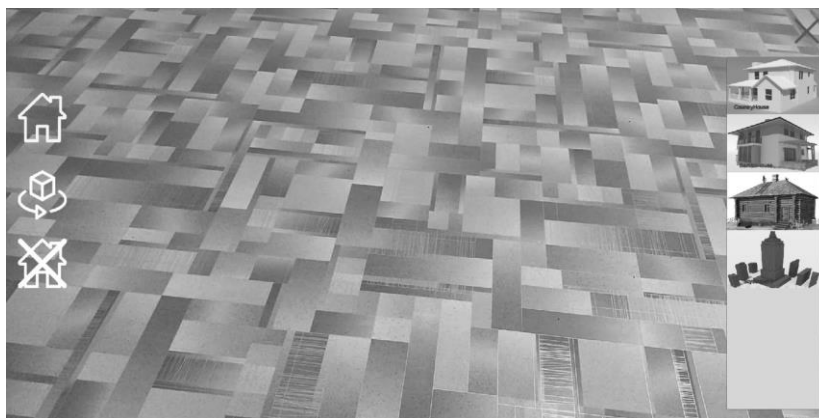


Рис. 5. Выбор 3D-модели из списка

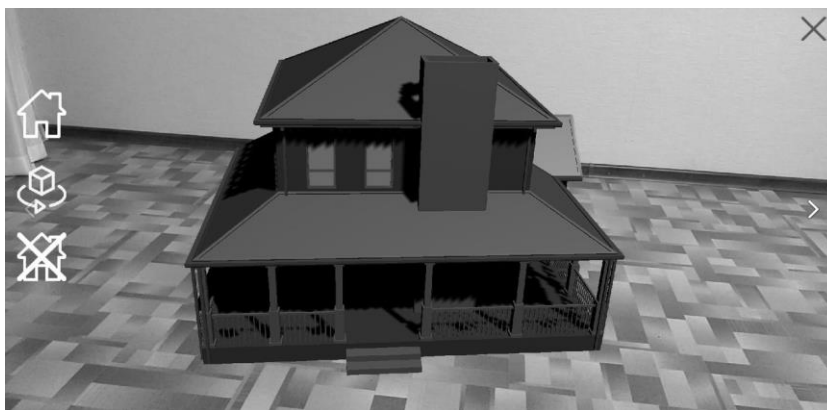


Рис. 6. Добавление 3D-модели

Заключение

В ходе выполнения данной работы были изучены в необходимом объеме области знаний: 3D-моделирование, AR; программные средства: Unity, ARCore, ARFoundation; найдена 3D-модель, спроектирована функциональность программы, и, как итоговый результат, реализовано AR-приложение для визуализации исторической застройки города Томска.

Разработанное приложение можно применять как для исторической, так и для не-исторической застройки любого города.

ЛИТЕРАТУРА

1. Unity [Электронный ресурс] // Unity, платформа разработки в реальном времени URL: <https://unity.com/ru> (дата обращения 16.05.2021).
2. AR – Дополненная реальность [Электронный ресурс] // Habr.com. URL: <https://habr.com/ru/post/419437/> (дата обращения 16.05.2021).
3. Свой AR. Основы векторной алгебры – Дополненная реальность [Электронный ресурс] // Habr.com. URL: <https://habr.com/ru/post/547876/> (дата обращения 16.05.2021).

VIII. ПРОГРАММНЫЕ И АППАРАТНЫЕ АСПЕКТЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

DOI: 10.17223/978-5-907442-42-9-2021-34

EVOLUTION OF COMPUTER VIRUSES

Chikalova S.V., Tkachuk P.R., Lavrinenko M.D.

Tomsk State University

chikalova.sveta@mail.ru, tkachuk_polina@bk.ru, mila-gygy@mail.ru

Introduction

Today, the massive use of personal computers, unfortunately, is associated with the emergence of self-replicating virus programs that interfere with the normal operation of the computer, destroy the file structure of disks and damage the information stored in the computer. Increasingly, there are reports in the media about all sorts of pirate tricks by computer hooligans, about the emergence of more and more sophisticated self-distributing programs.

A computer virus is a type of malware that can inject itself into the code of other programs, system memory areas, boot sectors, and distribute copies of itself through various communication channels.

The creation and distribution of malicious programs (including viruses) is prosecuted in Russia in accordance with the Criminal Code of the Russian Federation (Chapter 28, Article 273).

Nowadays, things are changing too quickly, so we consider it necessary to compare one of the first computer viruses with one of the last.

1. Formulation of the problem

This paper examines one of the first computer viruses, the Morris worm, and one of the latest computer viruses, WannaCry. On the basis of the studied theoretical aspects and generalization of theoretical material, viruses are compared, their similarities and differences are assessed. The aim of the work is to study the influence of computer viruses on the operation of computers, this study will help to deepen knowledge in the field of the theory of computer viruses.

2. Morris worm

One of the earliest computer viruses is the Morris Worm, which was written by Cornell University graduate student Robert Tappan Morris and launched at about 6:00 pm on November 2, 1988 at the Massachusetts Institute of Technology [1].

Morris's father was an outstanding programmer. He has worked in cryptography, worked at Bell Labs, and helped develop Multics and Unix. At the time of the cyberattack, Robert's father was in a senior position in the division responsible for IT security of federal government systems. That is, the future programmer had someone to learn from.

2.1. Morris worm attack method

The Morris worm only infected the Unix operating system, exploiting known vulnerabilities in programs such as: Sendmail – email of the time, Finger is a utility that allows users to obtain information about other users, as well as weak passwords, which were often a repetition of a login consisting of a name and the user's last name, or some popular word. The worm completely paralyzed the system by repeatedly rewriting its own code and launching several copies of itself at the same time, which clogged memory and clogged network channels. A peculiarity of the Morris worm is that the computer could become infected several times, gradually reducing the system's performance, eventually disabling it.

Attempts to investigate the problem and eliminate malicious files failed for a long time due to the fact that system files were frequently updated, changed, or information was destroyed.

2.2. The purpose of the Morris worm

Interestingly, the worm was not originally designed as a super-effective virus; according to Robert Morris, it was only for educational purposes, as an experimental project. It all happened because of the carelessness of the administrators and the mistake of the creator. The error was that the virus had a special spreading mechanism: the program, before introducing itself into the system, had to ask if there is already a copy of it in this system, but instead of giving a positive answer with an unambiguous command, Morris added a randomizer to the code, which in one of the seven cases of a positive answer still loaded the program into the system. This led to the fact that many devices were infected several times.

2.3. Removing the Morris worm

The activity of the worm was easy to detect due to specific security flaws in standard servers. As a result, within twelve hours after the beginning of the infection, the worm was able to catch, decompile, and develop a number of countermeasures (the simplest is to rename the cc compiler so that the worm cannot compile its "head"). For troubleshooting, leading American information technology specialists were assembled, who managed to completely fix the problem in two days, creating a special program that ultimately destroyed the virus.

2.4. Morris worm damage

Robert Morris really had no idea that he would create a virus of this magnitude and was himself afraid of what he had done. As follows from the report on the results of the investigation of the incident, he made some attempts to stop the epidemic, although the "switch" was not provided. Robert, as soon as he learned about the seriousness of the situation, called his friend with a request to publish an anonymous apology and a guide to "treatment", but he did not go into direct contact and did not inform those in charge.

The Morris worm practically paralyzed computers on the network for up to five days. Computers were paralyzed for at least 8 million hours and over 1 million hours [2] were spent restoring systems to work. Total losses in monetary terms were estimated at about \$98 million, they were the sum of direct and indirect losses.

Direct losses included (\$32 million): stopping, testing and rebooting 42700 machines; identification of the worm, deleting, cleaning memory and restoring the performance of 6200 machines (this is about 10% of all computers connected to the network at that time); analysis of the worm's code, disassembling and documenting; fixing UNIX systems and testing.

Indirect losses were attributed (\$66 million): loss of computer time due to lack of access to the network; loss of user access to the network.

As for the punishment of Robert himself, then in 1989, Robert appeared in court, becoming the very first person charged under a new article on computer crimes – the Computer Fraud and Abuse Act. He faced up to five years in prison and a quarter of a million dollars in fines. But he eventually received 3 years and a \$10 thousand fine, and was also sentenced to 400 hours of community service (the programmer considered this punishment excessive, but the appeal was rejected).

Why did Morris avoid the real deadline that the public demanded? Because Robert had no criminal plan and was not hiding anything.

3. WannaCry Computer Virus

Among the huge number of malicious programs, there are unique individuals. Having studied the most famous ones, we present you one of the most dangerous viruses of the last decade.

WannaCry (also known as WannaCrypt, WCry, WanaCrypt0r 2.0, and Wanna Decryptor) is a malware, network worm, and ransomware that only infects computers running Microsoft Windows. After infecting a computer, the worm's program code encrypts almost all files stored on the computer and offers to pay a ransom in cryptocurrency to decrypt them. If the ransom is not paid within 7 days from the moment of infection, the ability to decrypt files is lost forever [3].

An early version of this virus, called WeCry, appeared in February 2017 and extorted 0.1 bitcoin (\$177 at current exchange rates). WanaCrypt0r is an improved version of this malware, in which attackers can specify any amount and increase it over time. The developers of the virus are unknown and it is not a fact that they are behind the attacks. They may well sell malware to anyone who wants it for a one-time payment.

The mass distribution of WannaCry began on May 12, 2017 – first of all, they attacked computers in Spain, and then in other countries. In Russia, ministries (Ministry of Internal Affairs of Russia), MegaFon suffered. The information systems of Russian Railways were affected, but the worm quickly localized and did not affect train traffic. In addition, computers in the UK, Spain, Italy, Germany, Portugal, Turkey, Ukraine, Kazakhstan, Indonesia, Vietnam, Japan and the Philippines were affected by the virus.

Over 98% of WannaCry ransomware infections occur on computers running Windows 7, with over 60% of infections affecting the 64-bit version of the OS. Such data were released by Kaspersky Lab analysts. According to statistics, less than 1% of infected computers are running Windows Server 2008 R2 and Windows 10 (0.03%).

The WannaCry worm exploits a vulnerability in Windows operating systems to infect computers. The WannaCry worm itself was created and launched by unknown attackers using information stolen from the National Security Agency.

After penetrating the folder with documents and other files, the virus encrypts them, changing the extensions to .WNCRY. The malware then requires the purchase of a special key, which costs between \$300 and \$600, threatening to delete the files otherwise.

In general, WannaCry is an exploit that is used to infect and spread, plus an ransomware that is downloaded to the computer after the infection has occurred.

This is an important difference between WannaCry and most other ransomware. In order to infect your computer with an ordinary, let's say, ransomware, the user must make some mistake – click on a suspicious link, allow executing a macro in Word, download a dubious attachment from a letter. You can get infected with WannaCry without doing anything at all.

3.1. WannaCry attack method

The malware scans Internet sites for computers with open TCP port 445, which is responsible for supporting the SMBv1 protocol. Having found such a computer, the program makes several attempts to exploit the EternalBlue vulnerability on it and, if successful, installs the DoublePulsar backdoor, through which the executable code of the WannaCry program is loaded and launched. On every attempt to use it, the malware checks for the presence of DoublePulsar on the target computer and, if detected, downloads directly through this backdoor.

It encrypts files of various types, including office documents, photographs, films, archives and other file formats, which may contain potentially important information for the user. Encrypted files are given the .WCRY extension (hence the name of the ransomware) and become completely unreadable.

After that, he changes the desktop wallpaper, displaying there a notification about the infection and a list of actions that supposedly need to be performed in order to return the files. WannaCry scatters the same notifications in the form of text files among folders on the computer so that the user does not miss it.

After completing the encryption process, the program displays a window with a request to transfer a certain amount in bitcoins (equivalent to \$300) [4] to the specified wallet within three days. If the ransom is not received in a timely manner, its amount will be automatically

doubled. On the seventh day, if WannaCry is not removed from the infected system, the encrypted files are destroyed.

The ransom payments are pointless, as the individual keys will never be sent and the files will remain encrypted.

3.2. The purpose of WannaCry

The action of the Wanna Cry virus is aimed at getting on a computer, encrypting the files on it, and extorting a ransom for their decryption.

3.3. Removing WannaCry

The 22-year-old blogger known on the Internet as MalwareTech was analyzing the malware code and noticed that the software was trying to reach an unusual web address – iu-qerfsodp9ifjaposdfjhgosurijfaewrwergwea.com – but that address was not linked to the website because it was not identified [5].

Whenever the malware tried to connect to the mysterious website, an error would occur and the virus would act, causing damage.

As an experiment, MalwareTech decided to spend \$10 69 cents and registered the specified web address. As the owner of the web address, he gained access to analytical data and gained an understanding of the scale of online extortion.

But this discovery was not the only one – he soon found that after registering the web address, the spread of the malware stopped.

MalwareTech believes that the malware did not self-destruct, but was "frightened" by the so-called virtual machine – a protected platform used, among other things, to detect viruses and read their program code.

A real computer cannot access the meaningless address iu-qerfsodp9ifjaposdfjhgosurijfaewrwergwea.com, unlike a virtual machine, which is able to simulate connecting to it as a genuine website.

Recently, malware creators have been embedding a special verification code in them to see if the machine they have entered is virtual. If so, the virus is not activated so as not to be destroyed.

"The malware stops immediately to prevent further analysis," MalwareTech writes on its blog. "Because of my logging, all these viruses thought they had entered the virtual machine and logged out. ... So we are inadvertently prevented the spread of ransomware. "

The WannaCry virus can be removed, but all files that were encrypted with WannaCry are still not decrypted.

Experts also warn of the possible appearance of new, more advanced modifications of WannaCry.

Experts from Kaspersky Lab and the antivirus company Symantec, based on a tweet published by Google researcher Neil Mehta, noted that the WannaCry code signatures coincide with a code signature allegedly used in February 2015 by the Lazarus Group, an alleged connection with the DPRK government.

Linguistic analysis of the text of the ransom demand by specialists from the American cybersecurity company Flashpoint showed that the main language of the authors of WannaCry is most likely the southern dialect of Chinese.

3.4. Damage from WannaCry

According to experts, in the first four days of a large-scale cyberattack, about 300 thousand users in 150 countries of the world were affected. The total damage is estimated at \$1 billion.

4. Comparison and analysis of these two viruses

The development of technology does not stand still, computers have changed a lot since the beginning of their first appearance (1927), their capabilities have expanded, they have become more productive, faster, and multifunctional.

Thanks to technological progress, PCs have fewer weaknesses, which is why viruses are becoming more dangerous and sophisticated.

We are comparing two network worms, the Morris worm and the WannaCry worm, for more accurate analysis.

Similarity of viruses:

The Morris worm and WannaCry belong to the same type of malware. Both viruses are worms.

Explanation: A worm is a subclass of a virus, meaning every worm is a virus, but not every virus is a worm. The virus multiplies on its own, it can gradually infect files not only within one computer, but also spread to other devices. The worm, on the other hand, multiplies without infecting other files; it is injected onto a specific PC. The main difference is in the code. The worm has its own code, while the virus injects itself into files that are already in the system [6].

Differences between viruses:

1. The WannaCry code is ahead of the Morris worm in complexity. As described earlier, a bug was made in the code of the first virus, leading to an overflow of memory, which simply showed the imperfection of computers of that time. The WannaCry code was deliberately complex.

2. The goals of the creators of network worms were different. The creators of WannaCry demanded a ransom for decrypting the data, and Robert Morris, creating the Morris worm, acted solely for educational purposes.

3. Elimination of WannaCry happened only partially, since the virus is new and its modifications are still taking place. If we talk about the Morris worm, after the discovery of the problem, the specialists found a solution within 2 days.

4. The consequences of the impact of WannaCry are more global, since even if the network worm is removed, the files will not be decrypted. To remove the Morris worm, it was necessary to take a number of countermeasures, after which the system was not disrupted or data was lost.

5. The total monetary damage caused by the Morris worm and WannaCry, adjusted for inflation, is different. At the 2017 rate for the Morris worm, the damage would be approximately \$217 million, which is about 21.7% of WannaCry's total damage.

Conclusion

Thus, with each passing year, viruses become more sophisticated and more difficult to get rid of. They cause serious financial damage and cause PC crashes, so users need to be vigilant. There are viruses, but there are also antiviruses that must be installed and used to avoid data loss, collection of personal information about you, and computer malfunctions. After collecting data and conducting a comparative analysis, we came to the conclusion that our hypothesis was indeed confirmed. The WannaCry virus is more dangerous than the Morris worm.

REFERENCES

1. Morris worm [Electronic resource]: website / Wikipedia – URL: https://en.wikipedia.org/wiki/Morris_worm (date accessed: 29.04.2021)
2. Morris worm — он был первым [Electronic resource]: website / Habr – URL: <https://habr.com/ru/post/160337/> (date accessed: 29.04.2021)
3. WannaCry ransomware attack [Electronic resource]: website / Wikipedia – URL: https://en.wikipedia.org/wiki/WannaCry_ransomware_attack (date accessed: 29.04.2021)
4. Эпидемия шифровальщика WannaCry: что произошло и как защититься [Electronic resource]: website / kaspersky.ru – URL: <https://www.kaspersky.ru/blog/wannacry-ransomware/16147/> (date accessed: 30.04.2021)
5. "Случайный герой" рассказал, как он остановил вирус WannaCry [Electronic resource]: website / bbc.com – URL: <https://www.bbc.com/russian/news-39909763> (date accessed: 01.05.2021)
6. Eugene H. Spafford The Internet Worm Program: An Analysis [Electronic resource]: website / spaf.cerias.purdue.edu – URL: <https://spaf.cerias.purdue.edu/tech-reps/823.pdf> (date accessed: 03.05.2021)

EFFICIENT FPGA IMPLEMENTATION OF A RECONFIGURABLE FSM BASED ON SUBSTITUTIONS

Razekov S.I.

Tomsk State University
sirazekov@stud.tsu.ru

Introduction

Reconfigurable finite-state machines (R-FSMs) are used for designing complex digital systems with modifiable behavior. There exist various kinds of R-FSMs: Self-R-FSM [1], transition-based R-FSM [2], R-FSM with input multiplexing and state-based input selection architecture [3], R-FSM based on substitutions [4]. The R-FSM based on substitutions can find application in cryptography.

In this paper, R-FSM based on substitutions is implemented on an FPGA platform. Substitution representation techniques are compared, namely table representation, system of Boolean functions (disjunctive/algebraic normal forms), formulas with specific operations (ARX, butterfly structure). To estimate performance and resource utilization of a design we use following parameters: maximum path delay, maximum clock frequency, number of Look-Up-Tables (LUTs) used in design implementation.

1. Reconfigurable finite-state machine based on 4-bit substitutions

Architecture of the R-FSM based on substitutions is presented on Fig. 1. Substitutions *Sub X*, *Sub Y*, *Sub S* are modifiable and *Sub 1*, *Sub 2*, ..., *Sub N* are fixed. Two multiplexers *M1* and *M2* are used for building the output and state-transition functions, respectively. The current state of the R-FSM is stored in the register *Reg*. Input alphabet, set of states and output alphabet and are all sets of numbers with $\log_2 N$ bit size.

Configuration of the R-FSM is done by initializing the modifiable substitutions with the keys (K_X, K_Y, K_S). As a result, we have a strongly connected, reduced and reversible FSM [4], which can be used in FSM-based ciphers, particularly in Zakrevskij cipher [5].

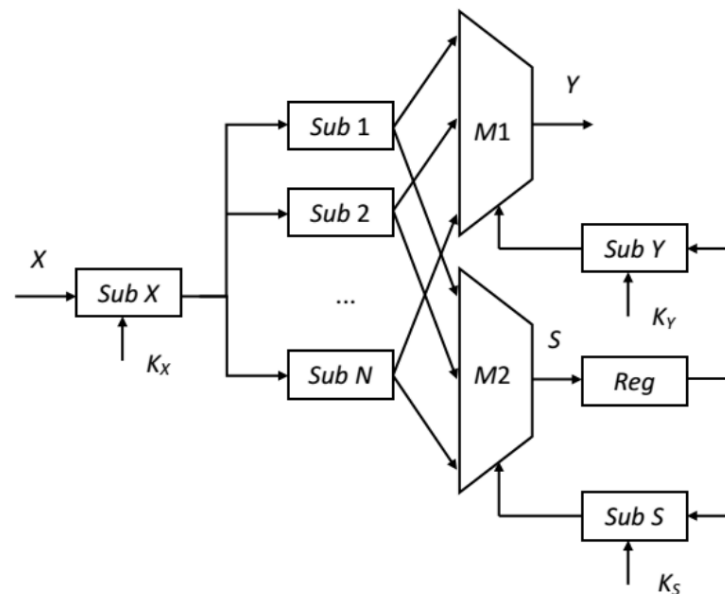


Fig. 1. Structure of the R-FSM based on substitutions

Input alphabet, set of states and output alphabet in our FPGA implementation are sets of 4-bit numbers with cardinality equal to 16. Thus, each substitution in the R-FSM takes a 4-bit input and transforms it into a 4-bit output. The modifiable substitution is implemented as a block of RAM memory [6]. Also a multiplexer with 16 4-bit data inputs and a single 4-bit select input was designed. The current state of the R-FSM is stored in a register with 4 flip-flops.

