

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Том 27

2021

№ 6

САПР

КОМПЬЮТЕРНАЯ ГРАФИКА

МЕТОДЫ ПРОГРАММИРОВАНИЯ

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

ТЕЛЕКОММУНИКАЦИИ
И ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

НЕЙРОСЕТИ И
НЕЙРОКОМПЬЮТЕРЫ

СТРУКТУРНЫЙ СИНТЕЗ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ
СИСТЕМЫ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ

ИТ В ОБРАЗОВАНИИ

ГИС

Рисунки к статье А. В. Щекина

«АЛГЕБРА ЭЛЕМЕНТОВ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА»

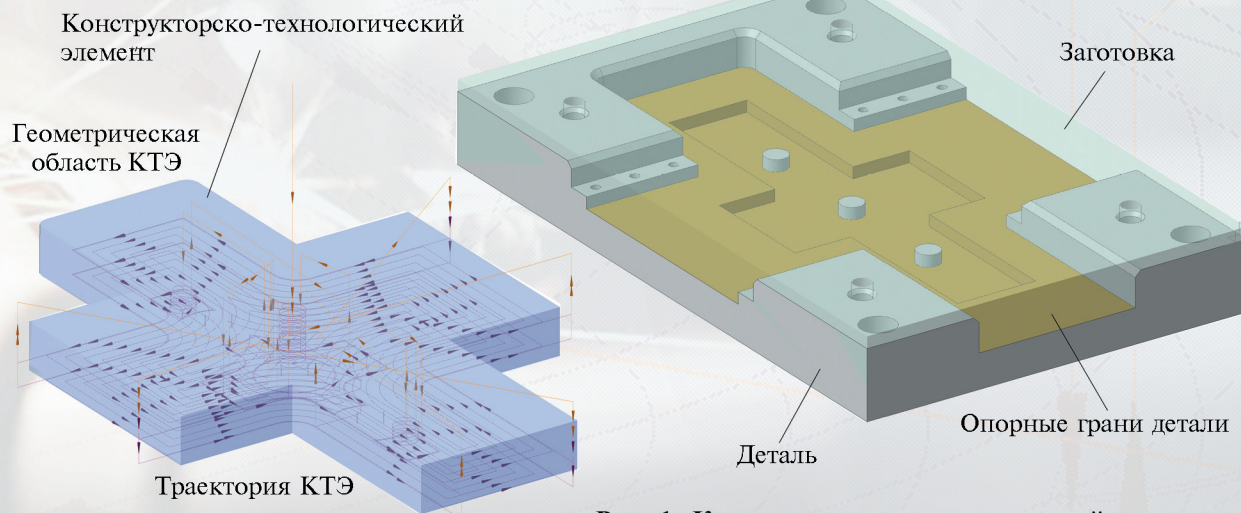
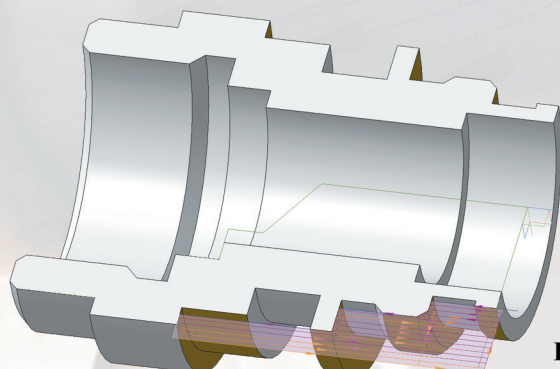


Рис. 1. Конструкторско-технологический элемент (КТЭ) фрезерной обработки



КТЭ = {Геометрическая область обработки,
Траектория обработки}

Рис. 2. КТЭ токарной обработки

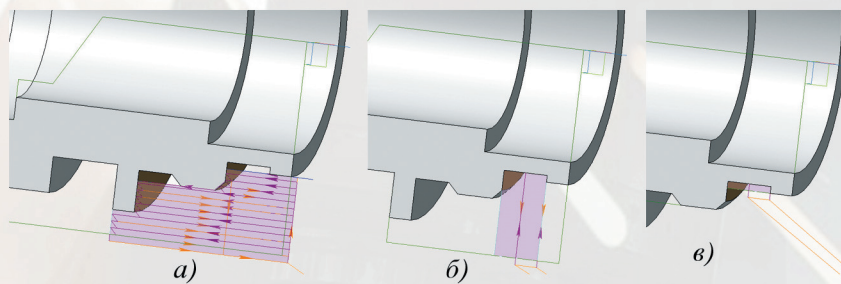


Рис. 3. Сложение двух элементов обработки

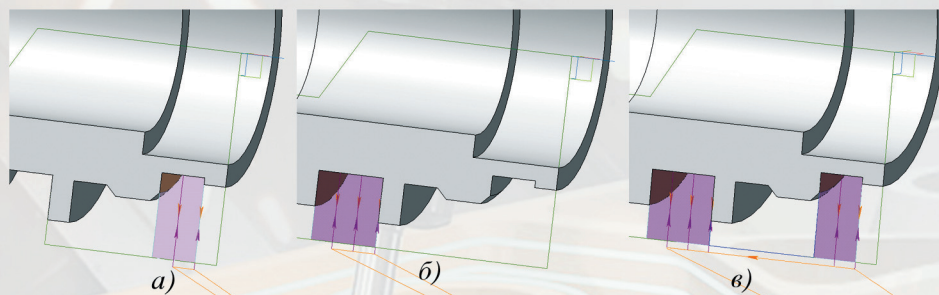


Рис. 4. Умножение двух элементов обработки

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 27
2021
№ 6

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

Щекин А. В. Алгебра элементов технологического процесса 283

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

Тарасов В. Н. Системы массового обслуживания с запаздыванием во времени 291

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

Фахми Ш. С., Давидчук А. Г., Костикова Е. В. Новые алгоритмы сжатия без потерь транспортных сюжетов 299