2. Fixed substitutions representation techniques

2.1. Table representation

The most common way to represent a substitution is a table which shows the mapping directly. In hardware description language notation it can be described as: RAM memory, ROM memory (case statement) [6], and directly as a substitution table (selected signal assignment statement [6]). It turned out that the last two approaches synthesize in a same circuit. When we describe a substitution as RAM memory, the specially configured LUT (Look-Up-Table) elements are used, which behave as distributed RAM memory [7]. In all other cases presented in this paper we simply obtain a combinational circuit made of LUT elements. LUT is the main resource of an FPGA. It is a logic element which can implement an arbitrary Boolean function of a fixed number of inputs. Here is an example of table representation:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 13 & 5 & 0 & 6 & 7 & 15 & 14 & 8 & 4 & 2 & 9 & 1 & 10 & 12 & 3 & 11 \end{pmatrix}. \quad (1)$$

2.2. System of Boolean functions

Boolean function is a function which maps a set of Boolean values to a single Boolean value: $\{0,1\}^n \rightarrow \{0,1\}$, where n is a non-negative integer. System of Boolean functions is a finite set of Boolean functions. If we take a substitution table and change the decimal number representation to the binary form, we obtain a set of Boolean functions each corresponding to a bit of the substitution result. The number of arguments that each function takes equals the size of the substitution argument in binary form. In this paper, we represented the Boolean functions in full disjunctive normal form (FDNF) and algebraic normal form (ANF). For example, substitution (1) can be represented as:

$$\begin{cases} S_0(X) = 1 \oplus x_1 \oplus x_3, \\ S_1(X) = (x_0 \wedge x_1) \oplus x_2 \oplus (x_0 \wedge x_3), \\ S_2(X) = 1 \oplus x_1 \oplus (x_0 \wedge x_1) \oplus (x_1 \wedge x_2) \oplus (x_0 \wedge x_3) \oplus (x_2 \wedge x_3), \\ S_3(X) = 1 \oplus x_0 \oplus x_1 \oplus (x_0 \wedge x_1) \oplus x_2 \oplus x_3 \oplus (x_0 \wedge x_3), \end{cases}$$

where $X = (x_3, x_2, x_1, x_0)$ is a 4-bit argument and $S(X) = (S_3(X), S_2(X), S_1(X), S_0(X))$ is the substitution result; x_i and $S_j(x)$ are the i -th and j -th bits of the argument x and result $S(x)$ respectively, counting from the least significant bit.

2.3. Formula (ARX, butterfly structure)

Some substitutions can be described as formulas with specific operations: modular addition, bitwise XOR, finite field multiplication, etc. In this paper, we implemented ARX-based substitutions [8] and substitutions with butterfly structure [9]. However, instead of proposed 8-bit substitutions, we implemented 4-bit substitutions with similar structure.

For example, substitution (1) can be described with a round function $g(X)$ which runs 5 times:

$$\begin{aligned} g(X) &= f_2 \circ f_1(X), \\ f_2(X) &= \left(b(X_1), b\left(\left(\left(\left(X_2 \square 1\right) + 2\right) \bmod 4\right) \oplus b(X_1)\right) \right), \\ f_1(X) &= \left(b\left(\left(\left(X_1 \square 1\right) + X_2\right) \bmod 4\right), b(X_2) \right), \end{aligned}$$

where $g(X)$ is a composition of functions f_1 and f_2 , X is a 4-bit argument, $b(X) = (x_3, x_2, x_1, x_0)$ is the binary form of number X , $X \ll i$ is right circular shift of number X by i bits, argument X is represented here as a concatenation of two 2-bit binary vectors $b(X_1) = (x_3, x_2)$ and $b(X_2) = (x_1, x_0)$: $X = (b(X_1), b(X_2))$.

3. FPGA implementation of R-FSM

The project has been synthesized on Xilinx Artix-7 FPGA. VHDL [6,10] description of the R-FSM was made and simulated in Vivado Design Suite developer tool [11].

S-boxes of various ciphers (DES, Gift, PRIDE, PRINCE, KLEIN, NOEKEON, Piccolo) and some randomly created substitutions which, in contrast, can also be described with formulas, are taken and used as fixed substitutions. In cryptography, an S-box (substitution-box) is a basic component of block ciphers. There are many ciphers with S-boxes mapping 4-bit input into 4-bit output. The ciphers use tables to represent S-boxes.

We use tool for creating a substitution table if there is given an ARX-formula or butterfly structure and an automatic VHDL code generator which produces VHDL code for a substitution table.

Various substitution representation techniques (table, system of Boolean functions, formula) are compared. Efficiency is determined by evaluating timing (Tab. 1) and utilization parameters (Tab. 2) of the implemented design which include: maximum path delay, maximum clock frequency, number of LUTs used in design.

Table 1

Timing parameters

Substitutions	Representation technique	Max. delay	Max. clock frequency
ARX-based substitutions	RAM memory	14.385 ns	69.516 MHz
	ROM memory / Substitution table	14.285 ns	70.003 MHz
	Formula	13.805 ns	72.437 MHz
	FDNF	14.035 ns	71.250 MHz
	ANF	14.125 ns	70.796 MHz
Substitutions with butterfly structure	RAM memory	14.385 ns	69.516 MHz
	ROM memory / Substitution table	14.186 ns	70.492 MHz
	Formula	13.859 ns	72.155 MHz
	FDNF	13.808 ns	72.421 MHz
	ANF	14.312 ns	69.871 MHz
S-boxes	RAM memory	14.385 ns	69.516 MHz
	ROM memory / Substitution table	14.350 ns	69.686 MHz
	FDNF	13.766 ns	72.642 MHz
	ANF	13.812 ns	72.400 MHz

Table 2

Utilization parameters

Substitutions	Representation technique	LUT
ARX-based substitutions	RAM memory	109
	ROM memory / Substitution table	74
	Formula	
	FDNF	
	ANF	
Substitutions with butterfly structure	RAM memory	109
	ROM memory / Substitution table	71
	Formula	
	FDNF	
	ANF	
S-boxes	RAM memory	109
	ROM memory / Substitution table	76
	FDNF	
	ANF	

All implementations also use 4 flip-flops and dedicated multiplexers F7 MUX, F8 MUX [7] (16 and 8, respectively).

Conclusion

This article presents research results on different substitution representation techniques for FPGA implementation of a reconfigurable finite-state machine based on 4-bit substitutions. Various substitution representation techniques are used, namely table representation (memory, substitution table), ARX-formula, butterfly structure, system of Boolean functions (disjunctive/algebraic normal forms). Nevertheless, all designs, except the one, where substitutions are described as RAM memory, have roughly the same utilization and timing parameters. However, when the fixed substitutions are designed as RAM memory, the R-FSM implementation requires significantly more LUTs and has a slightly lower maximum clock frequency.

REFERENCES

1. *Koster M., Teich J.* (Self-)reconfigurable finite state machines: theory and implementation // Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition – 2002 – P. 559–566.
2. *Glaser J., Damm M., Haase J., Grimm C.* TR-FSM: Transition-based Reconfigurable finite state machine // ACM Transactions on Reconfigurable Technology and Systems (TRETS). – 2011. – V. 4. – № 3. – Article №23.
3. *Das N., Aruna Priya P.* FPGA Implementation of Reconfigurable Finite State Machine with Input Multiplexing Architecture Using Hungarian Method // International Journal of Reconfigurable Computing. – 2018. – V. 2018. – Article ID 6831901 – 15 p.
4. *Trenkaev V.N.* Reconfigurable finite-state machine based on substitutions // *Prikladnaya Diskretnaya Matematika. Prilozhenie.* – 2019. – № 12. – P. 192–193.
5. *Trenkaev V.N.* Zakrevskij's cipher based on reconfigurable FSM // *Prikladnaya Diskretnaya Matematika.* – 2010. – № 3(9). – P. 69–76.
6. *Harris D.M., Harris S.L.* Digital Design and Computer Architecture. Second edition. – Morgan Kaufmann, 2013. – 720 p.
7. Series FPGAs Configurable Logic Block (UG474) – URL: https://www.xilinx.com/support/documentation/user_guides/ug474_7Series_CLB.pdf
8. *Komissarov S.M.* On algorithmic implementation of 16-bit S-boxes with ARX and Butterfly structures // *Prikl. Diskr. Mat. Suppl.* – 2019 – 12 – P. 101–107.
9. *Fomin D.B.* New classes of 8-bit permutations based on a butterfly structure // *Mat. Vopr. Kriptogr.* – 2019. – V. 10. – № 2. – P. 169–180.
10. *Ashenden P.J.* The designer's guide to VHDL. 3rd ed. – Elsevier Inc., 2008.
11. Vivado Design Suite User Guide : Synthesis (UG901) – URL: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_2/ug901-vivado-synthesis.pdf

DOI: 10.17223/978-5-907442-42-9-2021-36

ЧТЕНИЕ ДАННЫХ С ЭКСПЕРИМЕНТАЛЬНОЙ УСТАНОВКИ

Безходарнов Н.И., Самохина С.И.

*Томский государственный университет
nblaaa@mail.ru*

Введение

В одной из лабораторий Томского кардиологического центра ведётся работа по снятию показаний с нагруженной сердечной мышцы. Для этого в лаборатории имеется оборудование, предназначенное для экспериментов с сердечной мышцей. Именно для этой установки была поставлена задача разработать информационную систему, в которой выполняется снятие показаний при проведении эксперимента, их оцифровка и математическая обработка. Установка для экспериментов состоит из считывающего устройства, модификаторов сигнала и преобразователя аналогового сигнала в цифровой. Для считывания данных необходимо связаться с преобразователем. Далее полученная кривая аппроксимируется с помощью метода сетки, выполняется анализ полученных данных и визуализация.

1. Постановка задачи

Рассматривается экспериментальная установка, предназначенная для чтения показателей напряжённости сердечных мышц. Чтобы получить с этой установки данные, требуется понять, как она передаёт выходные сигналы, и разработать модуль, предназначенный для чтения данных. На выходе должна быть получена кардиологическая кривая, которая в дальнейшем подвергается обработке, а именно – аппроксимации двумя компонентами экспоненциального вида. Задачей данной работы является получение с экспериментальной установки кардиологической кривой и её двухкомпонентное разложение [1–3].

Основной целью данного проекта является разработка информационной системы, реализующей чтение данных с установки и 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 задачи:

1. Изучение экспериментальной установки.
2. Разработка модуля чтения данных с установки.
3. Реализация модуля аппроксимации функции для реализации алгоритма.
4. Объединения всех модулей в информационную систему.

2. Функциональные возможности информационной системы

В качестве средств реализации выбрана среда 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 эксперимента, или из файлов, содержащих результаты прошлых экспериментов; отображение графиков на основе полученных данных; нахождение разложения кардиологической кривой, её построение и вывод подробной информации о результатах. Для реализации данного функционала были разработаны модули чтения данных с установки и написан алгоритм, позволяющий найти аппроксимирующую кривую к кардиологической.

3. Описание алгоритма получения данных с экспериментальной установки

Для реализации модуля чтения данных, прежде всего, требовалось понять, как устроен вывод сигнала в экспериментальной установке. Установка состоит из устройства, считывающего показания напряжённости, множества модификаторов сигнала и преобразователя аналогового сигнала в цифровой. Для получения сигнала достаточно установить связь с этим преобразователем и получить от него данные в числовом виде. Экспериментальная установка представляет собой устройство модели DT9802, которое преобразовывает аналоговый сигнал в числовой вид. Данное устройство было разработано компанией Measurement Computing. Также данной компанией была разработана библиотека Universal Library [5], которая позволяет взаимодействовать с подобными преобразователями. Эта библиотека и была использована при разработке модуля. Для полного взаимодействия с устройством необходимо осуществить следующие действия:

1. Получить список устройств.
2. Подключиться к нужному устройству.
3. Выставить все необходимые параметры.
4. Запустить процесс чтения данных.
5. Получить данные с устройства.
6. Остановить процесс чтения данных.
7. Отключиться от устройства.

Получение списка устройств осуществляется функцией `olDaEnumBoards`, которая принимает функцию пополнения списка на 1 элемент и указатель на сам список. В случае отсутствия подключенных устройств эта функция возвращает ошибку.

Подключение к устройству осуществляется функцией `olDaInitialize`, которая принимает на вход название устройства и структуру для работы с драйвером. Пока программа не подключится к устройству, дальнейшая работа с ним невозможна. При подключении требуется подготовить устройство к чтению данных, а именно: выставить режим работы `OL_DF_CONTINUOUS` для непрерывного чтения данных командой `olDataSetDataFlow`, выставить режим работы буферов `OL_WRP_MULTIPLE` для возможности их повторного использования командой `olDataSetWrapMode` и выставить частоту 1000 Гц командой `olDaSetClockFrequency`.

Далее, необходимо выставить все параметры устройства, а именно: канал (`olDaSetChannelListEntry`), фильтр (`olDaSetChannelFilter`), границы (`olDaSetRange`) и множитель (`olDaSetGainListEntry`). Получить всевозможные вариации данных параметров можно с помощью функции `olDaEnumSSCaps`, в которой требуется указать необходимый тип параметра, функцию для пополнения списка на 1 элемент и указатель на список. Затем, получив и выбрав из предоставленных вариантов нужные, требуется убедиться в том, что все параметры были выставлены верно. Данная процедура осуществляется функцией `olDaConfig`. После этого уже можно начать считывать данные.

Для запуска чтения данных прежде всего нужно подать на устройство дескриптор окна, т.к. передача данных осуществляется посредством системных сигналов, отправляемых приложению. Осуществляется это командой `olDaSetWndHandle`. Далее требуется выделить на устройстве память для хранения прочитанных данных, чтобы их можно было передать компьютеру. Выделением памяти занимается функция `olDmCallocBuffer`, а после неё требуется вызвать функцию `olDaPutBuffer`, чтобы привести созданный буфер в состояние готовности к записи. После этого можно запустить чтение данных с помощью функции `olDaStart`.

Передача данных осуществляется посредством системных сигналов. Для того, чтобы получить эти сигналы, нужно использовать функцию `nativeEvent`, являющуюся частью библиотеки Qt. Т.к. сигналы бывают разные, то сначала требуется убедиться, что мы получили нужный. При получении сигнала приходит его идентификатор, по которому можно определить тип сообщения. Для считывания данных с устройства нужно получить сигнал с идентификатором `OLDA_WM_BUFFER_DONE`, в этом случае можно воспользоваться буфером, возвращаемым функцией `olDaGetBuffer`, затем узнать количество валидных элементов функцией `olDmGetValidSamples` и получить указатель на массив данных функцией `olDmGetBufferPtr`. Далее данные считываются с устройства и сохраняются в программе для дальнейшей обработки.

После того, как все данные были получены, нужно остановить чтение. Для этого достаточно вызвать функцию `olDaStop`, затем циклом получить все буферы функцией `olDaGetBuffer` и освободить их функцией `olDmFreeBuffer`.

Остаётся только отключиться от устройства с помощью следующих 2 функций: `olDaReleaseDASS` (освобождает устройство) и `olDaTerminate` (отключает программу от драйвера).

Таким образом осуществляется взаимодействие с устройством. Описанная выше концепция работы позволяет получить валидные данные, которые могут в дальнейшем быть подвергнуты обработке.

4. Описание алгоритма аппроксимации

Алгоритм, предложенный в [6], 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).

Для решения этой задачи используется метод сетки, который заключается в следующем: задаётся начальное приближение и некоторый шаг для отступа, 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.

5. 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 загрузить данные о кардиологической кривой. Это можно сделать, нажав кнопку «Построить график», предварительно выбрав в списке выше то, откуда считать данные. Есть возможность получать данные либо из файлов, либо с экспериментальной установки. В главном окне появится специальное поле, где отобразится график, с которым можно взаимодействовать с помощью указателя мыши. Для удобства задания промежутка аппроксимации 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 станет неактивным, и на нём появится окно с просьбой подождать. После вычислений отображаются результаты аппроксимации (рис. 1).

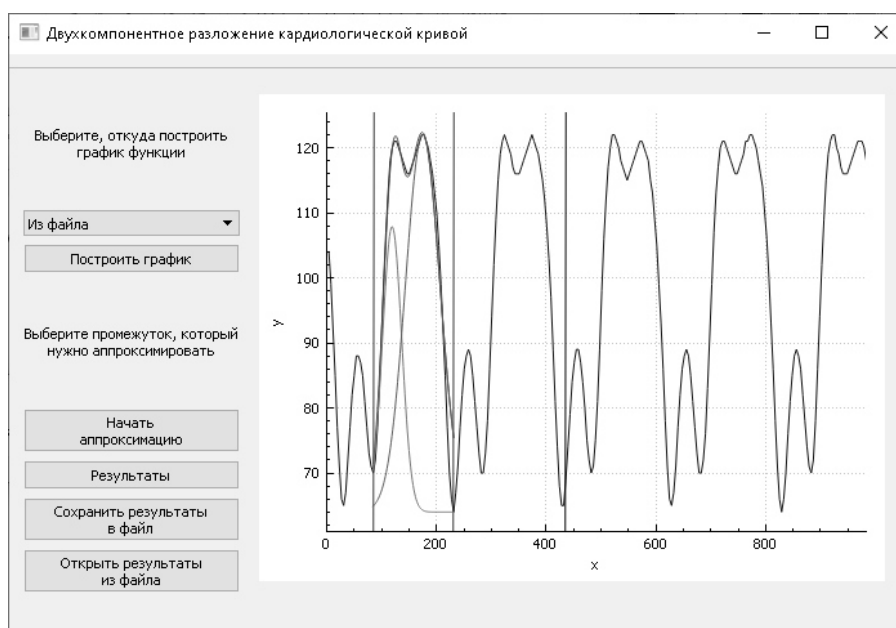


Рис. 1. Главное окно программы после вычислений

Также после вычислений станет активной кнопка «Результаты», нажав которую, можно открыть отдельное окно с различной информацией о найденной функции: график функции, её компонент и её первых двух производных (рис. 2), таблица значений функции и её производных, аналитический вид и другое. Можно сохранить результаты вычислений в файл, чтобы позже загрузить их в программу без повторного пересчёта.

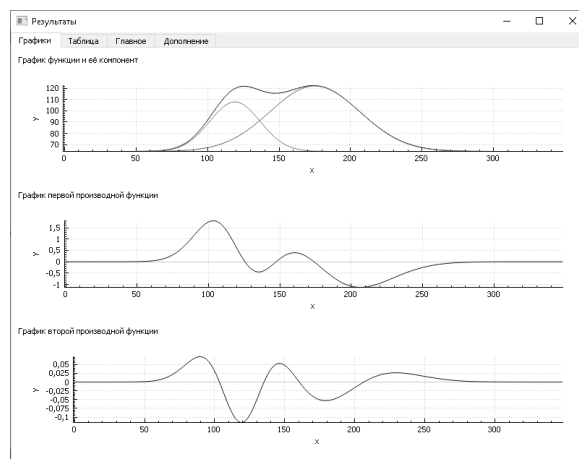


Рис. 2. Окно информации о результатах

В результате реализована информационная система с удобным графическим интерфейсом, которая решает поставленную задачу.

Заключение

В данной работе создана информационная система, которая осуществляет чтение данных с экспериментальной установки, сохраняет данные в виде кардиологической кривой и её разложение в экспоненциальную функцию. Данные с устройства считываются корректно и целостно сохраняются в файл, аппроксимация находится достаточно близкая к оригинальной функции, результаты выводятся на экран наглядно и информативно. Интерфейс разработан интерактивный и понятный. Алгоритм аппроксимации протестирован на множестве входных данных, и получившиеся результаты оказались достаточно корректными.

На данный момент система находится на этапе внедрения в производственный процесс проведения эксперимента с сердечной мышцей. Надеемся, что в будущем данная информационная система 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. – Томск, 1991. – 184 с.
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 экономических 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. Data Translation, Inc. Dtx-EZ™ Getting Started Manual [Текст]: руководство пользователя / Data Translation, Inc. – Marlboro.: Data Translation, Inc., 2006. – 234 с.
6. Безходарнов Н.И., Самохина С.И. Двухкомпонентное разложение кардиологической кривой // Математическое и программное обеспечение информационных, технических и экономических систем : материалы Международной научной конференции, Томск, 28–30 мая 2020 г. – Томск: Изд-во Том. ун-та, 2020. – С. 86–90.

ПОДХОД К ПОВЫШЕНИЮ БЕЗОПАСНОСТИ ВСТРАИВАНИЯ ИНФОРМАЦИИ В ФАЗОВЫЙ СПЕКТР ДПФ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ*

Мельман А.С.

*Национальный исследовательский университет «Высшая школа экономики»
amelman@hse.ru*

Введение

Широкое распространение мультимедиа-технологий открыло возможности для возникновения нового канала передачи данных, незаметного для потенциального злоумышленника. Для эффективного сокрытия самого факта наличия каких-либо секретных сведений их встраивают в различные цифровые контейнеры, такие как изображения, аудио- и видеофайлы, текстовые документы. Для этого применяются методы цифровой стеганографии. В частности, в настоящей работе рассматривается сокрытие информации в цифровых изображениях.

Основная идея стеганографии заключается в том, чтобы сделать секретное вложение незаметным для третьих лиц. На сегодняшний день цифровая стеганография является сложившимся научным направлением, и многие исследовательские группы со всего мира регулярно предлагают новые алгоритмы стеганографического встраивания, отличающиеся разными характеристиками качества. Подавляющее большинство современных алгоритмов обеспечивают незаметность встроенного сообщения для человеческого глаза. Однако более строгим требованием к безопасности стеганографических алгоритмов является обеспечение устойчивости к стегоанализу, т.е. к попыткам обнаружить наличие встроенного сообщения в конкретном контейнере. Это значит, что изображение-контейнер и стегоизображение должны быть статистически неразличимы. Выполнение данного требования представляет собой сложную научную задачу, поскольку встраивание дополнительной информации всегда вносит искажения в естественную модель цифрового изображения.

В данной работе предлагается новый подход к повышению безопасности стеганографического встраивания в фазовый спектр дискретного преобразования Фурье (ДПФ) цифрового изображения за счёт устранения пиков на гистограмме распределения фазовых значений коэффициентов ДПФ.

1. Встраивание информации в фазовый спектр ДПФ

Методы сокрытия информации в цифровых изображениях делятся на два больших класса в зависимости от того, какие именно элементы данных изображения будут изменяться в процессе встраивания. Методы пространственного встраивания оперируют пикселями цифровых изображений [1,2], в то время как методы частотного встраивания работают с коэффициентами частотного преобразования, применённого к матрице пикселей [3,4].

В работе [5] одного из авторов настоящего исследования был предложен алгоритм частотного встраивания информации в фазовый спектр ДПФ цифровых изображений. Основной особенностью этого алгоритма является возможность безошибочного извлечения встроенной информации. Многие существующие алгоритмы частотного встраивания не позволяют избежать искажения встроенного сообщения при извлечении, поскольку округление вещественных значений коэффициентов при переходе к целочисленным пикселям приводит к необратимой потере части информации. Алгоритм [5]

* Работа выполнена при финансовой поддержке гранта РФФИ и Томской области в рамках проекта № 19-47-703003.

успешно справляется с этой проблемой за счёт использования итеративной процедуры встраивания. Это значит, что ещё на этапе встраивания происходит проверка на возникновение ошибок при последующем извлечении. Если такие ошибки возникают, происходит повторное встраивание информации в текущий блок коэффициентов. В итоге либо удаётся достичь устойчивого безошибочного извлечения для данного блока, либо его элементы изменяются таким образом, чтобы ни один из них не использовался при извлечении информации.

Для встраивания используется 21 элемент фазового спектра. При этом изменениям подвергаются не все элементы из области встраивания, конкретные значения, которые будут содержать биты информации, выбираются таким образом, чтобы минимизировать число изменений, вносимых в блок.

Рассмотрим подробнее процедуры встраивания и извлечения информации. Фазовые значения коэффициентов ДПФ могут принимать значения из диапазона $(-\pi; \pi]$. Выберем такие два значения из данного диапазона, что $\varphi_0 = -\varphi_1$. Тогда встраивание очередного бита информации b в фазовое значение φ происходит по формуле

$$\varphi' = \begin{cases} \varphi_0, & \text{если } b = 0, \\ \varphi_1, & \text{если } b = 1, \end{cases}$$

где φ' – фазовое значение коэффициента ДПФ после встраивания.

На этапе извлечения данных выполняется обратное ДПФ, округление значений для получения целочисленных пикселей, и повторное ДПФ. При этом фазовые значения блока коэффициентов могут искажаться, поэтому при извлечении используется интервал значений, ширина которого определяется параметром алгоритма ε . Тогда формула извлечения выглядит следующим образом:

$$b = \begin{cases} 0, & \text{если } \varphi'' \in (\varphi_0 - \varepsilon; \varphi_0 + \varepsilon), \\ 1, & \text{если } \varphi'' \in (\varphi_1 - \varepsilon; \varphi_1 + \varepsilon), \end{cases}$$

где φ'' – фазовое значение коэффициента ДПФ при извлечении.

2. Постановка проблемы

Алгоритм [5] характеризуется визуальной незаметностью встраивания при средней ёмкости 0.17 бит/пиксель. Рис. 1 иллюстрирует пример изображения-контейнера и соответствующего стегоизображения для классического изображения «Peppers» из базы The USC-SIPI [6].

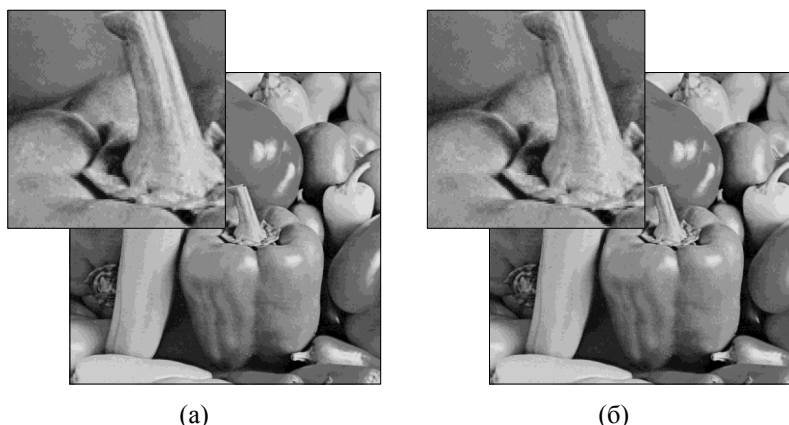


Рис. 1. Изображение-контейнер (а) и соответствующее стегоизображение (б)

Пример на рис. 1 показывает, что даже при увеличении масштаба изображения факт наличия секретного сообщения остаётся скрытым. Однако это наблюдение не гарантирует статистической неразличимости изображений до и после встраивания.

Построим гистограммы распределения фазовых значений коэффициентов ДПФ для изображения-контейнера и соответствующего стегоизображения. Такие гистограммы для изображений, представленных на рис. 1, приводятся на рис. 2.

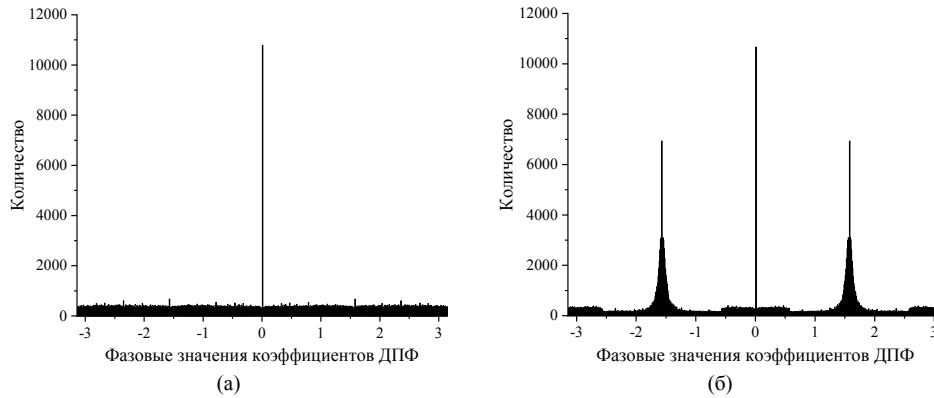


Рис. 2. Гистограмма распределения фазовых значений коэффициентов ДПФ изображения-контейнера (а) и соответствующего стегоизображения (б)

Сравнение гистограммы на рис. 2б с гистограммой на рис. 2а делает очевидным наличие стеганографического вложения. При встраивании информации по алгоритму [5] фазовые значения заменяются фиксированной величиной, определяемой параметром алгоритма встраивания. В данном случае при встраивании нулевого бита фазовое значение коэффициента ДПФ меняется на $\varphi_0 = -1.57$, единичного бита – на $\varphi_1 = 1.57$. При выполнении обратного ДПФ, округления и повторного выполнения ДПФ данные значения могут изменяться, однако их значения по-прежнему остаются близки к исходным, именно это позволяет впоследствии извлекать встроенную информацию. Поэтому на гистограмме, изображённой на рис. 2б, видны отчётливые пики в районе значений ± 1.57 . Гистограммы других стегоизображений, полученных по алгоритму [5], имеют аналогичный вид. Это серьёзная уязвимость, которая позволяет элементарным образом обнаружить наличие встроенного сообщения и скомпрометировать скрытый канал передачи данных. Поэтому необходима разработка подхода, позволяющего устранить демаскирующие вложение признаки.

3. Предлагаемый подход

Характерные пики на гистограмме стегоизображения возникают по причине возникновения большого количества значений, близких к φ_0 и φ_1 , после встраивания дополнительной информации. Наличие пиков в районе фиксированного значения, равного параметру алгоритма встраивания, свидетельствует о необходимости использовать переменное значение параметра. В настоящей работе предлагается отказаться от использования фиксированного значения параметров φ_0 и φ_1 и выбирать новые значения, удовлетворяющие соотношению $\varphi_0 = -\varphi_1$ из $(-\pi; \pi]$, при переходе от блока к блоку. Необходимо принять во внимание, что интервалы извлечения $(\varphi_0 - \varepsilon; \varphi_0 + \varepsilon)$ и $(\varphi_1 - \varepsilon; \varphi_1 + \varepsilon)$ не должны пересекаться, чтобы избежать ошибок при извлечении встроенной информации. Однако в случае реализации предлагаемого изменения возможна ситуация, при которой границы интервала извлечения пересекаются или выхо-

дуют за пределы диапазона $(-\pi; \pi]$. С учётом этого замечания, схему извлечения необходимо скорректировать. Новые правила извлечения сформулированы в табл. 1.

Значения фазового спектра коэффициентов ДПФ имеют распределение, близкое к равномерному, за исключением большого количества значений в районе нуля. Поэтому, чтобы избежать заметных искажений гистограммы стегоизображения, в настоящем исследовании предлагается для выбора φ_0 и φ_1 использовать псевдослучайную последовательность значений из интервала $(-\pi; \pi]$, сгенерированную в соответствии с равномерным распределением.

Таблица 1

Новые правила извлечения встроенной информации

b	Условие для $\varphi_0 \pm \varepsilon$ или $\varphi_1 \pm \varepsilon$	Условие для φ''
0	$\varphi_0 - \varepsilon > -\pi$ и $\varphi_0 + \varepsilon < 0$	$\varphi'' \in (\varphi_0 - \varepsilon; \varphi_0 + \varepsilon)$
0	$\varphi_0 - \varepsilon < -\pi$ и $\varphi_0 + \varepsilon < 0$	$\varphi'' \in (-\pi; \varphi_0 + \varepsilon) \cup (\varphi_0 - \varepsilon + \pi; 0)$
0	$\varphi_0 - \varepsilon > -\pi$ и $\varphi_0 + \varepsilon > 0$	$\varphi'' \in (-\pi; \varphi_0 + \varepsilon - \pi) \cup (\varphi_0 - \varepsilon + \pi; 0)$
1	$\varphi_1 - \varepsilon > 0$ и $\varphi_1 + \varepsilon < \pi$	$\varphi'' \in (\varphi_1 - \varepsilon; \varphi_1 + \varepsilon)$
1	$\varphi_1 - \varepsilon < 0$ и $\varphi_1 + \varepsilon < \pi$	$\varphi'' \in (0; \varphi_1 + \varepsilon) \cup (\varphi_1 - \varepsilon + \pi; \pi)$
1	$\varphi_1 - \varepsilon > 0$ и $\varphi_1 + \varepsilon > \pi$	$\varphi'' \in (0; \varphi_1 + \varepsilon - \pi) \cup (\varphi_1 - \varepsilon; \pi)$

Для обеспечения безошибочного извлечения встроенной информации данная последовательность должна также быть известна на стороне получателя секретного сообщения. Однако передача вместе с каждым стегоизображением большого объёма дополнительной информации может привлечь внимание злоумышленника и скомпрометировать наличие скрытого канала передачи данных. В случае псевдослучайной последовательности отправителю и получателю данных достаточно заблаговременно обменяться начальным значением, которое использовалось для генерации чисел. Это значение вместе со значением ε можно использовать как ключ и многократно использовать при организации связи между одной парой пользователей. В этом случае не требуется передавать большой объём информации, уникальной для каждой пары контейнер-вложение.

4. Результаты экспериментов

Для выполнения вычислительных экспериментов из классической базы тестовых изображений [6] были выбраны 20 изображений в градациях серого размером 512×512 пикселей.

Для того, чтобы убедиться в эффективности предложенного подхода к корректровке искажений гистограммы с точки зрения стегоанализа, была выполнена проверка статистической значимости различий между гистограммами контейнера и стегоизображения с помощью непараметрического критерия Уилкоксона. В табл. 2 представлены значения p -уровня значимости для данного критерия. Чтобы две гистограммы считались статистически неразличимыми, должно выполняться условие $p > 0.05$.

Как следует из таблицы, при встраивании информации по алгоритму [5] до применения нового подхода, обеспечивающего сглаживание пиков на гистограмме фазовых значений коэффициентов ДПФ, во всех случаях $p < 0.05$, т.е. стегоизображение и изображение-контейнер статистически различаются. Применение предложенного подхода кардинально меняет ситуацию: только в 4-х случаях из 20-ти стегоизображение и контейнер оказались статистически различимы. В остальных случаях предложенный подход успешно решил проблему обнаружения стеганографического вложения с помощью анализа гистограмм. Это позволяет говорить о повышении безопасности стеганографи-

ческого встраивания дополнительной информации в фазовый спектр ДПФ цифровых изображений.

На рис. 3 представлен пример стегоизображения, полученного с применением нового подхода, а также соответствующая гистограмма распределения фазовых значений коэффициентов ДПФ. Можно отметить, что заметные искажения гистограммы, демаскирующие вложение, успешно устранены.

Таблица 2

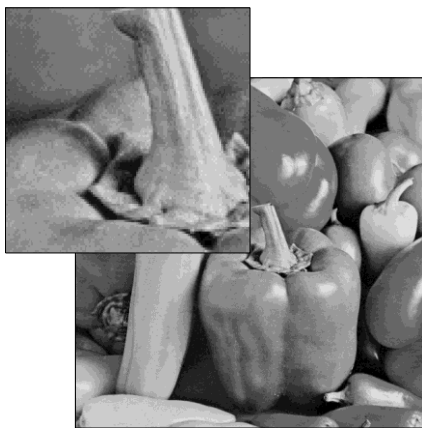
Сравнение статистической неразличимости гистограмм контейнера и стегоизображения до и после применения предлагаемого подхода

Номер	Изображение	p-уровень значимости критерия Уилкоксона	
		До применения нового подхода	После применения нового подхода
1	Aerial	2.1854×10^{-38}	0.7998
2	Airplane	1.7984×10^{-29}	0.0632
3	Airport	2.7428×10^{-52}	0.9402
4	Aps	4.2270×10^{-45}	0.1780
5	Baboon	7.2229×10^{-45}	0.7341
6	Barbara	1.2037×10^{-34}	0.2617
7	Boat	1.3879×10^{-44}	0.2542
8	Couple	2.6993×10^{-36}	0.2593
9	Elaine	3.8320×10^{-49}	0.5765
10	Goldhill	1.1670×10^{-43}	0.8652
11	House	7.4557×10^{-30}	0.0056
12	Lake	1.3508×10^{-42}	0.5016
13	Lena	1.8444×10^{-35}	0.0941
14	Man	2.8794×10^{-35}	8.7503×10^{-9}
15	Peppers	2.3284×10^{-44}	0.3847
16	Splash	3.1147×10^{-21}	0.0017
17	Stream and Bridge	5.5202×10^{-44}	0.0002
18	Tank	4.1831×10^{-49}	0.4102
19	Tiffany	8.5237×10^{-35}	0.2350
20	Truck	5.2274×10^{-35}	0.1812

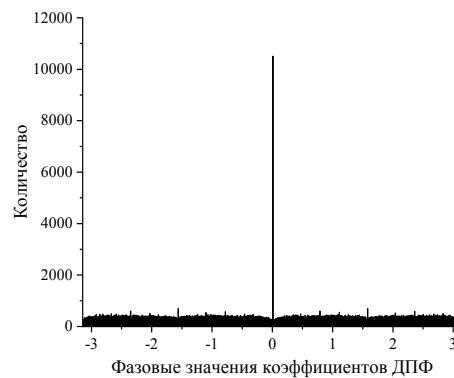
Рис. 3а демонстрирует визуальную незаметность вложения. Для численной оценки незаметности встраивания в стеганографии применяется метрика пиковое отношение сигнала к шуму (PSNR), которая вычисляется по формуле

$$PSNR = 20 \log_{10} \left(\frac{255}{RMSE} \right), \quad RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (P_i - Q_i)^2},$$

где n – общее количество пикселей, P_i – значение пикселя контейнера, Q_i – значение пикселя стегоизображения.



(а)



(б)

Рис. 3. Стегоизображение, полученное с использованием нового подхода, (а) и соответствующая гистограмма (б)

На рис. 4а представлена гистограмма значений PSNR для используемых изображений. Эти значения PSNR соответствуют максимальному заполнению контейнера, соответствующие значения ёмкости отражены на рис. 4б.

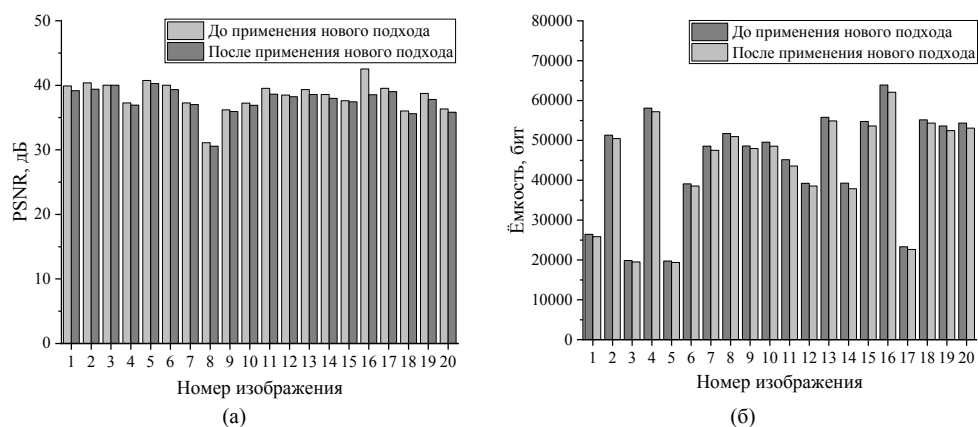


Рис. 4. Значения метрики PSNR (а) и соответствующие значения максимальной ёмкости (б)

Из рис. 4 следует, что использование предложенного подхода привело к уменьшению максимальной ёмкости и значения PSNR. Однако данное уменьшение является небольшим. При этом безопасность встраивания, связанная с увеличением уровня устойчивости к стегоанализу, существенно повышается. Можно сделать вывод, что применение нового подхода к сглаживанию гистограмм фазовых значений коэффициентов ДПФ эффективно решает проблему уязвимости к статистическому стегоанализу без значительных потерь ёмкости и незаметности встраивания.