Ефанов Д. В., Сапожников В. В., Сапожников Вл. В. Организация схем встроенного контроля на основе метода логического дополнения с предварительным преобразованием рабочих функций в контрольные векторы кодов Бергера 306

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

Саксонов Е. А. Способ обезличивания персональных данных 314

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Гостева Е. А., Ланин В. В. Реализация системы интеллектуального поиска информации в промышленных стандартах 322

Израилов К. Е. Обобщенная классификация уязвимостей интерфейсов транспортной инфраструктуры Умного города 330

Главный редактор:

СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.
ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.
КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.
ПОПКОВ Ю. С.,
акад. РАН, д. т. н., проф.
РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.
РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.
СОЙФЕР В. А.,
акад. РАН, д. т. н., проф.
СОКОЛОВ И. А.,
акад. РАН, д. т. н., проф.
СУЕТИН Н. В., д. ф.-м. н., проф.
ЧАПЛЫГИН Ю. А.,
акад. РАН, д. т. н., проф.
ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.
ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.
ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.
БАРСКИЙ А. Б., д. т. н., проф.
ВАСЕНИН В. А., д. ф.-м. н., проф.
ВАСИЛЬЕВ В. И., д. т. н., проф.
ВИШНЕКОВ А. В., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.
ДОМРАЧЕВ В. Г., д. т. н., проф.
ЗАБОРОВСКИЙ В. С., д. т. н., проф.
ЗАРУБИН В. С., д. т. н., проф.
КАРПЕНКО А. П., д. ф.-м. н., проф.
КОЛИН К. К., д. т. н., проф.
КУЛАГИН В. П., д. т. н., проф.
КУРЕЙЧИК В. В., д. т. н., проф.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МАРТЫНОВ В. В., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
ПОЛЕШУК О. М., д. т. н., проф.
ПРОХОРОВ С. А., д. т. н., проф.
САКСОНОВ Е. А., д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
СОЛОВЬЕВ Р. А., д. т. н., в. н. с.
ТИМОНИНА Е. Е., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ФОМИЧЕВ В. А., д. т. н., проф.
ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

INFORMATION TECHNOLOGIES INFORMACIONNYYE TEHNOLOGII

Vol. 27
2021
No. 6

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

DOI 10.17587/issn.1684-6400

ISSN 1684-6400

CONTENTS

CAD-SYSTEMS

Shchekin A. V. Algebra of Technological Process 283

COMPUTING SYSTEMS AND NETWORKS

Tarasov V. N. Queuing Systems with a Time Lag 291

DIGITAL PROCESSING OF SIGNALS AND IMAGES

Fahmi S. Sh., Davidchuk A. G., Kostikova E. V. New Lossless Compression Algorithms for Transport Images 299

Efanov D. V., Saposhnikov V. V., Saposhnikov V. V. The Concurrent Error-Detection Systems Organization Based on the Boolean Complement Method with the Preliminary Transformation of Operating Functions into Check Vectors of Berger Codes 306

INFORMATION SECURITY

Saksonov E. A. Method of Depersonalization of Personal Data 314

INTELLIGENT SYSTEMS AND TECHNOLOGIES

Gosteva E. A., Lanin V. V. System Development for Intelligent Search in Industrial Standards 322

Izrailov K. E. Generalized Classification of Vulnerabilities of Interfaces to the Transport Infrastructure of the Smart City 330

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Vasiliev V. I., Dr. Sci. (Tech.), Prof.
Vishnekov A. V., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. V., Dr. Sci. (Tech.), Prof.
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Martynov V. V., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Poleschuk O. M., Dr. Sci. (Tech.), Prof.
Prokhorov S. A., Dr. Sci. (Tech.), Prof.
Saksonov E. A., Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Solovyev R. A., Dr. Sci. (Tech.)
Timonina E. E., Dr. Sci. (Tech.), Prof.
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ CAD-SYSTEMS

УДК 004.9:658.512

DOI: 10.17587/it.27.283-290

А. В. Щекин, зав. науч.-иссл. лаб. "Автоматизация программирования станков с ЧПУ",
e-mail: schekin@inbox.ru,

Национальный исследовательский Мордовский государственный университет имени Н. П. Огарева

Алгебра элементов технологического процесса

Представлен формальный аппарат моделирования структуры технологического процесса механической обработки на основе алгебры конструкторско-технологических элементов. Дано расширенное определение понятия «конструкторско-технологический элемент» с учетом объемного характера обработки материала в процессе изготовления детали. Конструкторско-технологический элемент рассматривается как совокупность геометрической области обработки и применяемой к ней траектории движения инструмента, задаваемой набором технологических параметров. Алгебра технологического процесса включает в себя операцию сложения (добавления элемента в структуру технологического процесса) и операцию умножения (слияния элементов). Множество элементов обработки образует ассоциативную и в общем случае некоммутативную алгебраическую группу. Показана возможность применения данной алгебры в задачах автоматизации синтеза структуры технологического процесса.