Заключение

Методы стеганографии позволяют незаметным образом скрывать дополнительную информацию в цифровых изображениях. Важным требованием к стеганографическим алгоритмам является устойчивость перед стегоанализом. В настоящей работе представлен подход к повышению устойчивости перед стегоанализом встраивания информации в фазовый спектр ДПФ цифровых изображений за счёт использования разных значений параметра встраивания для каждого блока коэффициентов. Результаты экспериментов демонстрируют, что псевдослучайный выбор нового значения фазы при встраивании успешно сглаживает демаскирующие признаки на гистограмме распределения фазовых значений коэффициентов ДПФ стегоизображения и делает изображение-контейнер и стегоизображение статистически неразличимыми.

ЛИТЕРАТУРА

1. Verma V., Muttoo S.K., Singh V.B. Enhanced payload and trade-off for image steganography via a novel pixel digits alteration // *Multimedia Tools and Applications*. – 2020. – V. 79. – № 11-12. – P. 7471–7490.
2. Shukur W.A., Jabbar K.K. Information hiding using LSB technique based on developed PSO algorithm // *International Journal of Electrical and Computer Engineering*. – 2018. – V. 8. – № 2. – P. 1156–1168.
3. Maheswari S.U., Hemanth D.J. Performance enhanced image steganography systems using transforms and optimization techniques // *Multimedia Tools and Applications*. – 2017. – V. 76. – № 1. – P. 415–436.
4. Evsutin O., Dzhanaashia K. Algorithm of Information Embedding into Digital Images Based on the Chinese Remainder Theorem for Data Security // *Cryptography*. – 2020. – V. 4. – № 4. – Art. no. 35.
5. Evsutin O., Kokurina A., Meshcheryakov R., Shumskaya O. The adaptive algorithm of information unmistakable embedding into digital images based on the discrete Fourier transformation // *Multimedia Tools and Applications*. – 2018. – V. 77. – P. 28567–28599.
6. The USC-SIPI Image Database. URL: <https://www.sipi.usc.edu/database/> (дата обращения: 27.04.2021).

О ВОЗМОЖНОСТЯХ ВВЕДЕНИЯ ДОПОЛНИТЕЛЬНЫХ ВЕНТИЛЕЙ С ЦЕЛЬЮ ОБФУСКАЦИИ ЛОГИЧЕСКИХ СХЕМ

Провкин В.А., Матросова А.Ю., Енютина А.И.

Томский государственный университет

prowkan@mail.ru, mau11@yandex.ru, arina.enyutina@mail.ru

Введение

В связи с расширением аутсорсинга в производстве интегральных схем возникает возможность внедрения в них вредоносных подсхем с целью искажения функционирования или извлечения конфиденциальной информации. Это приводит к необходимости блокирования внешнего вмешательства в создаваемые схемы. Способы блокирования (обфускации) могут быть различными. Суть их заключается в том, что информация о функционировании схемы становится доступной только авторизованному пользователю. Один из подходов к обфускации заключается в использовании ключа, применение которого обеспечивает корректное функционирование схемы. Чтобы защитить комбинационную схему с помощью k -разрядного ключа, предлагается включать в схему k дополнительных вентилях [1]. Выбор линии для включения вентиля, тип вентиля существенно влияют на эффективность искажения поведения схемы в отсутствие ключа. На рис. 1а приведен фрагмент логической схемы, а рис. 1б поясняет основную идею искажения поведения схемы. Выход элемента C_1 подключается к дополнительному вентилю XOR, на второй вход которого поступает внешний входной сигнал K_1 однобитового ключа. Схема работает в требуемом режиме, если сигнал на входе K_1 равен 0. В противном случае на выходе элемента XOR CC_1 формируется сигнал, инверсный правильному. Вместо элемента CC_1 типа XOR может быть установлен элемент XNOR. В этом случае на однобитовый правильный ключ K_1 поступает сигнал, равный 1.

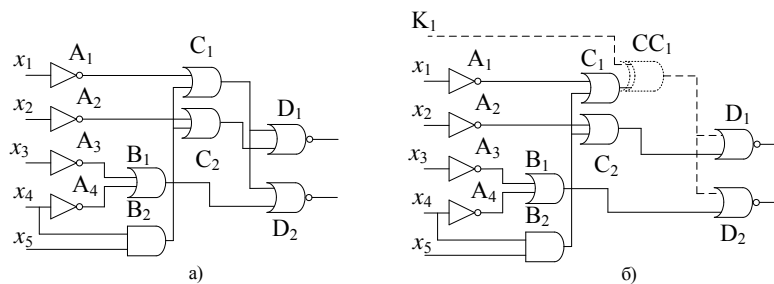


Рис. 1. Фрагмент логической сети:
а) – исходная комбинационная схема, б) – схема с однобитовым ключом

При воздействии входного набора $X = (00000)$ и неправильного ключа $K_1 = 1$ (рис. 1) на выходах схемы D_1, D_2 формируются сигналы (11), в то время как при правильном ключе $K_1 = 0$ на тех же выходах формируются сигналы (00).

Итак, для сокрытия функциональности схемы необходимо добавить в некоторые линии схемы дополнительные элементы и знать правильное кодовое слово, поступающее на входы введенных в схему элементов. Искажение кодового слова выводит схему из области правильного функционирования.

Основная задача, которую необходимо решить в рамках реализации рассматриваемой общей идеи, заключается в определении подходящего множества внутренних линий схемы с целью создания максимальных трудностей для злоумышленника по поиску правильного ключа.

Будем иметь в виду, что каждому внутреннему полюсу v (внутренней линии) комбинационной схемы сопоставляется частичная функция. Множество M_1 единичных наборов этой функции есть множество тестовых наборов для неисправности константа 0 этого полюса, а множество M_0 нулевых наборов этой функции есть множество тестовых наборов для неисправности константа 1 рассматриваемого полюса (линии). Введение в линию элемента XOR (NXOR) позволяет с помощью соответствующего ключа менять выходной сигнал линии на инверсный сигнал. Однако изменения сигнала на линии проявляется на выходах схемы только на тестовых наборах для константных неисправностей этой линии. Чем выше доля определенных наборов частичной функции, тем больше искажений вносится в схему при неправильном ключе дополнительного элемента, вводимого в линию. В работах [1–9] подмножества тестовых наборов для различных линий схемы находятся путем ее моделирования на входных наборах в условиях неисправности каждого из рассматриваемых полюсов (линий) и сравнения результатов моделирования с исправной схемой. Это чрезвычайно затратная процедура. В перечисленных работах предлагаются различные возможности сокращения этих затрат. При таком подходе, в частности, для схем с десятками и сотнями входов возможно нахождение только подмножеств тестовых наборов и, соответственно, получение только приблизительной оценки качества линий, выбираемых для включения в них дополнительных элементов. В данной работе предлагается вычислять частичные функции для линий и тем самым точно оценивать множества входных наборов схемы, на которых отсутствие правильного ключа приводит к искажению ее поведения за счет введения в линию дополнительного элемента.

Для многих логических схем само появление элементов XOR, NXOR может оказаться подозрительным и облегчить поиск относящихся к ключу входов. В работе предлагается для таких схем включать в выбранные линии элементы AND, OR. На рис. 2а приведен пример введения в линию элемента AND и ключа $K_1 = 0$, а на рис. 2б – введение в линию элемента OR и ключа $K_1 = 1$. В случае введения элемента AND и соответствующего ключа в линию вносится неисправность константа 0, а в случае введения элемента OR и соответствующего ключа в линию вносится неисправность константа 1. Следовательно, введение элемента AND приводит к искажению поведения схемы на множестве M_1 частичной функции, а введение элемента OR приводит к искажению поведения схемы на множестве M_0 частичной функции.

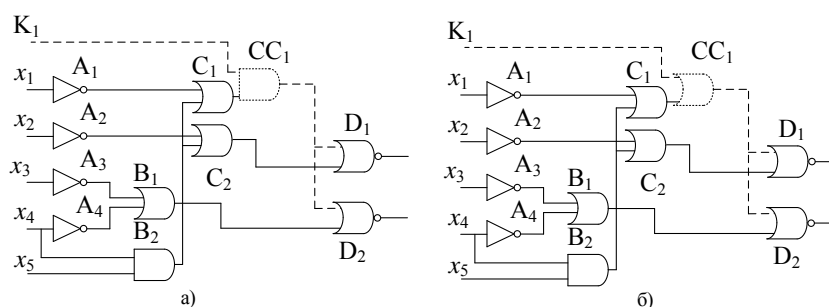


Рис. 2. Шифрование схемы однобитовым ключом:
а) – введение элемента AND, б) – введение элемента OR

1. Постановка задачи

Имеется многовыходная комбинационная схема C , реализующая систему булевых функций $f_i(x_1, x_2, \dots, x_n)$, $i = \overline{1, m}$. Требуется ввести в эту схему дополнительные вентили таким образом, чтобы новая схема имела $n + k$ входов, где n – число входов в исходной схеме, а k – длина ключа для защиты схемы. Полученная схема реализует си-

стему функций $F_i(u_1, u_2, \dots, u_k, x_1, x_2, \dots, x_n)$, $i = \overline{1, m}$. При этом должен существовать такой двоичный набор длины k ($d_1 d_2 \dots d_k$), называемый ключом схемы, что $F_i(u_1 = d_1, u_2 = d_2, \dots, u_k = d_k, x_1, x_2, \dots, x_n) = f_i(x_1, x_2, \dots, x_n)$ для всех $i = \overline{1, m}$, и $F_i(u_1, u_2, \dots, u_k, x_1, x_2, \dots, x_n) \neq f_i(x_1, x_2, \dots, x_n)$ хотя бы для одного i хотя бы на одном наборе входных переменных ($x_1 x_2 \dots x_n$), если хотя бы один из u_i не равен d_j . Желательно, чтобы множество наборов ($x_1 x_2 \dots x_n$), для которых $F_i(u_1, u_2, \dots, u_k, x_1, x_2, \dots, x_n) \neq f_i(x_1, x_2, \dots, x_n)$ при $u_j \neq d_j$ было как можно меньше (но при этом оно, очевидно, не должно быть пустым, т.к. в таком случае защита не имеет смысла). Выбор линий связи для введения вентилях основывается на информации о тестовых наборах для константных неисправностей на этих линиях.

2. Определение множеств тестовых наборов константных неисправностей внутренних полюсов схемы

Рассмотрим многовыходную комбинационную схему C (комбинационную часть последовательностной схемы), обозначим символом v внутреннюю линию, для которой строится частично определённая функция. Выделим подсхему, выходом которой является линия v , а входами – входы схемы C . Эта схема реализует некоторую функцию $\phi(x_1, \dots, x_n)$. Также выделим одно выходных подсхем, в которых входами является линия v и полюсы схемы C , а выходом является один из выходов схемы C . Всего таких подсхем m_v , где m_v – число выходов схемы C , которые связаны с линией v , т.е. существует путь (возможно, не единственный), началом которого является линия v , а концом – один из выходов схемы C . Обозначим функцию, реализуемую такой подсхемой, как $f_i(v, x_1, \dots, x_n)$, $i = \overline{1, m_v}$.

Набор значений входных полюсов схемы является тестом для одиночной константной неисправности внутренней линии одно выходной схемы, если смена значения на этой линии приводит к изменению значения на выходе схемы. Из этого следует, что тестовые наборы для неисправности на линии v являются корнями уравнения $f_i(0, x_1, \dots, x_n) \oplus f_i(1, x_1, \dots, x_n) = 1$. Набор значений входных полюсов для многовыходной схемы является тестовым, если смена значения линии приводит к изменению значения хотя бы на одном из выходов схемы (возможно, сразу на нескольких). Таким образом, тестовые наборы для одиночной константной неисправности линии v многовыходной схемы являются корнями уравнения $\bigvee_{i=1}^{m_v} [f_i(0, x_1, \dots, x_n) \oplus f_i(1, x_1, \dots, x_n)] = 1$.

Для разделения множества всех тестовых наборов на множества тестовых наборов неисправности «константа 0» и неисправности «константа 1» используем функцию $\phi(x_1, \dots, x_n)$. Множества тестовых наборов для константной неисправности линии v определяются следующим образом:

$$M_1 = \left\{ (x_1, \dots, x_n) \in B_2^n : \phi(x_1, \dots, x_n) \wedge \left(\bigvee_{i=1}^{m_v} [f_i(0, x_1, \dots, x_n) \oplus f_i(1, x_1, \dots, x_n)] \right) = 1 \right\},$$

$$M_0 = \left\{ (x_1, \dots, x_n) \in B_2^n : \overline{\phi(x_1, \dots, x_n)} \wedge \left(\bigvee_{i=1}^{m_v} [f_i(0, x_1, \dots, x_n) \oplus f_i(1, x_1, \dots, x_n)] \right) = 1 \right\}.$$

Здесь $B_2 = \{0, 1\}$, $B_2^n = \underbrace{\{0, 1\} \times \{0, 1\} \times \dots \times \{0, 1\}}_{n \text{ раз}}$.

Множества M_1 и M_0 могут быть построены с использованием ROBDD-графов или SAT-задачи [10, 11]. Знание этих множеств для каждой линии связи позволяет опреде-

лить область искажения поведения схемы за счет неправильного значения ключа и связанного с ним дополнительного вентиля, вводимого в линию.

3. Введение дополнительных вентилях в схемы для искажения частичных функций

Теперь опишем процесс обфускации схемы с помощью введения дополнительных вентилях, а также исследуем, как дополнительные вентилях влияют на поведение схемы при неправильных значениях ключа. Рассмотрим следующий фрагмент комбинационной схемы:

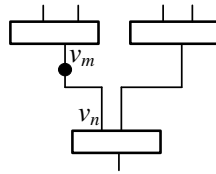


Рис. 3. Схема с выделенной линией связи, в которую необходимо ввести вентиль для защиты схемы

В схеме выделена линия связи, в которую предлагается ввести вентиль И или вентиль ИЛИ с целью обфускации схемы. Здесь v_m является выходом некоторого вентилях и началом линии связи, в которую необходимо ввести дополнительный вентиль; v_n является одним из входов некоторого вентилях и концом линии связи. При нормальном режиме функционирования схемы $v_n = v_m$.

1. Введение элемента AND/подсхемы из элементов AND и NOT:

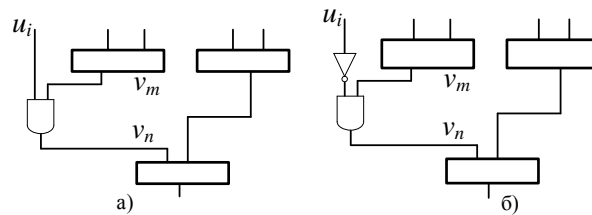


Рис. 4. Защита логической схемы:
а) введение элемента AND, б) введение подсхемы из элементов AND и NOT

При введении в схему элемента AND (рис. 4а) значение на полюсе $v_n = u_i \wedge v_m$. Если $u_i = 1$, то $v_n = u_i \wedge v_m = 1 \wedge v_m = v_m$, т.е. элемент AND обеспечивает прохождение значения на полюсе v_m и схема функционирует корректно. Если же $u_i = 0$, то $v_n = u_i \wedge v_m = 0 \wedge v_m = 0$. Это эквивалентно тому, что на полюсе v_n имеет место константная неисправность типа «константа 0». Следовательно, при $u_i = 0$ поведение схемы искажается на множестве M_1 , т.е. на множестве тестовых наборов для неисправности «константа 0» в этом полюсе. Таким образом, при введении в схему элемента И для корректного функционирования схемы значение i -й компоненты ключа должно быть равно 1.

В дополнении к элементу И на одном из входов этого элемента может быть добавлен вентиль НЕ (рис. 4б)). Значение на полюсе в этом случае $v_n = \overline{u_i} \wedge v_m$. При неправильном значении u_i поведение схемы искажается на множестве M_1 (множество тестовых наборов неисправности «константа 0» на этой линии связи). Но, в отличие от слу-

чая, когда вводится только элемент И, правильное значение ключа инвертируется: $v_n = v_m$, только если $\bar{u}_i = 1$, т.е. $u_i = 0$.

2. Введение элемента OR/подсхемы из элементов OR и NOT:

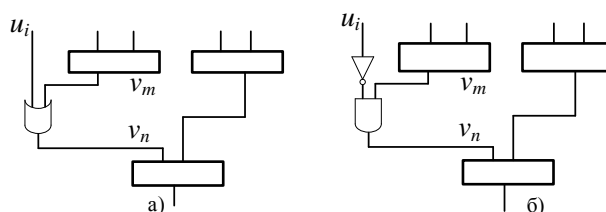


Рис. 5. Защита логической схемы:
а) введение элемента OR, б) введение подсхемы из элементов OR и NOT

При введении в схему элемента ИЛИ (рис. 5а) значение на полюсе $v_n = u_i \vee v_m$. В этом случае значение i -й компоненты ключа должно быть равно 0, чтобы схема функционировала корректно, поскольку $v_n = u_i \vee v_m = 0 \vee v_m = v_m$. При неправильном же значении ключа (т.е. при $u_i = 1$) элемент И обеспечивает прохождение значения на полюсе v_m и схема функционирует корректно. Если же $u_i = 0$, то $v_n = u_i \vee v_m = 1 \vee v_m = 1$, т.е. проявляется неисправность «константа 1», и поведение схемы искажается в области тестовых набором M_0 .

В дополнении к элементу ИЛИ на одном из входов этого элемента может быть добавлен вентиль НЕ (рис. 5б)). Значение на полюсе в этом случае $v_n = \bar{u}_i \vee v_m$. Аналогично, как и в случае 1, область искажения сохраняется, но правильное значение ключа инвертируется.

Использование однотипных искажений с помощью противоположных значений ключа усложняет поиск дополнительных элементов схемы, связанных с использованием ключа.

4. Исследование возможностей обфускации схем на тестовых примерах

Для тестовых схем из набора MCNC были вычислены доли множеств тестовых наборов для константных неисправностей внутренних линий относительно множества всех входных наборов схемы. В результате предварительных исследований получено, что неисправности на большинстве (около 30–40%) линий связи приводят к искажению функционирования схемы примерно на 1–10% всех входных наборов, число линий связи, на которых неисправность приводит к искажению не более чем на 0.1% все входных наборов, обычно составляет не более 20. Однако были вычислены только доли тестовых наборов, соотношение же между множествами тестовых наборов для разных линий (в частности, как много общих наборов между множествами тестов для разных линий) неизвестно и является предметом дальнейших исследований.

Задав длину ключа, можно выбрать для рассматриваемой схемы линии, каждая из которых обеспечивает по возможности меньшее искажение поведения схемы, с тем, чтобы затруднить обнаружение некорректного поведения. Использование предлагаемых в данной работе элементов и подсхем позволяет вносить в схему меньшие искажения по сравнению с элементами XOR, XNOR. Применение последних элементов обеспечивает искажение поведения схемы как на множестве тестовых наборов для неисправности константа 1, так и константа 0.

Заключение

Предложен подход к обфускации комбинационных схем на основе введения более простых элементов и подсхем по сравнению с традиционно используемыми элементами XOR, XNOR. Подход ориентирован как на затруднение обнаружения искажений поведения схемы, так и на последующий поиск ключа. Для поиска линий с целью введения в них дополнительных элементов предлагается использовать вычисление частичных функций вместо традиционно применяемого моделирования. Моделирование, в отличие от предлагаемого подхода, позволяет лишь приближенно оценивать подходящие линии, в особенности для схем с несколькими десятками и более входов.

ЛИТЕРАТУРА

1. Золоторевич Л.А. Аппаратная защита цифровых устройств // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2020. – № 50. – С. 69–78.
2. Yasin M., Rajendran J., Sinanoglu O., Karri R. On Improving the Security of Logic Locking // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2016. – V. 35. – No. 9. – P. 1411–1424.
3. Karousos N., Pexaras K., Karybali I.G., Kalligeros E. Weighted Logic Locking: a New Approach for IC Piracy Protection // IEEE 23rd Int. Symposium on On-Line Testing and Robust System Design (IOLTS). – 2017. – P. 221–226.
4. Chakraborty R.S., Bhunia S. Security against Hardware Trojan through a Novel Application of Design Obfuscation // IEEE/ACM Int. Conference on Computer-Aided Design. – 2009. – P. 113–116.
5. Roy J.A., Koushanfar F., Markov I.L. EPIC: Ending Piracy of Integrated Circuits // IEEE Computer. – 2010. – V. 43. – No. 10. – P. 30–38.
6. Dupuis S., Rouzeyre B., Flottes M.-L., Natale G.D., Ba P.-S. New Testing Procedure for Finding Insertion Sites of Stealthy Hardware Trojans // DATE: Design, Automation and Test in Europe. Grenoble, 2015. – P. 776–781.
7. Xiao K., Forte D., Jin Y., Karri R., Bhunia S., Tehranipoor M. Hardware Trojans: Lessons learned after one decade of research // ACM transactions on design automation of electronic system. – 2016. – V. 22. – No. 1. – P. 1–23.
8. Shakya B., Salmani T.H., Forte D., Bhunia S., Tehranipoor M. Benchmarking of hardware Trojans and maliciously affected circuits // J. Hardw. Syst. Secur. (HaSS). – 2017. – V. 1 (1). – P. 85–102.
9. Сергейчик В.В., Иванюк А.А. Методы лексической обфускации VHDL-описаний // Information Technologies and Systems 2013 (ITS 2013) : Proc. of The Int. Conference. BSUIR. Minsk, 2013. – С. 198–199.
10. Голубева О.И. Алгоритмы получения функций тестопригодности полюса элемента комбинационной схемы // Современные проблемы физико-математических наук : материалы VI Всероссийской научно-практической конференции с международным участием (4–5 декабря 2020 г., г. Орел) : научное электронное издание. Орел: ОГУ им. И.С. Тургенева, 2020. – С. 327–335.
11. Golubeva O. BDD and DNF Based Algorithms for Constructing All Testability Functions of Combinational Circuit // 2021 International Siberian Conference on Control and Communications (SIBCON), 2021, – P. 1–6. – doi: 10.1109/SIBCON50419.2021.9438879.

DOI: 10.17223/978-5-907442-42-9-2021-39

СОЗДАНИЕ ПРОГРАММЫ ДЛЯ АВТОМАТИЗАЦИИ УДАЛЕНИЯ ПУБЛИКАЦИЙ В СООБЩЕСТВЕ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ»

Савенкова М.М., Пахомова Е.Г.

Томский государственный университет
marina1412_11@mail.ru, PakhomovaEG@yandex.ru

Введение

В связи с тем, что в настоящее время социальные сети стали площадками не только для общения и развлечения, но и широко используются в качестве мест продвижения услуг и товаров, приёма коммерческих заказов и осуществления работы с огромной аудиторией, встает вопрос об автоматизации управления публикациями в сообществах. Проблема тем более представляется актуальной, что со временем количество контента в социальных сетях увеличивается, а поисковые системы ужесточают требования к качеству информации на страницах социальных сетей. Их алгоритмы ранжирования информации выдвигают на первые страницы в выдаче только свежий и наиболее востребованный контент.

Несмотря на относительное многообразие существующих программных решений в контексте общей задачи бывает затруднительно выбрать подходящее, т.к. требования к логике приложения для конкретного заказчика могут быть уникальными и оригинальными. Таким образом, в целях управления большими объемами сообщений на стене сообщества, поддержания информации на стене в актуальном состоянии, управления частными объявлениями пользователей сети требуется создание программного обеспечения, предназначенного для реализации конкретных требований.

В данной работе будет рассмотрено создание бота, поддерживающего в актуальном состоянии информацию на стене сообщества социальной сети «ВКонтакте», которая является одной из наиболее популярных социальных сетей в русскоязычном сегменте интернета. Поддержание актуальности будет осуществляться путем своевременного удаления утративших важность публикаций согласно настройкам пользователя. Также дается краткий обзор инструментов для реализации данной задачи, которые предоставляет эта социальная сеть.

1. Описание API «ВКонтакте»

Поскольку «ВКонтакте» представляет собой интернет-сообщество, то бота лучше разрабатывать на web-языке программирования и размещать в Интернете на хостинге. Хостинг (hosting) – это услуга, предоставляемая компаниями хостинг-провайдеринга, которая заключается в размещении информационных ресурсов пользователей на серверах компаний. Хостинги отличаются наличием тех или иных служб, возможностью поддержки различных технологий, языков программирования и скриптов.

В соответствии с рекомендациями команды разработчиков «ВКонтакте», можно использовать два подхода – использовать SDK на Java и PHP [1]. Кроме всего прочего, для реализации бота нужен web-сервер в сети Интернет с публичным IP-адресом. На данный момент самым распространенным видом хостинга является хостинг с поддержкой PHP, поэтому бот и будет реализован на PHP.

«ВКонтакте» предоставляет управление сообществом с помощью API (англ. Application Programming Interface). Для разработки бота потребовалось использовать Callback API и LongPoll API [1]. Кроме этого, понадобилось использовать авторизацию с помощью StandAlone приложения (способ OAuth) [1].

Callback API представляет из себя инструмент, который позволяет отслеживать все новые события, происходящие в сообществе. Для начала работы с Callback API необходимо подключить сервер в настройках сообщества («Управление сообществом» → «Настройки» → «Работа с API») и выбрать типы событий, данные о которых требуется получать (рис. 1а).

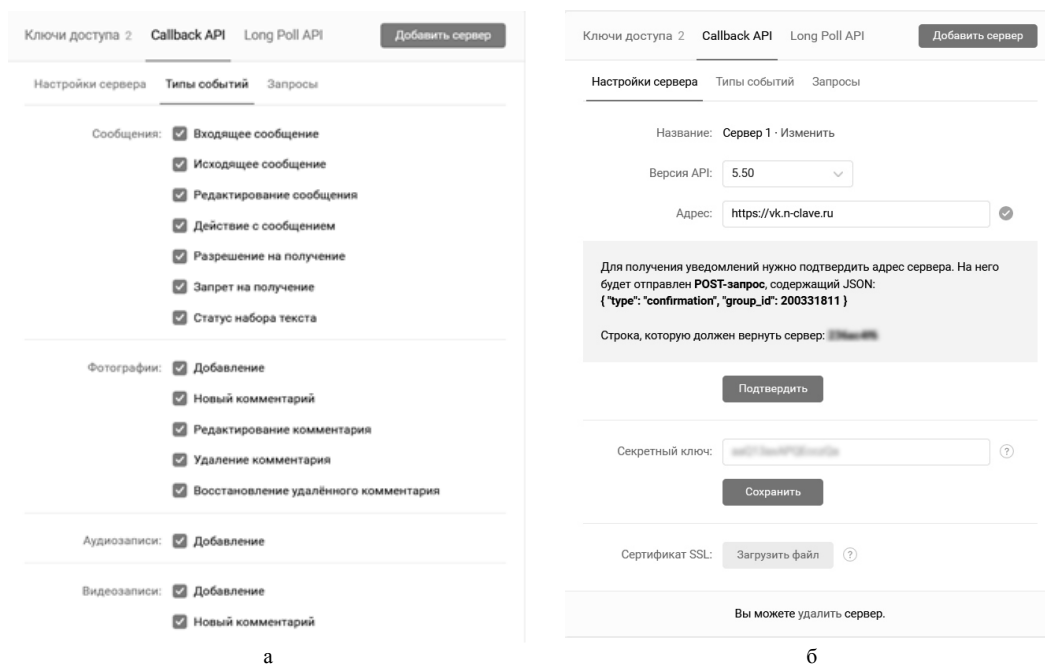


Рис. 1. Настройки API в сообществе

Когда в сообществе происходит заданное событие, «ВКонтакте» отправляет на сервер запрос с данными в формате JSON, заключающем в себе информацию об объекте, вызвавшем событие (например, добавленный комментарий). JSON (англ. JavaScript Object Notation) – текстовый формат обмена данными, основанный на JavaScript. Несмотря на происхождение от JavaScript, формат считается независимым от языка и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON [2].

Long Polling – это технология, которая позволяет получать информацию о новых событиях с помощью «длинных запросов». Сервер получает запрос, но отправляет ответ на него не сразу, а лишь тогда, когда произойдет какое-либо событие (например, поступит новое входящее сообщение), либо истечет заданное время ожидания.

Другими словами, получая запрос, сервер ждет, когда произойдут события, о которых он должен вас уведомить, и, когда они произойдут, LongPoll-сервер отправляет ответ на ваш запрос, содержащий информацию о случившихся событиях. Используя этот подход, можно отображать в своем приложении важные события.

Настройки сервера производятся в специальном разделе на странице сообщества. Также в разделе под названием «Управление» необходимо получить ключи доступа (токены) для взаимодействия с сервером и использования Callback API (рис. 1б).

2. Анализ программного кода

В данной работе реализация бота представлена тремя файлами:

- *index.php* – страница для для взаимодействия с Callback API и сбора статистики.
- *bot.php* – взаимодействие с LongPoll API и просмотр новых сообщений.
- *user.php* – задание параметров управления публикациями.

Также на сервере имеются некоторые вспомогательные файлы, о которых будет упомянуто позже.

2.1. Файл *index.php*

В файле *index.php* содержится скрипт, логика которого построена на основании значения суперглобального массива `$_GET`. `$_GET` – ассоциативный массив перемен-

ных, переданных скрипту через параметры URL, известные также как строка запроса [3]. Если значение не установлено, то происходит проверка массива `$_REQUEST` [3].

В случае, если значения `$_REQUEST` определены, происходит получение и декодирование уведомления. Если это уведомление кода ошибки и требуется подтверждение адреса, то необходимо отправить строку для подтверждения и закончить выполнение скрипта. Если же уведомление содержит информацию о новом сообщении, при помощи методов API «ВКонтакте» мы можем отправить ответное сообщение пользователю, проверив таким образом работу Callback API.

И во всех случаях без исключения возвращаем серверу Callback API ответ «ok», как того требуют правила работы с API «ВКонтакте» [1]. После этого делаем запись входящих сообщений в обычный текстовый файл, также хранящийся на сервере. Название файла задается в виде `YYYYMMDD.txt` (рис. 2), что в дальнейшем будет использовано в сортировке, необходимой для удобного размещения информации о событиях на странице.

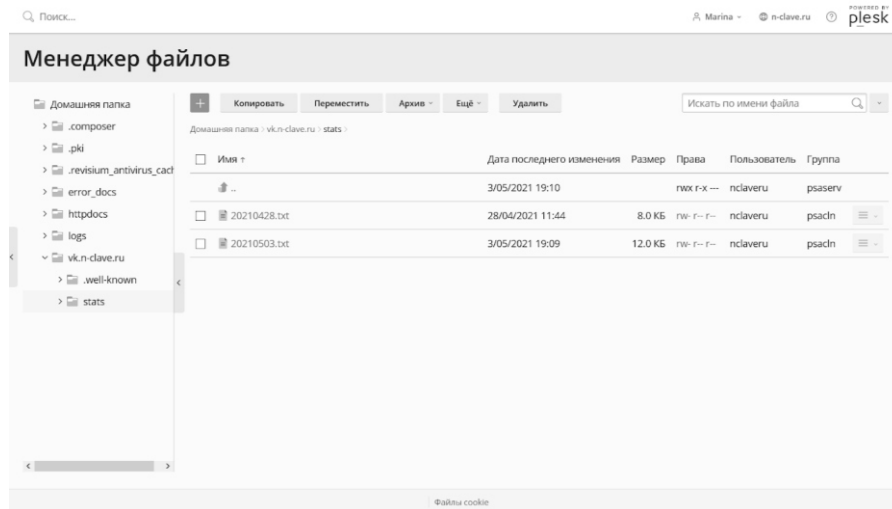


Рис. 2. Файлы входящих сообщений

Если задано значение «view» массива `$_GET`, то на странице идет отображение прошедших событий.

Сначала получаем список файлов и каталогов, расположенных в папке «stats» и сортируем его в обратном порядке так, чтобы последнее произошедшее событие было первым в списке. Далее в цикле осуществляется перебор имеющихся файлов со списком событий, создание текстовых массивов на основе их содержимого и последующее инвертирование (т.к. удобнее для просмотра располагать новые события выше старых).

Также на сервере хранится файл `events.json`, представляющий из себя перечисление типов событий и их значений на русском языке. Благодаря этому типы событий на странице отображаются на русском языке.

2.2. Файл `bot.php`

Логика скрипта, содержащегося в файле `bot.php`, также разбивается на несколько различных ситуаций в зависимости от параметров строки запроса.

Перед подключением к LongPoll-серверу необходимо получить данные сессии («server», «key», «ts») методом `messages.getLongPollServer`. Чтобы начать получать ответы от сервера, необходимо получить три обязательных параметра, необходимых для работы LongPoll: «server», «key» и «ts»:

- «key» – секретный ключ сессии,
- «server» – адрес сервера,

- «ts» – номер последнего события, начиная с которого нужно получать данные.

Полученные параметры хранятся в файле *data.json* и периодически обновляются, т.к. ключ сессии и адрес сервера меняются. Также для работы с LongPoll понадобились ключ доступа управления сообществом и id группы.

Итак, рассматривается несколько случаев.

1. Получен новый секретный ключ сессии («newkey»).

а. Перезаписываем параметр «key» в файле *data.json*. Сообщаем, что ключ доступа получен, и отображаем на странице гиперссылку на адрес <https://vk.n-clave.ru/bot.php> с текстом «Проверка новых событий».

б. Если не удалось сформировать файл с ключом доступа, сообщаем об этом и предлагаем попробовать ещё раз (гиперссылка <https://vk.n-clave.ru/bot.php?newkey> с текстом «Попробовать ещё раз»), при этом проверка новых событий по-прежнему остается доступной.

2. В URL-строке задан параметр «settings».

В этом случае ответ, полученный с сервера, отображается в виде ассоциативного массива на странице, что позволяет посмотреть текущие настройки LongPoll для группы.

3. Получен номер последнего события («ts»).

а. Проверяем, имеется ли на нашем сервере файл с ключом доступа. Если нет, выводим сообщение, что файл не найден, и прерываем выполнение скрипта.

б. Перезаписываем номер последнего события в файл. В результате отображаем сообщение об успешности или неудачи установки номера.

4. В остальных случаях:

а. В ответ на запрос сервер может вернуть одну из ошибок, в зависимости от значения которой отображается соответствующая информация и гиперссылка:

- «failed»:1 – история событий устарела или была частично утеряна. Бот может получать события далее, используя новое значение «ts» из ответа.
- «failed»:2 – истекло время действия ключа. Нужно получить новый ключ доступа, поэтому на странице отображается гиперссылка <https://vk.n-clave.ru/bot.php?newkey> с текстом «Получить».
- «failed»:3 – информация о пользователе утрачена, нужно запросить новые ключ доступа и точку отсчёта событий.

б. Если ошибок не было, то обновляем номер последнего события. Появляется текстовое поле для установки номера последнего события, который можно при желании изменить (рис. 3). В случае, когда номер обновить не удалось, выводим соответствующее сообщение.

в. Если в ответе на запрос задан параметр «updates», но их количество равно нулю, то сообщаем, что новых событий нет, иначе отображаем гиперссылку <https://vk.n-clave.ru/bot.php> с текстом «Проверка новых событий».

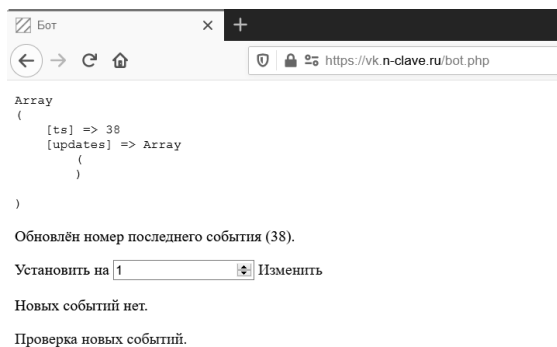


Рис. 3. Установка номера события

Новые события отображаются в качестве ассоциативного массива (рис. 4).

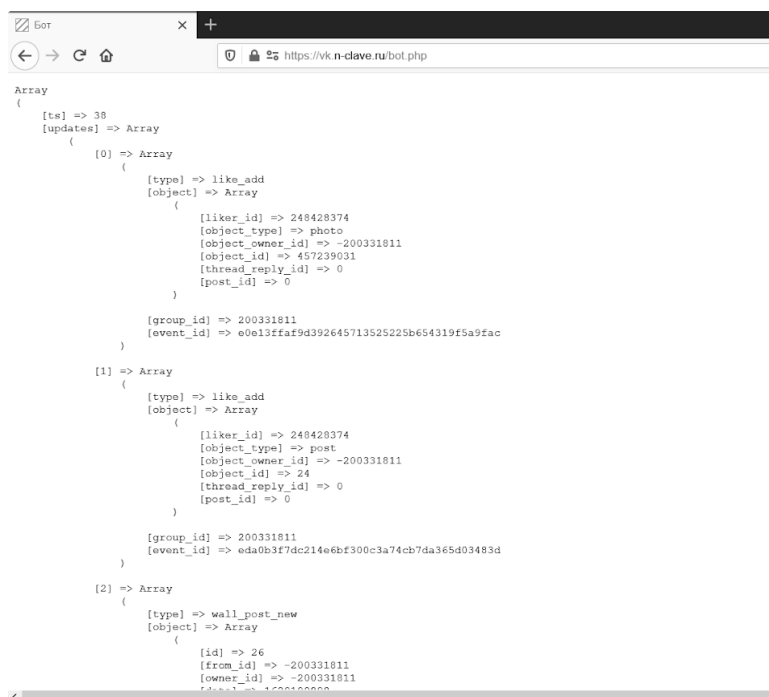


Рис. 4. Отображение новых событий

2.3. Файл *user.php*

В файле *user.php* производятся действия по удалению публикаций на стене сообщества и происходит задание пользовательских настроек, установка которых и происходит на web-странице <https://vk.n-clave.ru/user.php> (рис. 5).

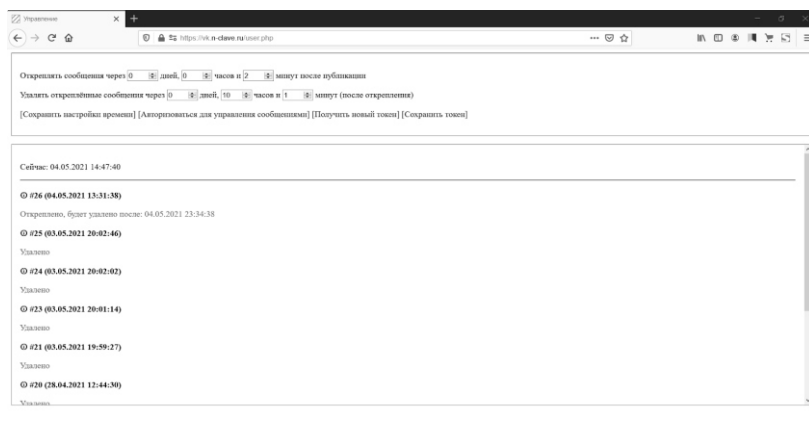


Рис. 5. Задание пользовательских настроек

Алгоритм состоит в следующем. Файлы, хранящиеся на сервере в папке «stats», содержат информацию о прошедших событиях в формате JSON, в том числе сведения о времени публикации. В цикле осуществляется перебор файлов, и время прошедшего события, для которого типом события является «wall_post_new» (новая публикация на стене сообщества), сравнивается с текущим временем. Если текущая дата больше или равна дате, когда событие должно быть откреплено (или удалено), то запись открепляется (или удаляется) со стены сообщества посредством методов API «ВКонтакте» в соответствии с выбранными настройками.

Обработка пользовательских настроек включает в себя рассмотрение нескольких ситуаций. На сервере хранятся файлы с названиями *last.json*, *settings.json* и *app.json*. Файл *last.json* хранит id прошедших событий и их статус – откреплено или удалено, *settings.json* – интервал времени в секундах, через которое должно произойти открепление или удаление публикации, *app.json* – токены и ключи для управления публикациями.

Итак, возможны следующие варианты в зависимости от параметров URL-строки:

1. Если в суперглобальном массиве `$_POST` (ассоциативный массив переменных, переданных скрипту через параметры URL) имеется значение токена, то файлы *last.json*, *app.json* удаляются, и выводится сообщение о том, что получен новый токен для управления сообществом. Формируется массив, содержащий по умолчанию текущий токен, если он был не заменен на новый, id группы и ключи безопасности. Созданный массив в виде строки помещается в файл *app.json*.
2. Если массив `$_POST` содержит параметры «unpin» и «delete», то в файле *settings.json* происходит перезапись текущих настроек.