Ключевые слова: технологический процесс, автоматизация, план обработки, САМ-система, САПР ТП, конструкторско-технологический элемент

Введение

Проектирование технологических процессов механической обработки на современных промышленных предприятиях реализуется системами подготовки управляющих программ для станков с ЧПУ (САМ-системами) и выпуска технологической документации (САПР ТП). Но несмотря на активное внедрение указанных систем в информационное пространство предприятий уровень автоматизации технологической подготовки производства остается недостаточно высоким и не позволяет создавать технологические процессы полностью в автоматическом режиме. В значительной степени это обусловлено наличием в технологическом проектировании такой трудно формализуемой и многовариантной задачи, как построение структуры технологического процесса и планов обработки на отдельных операциях. В связи с этим разработка научно-обоснованных подходов к синтезу структуры технологического процесса продолжает оставаться актуальной проблемой автоматизации машиностроения.

Начало разработки методов автоматизированного проектирования технологических процессов приходится на 70–80-е годы 20 века. Тогда были сформулированы основные принципы построения САПР ТП, обозначены перспективные задачи в области автоматизации

технологического проектирования, разработаны первые диалоговые САПР [1]. В отечественной науке совершенствование технологической подготовки производства развивалось преимущественно в направлении унификации деталей, стандартизации средств технологического оснащения и типизации технологических процессов, т. е. в основном в рамках вариантного подхода. Во второй половине прошлого столетия сформировался основной набор математических методов для моделирования технологических процессов на базе теории множеств, теории графов, логики предикатов первого порядка, теории автоматов. Эти методы используются и в настоящее время для формализованного описания и решения задач технологического проектирования [2–5]. Для моделирования маршрутных процессов наиболее активно применяются графы, которые наглядно отображают альтернативные варианты технологического процесса.

В зарубежных САПР автоматизация технологического проектирования изначально развивается преимущественно по пути генеративного подхода. Этот подход подразумевает моделирование обработки деталей в САМ-системах на основе конструктивных элементов и получение управляющих программ, ассоциативно связанных с этими элементами [6–10]. В зарубежной литературе конструктивные

элементы обозначаются термином *features* [11] и представляют собой наборы из одной или нескольких поверхностей детали. В отечественной терминологии аналогом слова *feature* в контексте технологического проектирования является термин "конструкторско-технологический элемент". При этом проблема структурного синтеза технологического процесса решается на этапе интеграции САМ-систем с системами конструкторского проектирования CAD и сводится к двум основным задачам: распознаванию конструктивных элементов (задача *feature recognition* [6–7]) и составлению последовательности их обработки (задача *CAPP-планирования* [8, 9]).

В ходе синтеза структуры технологической операции формируется исходное множество проектных альтернатив, выполняется отсев нерациональных вариантов и выбор оптимальной последовательности переходов. В качестве критерия оптимальности могут приниматься различные целевые функции, например, максимум производительности процесса, минимум приведенных затрат или различные комплексные подходы [1]. Для поиска оптимального маршрута обработки обычно применяются алгоритмы вычисления кратчайших путей на графах. При этом в качестве вершин графа рассматриваются отдельные поверхности детали, а в качестве ребер — связи между поверхностями (например, время обработки). Далее в статье показано, что если учитывать пересечения элементов обработки, то в зависимости от маршрута движения по графу весовые коэффициенты ребер могут принимать переменные значения. Это обстоятельство делает проблематичным применение стандартных алгоритмов поиска оптимального пути на графе. В настоящей работе предлагается оригинальный подход для моделирования структуры технологического процесса на основе алгебры конструкторско-технологических элементов. Этот формальный аппарат можно использовать для анализа и синтеза планов обработки в рамках генеративного подхода.

1. Формальная модель конструкторско-технологического элемента

Понятие конструкторско-технологического элемента (КТЭ) в литературе часто определяется как совокупность поверхностей детали, характеризующих общностью применяемых методов формообразования [11, 12]. Но это определение не учитывает, что в процессе получения отдельных поверхностей детали происходит удаление

припуска, который представляет собой объем материала между заготовкой и указанными поверхностями. Поэтому дадим расширенное определение понятию КТЭ.

Определение 1. Конструкторско-технологический элемент — это замкнутая в пространстве геометрическая область, ассоциативно связанная с конструктивными элементами детали и характеризующаяся набором технологических параметров, задающих способ ее обработки.

В этом определении, во-первых, присутствует объемный характер изготовления детали, во-вторых, подчеркивается ассоциативная связь с ее конструктивными элементами и, в-третьих, предусматривается технологическая составляющая.

В общем случае КТЭ — это трехмерный объем удаляемого материала. Например, при объемной фрезерной обработке (рис. 1, см. вторую сторону обложки) КТЭ представляет собой твердое тело в пространстве, расположенное внутри тела заготовки и граничащее с его оболочкой (при этом под заготовкой понимается текущая форма детали перед началом обработки данного КТЭ). В некоторых случаях можно использовать упрощенные модели КТЭ, например, для токарных переходов КТЭ достаточно описать плоскими фигурами (рис. 2, см. вторую сторону обложки).

Ассоциативность является необходимым условием повторного использования конструкторско-технологических проектов [13]. Ассоциативность реализуется через наследование информации, которое означает не просто передачу данных от детали в систему технологической подготовки, а наличие в этом процессе "родителей" и "потомков". В общем случае КТЭ может унаследовать от конструктивных элементов детали не только геометрию, но и дополнительные данные, например, аннотации и свойства поверхностей. Некоторые аспекты наследования информации в интеграции систем конструкторско-технологической подготовки производства описаны в работах [14, 15]. Ассоциативная связь КТЭ с обрабатываемой деталью реализуется через ссылки на опорные объекты 3D-модели, в качестве которых могут использоваться грани, ребра или вершины детали, а также дополнительные операции построения в дереве модели.