Для того, чтобы иметь возможность управлять событиями, в данном случае, публикациями, в сообществе было создано StandAlone-приложение, которое позволяет получить специальный токен, после чего методы API «ВКонтакте» становятся доступными для использования в боте.

Существует два способа запуска бота. Первый – запуск вручную, открыв страницу в браузере, или посредством таймера, который бы запускал на выполнение скрипт. Второй осуществляется путем создания задания в планировщике сервера, которое в назначенное время будет запускать скрипт <https://vk.n-clave.ru/user.php> (рис. 6).

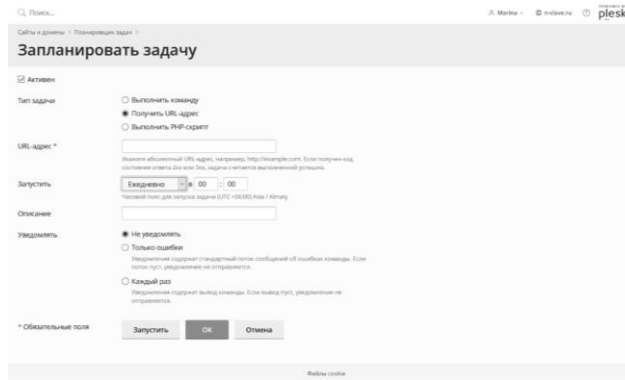


Рис. 6. Планировщик задач на сервере

Заключение

Таким образом, поставленная задача по созданию программы, которая поддерживает в актуальном состоянии информацию на стене сообщества социальной сети «ВКонтакте» путем удаления публикаций по таймингу, была выполнена. Благодаря предложенному инструментарию API «ВКонтакте», рассмотренному в работе, предоставляется возможность автоматизации взаимодействия со страницей сообщества, что значительно облегчает создание приложений, предназначенных для управления событиями в рамках данной социальной сети.

В свою очередь использование бота, подобного реализованному, значительно облегчает администрирование сообществ, т.к. избавляет пользователя от необходимости контроля расписания и удаления публикаций вручную.

ЛИТЕРАТУРА

1. Документация API «ВКонтакте» [Электронный ресурс] // VK.COM: социальная сеть «ВКонтакте». URL: <https://vk.com/dev/manuals>
2. Работа с данными в формате JSON [Электронный ресурс] // INTUIT.RU: Национальный Открытый Университет «ИНТУИТ». URL: <https://intuit.ru/studies/courses/19522/1309/lecture/28910>
3. Руководство по PHP [Электронный ресурс] // PHP.NET: сообщество PHP-программистов. URL: <https://www.php.net/manual/ru/index.php>

DOI: 10.17223/978-5-907442-42-9-2021-40

ПРИМЕНЕНИЕ 2-SAT ЗАДАЧИ ДЛЯ ОБНАРУЖЕНИЯ УТЕЧКИ РЕСУРСОВ В СПИСОЧНЫХ СТРУКТУРАХ

Серебрянникова К.А., Андреева В.В.

Томский государственный университет
krisert@mail.ru, avv.21@mail.ru

Введение

Тестирование программного обеспечения (далее – ПО) является неотъемлемой частью жизненного цикла разработки ПО. Верификация ПО является одним из важнейших, и, в то же время, трудозатратных этапов разработки. Ошибки, допущенные на стадии верификации ПО, могут привести к существенным потерям в будущем.

Разработка программного обеспечения на языках Си/C++ дает широкие возможности для управления памятью в процессе работы программы. В частности, хранение и обработка данных в списочных структурах позволяет динамически размещать элементы в памяти по мере их необходимости. Здесь самое главное – не допустить утечку ресурсов, т.е. доступ к данным. Потеря доступа приводит к тому, что становится невозможным освобождение памяти из-под таких элементов, что, в свою очередь, приводит

к исчерпанию динамической памяти и аварийному останова программы. Обнаружение таких ситуаций невозможно на уровне компиляции программы, т.к. они являются логическим дефектом и дают о себе знать исключительно в момент работы. Обнаружение такого вида дефектов программы возможно с помощью специальных средств, таких как статические анализаторы. Статические анализаторы позволяют обнаруживать обширное количество ошибок в коде без непосредственного запуска программы, но, к сожалению, статические анализаторы не способны обнаруживать утечки ресурсов, порождаемых неверными действиями пользователей при обработке данных в списочных структурах.

Данная работа посвящена разработке подхода статического анализа, позволяющего обнаруживать такие дефекты. Прилагаемый подход статического анализа кода анализирует именно тот участок, в котором непосредственно выполняются операции над списочными структурами. Сами операции моделируются с помощью рассматриваемой модели 2-SAT, которая представляет логическую формулу вида КНФ, состоящей из дизъюнктов только от двух переменных. С помощью дизъюнктов описываются отношения между элементами динамических структур. Результатом анализа является проверка такой формулы на выполнимость с помощью SAT-решателя. Невыполнимая формула демонстрирует факт отсутствия утечек, т.к. построена модель, в которой нет разрывов между связями элементов.

1. Построение среза кода в виде синтаксического дерева

С целью реализации статического анализа исходный код программы представляется в виде абстрактного синтаксического дерева, или, другими словами, преобразуется в структуру, удобную для дальнейшего анализа (от англ. AST – abstract syntax tree). AST в информатике – конечное помеченное ориентированное дерево, в котором внутренние вершины сопоставлены (помечены) с операторами языка программирования, а листья – с соответствующими операндами [1]. Иными словами, это исходный код программы, представленный в виде структуры, удобной для дальнейшего анализа. В листингах 1 и 2 продемонстрированы код метода **PushBack**, который добавляет элемент в конец списка, и код в виде синтаксического дерева.

<pre> void PushBack(Node* head, int value) { Node*last = GetLast(head); Node* newNode (Node*)malloc(sizeof(Node)); newNode->value = value; if (last == NULL) { newNode->next = newNode; } else{ last->next = newNode; } } </pre>	<pre> = </pre>	<pre> PushBack(method):in "head"<Node*>, in "value"<int> ["last"<Node*>: { value: getLast("head") next : getLast("head") }; if ("last" == NULL) { "head"<Node*> { next:newNode<Node*> }; } else { "last" <Node*>: { next:newNode<Node*> }; }] </pre>
--	----------------	--

Листинг 1. Исходный код метода Push Back

Листинг 2. Код метода Push Back, представленный в виде синтаксического дерева

В качестве метода построения среза кода с последующим представлением его в виде синтаксического анализа воспользуемся подходом, предложенным в [2]. Данный подход позволяет выделить подмножество кода, необходимое для анализа, в частности, именно те участки, которые относятся обработке списочной структуры.

2. Построение графов списочных структур

Метод анализа списочных структур базируется на последовательном анализе синтаксического дерева и формировании для каждого блока кода, связанного с выполнением некоторых действий над элементами структуры, особого графа G – графа списочных структур. Граф G состоит из множества вершин, являющихся элементами списочной структуры, ребра соответствуют связям между элементами. Здесь необходимо заметить, что при работе со списочными структурами элементы создаются в динамической памяти (куче), причем для организации корректной работы обязательно нужно сохранять адреса этих объектов. В случае, например, таких структур, как линейные списки, достаточно хранить адрес на первый элемент. Следовательно, мы можем выделить два типа вершин: вершины, соответствующие переменным, хранящим адреса объектов и объекты, непосредственно хранящиеся в динамической памяти.

Пусть $V = \{V_i\}$ – множество переменных, хранящих адреса элементов динамической структуры, $N = \{N_i\}$ – множество адресов объектов, хранящихся в динамической памяти.

Построение графа выполняется в результате анализа операций, проводимых над списками. В частности, создание элементов списочной структуры, встраивание их в структуру порождает формирование узлов графа и проведение ребер между вершинами, соответствующими связям между элементами в списочной структуре. Рассмотрим построение графа на конкретном примере.

На рис. 1 представлен граф после создания «головного» указателя типа $Node^* head$, где V_1 – переменная, хранящая адреса элементов динамической структуры. Т.к. в списке отсутствуют элементы, то «головной» указатель ссылается на NULL.

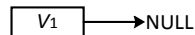


Рис. 1. Граф после создания «головного» элемента

Далее на рис. 2 и 3 продемонстрированы графы после добавления элементов с помощью метода *PushBack* (Листинг 1). Рассмотрим, каким образом элементы связываются между собой в списочную структуру данных. Связь элементов происходит с помощью указателя *next* (поле структуры *Node*). При добавлении элемента в «хвостовую» часть списка могут возникнуть 2 случая:

1. Изначально список пустой. В данном случае для корректной работы программы необходимо переопределить указатель *head* на новый элемент, а новый элемент направить в NULL.

2. Изначально список непустой. Если в списочной структуре уже содержатся элементы, то необходимо перенаправить предыдущий последний элемент на новый, а новый направить в NULL.

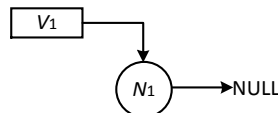


Рис. 2. Граф после добавления первого элемента

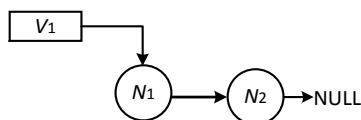


Рис. 3. Граф после добавления второго элемента

3. Формирование логической формулы для графов списочных структур в виде КНФ, проверка на выполнимость, формулировка выводов

Рассмотрим логическую модель, описанную в [3], которую мы можем применить к проблеме утечки памяти в списочных структурах данных. Для поиска утечки необходимо замкнуть последний элемент с «головным». Таким образом, проблема поиска утечек памяти сводится к проверке связности графа. Связный граф – граф, между любой парой вершин которого существует как минимум один путь [4].

Как показано в [3], ориентированный граф можно представить с помощью логической формулы КНФ, состоящей из дизъюнктов от двух переменных. Эти переменные советуют вершинами графа. Переменная, из которой выходит ребро, входит в дизъюнкт без знака инверсии, переменная в которую входит ребро, входит в дизъюнкт со знаком инверсии. Воспользуемся выводами из [3]: граф является связным тогда и только тогда, когда задача 2-SAT имеет ровно два решения: первое – это то, где каждая логическая переменная истинна (что называется присвоением белого), второе – то, где каждая логическая переменная ложна (что называется присвоением черного). Такую задачу авторы называют «черно-белой» SAT-задачей.

Следовательно, при добавлении отрицания «белого» и «черного» решений к сгенерированной задаче SAT она становится невыполнимой.

Замечание 1. Если вершина x_1 имеет ребра к вершинам x_2 и x_3 , то логическая модель будет следующей: $(x_1 \rightarrow x_2) \wedge (x_1 \rightarrow x_3)$.

Замечание 2. Формула $(x_1 \rightarrow x_2) \wedge (x_1 \rightarrow x_3)$ преобразовывается в задачу 2-SAT с помощью следующего правила: $x \rightarrow y = x \vee \bar{y}$.

Для поиска решения используется SAT-решатель CryptoMiniSat [5]. Применим данную модель к задаче поиска утечек памяти в линейном списке.

Т.к. модель предполагает проверку связности графа, необходимо замкнуть следующим образом: последний элемент списка, направленный в NULL, замыкаем с «головным». Получаем граф, изображенный на рис. 6.

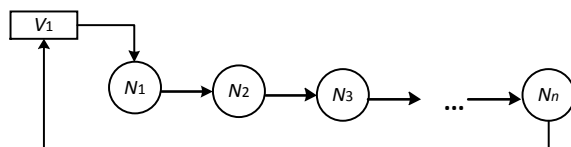


Рис. 6. Граф линейного списка: V_1 – указатель head на первый элемент списка, $N_1 - N_n$ – узлы (элементы) списка

Т.к. ребра графа отражают связи между элементами списка, при отсутствии хотя бы одного ребра между вершинами графа, он становится несвязным.

Перепишем граф в виде формул: $V_1 \rightarrow N_1, N_1 \rightarrow N_2, N_2 \rightarrow N_3, \dots, N_{n-1} \rightarrow N_n, N_n \rightarrow V_1$. Составим КНФ для SAT-решателя, используя замечание 2:

$$(\bar{V}_1 \vee N_1) \wedge (\bar{N}_1 \vee N_2) \wedge (\bar{N}_2 \vee N_3) \wedge \dots \wedge (\bar{N}_{n-1} \vee N_n) \wedge (\bar{N}_n \vee V_1). \quad (1)$$

Согласно [2], если граф связный, т.е. отсутствуют разрывы между элементами списка, формула (1) имеет 2 решения («черное» и «белое»). Добавим в формулу (1) «белое» и «черное» решения в виде двух последних конъюнктов:

$$(\bar{V}_1 \vee N_1) \wedge (\bar{N}_1 \vee N_2) \wedge (\bar{N}_2 \vee N_3) \wedge \dots \wedge (\bar{N}_{n-1} \vee N_n) \wedge (\bar{N}_n \vee V_1) \wedge \\ \wedge (V_1 \vee N_1 \vee N_2 \vee N_3 \vee \dots \vee N_n) \wedge (\bar{V}_1 \vee \bar{N}_1 \vee \bar{N}_2 \vee \bar{N}_3 \vee \dots \vee \bar{N}_n).$$

Таким образом, опираясь на описанную выше модель, сделаем выводы о наличии/отсутствии утечек памяти в программном коде:

– Если полученная формула невыполнима, то граф связный. Из этого можно сделать вывод, что разрыва связей между элементами нет и соответственно утечки памяти не обнаружены.

Если формула выполнима, то граф несвязный, т.е. утечка памяти обнаружена.

Сформулируем алгоритм поиска утечек памяти в списочных структурах данных. Алгоритм состоит из следующих этапов

1. Построение среза кода в виде синтаксического дерева.
2. Построение графов списочных структур.
3. Формирование логической формулы для графов списочных структур в виде КНФ.
4. Анализатор на основе SAT-решателя.

Предложенный алгоритм программно реализован. Тестирование выполнялось на специально подготовленных примерах, включающих в себя многочисленные варианты утечек. В частности, рассматривались все операции, приводящие к утечкам ресурсов. К таким операциям относятся, сдвиг указателя, переопределение указателя и неверная установка связей между элементами.

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. В результате, был разработан алгоритм, который упрощает поиск утечек памяти в программах, использующих динамические структуры списочного типа.

ЛИТЕРАТУРА

1. Абстрактное синтаксическое дерево [Электронный ресурс] URL: Абстрактное синтаксическое дерево – Википедия (wikipedia.org); (дата обращения 01.05.2021).
2. *Takahashi M., Mizukoshi N., Tsuda K.* Efficient Program Verification Using Binary Trees and Program Slicing // In: Negoita M.G., Howlett R.J., Jain L.C. (eds) Knowledge-Based Intelligent Information and Engineering Systems. KES 2004. – Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. – 2004. – V. 3213. – https://doi.org/10.1007/978-3-540-30132-5_69
3. *Biro C., Kasper G.* Equivalence of strongly connected graphs and black-and-white 2-SAT problems // *Miskolc Mathematical Notes.* – 2018.
4. Связный граф [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%8F%D0%B7%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84; (дата обращения 20.04.2021).
5. CryptoMiniSAT [Электронный ресурс] URL: https://msoos.github.io/cryptominisat_web/#/; (дата обращения 03.05.2021).

МОДЕЛИРОВАНИЕ КИНЕТИКИ ТОЧЕЧНЫХ, ЛИНЕЙНЫХ И ПОВЕРХНОСТНЫХ ДЕФЕКТОВ И ПОРОГОВ НА ДВИЖУЩЕЙСЯ ВИНТОВОЙ И КРАЕВОЙ ДИСЛОКАЦИИ В ГЦК МЕТАЛЛАХ

Трифонов С.А.¹, Самохина С.И.¹, Черепанов Д.Н.²

¹Томский государственный университет,

²Томский государственный архитектурно-строительный университет
stasta956@gmail.com

Введение

Изучение поведения металлов, а именно – явления пластической деформации, 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 стационарной плотности порогов

В [1,2] были подробно описаны математическая модель кинетики точечных, линейных и поверхностных дефектов, а также математическая модель кинетики порогов на движущейся винтовой и краевой дислокации. Модели основаны на взаимодействии двух противоположных процессах: накопление деформационных дефектов и их исчезновение.

Исследовав рассмотренные ранее модели по отдельности, нами была установлена связь между ними. Эти две модели связывает такой параметр как стационарная плотность порогов c_j .

В модели генерации и аннигиляции точечных, линейных и поверхностных дефектов в слагаемых, описывающих интенсивности генераций точечных дефектов (G_i, G_{1v}, G_{2v}) учитывается стационарная плотность порогов, а точнее – интенсивность генерации точечных дефектов прямо пропорциональна стационарной плотности порогов. Следовательно, данный параметр должен иметь по возможности наивысшую степень точности, для того, чтобы получить точные результаты.

Как известно, во время движения дислокации ее фрагменты могут пересечься с фрагментом другой дислокации, что приведет к возникновению порога, т.е. часть дислокации переползет в соседнюю атомную плоскость. Это привело к исследованиям различных механизмов возникновения, аннигиляции и движения порогов. В данной работе нас интересует модель движения порогов и, в частности, выражение для стационарной плотности порогов. Стационарной плотностью порогов называют количество порогов в единице длины дислокации.

Первое выражение для стационарной плотности порогов на движущейся дислокации было получено из решения системы дифференциальных уравнений Хирша – Мотта [3]:

$$\begin{cases} \frac{dc_j^v}{dx} = 0.5 \xi \rho - k c_j^i c_j^v, \\ \frac{dc_j^i}{dx} = 0.5 \xi \rho - k c_j^i c_j^v. \end{cases} \quad (1)$$

Здесь x – пробег дислокации, ξ – доля дислокаций “леса”, $k \approx 1$ (что не обосновано), (ρ – плотность дислокаций. Функции $c_j^i = c_j^v = (\xi\rho/2k)^{1/2} \text{th}(k\xi\rho/2)^{1/2} x$ будут решением этой системы уравнений при условии $c_j^i = c_j^v = 0$ в первоначальном положении дислокации. В итоге, путем преобразований была получена формула для стационарной линейной плотности порогов: $c_j = (\xi\rho/2)^{1/2}$. Однако в данной формуле не учитываются ни скорость движения порогов вдоль движения дислокации, ни сила взаимодействия порогов между собой.

Далее была выведена система уравнений, в которой учтен механизм слияния порогов, т.е. система (1) преобразовалась в следующую систему:

$$\begin{cases} \frac{dc_j^v}{dx} = 0.5\beta_j \xi \rho - k c_j^i c_j^v - k (c_j^v)^2, \\ \frac{dc_j^i}{dx} = 0.5\beta_j \xi \rho - k c_j^i c_j^v - k (c_j^i)^2. \end{cases}$$

Поскольку рассматриваются дислокации, генерируемые источниками Франка – Рида, то дислокационные сегменты всегда рождаются парами с различными знаками – противоположными сегментами петли, следовательно, на ранних стадиях деформирования, когда дислокации некомпланарных систем скольжения распределены однородно по знаку, вакансионные и межузельные пороги на винтовых сегментах появляются приблизительно в равном количестве, поэтому принимая $c_j^i = c_j^v$ и суммируя $c_j = c_j^i + c_j^v$, нетрудно получить следующее уравнение:

$$\dot{c}_j = (\beta_j \xi \rho - k (c_j)^2) v.$$

Полученное уравнение описывает суммарную плотность порогов на движущейся дислокации со скоростью $v = \dot{x} = dx/dt$. Однако и данное уравнение не является удовлетворительным, поскольку здесь присутствует коэффициент k . Данный коэффициент не расшифрован и не является постоянным на разных этапах деформирования. Вместо этого коэффициента должно быть обоснованное выражение, которое будет зависеть от скорости движения порогов и учитывать неоднородность распределения дислокаций.

Чтобы исключить эти неясности, было проведено исследование кинетики порогов на сдвигобразующих дислокациях. В результате для плотности порогов на движущейся винтовой дислокации было получено следующее дифференциальное уравнение:

$$\dot{c}_j = G_j(\rho) - A_j(c_j) = \beta_j \xi \rho v - (c_j)^2 v_j.$$

Здесь $\beta_j \approx (1 - \beta_r) 0.5 \approx 0.43$ – доля порогаобразующих дислокаций, v – скорость движения винтовой дислокации, v_j – скорость движения порогов вдоль дислокации. Найдя решение данного уравнения, получили новое выражение для стационарной плотности порогов: $c_j = \sqrt{\beta_j \xi \rho v / v_j}$. В данном уравнении уже учтена скорость движения порогов, однако вновь не учитывается неоднородность распределения дислокаций разного знака. Следующим шагом в развитии данного исследования стало описание математических моделей движения винтовых и краевых сегментов дислокации.

Доказано, что механизмы генерации дефектов деформации существенно зависят не только от общей площади, заматаемой дислокациями, но и от характера движения краевых и винтовых сегментов дислокаций, которые движутся в разных направлениях и на разное расстояние. Путем обоснованных предположений была получена модель движения винтовых сегментов дислокации [4], которая имеет вид:

$$\dot{e}_k^s = v_s \left(\tau_{\text{eff}} b - 0.25 G b^2 c_j - B_v v_s - (\mu + e_k^s) R^{-1} \right), \quad (2)$$

$$\dot{R} = c_t \sqrt{1 - (1 + e_k^s e_{\text{уп}}^{-1})^{-2}}, \quad (3)$$

$$\dot{c}_j = \beta_j \xi \rho v_s - (w_j c_j)^2 v_j, \quad (4)$$

где $\tau_{\text{eff}} = \tau - \tau_R \approx 0.4 G b \rho^{1/2}$, $e_{\text{уп}} \approx \mu \approx 0.5 G b^2$ – собственная энергия единицы длины сегмента, c_t – скорость звука, ξ – доля дислокаций леса, $\beta_j = 0.5(1 - \beta_r)$ – доля порогообразующих дислокаций леса, β_r – доля реагирующих дислокаций; v_j – скорость движения порогов при вдоль линии дислокации.

Уравнения (2)–(4) описывают кинетику порогов на движущихся сегментах винтовой ориентации. Структура дислокаций леса в уравнениях модели описывается скалярной плотностью дислокаций ρ и параметром w_j , который учитывает долю дислокаций одного знака, содержащихся в дислокационном лесе. Этот параметр в рассматриваемой модели варьируется.

Уравнение (4) является достаточно точным для расчета стационарной плотности порогов, т.к. в нем учитывается неоднородность дислокаций разного знака и скорость движения порогов. Однако для того, чтобы воспользоваться данным уравнением, необходимо знать, чему равна скорость движения порогов v_j . В ранних работах эта скорость либо принималась равной скорости звука, либо равной скорости движения винтовых сегментов, либо делались какие-либо другие предположения.

Требуется проведение дальнейшего исследования в данном направлении и разработка математической модели для вычисления скорости движения порогов в зависимости от времени. Получение этих данных позволит получить обоснованное значение или выражение для средней скорости движения порогов и сделает дальнейшие исследования по движению винтовых сегментов дислокационной петли максимально точными.

2. Математическая модель скорости движения порогов

Порог на винтовой дислокации можно рассматривать как краевой сегмент длиной приблизительно в одно межатомное расстояние, поэтому можно модифицировать первое уравнение в модели движения краевых сегментов

$$\dot{e}_k^e = v_e \left(\tau_{\text{eff}} b - B_v v_e - (\mu + e_k^e) R^{-1} \right), \quad (5)$$

$$\dot{R} = v_e = c_t \sqrt{1 - (1 + e_k^e e_{\text{уп}}^{-1})^{-2}} \quad (6)$$

таким образом, чтобы движение порога произошло в направлении равнодействующей силы взаимодействия порогов $\mu(4\pi(1 - \nu)x)^{-1}$, где $2x$ уменьшается от среднего расстояния между порогами c_j^{-1} до b , силы трения, сопротивляющейся движению порога $\tau_j b$, и проекций линейных натяжений $\mu \cos \varphi_1$ и $\mu \cos \varphi_2$ винтовых сегментов по обе стороны от порога, где φ_1 и φ_2 – углы между направлениями линейных натяжений и линией дислокации.

Поскольку проекции линейных натяжений требуется выразить через $2x$, то воспользуемся формулой $\tau = \mu x^{-1} b^{-1} \sin \varphi_1 = \mu (c_j^{-1} - x)^{-1} b^{-1} \sin \varphi_2$. Тогда равнодействующая проекций линейных натяжений примет вид

$$\mu(\cos \varphi_1 - \cos \varphi_2) = \mu \left(\sqrt{1 - (\tau x b \mu^{-1})^2} - \sqrt{1 - (\tau(c_j^{-1} - x) b \mu^{-1})^2} \right).$$

Следует отметить, что если все пороги находятся на одном и том же расстоянии друг от друга и строго чередуются по знаку, то вся система находится в равновесии, однако это равновесие неустойчиво. Если в результате термических флуктуаций расстояние между некоторыми порогами уменьшается, то система, состоящая только из этих двух порогов, выходит из равновесия, после чего начинается движение порогов навстречу друг другу.

Кроме того, при малом расстоянии между порогами скорость их движения может и близко не приближаться к скорости звука, поскольку пороги аннигилируют быстрее, чем они разгонятся до высоких скоростей. В этом случае, вместо уравнения (5) можно использовать для кинетической энергии движения порога выражение $0.5m v_j^2$, где $m \approx E_0 c_e^{-2} \approx 0.1 \mu c_i^{-2} (1 - \nu)^{-1}$ – масса движущего порога. Однако тогда требуется найти наибольшее расстояние между порогами, при котором это допущение правомерно.

Порог на винтовой дислокации можно рассматривать как краевой сегмент длиной приблизительно в одно межатомное расстояние, поэтому можно модифицировать равенство (5) следующим образом:

$$\dot{e}_k^e = v_e \left(\mu \left((4\pi(1-\nu)x)^{-1} + \sqrt{1 - (\tau x b \mu^{-1})^2} - \sqrt{1 - (\tau(c_j^{-1} - x) b \mu^{-1})^2} \right) - \tau_f b - B_0 v_e \right). \quad (7)$$

Таким образом, мы получаем модель движения порогов вдоль линии дислокации в виде двух дифференциальных уравнений: уравнение (6) в виде

$\dot{x} = v_e = c_i \sqrt{1 - (1 + e_k^e e_{\text{упр}}^{-1})^{-2}}$ и уравнение (7) с начальными условиями $v_e(0) = 0$ и $x(0) = 0,5c_j^{-1}$.

В результате получили следующую систему обыкновенных дифференциальных уравнений:

$$\begin{aligned} \dot{e}_k^e &= v_e \left(\mu \left((4\pi(1-\nu)x)^{-1} + \sqrt{1 - (\tau x b \mu^{-1})^2} - \sqrt{1 - (\tau(c_j^{-1} - x) b \mu^{-1})^2} \right) - \tau_f b - B_0 v_e \right), \\ \dot{x} &= v_e = c_i \sqrt{1 - (1 + e_k^e e_{\text{упр}}^{-1})^{-2}}. \end{aligned}$$

Заключение

В данной работе был описан процесс моделирования стационарной плотности порогов. Наиболее точное и обоснованное выражение для стационарной плотности порогов было найдено в модели кинетики движения винтовых сегментов дислокации. Для возможности использования данного выражения была сформулирована математическая модель скорости движения порогов. Данная модель была получена из математической модели кинетики движения краевых сегментов дислокации, путем преобразований. В дальнейшем необходимо провести вычислительные эксперименты, чтобы убедиться в достоверности и точности сформулированной модели.

ЛИТЕРАТУРА

1. Математическое моделирование от междислокационных взаимодействий до макроскопической деформации : монография / под ред. В.А. Старенченко. – Томск: Изд-во Том. гос. архит.-строит. ун-та, 2015. – 540 с.
2. Черепанов Д.Н., Старенченко В.А., Селиванникова О.В. Генерация межузельных атомов в монокристаллах с ГЦК - структурой // Изв. ВУЗов. Физика. – 2015. – Т. 58. – № 4. – С. 16–23.
3. Muller H., Leibfrid G.Z. Oberflächenscheinungen auf gedehntem Aluminium-Einkristaller in ihrer Abhängigkeit von der Dehngeschwindigkeit // Z. Phys. – 1955. – Bd. 142. – H. 2. – S. 87–115.
4. Старенченко В.А., Черепанов Д.Н., Слободской М.И. Кинетика порогов на движущейся винтовой дислокации в ГЦК-кристалле // Изв. вузов. Физика. – 2009. – № 9/2. – С. 108–117.

МЕТОД ИСКЛЮЧЕНИЯ ПЕРЕМЕННОЙ ИЗ КНФ ЦЕЙТИНА

Тычинский В.З.

Томский государственный университет
tvz.041@gmail.com

Введение

Непрерывное развитие микроэлектроники и увеличение сложности и компактности современных интегральных схем приводит к усложнению процесса их диагностики. Разработка проверяющих и диагностирующих тестов занимает всё больше времени при проектировании цифровых устройств. Одной из актуальных проблем диагностики является получение тестовых наборов для обнаружения неисправностей задержек путей [1,2].

В [3] один из подходов к решению задачи получения всех тестовых пар пути сводится к задаче выполнимости булевых формул, и использованию SAT-решателей. Развитие SAT-технологий позволяет обеспечить проверку выполнимости всё более сложных КНФ. В большинстве SAT-решателей используется представление логической схемы в виде КНФ Цейтина (также называемой «КНФ разрешения схемы» [4,5]).

КНФ Цейтина для элемента $y = f(x_1, x_2, \dots, x_n)$ некоторой комбинационной схемы есть КНФ от функции $g(y, f) = y \sim f(x_1, x_2, \dots, x_n)$, представляющая собой сингулярное покрытие исходной функции y [6]. Одним из главных достоинств данной КНФ является относительная простота её построения – сложность данной КНФ линейно зависит от числа элементов рассматриваемой схемы. Однако исключение переменной из данной КНФ обычным способом (исключение всех вхождений данной переменной в КНФ) недопустимо. КНФ Цейтина является представлением структуры схемы, содержащим сведения о связях между её вентилями. Как следствие, полученная после исключения переменной КНФ не будет соответствовать КНФ Цейтина булевой функции с исключённой переменной.

1. Исключение переменной из КНФ Цейтина

Для определения правил исключения переменной из КНФ Цейтина, необходимо проанализировать процесс исключения соответствующих переменной вентиля и связей из схемы:

1. Исключение переменной x из схемы S (рис. 1, 2) есть удаление подсхемы с выходом, соответствующим переменной x . При этом также требуется удалить все линии, связывающие выход исключаемой подсхемы с другими элементами схемы. Как следствие, число входов у связанных с ними элементов схемы уменьшается, что требует их преобразования (и изменения соответствующих им КНФ Цейтина).

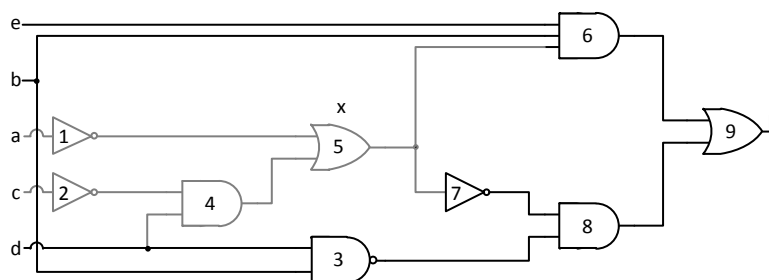


Рис. 1. Комбинационная схема S . Исключается подсхема с выходом u_5 .

2. Для одноходового элемента исключение входа эквивалентно исключению самого элемента (переменной, соответствующей его выходу).

3. Для многоходового элемента исключение входа эквивалентно замене этого элемента на элемент, реализующий булеву функцию исходного элемента с исключённой переменной.

Для реализации корректного удаления подсхем из КНФ Цейтина требуется чётко определить соответствия дизъюнктов их элементам. Для этого предлагается использовать подход деления схемы на уровни [7], при котором элементы соседних уровней связаны отношением предшествования. В дальнейшем предполагается, что нумерация уровней производится от входов схемы, а не от выходов. Первыми нумеруются элементы первого уровня, затем элементы второго уровня и т.д. Порядок нумерации элементов внутри одного уровня может быть произвольным.

Таким образом, дизъюнкты, которые относятся к рассматриваемому элементу x , содержат только переменные, соответствующие:

- входам схемы;
- внутренним элементам, номера которых меньше номера элемента, соответствующего переменной x .

Кроме того, при удалении требуется определить, какие из предшествующих данному вентилю элементов встречались в схеме (и, как следствие, в КНФ) один раз, и исключить соответствующие им дизъюнкты из выражения. Для этого предлагается при построении КНФ Цейтина не удалять повторяющиеся дизъюнкты. В то же время для сокращения записи предлагается указывать для каждого дизъюнкта степень, равную количеству повторений данного дизъюнкта в КНФ.

Рассмотрим в качестве примера исключение переменной u_5 из схемы C (рис. 1). Соответствующая схеме КНФ Цейтина имеет следующий вид:

$$TCNF(F) = (\bar{u}_6 \vee u_9) \wedge (\bar{u}_8 \vee u_9) \wedge (u_6 \vee u_8 \vee \bar{u}_9) \wedge (u_7 \vee \bar{u}_8) \wedge (u_3 \vee \bar{u}_8) \wedge (\bar{u}_7 \vee \bar{u}_3 \vee u_8) \wedge (u_5 \vee u_7) \wedge (\bar{u}_5 \vee \bar{u}_7) \wedge (e \vee \bar{u}_6) \wedge (b \vee \bar{u}_6) \wedge (u_5 \vee \bar{u}_6) \wedge (\bar{e} \vee \bar{b} \vee \bar{u}_5 \vee u_6) \wedge (\bar{u}_1 \vee u_5)^2 \wedge (\bar{u}_4 \vee u_5)^2 \wedge (u_1 \vee u_4 \vee \bar{u}_5)^2 \wedge (u_2 \vee \bar{u}_4)^2 \wedge (d \vee \bar{u}_4)^2 \wedge (\bar{u}_2 \vee \bar{d} \vee u_4)^2 \wedge (d \vee u_3) \wedge (b \vee u_3) \wedge (\bar{d} \vee \bar{b} \vee \bar{u}_3) \wedge (c \vee u_2)^2 \wedge (\bar{c} \vee \bar{u}_2)^2 \wedge (a \vee u_1)^2 \wedge (\bar{a} \vee \bar{u}_1)^2.$$

При исключении потребуется изменить элемент u_6 и исключить элемент u_7 . Далее, исключение переменной u_7 потребует изменения элемента u_8 . Для исключения переменных требуется удалить по одному дизъюнкту для элементов подсхем с выходами u_7 и u_5 . Результирующая КНФ Цейтина будет соответствовать схеме на рис. 2:

$$TCNF^*(F) = (\bar{u}_6 \vee u_9) \wedge (\bar{u}_8 \vee u_9) \wedge (u_6 \vee u_8 \vee \bar{u}_9) \wedge (u_3 \vee \bar{u}_8) \wedge (\bar{u}_3 \vee u_8) \wedge (e \vee \bar{u}_6) \wedge (b \vee \bar{u}_6) \wedge (\bar{e} \vee \bar{b} \vee u_6) \wedge (d \vee u_3) \wedge (b \vee u_3) \wedge (\bar{d} \vee \bar{b} \vee \bar{u}_3).$$

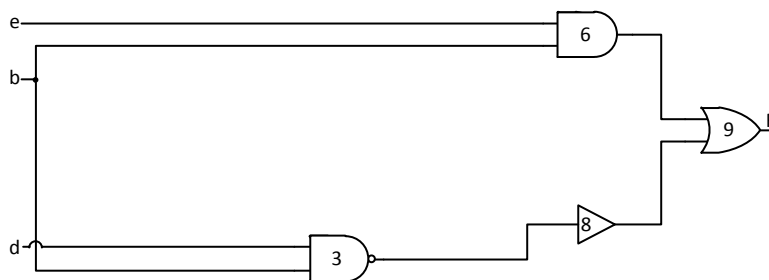


Рис. 2. Комбинационная схема C после исключения подсхемы с выходом u_5

2. Исключение переменной из КНФ Цейтина основных логических элементов

КНФ Цейтина для простейших вентилей имеет следующий вид:

$$y = x : (x \vee \bar{y}) \wedge (\bar{x} \vee y) ;$$

$$y = \bar{x} : (x \vee y) \wedge (\bar{x} \vee \bar{y}) ;$$

$$y = x_1 \oplus x_2 : (\bar{x}_1 \vee x_2 \vee y) \wedge (x_1 \vee \bar{x}_2 \vee y) \wedge (x_1 \vee x_2 \vee \bar{y}) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{y}) ;$$

$$y = x_1 \oplus x_2 : (x_1 \vee x_2 \vee y) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee y) \wedge (\bar{x}_1 \vee x_2 \vee \bar{y}) \wedge (x_1 \vee \bar{x}_2 \vee \bar{y}) ;$$

$$y = x_1 \wedge x_2 \wedge \dots \wedge x_n : (x_1 \vee \bar{y}) \wedge (x_2 \vee \bar{y}) \wedge \dots \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \dots \vee y) ;$$

$$y = \overline{x_1 \wedge x_2 \wedge \dots \wedge x_n} : (x_1 \vee y) \wedge (x_2 \vee y) \wedge \dots \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \dots \vee \bar{y}) ;$$

$$y = x_1 \vee x_2 \vee \dots \vee x_n : (\bar{x}_1 \vee y) \wedge (\bar{x}_2 \vee y) \wedge \dots \wedge (x_1 \vee x_2 \vee \dots \vee \bar{y}) ;$$

$$y = \overline{x_1 \vee x_2 \vee \dots \vee x_n} : (\bar{x}_1 \vee \bar{y}) \wedge (\bar{x}_2 \vee \bar{y}) \wedge \dots \wedge (x_1 \vee x_2 \vee \dots \vee y) .$$

Рассмотрим процесс исключения переменных для этих элементов:

1. Для одноходовых элементов вида «НЕ» / «Буфер» исключение входа эквивалентно исключению самого элемента (переменной, соответствующей его выходу):

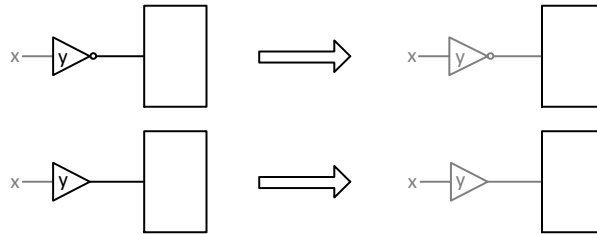


Рис. 3. Исключение переменной из одноходовых элементов

2. Для многоходовых (3 входа и более) элементов вида «И» / «ИЛИ» / «НЕ И» / «НЕ ИЛИ» исключение одного из входов эквивалентно замене элемента таким же элементом, с числом входов на единицу меньше:

$$TCNF(y) = (x_1 \vee \bar{y}) \wedge (x_2 \vee \bar{y}) \wedge (x_3 \vee \bar{y}) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee y)$$

$$TCNF^*(y) = (x_2 \vee \bar{y}) \wedge (x_3 \vee \bar{y}) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee y)$$

$$TCNF(y) = (\bar{x}_1 \vee y) \wedge (\bar{x}_2 \vee y) \wedge (\bar{x}_3 \vee y) \wedge (x_1 \vee x_2 \vee x_3 \vee \bar{y})$$

$$TCNF^*(y) = (\bar{x}_2 \vee y) \wedge (\bar{x}_3 \vee y) \wedge (x_2 \vee x_3 \vee \bar{y})$$

$$TCNF(y) = (x_1 \vee y) \wedge (x_2 \vee y) \wedge (x_3 \vee y) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{y})$$

$$TCNF^*(y) = (x_2 \vee y) \wedge (x_3 \vee y) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee \bar{y})$$

$$TCNF(y) = (\bar{x}_1 \vee \bar{y}) \wedge (\bar{x}_2 \vee \bar{y}) \wedge (\bar{x}_3 \vee \bar{y}) \wedge (x_1 \vee x_2 \vee x_3 \vee y)$$

$$TCNF^*(y) = (\bar{x}_2 \vee \bar{y}) \wedge (\bar{x}_3 \vee \bar{y}) \wedge (x_2 \vee x_3 \vee y)$$

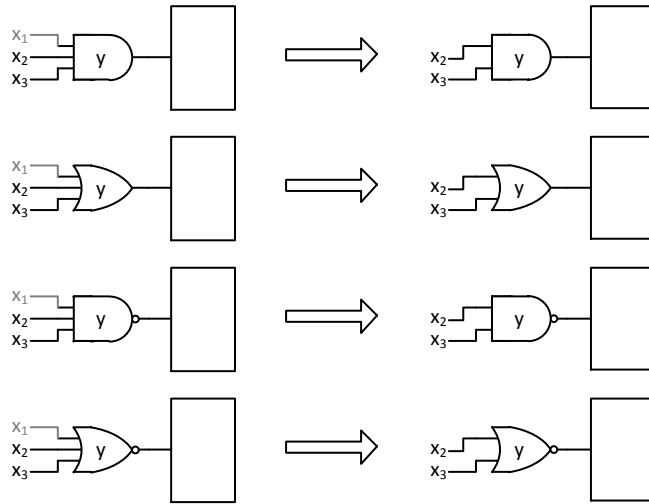


Рис. 4. Исключение переменной из многовходовых элементов

Для исключения входа x_i из КНФ Цейтина в этом случае необходимо и достаточно:
 – исключить дизъюнкт, содержащий только вход x_i и выход элемента y ;
 – исключить переменную x_i из дизъюнкта, содержащего все входы схемы и выход элемента y .