Технологическая часть КТЭ — это набор движений режущего инструмента по заданной траектории. Для обозначения технологической составляющей будем использовать термин "траектория", подразумевая под ним всю совокупность числовых параметров, описывающих свойства инструмента, параметры траектории

и режимы резания. Далее для обозначения КТЭ будем использовать также сокращенный термин "элемент обработки". В формулах элемент обработки будем изображать символами MF (от англ. *manufacturing feature*).

Таким образом, КТЭ представляет собой совокупность обрабатываемой геометрической области и применяемой к ней траектории обработки. Формально КТЭ можно определить формулой

$$MF = \{Feature, Trajectory\},$$

где *Feature* — это геометрическая область обработки, которая включает в себя ее форму, расположение и ориентацию в пространстве. Для описания формы могут применяться различные способы математического описания: для трехмерных областей, например, Врепредставление, для 2D-областей — описания контурами или регионами.

Trajectory — это набор числовых параметров, задающих траекторию внутри элемента обработки. В состав *Trajectory* могут входить также номер инструментальной позиции при обработке на станке с ЧПУ, параметры инструмента и режимы обработки. *Trajectory* зависит от геометрии элемента обработки, т. е. представляет собой функцию *Trajectory(Feature)*.

К *Feature* и к *Trajectory* применимы операции над множествами.

Формирование набора КТЭ в различных системах технологического проектирования может осуществляться разными способами: ручным выбором обрабатываемых поверхностей, автоматизированным или автоматическим их распознаванием по геометрии детали. Для построения траекторий обработки с учетом текущей формы заготовки необходимо, чтобы область обработки замыкалась на геометрию, которую имеет заготовка перед началом обработки данного элемента. Замыкание элемента на заготовку представляет собой отдельную задачу в процессе формирования КТЭ и может выполняться разными способами, например, касательным продолжением контуров или граней элемента. Далее процедуру замыкания элемента на заготовку будем обозначать функцией *Extension(Feature)*.

Согласно ГОСТ 3.1109—82 "ЕСТД" [11] технологический процесс состоит из операций, установов, позиций, переходов и т. д. Однако первичными элементами структуры технологического процесса следует считать не переходы, а КТЭ. Любой объект структуры технологического процесса, который связан с удалением материала с заготовки (переход, операция или технологический процесс в целом), всегда имеет геометрическую область. Поэтому первич-

ной структурой технологического процесса является структура, составленная из элементов обработки. Первичная структура может быть отображена в другие способы описания технологического процесса, в том числе и в терминах ЕСТД. Для составления первичной структуры технологического процесса предлагается алгебра на множестве элементов обработки.

2. Алгебра технологического процесса

Алгебра технологического процесса имеет следующий вид: $A = \{E, \Omega\}$, где E — множество элементов обработки; Ω — множество алгебраических операций, обозначаемых набором операторов $\{+, -, \times, /\}$.

Операция сложения элементов обработки обозначается знаком $+$. Поскольку результат технологического процесса механической обработки представляет собой также элемент обработки, то операция сложения одновременно является и операцией добавления элемента в структуру технологического процесса. Операция сложения выполняется согласно следующим правилам:

$$MF_1 + MF_2 = MF_3;$$

$$Feature_3 = Feature_1 \cup Feature_2; \quad (1)$$

$$Trajectory_3 = Trajectory_1 \cup Trajectory_2 (Feature_2 \setminus Feature_1); \quad (2)$$

$$Feature_2 = Feature_2 \setminus Feature_1. \quad (3)$$

В результате операции сложения происходит объединение геометрических областей элементов. Траектория суммарного элемента представляет собой множество не зависящих друг от друга траекторий элементов (на рис. 3, а (см. вторую сторону обложки) показан первый элемент и его траектория, на рис. 3, б — второй элемент в исходном состоянии). При этом область второго элемента отсекается областью первого элемента (рис. 3, в, см. вторую сторону обложки).

Аксиома 1. Операция сложения элементов обработки ассоциативна:

$$MF_1 + (MF_2 + MF_3) = (MF_1 + MF_2) + MF_3.$$

Утверждение 1. Любая непустая структура технологического процесса может быть представлена суммой ее элементов обработки, равной результирующему элементу обработки всего технологического процесса:

$$MF_{\text{ТП}} = MF_1 + MF_2 + MF_3 + \dots \\ \dots + MF_i + \dots + MF_n.$$

Порядок записи элементов в сумме задает последовательность обработки элементов в структуре технологического процесса. Аксиома 1 следует из индуктивного соображения, что расстановка скобок внутри суммы не меняет порядок изготовления элементов, скобки позволяют просто группировать некоторые элементы в наборы.

Утверждение 2. Операция сложения элементов в общем случае не коммутативна:

$$MF_1 + MF_2 \neq MF_2 + MF_1.$$

Коммутативность операции сложения означает, что можно изменить порядок изготовления элементов. Множество элементов обработки может содержать как коммутирующие, так и некоммутирующие элементы по операции сложения. Сочетаемость элементов определяется большим числом различных факторов (геометрических, технологических, организационно-экономических), главным из которых является геометрическая доступность элемента.

Определение 2. Нейтральным элементом множества элементов обработки является такой нулевой элемент, у которого не задана либо область обработки ($Feature = \emptyset$), либо траектория ($Trajectory = \emptyset$).

Теорема 1. Сложение элемента обработки с самим собой и его умножение на число равно этому элементу:

$$MF + MF + MF + \dots = n \cdot MF = MF. \quad (4)$$

Доказательство. Результат выполнения формул (1) и (2) с двумя элементами показывает, что повторная обработка по ранее обработанной области не добавляет новых элементов:

$$\begin{aligned} Feature \cup Feature &= Feature; \\ Trajectory(Feature) \cup Trajectory(Feature \setminus Feature) &= \\ &= Trajectory \cup \emptyset = Trajectory. \end{aligned}$$

Операция вычитания элементов обработки обозначается знаком $-$. Операция вычитания одновременно является и операцией удаления элемента из структуры технологического процесса. Результирующая область обработки замыкается операцией *Extension* на область заготовки. Операция вычитания выполняется согласно следующим правилам:

$$\begin{aligned} MF_1 - MF_2 &= MF_3; \\ Feature_3 &= Extension(Feature_1 \setminus Feature_2); \quad (5) \\ Trajectory_3 &= Trajectory_1 \setminus Trajectory_2. \quad (6) \end{aligned}$$

Обратный элемент обработки MF^{-1} для операции сложения представляет собой этот же элемент со знаком минуса.

Теорема 2. Сложение элемента с его обратным элементом равно нулевому элементу:

$$MF + (-MF) = MF - MF = \emptyset.$$

Доказательство этой теоремы следует из формул (5) и (6).

На множестве элементов обработки были определены операция сложения, нулевой элемент и обратный элемент. Таким образом, множество элементов обработки представляет собой алгебраическую группу G^+ . В общем случае эта группа некоммутирующая и может содержать как коммутирующие, так и некоммутирующие элементы. Подмножество коммутирующих элементов внутри группы G^+ образует абелеву группу G_a^+ , которая задает множество всех допустимых перестановок в структуре технологического процесса. Поэтому для анализа и синтеза плана обработки в первую очередь представляет интерес свойство коммутативности операции сложения элементов обработки.

Первым и главным фактором, влияющим на сочетаемость элементов, является геометрическая доступность области обработки элемента. Очевидно, что нельзя поменять местами точение наружной поверхности резьбы и собственно нарезание резьбы, точно так же нельзя изготовить внутреннюю канавку, не подготовив соответствующего отверстия. Но геометрически независимые элементы (если на них не влияют дополнительные факторы) вполне могут являться коммутирующими. Например, растачивание и наружное точение можно переставить местами, можно также изменить последовательность сверления и наружного точения. Выявление геометрических и технологических факторов, влияющих на сочетаемость элементов в операции сложения, требует отдельного исследования и будет представлено в следующих публикациях.

Операция умножения элементов обработки обозначается знаком $\times$. Операция умножения представляет собой слияние элементов, в результате которого объединяются геометрические области обработки, а параметры траектории результирующего элемента приравниваются к параметрам траектории первого элемента (на рис. 4, *a* и 4, *б* — исходные элементы, на рис. 4, *в* — результат умножения (см. вторую сторону обложки)). В результате операции умножения второй элемент полностью поглощается первым элементом. Операция умножения имеет приоритет над операцией сложения. Операция умножения выполняется согласно следующим правилам:

$$MF_1 \times MF_2 = MF_3;$$

$$Feature_3 = Feature_1 \cup Feature_2;$$

$$Trajectory_3 = Trajectory_1.$$

Аксиома 2. Операция умножения элементов обработки ассоциативна:

$$MF_1 \times (MF_2 \times MF_3) = (MF_1 \times MF_2) \times MF_3.$$

Утверждение 3. Операция умножения элементов в общем случае не коммутативна:

$$MF_1 \times MF_2 \neq MF_2 \times MF_1.$$

Необходимым условием коммутативности операции умножения двух элементов является пересечение допустимых интервалов числовых значений параметров их траекторий, т. е. условие $Trajectory_1 \cap Trajectory_2 \neq \emptyset$.

Так же, как и для операции сложения, множество элементов обработки для операции умножения представляет собой алгебраическую группу $G^{\times}$ (в общем случае не коммутативную), в которой определены операция умножения, обратный элемент MF^{-1} , получаемый через операцию деления, и нейтральный элемент I .

Операция деления обозначается знаком $/$ и выполняется согласно следующим правилам:

$$MF_1/MF_2 = MF_3;$$

$$Feature_3 = Feature_1 \setminus Feature_2;$$

$$Trajectory_3 = Trajectory_1.$$

Исходными данными для практического применения алгебры являются таблицы Кэли ее операций, которые отображают структуру алгебраической системы, применимость операций к парам элементов и их свойство коммутативности. Автоматическому составлению таблиц Кэли предшествует декомпозиция области обработки технологического процесса (эта область получается вычитанием детали из заготовки) на множество исходных элементов. На данном множестве для каждой пары исходных элементов выполняется проверка применимости операций и их свойства коммутативности, в результате которой окончательно формируются таблицы Кэли.

Применение операций алгебры к каждой паре элементов позволяет:

- с учетом свойства коммутативности получить комбинации элементов обработки в виде алгебраических формул и одновременно провести проверку корректности

структурной упорядоченности элементов алгебраическими методами;

- на базе совокупности полученных алгебраических формул сформировать граф возможных структур технологического процесса, который в дальнейшем можно использовать для поиска оптимального плана обработки.