3. Для двухвходовых элементов вида «И» / «ИЛИ» («НЕ И» / «НЕ ИЛИ») исключение одного из входов эквивалентно замене элемента на буферный элемент (инвертор), при этом преобразование КНФ Цейтина аналогично п. 2:

$$TCNF(y) = (x_1 \vee \bar{y}) \wedge (x_2 \vee \bar{y}) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee y) \quad TCNF(y) = (\bar{x}_1 \vee y) \wedge (\bar{x}_2 \vee y) \wedge (x_1 \vee x_2 \vee \bar{y})$$

$$TCNF^*(y) = (x_2 \vee \bar{y}) \wedge (\bar{x}_2 \vee y) \quad TCNF^*(y) = (\bar{x}_2 \vee y) \wedge (x_2 \vee \bar{y})$$

$$TCNF(y) = (x_1 \vee y) \wedge (x_2 \vee y) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{y}) \quad TCNF(y) = (\bar{x}_1 \vee \bar{y}) \wedge (\bar{x}_2 \vee \bar{y}) \wedge (x_1 \vee x_2 \vee y)$$

$$TCNF^*(y) = (x_2 \vee y) \wedge (\bar{x}_2 \vee \bar{y}) \quad TCNF^*(y) = (\bar{x}_2 \vee \bar{y}) \wedge (x_2 \vee y)$$

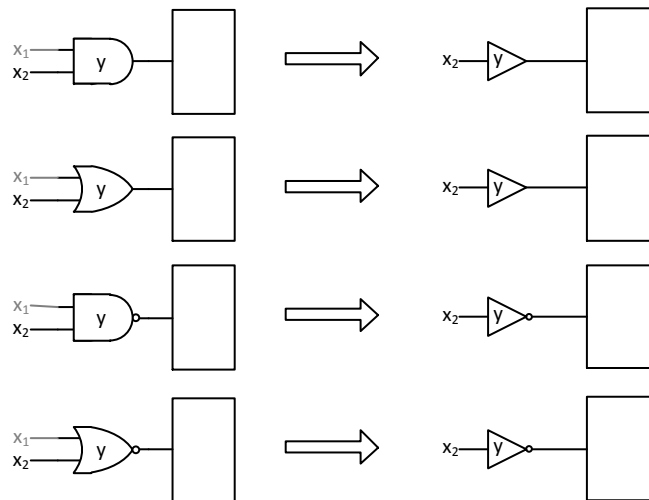


Рис. 5. Исключение переменной из двухвходовых элементов вида «И» / «ИЛИ» («НЕ И» / «НЕ ИЛИ»)

4. Для двухвходовых элементов вида «Исключающее ИЛИ» («Эквивалентность») исключение одного из входов эквивалентно замене элемента на комбинацию, дающую константу 1:

$$y | x_1 = (x_1 \oplus x_2) | x_1 = (\bar{x}_1 x_2 \vee x_1 \bar{x}_2) | x_1 = x_2 \vee \bar{x}_2$$

Таким образом, преобразование элемента схемы выглядит следующим образом:

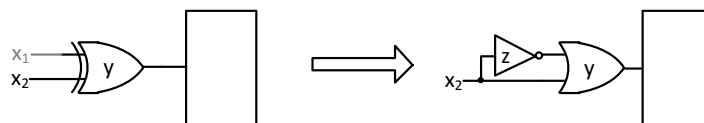


Рис. 6. Преобразование элемента «Исключающее ИЛИ» при исключении одного из входов

Обозначим в качестве $TCNF(x)$ КНФ Цейтина элемента x . Полную КНФ Цейтина подсхемы с выходом x обозначим через $TCNF^*(x)$. Тогда выражение КНФ Цейтина для подсхемы с выходом y будет иметь следующий вид:

$$\begin{aligned} TCNF^*(y) &= TCNF(y) \wedge TCNF^*(x_2) \wedge TCNF^*(z) = \\ &= TCNF(y) \wedge TCNF^*(x_2) \wedge (TCNF(z) \wedge TCNF^*(x_2)); \end{aligned}$$

$$TCNF(y) = (\bar{x}_2 \vee y) \wedge (\bar{z} \vee y) \wedge (x_2 \vee z \vee y);$$

$$TCNF(z) = (x_2 \vee z) \wedge (\bar{x}_2 \vee \bar{z});$$

$$TCNF^*(y) = (\bar{x}_2 \vee y) \wedge (\bar{z} \vee y) \wedge (x_2 \vee z \vee y) \wedge (x_2 \vee z) \wedge (\bar{x}_2 \vee \bar{z}) \wedge TCNF^*(x_2).$$

Поскольку переменная z введена при преобразовании и является дополнительной, она не будет исключаться из схемы самостоятельно. По этой причине второе выражение $TCNF^*(x)$ можно опустить, не увеличивая соответствующие ей счётчики дизъюнктов.

При попытке исключения переменной x_1 из данной подсхемы, в соответствии с правилами исключения, введёнными ранее, элемент с выходом y будет преобразован в буферный элемент, а элемент с выходом z потребует исключить, что в свою очередь приведёт к необходимости исключения элемента с выходом y .

Таким образом, полученную в результате преобразования можно представить в виде одноходового элемента (поскольку ввиду построения, исключение переменной z невозможно, а при исключении переменной x_1 исключается вся подсхема) со следующей таблицей истинности

x	y
0	1
1	1

с соответствующим сингулярным покрытием и КНФ Цейтина:

$$TCNF^*(y) = (\bar{x}_2 \vee y) \wedge (x_2 \vee y).$$

Аналогично, для элемента вида «Эквивалентность»:

$$y | x_1 = \overline{(x_1 \oplus x_2)} | x_1 = (x_1 x_2 \vee \bar{x}_1 \bar{x}_2) | x_1 = x_2 \vee \bar{x}_2;$$

$$TCNF(y) = (x_1 \vee x_2 \vee y) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee y) \wedge (\bar{x}_1 \vee x_2 \vee \bar{y}) \wedge (x_1 \vee \bar{x}_2 \vee \bar{y});$$

$$TCNF^*(y) = (x_2 \vee y) \wedge (\bar{x}_2 \vee y).$$

Таким образом, для исключения входа x_i из КНФ Цейтина в случае элементов «Исключающее ИЛИ» («Эквивалентность») необходимо и достаточно:

- исключить дизъюнкты, содержащие выход элемента y с инверсией (без инверсии);
- исключить переменную x_i из оставшихся дизъюнктов, независимо от знака инверсии.

Заключение

В данной работе был предложен алгоритм исключения переменной из КНФ Цейтина. С его помощью можно избежать преобразований КНФ Цейтина к другим формам

представления булевых функций для реализации исключения переменной. В частности, данный алгоритм может быть использован для исключения переменной в алгоритме формирования всех тестовых пар для робастно тестируемых неисправностей задержек путей [3]. Благодаря этому становится возможным гарантированно получать не только все тестовые пары, но любое требуемое число пар. Таким образом, становится возможным получение тестовой пары уже после первого решения полученного выражения SAT решателем.

Следует отметить, что в области 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 [8], 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. *Lyu Y., Mishra P.* Automated Test Generation for Trojan Detection using Delay-based Side Channel Analysis // 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020. – P. 1031–1036.
2. *Exurville I., Zussa L., Rigaud J., Robisson B.* Resilient hardware Trojans detection based on path delay measurements // 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2015. – P. 151–156.
3. *Матросова А.Ю., Тычинский В.З., Андреева В.В.* Построение тестовых последовательностей для робастно тестируемых неисправностей задержек путей с низкой потребляемой мощностью с использованием SAT-решателей и ROBDD-графов // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). – 2020. – № 2. – С. 43–49.
4. *Черемисинова Л.Д.* Поиск кратчайшей установочной последовательности схемы с памятью на D-триггерах // Весці Нацыянальнай акадэміі навук Беларусі. Серыя фізіка-матэматычных навук. – 2015. – С. 119–128.
5. *Kuehlmann A., van Eijk C.A.J.* Combinational and Sequential Equivalence Checking // Logic synthesis and Verification / eds S. Hassoun, T. Sasao, R. K. Brayton. – [S. l.]: Kluwer Academic Publishers, 2002. – P. 343–372.
6. *Roth J.P.* Diagnosis of Automata Failures: A Calculus and a Method // IBM Journal of Research and Development, July 1966. – V. 10. – No. 4. – P. 278–291.
7. *Armstrong D.B.* On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets. // IEEE Transactions on Electronic Computers, EC-15(1). – 1966. – P. 66–73.
8. *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". – Томск, 2019. – С. 190–196.

АВТОРСКИЙ УКАЗАТЕЛЬ

Авдеев Николай Геннадьевич.....	23
Агеева Анастасия Сергеевна	113
Аксёнов Сергей Владимирович	16
Андреева Валентина Валерьевна.....	214
Андриевская Анна Андреевна	48
Балашова Ольга Михайловна	71
Безходарнов Николай Ильич	191
Березовский Никита Владимирович	150
Бесчастный Виталий Александрович.....	106
Благинин Алексей Леонидович	130
Болдырева Кристина Константиновна.....	52
Буторина Наталья Борисовна.....	150
Васильева Ольга Владимировна.....	156
Веткина Анна Васильевна.....	63
Воркожиков Андрей Алексеевич	118
Гайдамака Юлия Васильевна.....	96
Гетц Кирилл Евгеньевич	162
Голос Елизавета Сергеевна	96
Гольшев Валерий Константинович	28
Горцев Александр Михайлович	63, 89
Джегоеде Адейеми Марк Ауреле Эммануэль	103
Дмитренко Анатолий Григорьевич	71
Дорожкин Антон Евгеньевич	3
Дуля Иван Сергеевич	32
Енютина Арина Ивановна.....	202
Ибрагимова Эллада Ибрагимовна	38
Иванова Ника Михайловна	121
Кабанова Татьяна Валерьевна	42
Кеба Анастасия Владимировна	78
Конюхов Роман Игоревич.....	106
Кришталева Никита Алексеевич	127
Куцазли Анна Ивановна.....	113
Лавриненко Милена Дмитриевна	183
Лопатин Иван Леонидович.....	118, 140
Литвинова Наталья Игоревна	166
Матросова Анжела Юрьевна	202
Мельман Анна Сергеевна.....	196
Моисеев Александр Николаевич.....	127
Молчанов Дмитрий Александрович	96, 106

Морозова Анна Сергеевна	156, 172
Морозова Мария Алексеевна.....	130
Мурзагулов Дамир Альбертович.....	32
Нежельская Людмила Алексеевна	78, 82
Нибасумба Эммануэль	121
Орел Иулиана Андреевна.....	172
Острикова Дарья Юрьевна.....	96
Павлюченко Мария Вячеславовна	42
Пауль Светлана Владимировна	136
Пахомова Елена Григорьевна.....	207
Петрива Наталья Владимировна	136
Полин Евгений Павлович.....	127
Попов Никита Сергеевич	177
Приступа Андрей Викторович.....	8
Провкин Виктор Алексеевич	202
Пупков Андрей Викторович	58
Разенков Семен Игоревич	188
Решетников Олег Алексеевич.....	140
Рудов Владислав Александрович	8
Савенкова Марина Максимовна	207
Салпагаров Солтан Исмаилович	103
Самохина Светлана Ивановна	191, 219
Самуйлов Константин Евгеньевич.....	106
Серебренникова Кристина Андреевна.....	214
Скворцов Алексей Владимирович	8
Степаненко Илья Денисович	82
Сыч Михаил Богданович.....	16
Тисленко Тимофей Иванович.....	144
Ткачук Полина Романовна	183
Трифонов Станислав Алексеевич.....	219
Тычинский Вячеслав Зиновьевич.....	223
Харин Петр Александрович.....	113
Черепанов Дмитрий Николаевич	219
Черняк Эдуард Михайлович	48
Чикалова Светлана Викторовна	183
Шкуркин Алексей Сергеевич	166, 177
Шманкеева Анастасия Вячеславовна.....	89
Якунина Елена Николаевна	162

СОДЕРЖАНИЕ

I. ИММЕРСИВНЫЕ ТЕХНОЛОГИИ, ТЕХНИЧЕСКОЕ ЗРЕНИЕ И ВИДЕОАНАЛИТИКА.....	3
Дорожкин А.Е. Руководство по самостоятельному проектированию и сборке масштабной модели с автономным управлением на примере Koenigsegg One:1	3
Рудов В.А., Приступа А.В., Скворцов А.В. Детектирование стандартных дорожных знаков на изображениях	8
Сыч М.Б., Аксёнов С.В. Обнаружение кальциатов в лёгких с помощью компьютерного зрения.....	16
II. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, МАШИННОЕ ОБУЧЕНИЕ, БОЛЬШИЕ ДАННЫЕ	23
Авдеев Н.Г. Комитетные методы нейронных сетей в задаче распознавания рукопечатных символов	23
Гольшев В.К. Нечёткий анализ формальных понятий: обзор подходов на примере	28
Дуля И.С., Мурзагулов Д.А. Применение методов глубокого обучения к задаче классификации временных рядов	32
Ибрагимова Э.И. О двух задачах кластеризации знакового графа	38
Павлюченко М.В., Кабанова Т.В. Анализ зависимости точности бинарной классификации текстов от применения мета-функций для различных алгоритмов классификации	42
III. МАТЕМАТИЧЕСКИЕ МЕТОДЫ И МОДЕЛИ ЦИФРОВОЙ ЭКОНОМИКИ	48
Андриевская А.А., Черняк Э.М. Анализ динамики показателей оборачиваемости оборотных средств производственного предприятия г. Томска	48
Болдырева К.К. Улучшенная оценка параметра авторегрессионного процесса с неизвестной дисперсией шумов	52
Пупков А.В. HAR-RV модель и прогноз волатильности	58
IV. ОБРАБОТКА ДАННЫХ, УПРАВЛЕНИЕ И ИССЛЕДОВАНИЕ СТОХАСТИЧЕСКИХ СИСТЕМ	63
Горцев А.М., Веткина А.В. Оценивание методом моментов параметра равномерного распределения длительности случайного непродлевающегося мертвого времени в рекуррентном полусинхронном потоке событий	63
Дмитренко А.Г., Балашова О.М. Исследование электромагнитного рассеяния на тонких соосных идеально проводящем и диэлектрическом цилиндрах	71
Кеба А.В., Нежелская Л.А. Плотность распределения вероятностей в обобщённом MAP-потоке событий	78
Нежелская Л.А., Степаненко И.Д. Статистические эксперименты на имитационной модели обобщенного полусинхронного потока событий с продлевающимся мертвым временем фиксированной длительности.....	82
Горцев А.М., Шманкеева А.В. Статистические эксперименты по оцениванию параметра распределения длительности случайного мёртвого времени в альтернирующем асинхронном потоке событий	89
V. ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ.....	96
Голос Е.С., Острикова Д.Ю., Молчанов Д.А., Гайдамака Ю.В. Моделирование эффекта микромобильности в беспроводных локальных сетях терагерцового диапазона	96

Джегюеде А.М.А.Э., Салпагаров С.И. О концепции P2P при передаче данных в сети с ограниченными возможностями	103
Конюхов Р.И., Бесчастный В.А., Молчанов Д.А., Самуйлов К.Е. К анализу непрерывности пользовательской сессии в условиях микромобильности и блокировки прямой видимости.....	106
Куцазли А.И., Агеева А.С., Харин П.А. Модель приоритетного доступа трафика URLLC с адаптивным изменением скорости трафика eMBB в сети 5G: Задача оптимизации	113
VI. ТЕОРИЯ МАССОВОГО ОБСЛУЖИВАНИЯ И ЕЕ ПРИЛОЖЕНИЯ.....	118
Воркожоков А.А., Лапатин И.Л. Исследование характеристик работы гибридной двухуровневой модели обработки запросов	118
Иванова Н.М., Нибасумба Э. Исследование надежности дублированной системы с полным восстановлением при произвольных длительностях жизни и ремонта её элементов.....	121
Кришталеv Н.А., Моисеев А.Н., Полин Е.П. Имитационное моделирование ресурсной СМО, функционирующей в случайной среде	127
Морозова М.А., Благинин А.Л. Имитационная модель тандемной системы массового обслуживания с повторными вызовами	130
Петрива Н.В., Пауль С.В. Гибридная система массового обслуживания с повторными вызовами	136
Решетников О.А., Лапатин И.Л. Исследование характеристик работы ненадежного узла обработки запросов.....	140
Тисленко Т.И. Задача MARL для светофора на перекрестке	144
VII. ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ АСПЕКТЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И DEVOPS-ИНЖЕНЕРИИ.....	150
Березовский Н.В., Буторина Н.Б. Разработка и анализ программной симуляции эволюционирующей водной экосистемы с процедурной генерацией ландшафта	150
Васильева О.В., Морозова А.С. Создание внешних обработок для переноса бухгалтерских проводок из различных конфигураций на платформе 1С: Предприятие	156
Гетц К.Е., Якунина Е.Н. Разработка экспертной системы поддержки принятия решений в киберспорте на примере серии игр FIFA	162
Литвинова Н.И., Шкуркин А.С. Разработка web-приложения для эффективного управления ресурсами компании по методологии O.K.R	166
Орел И.А., Морозова А.С. Разработка web-приложения по управлению предприятием с системой вознаграждений сотрудников.....	172
Попов Н.С., Шкуркин А.С. Разработка AR-приложения для визуализации исторической застройки города Томска.....	177
VIII. ПРОГРАММНЫЕ И АППАРАТНЫЕ АСПЕКТЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ.....	183
Chikalova S.V., Tkachuk P.R., Lavrinenko M.D. Evolution of Computer Viruses	183
Razenzov S.I. Efficient FPGA Implementation of a Reconfigurable FSM Based on Substitutions	188
Безходарнов Н.И., Самохина С.И. Чтение данных с экспериментальной установки	191

Мельман А.С. Подход к повышению безопасности встраивания информации в фазовый спектр ДПФ цифровых изображений.....	196
Провкин В.А., Матросова А.Ю., Енютина А.И. О возможностях введения дополнительных вентилях с целью обфускации логических схем.....	202
Савенкова М.М., Пахомова Е.Г. Создание программы для автоматизации удаления публикаций в сообществе социальной сети «ВКонтакте»	207
Серебренникова К.А., Андреева В.В. Применение 2-SAT задачи для обнаружения утечки ресурсов в списочных структурах.....	214
Трифонов С.А., Самохина С.И., Черепанов Д.Н. Моделирование кинетики точечных, линейных и поверхностных дефектов и порогов на движущейся винтовой и краевой дислокации в ГЦК металлах	219
Тычинский В.З. Метод исключения переменной из КНФ Цейтина.....	223
АВТОРСКИЙ УКАЗАТЕЛЬ	229

Научное издание

**МАТЕРИАЛЫ
VIII Международной молодежной научной
конференции
«МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ, ТЕХНИЧЕСКИХ
И ЭКОНОМИЧЕСКИХ СИСТЕМ»**

Томск, 26–30 мая 2021 г.

Под общей редакцией И.С. Шмырина

Издание подготовлено в авторской редакции

Подписано в печать 06.10.2021 г. Формат 70×108 1/16
Печ. л. 14,6; усл. печ. л. 20,4.
Тираж 500 экз. Заказ № 4805.

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 978-5-907442-42-9