3. Пример использования алгебры технологического процесса

Рассмотрим пример, который демонстрирует возможность применения алгебры технологического процесса в задаче синтеза плана обработки. На рис. 5 показано ступенчатое отверстие, состоящее из трех элементов A , B и C . Декомпозиция области обработки между деталью и заготовкой на исходные элементы выполнена исключительно из геометрических соображений (калитеты и шероховатости отверстий не учитывались). Все исходные элементы замкнуты на заготовку. Для отверстий A и B требуется предварительное центрование (элемент f на рис. 5). Отверстие C растачивается после предварительного засверливания с диаметром не менее 14 мм (элемент d). Локальные планы обработки отверстий имеют вид:

$$A = f + a, B = f + b, C = d + c.$$

Целью решения данной задачи является выбор оптимального плана изготовления отверстий по критерию минимального суммарного времени их обработки.

Составим таблицу Кэли для операции сложения по элементам A , B и C (табл. 1). В данном случае все три элемента образуют коммутативную группу, на базе которой можно составить $3! = 6$ вариантов плана обработки.

Алгебра технологического процесса позволяет вывести конечные формулы вариантов плана обработки из комбинаций элементов A , B и C в комбинации локальных элементов

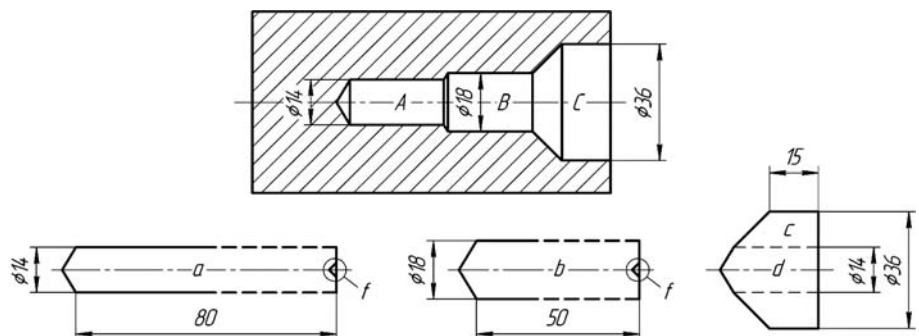


Рис. 5. Исходные элементы ступенчатого отверстия

Таблица 1

Таблица Кэли для операции сложения по элементам A , B и C

	A	B	C
A	A	$A + B$	$A + C$
B	$B + A$	B	$B + C$
C	$C + A$	$C + B$	C

Таблица 2

Таблица Кэли для операции сложения по локальным элементам

	f	a	b	d	c
f	f	$f + a$	$f + b$	$f + d$	$f + c$
a	—	—	$a + b$	a	$a + c$
b	—	$b + a$	—	b	$b + c$
d	$-(d)$	$d + a$	$d + b$	d	$d + c$
c	$-(c)$	$c + a$	$c + b$	—	—

с использованием простых символьных преобразований. Для парных сложений локальных элементов будем использовать таблицу Кэли по операции сложения (табл. 2).

При составлении табл. 2 учитывались формулы (1) и (3) для операции сложения и выше определенные геометрические правила определения коммутативности двух элементов. Для некоммутирующих элементов, а также элементов, области которых не пересекаются, в ячейки таблицы записываем полное обозначение суммы, для остальных пересекающихся элементов записываем результат их сложения. Например, поскольку элементы a и d находятся в разных планах обработки A и C , которые можно переставить местами, то a и d являются также коммутирующими элементами. При этом элемент a в результате прибавления к нему элемента d в силу формул (1) и (3) поглощает элемент d (в результате сложения $a + d$ из формулы (3) следует, что элемент d вообще перестает существовать), и в соответствующую ячейку таблицы Кэли записываем сразу результат их сложения, равный a . Но при обратном сложении $d + a$ элемент d полностью не исчезает, а только вычитается из области a , поэтому для $d + a$ записывается полное обозначение суммы. Аналогично и для элементов b и d . Жесткое следование a после f задано в локальном плане A , поэтому f и a являются некоммутирующими элементами. Диагональные элементы таблицы показывают возможность повторного выполнения элементов при отсутствии других элементов. Например, центрование f может быть выполнено повторно много раз без получения при этом новых результатов, кроме пустой траты времени станка. Поэтому

из формулы (4) следует, что $f + f = f$. Аналогично и для элемента d . Элементы a и b не могут быть выполнены без предварительного центрования, как и элемент c без засверливания, поэтому диагональные ячейки таблицы для элементов a , b и c остаются пустыми.

Отдельно остановимся на сумме $d + f$. Геометрическая область элемента f находится внутри элемента d , и в результате прибавления f к d из формулы (3) следует, что элемент f перестает существовать, и их сумма становится равной d . Таким образом, если исходить только из геометрических соображений, то получается, что сверление отверстия d заменяет центрование отверстий a и b . Оценить, хватит ли жесткости сверла для отверстия d для обеспечения точности расположения центрального отверстия по оси заготовки, можно только в том случае, если учесть дополнительные технологические факторы, такие как соотношение длины и диаметра сверла. Технологические факторы — это второй уровень факторов, влияющих на свойство коммутативности элементов. В данном примере будем считать, что жесткости сверла не хватает, чтобы заменить центрование, и тогда пары элементов f и d , а также f и c перестают быть коммутирующими.

Используя таблицу Кэли для операции сложения, получаем преобразования формул из комбинаций элементов A , B и C в комбинации локальных элементов (табл. 3). Две последние комбинации являются некорректными, так как содержат неприменимую операцию $c + f$.

Выбор оптимальной формулы осуществлялся путем перестроения плана обработки и рас-

Таблица 3

Формулы плана обработки

Комбинации	Формулы преобразования	Время, мин
$A + B + C$	$(f + a) + (f + b) + (d + c) =$ $= 2f + a + b + d + c =$ $= f + a + (b + d) + c = f + a + b + c$	2,41
$A + C + B$	$(f + a) + (d + c) + (f + b) =$ $= 2f + a + d + c + b =$ $= f + (a + d) + c + b = f + a + c + b$	2,47
$B + A + C$	$(f + b) + (f + a) + (d + c) =$ $= 2f + b + a + d + c =$ $= f + b + (a + d) + c = f + b + a + c$	2,03
$B + C + A$	$(f + b) + (d + c) + (f + a) =$ $= 2f + b + d + c + a =$ $= f + (b + d) + c + a = f + b + c + a$	2,03
$C + A + B$	$(d + c) + (f + a) + (f + b) = d + c + 2f +$ $+ a + b = d + (c + f) + a + b = \emptyset$	—
$C + B + A$	$(d + c) + (f + b) + (f + a) = d + c + 2f +$ $+ b + a = d + (c + f) + b + a = \emptyset$	—

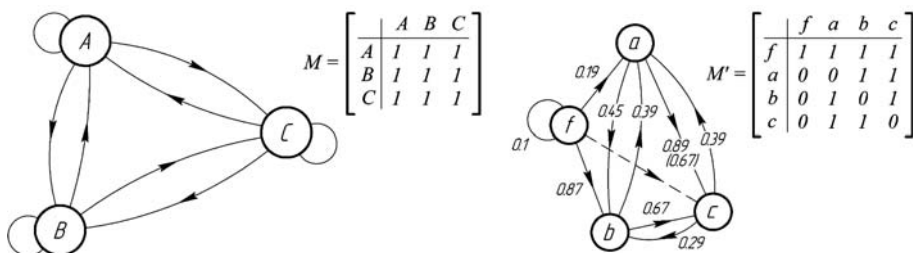


Рис. 6. Граф технологической структуры и его матрица смежности

чета каждого варианта в САМ-системе, разрабатываемой автором данной статьи для САПР-платформы КОМПАС-3D [16, 17]. Для расчета режимов резания использовались возможности конструкторско-технологической параметризации, встроенной в состав САМ-системы [18]. В результате численного эксперимента были получены две оптимальные формулы плана обработки $f + b + a + c$ и $f + b + c + a$ с временем обработки 2,03 мин (табл. 2).

Табл. 1 эквивалентна ориентированному графу и его матрице смежности M (рис. 6). Ребра графа обозначают операцию сложения двух элементов, вес ребра равен времени обработки второго элемента. Наличие петли у элемента означает, что данный элемент может присутствовать независимо от других элементов. Если вершина графа имеет петлю, то соответствующий диагональный элемент матрицы M равен 1, если петля отсутствует, то этот элемент равен 0. Отсутствие петли у вершины означает также, что элемент не может быть первым в плане обработки. Анализируя полученные формулы, можно построить граф по локальным элементам f , a , b и c и его матрицу смежности M' (рис. 6).

В общем случае время обработки отдельного элемента зависит от его места внутри последовательности обработки. Поэтому при движении по разным маршрутам веса некоторых ребер могут принимать переменные значения. Например, в маршрутах $f + a + c + b$ и $f + b + a + c$ вес ребра $a + c$ различный. Это объясняется тем, что во втором маршруте начальный диаметр растачивания отверстия c больше, чем в первом, и соответственно время обработки элемента c уменьшается. Динамический характер весовых коэффициентов ребер графа отражает тот факт, что при изменении последовательности обработки пересекающихся элементов происходит не только перестановка элементов в маршруте обработки, но и их геометрическая и технологическая трансформация. Данное обстоятельство делает затруднительным применение известных алгоритмов оптимизации на графах. Таким образом, поиск оптимального маршрута сводится к прямому перебору возможных ва-

риантов с их расчетом в системе технологического проектирования. В связи с этим возникает потребность в разработке способов минимизации как исходного числа допустимых формул плана обработки, так и времени их вычисления. В этом смысле применение алгебры технологического процесса совместно с ее таблицами Кэли позволяет под-

готовить исходное множество вариантов с использованием простых символьных вычислений.

Заключение

Алгебра технологического процесса представляет собой формальный аппарат для моделирования структуры технологического процесса, который можно использовать в задачах анализа и синтеза планов механической обработки. Практическая ценность предлагаемого подхода заключается в возможности его применения для автоматизации структурного синтеза технологических процессов. Алгебраическая запись маршрутов обработки отражает не только последовательность обрабатываемых элементов, но и характер связей между ними, и позволяет осуществлять преобразование формул посредством простых символьных вычислений. Этот подход в дальнейшем планируется развивать в составе интегрированной САМ-системы для платформы КОМПАС-3D. Программная реализация подсистемы автоматического синтеза структуры технологического процесса повышает эффективность проектируемых технологических процессов.

Список литературы

1. Корчак С. Н., Кошин А. А., Ракович Ф. Г., Синицын Б. И. Системы автоматизированного проектирования технологических процессов, приспособлений и режущих инструментов. Учеб. для вузов / Под общ. ред. С. Н. Корчака. М.: Машиностроение, 1988. 352 с.
2. Кузьмин В. В. Выбор технологических баз при обработке корпусных деталей // Вестник МГТУ "Станкин". 2008. № 2. С. 10–14.
3. Чигиринский Ю. Л. Возможность формализованного решения задач технологического проектирования // СТИН. 2009. № 12. С. 26–29.
4. Митин С. Г., Бочкарев П. Ю., Азиков Н. С. Метод генерации структур технологических операций для оборудования сверлильной группы // Проблемы машиностроения и надежности машин. 2018. № 2. С. 69–74.
5. Mitin S., Bochkarev P. Mathematical modelling in the computer-aided process planning // IOP Conference Series: Materials Science and Engineering. 124 (2016) 012077.
6. Babic B., Nesic N., Miljkovic Z. A Review of Automated Feature Recognition with Rule-Based Pattern Recognition // Computers in Industry. 2008. Vol. 59, Iss. 4. P. 321–337.
7. Mesbahi A. E., Jaider O., Rechia A. Automatic Recognition of Isolated and Interacting Manufacturing Features in Milling

Process // International Journal of Engineering Research and Applications. 2014. Vol. 4, Iss. 10. P. 57–72.

8. **Xun Xu, Lihui Wang, Stephen T. Newman.** Computer-aided process planning — A critical review of recent developments and future trends // International Journal of Computer Integrated Manufacturing. 2011. Vol. 24, N. 1. P. 1–31.

9. **Yusof Y., Latif K.** Survey on computer-aided process planning // International journal of advanced manufacturing technology. 2014. N. 75. P. 77–89.

10. **Sanfilippo E. M., Borgo S.** What are Features? An Ontology-Based Review of the Literature // Computer-Aided Design. 2016. Vol. 80. P. 9–18.

11. **ГОСТ 3.1109-82.** Единая система технологической документации (ЕСТД). Термины и определения основных понятий (с Изменением N 1).

12. **Зиннатуллин И. А., Горяинов Д. С.** Метод автоматизированного распознавания конструкторско-технологических элементов детали // Известия Самарского научного центра Российской академии наук. 2016. Т. 18, № 4 (2). С. 257–261.

13. **Щекин А. В., Сульдин С. П.** Ассоциативность траекторий в САМ-приложении "Модуль ЧПУ. Токарная обработка" // Мехатроника, автоматизация, управление. 2015. Т. 16, № 8. С. 570–575.

14. **Щекин А. В.** Автоматизация получения параметров детали для задач конструкторско-технологической параметризации // Инженерные технологии и системы. 2019. Т. 29, № 3. С. 345–365. DOI: 10.15507/2658-4123.029.201903.345-365.

15. **Щекин А. В.** Особенности наследования информации в задачах интеграции систем технической подготовки производства // Онтология проектирования. 2020. Т. 10, № 2(36). С. 201–217. DOI: 10.18287/2223-9537-2020-10-2-201-217.

16. **Модуль ЧПУ. Токарная обработка.** URL: <https://kompas.ru/kompas-3d/application/machinery/module-chpu>.

17. **Модуль ЧПУ. Фрезерная обработка.** URL: <https://kompas.ru/kompas-3d/application/machinery/module-chpu-fo>.

18. **Щекин А. В.** Конструкторско-технологическая параметризация в составе интегрированной САМ-системы // Информационные технологии. 2019. № 7. С. 34–54.

A. V. Shchekin, Chief of the Research Laboratory "Automation of CNC Programming",

e-mail: schekin@inbox.ru,

N. P. Ogarev Mordovia State University, Saransk, 430005, Russian Federation

Algebra of Technological Process

A formal apparatus for modeling the structure of the technological process of mechanical processing based on the algebra of design and technological elements is presented. Design and technological element (manufacturing feature) is considered as a set of geometric processing area and the tool trajectory applied to it, set by a set of technological parameters. Algebra includes an addition operation (adding an element to the process structure) and a multiplication operation (merging elements). The set of processing elements forms an associative and generally noncommutative algebraic group. The possibility of using algebra for analysis and synthesis of technological process structures is shown.

Keywords: processing plan, CAM system, CAD TP, COMPASS-3D, 3D model, design and technological element

DOI: 10.17587/it.27.283-290

References

1. **Korchak S. N., Koshin A. A., Rakovich F. G., Sinicyn B. I.** Computer-aided design systems for technological processes, fixtures and cutting tools, Moscow, Mashinostroenie, 1988, 352 p. (in Russian).

2. **Kuzmin V. V.** Selection of Technological Bases for Handling Details, *Vestnik MGTU "Stankin"*, 2008, no. 2, pp. 10–14 (in Russian).

3. **Chigirinskij Yu. L.** Possibility of formalized solution of technological design problems, *STIN*, 2009. no. 12, pp. 26–29 (in Russian).

4. **Mitin S. G., Bochkarev P. Y., Azikov N. S.** The technique for generating structures of manufacturing operations for equipment for a drilling group, *Journal of Machinery Manufacture and Reliability*, 2018. vol. 47, no. 2, pp. 181–186 (in Russian).

5. **Mitin S., Bochkarev P.** Mathematical modelling in the computer-aided process planning, *IOP Conference Series: Materials Science and Engineering*, 124 (2016), 012077.

6. **Babic B., Nesic N., Miljkovic Z.** A Review of Automated Feature Recognition with Rule-Based Pattern Recognition, *Computers in Industry*, 2008, vol. 59, iss. 4, pp. 321–337.

7. **Mesbahi A. E., Jaider O., Rechia A.** Automatic Recognition of Isolated and Interacting Manufacturing Features in Milling Process, *International Journal of Engineering Research and Applications*, 2014, vol. 4, iss. 10, pp. 57–72.

8. **Xun Xu, Lihui Wang, Stephen T. Newman.** Computer-aided process planning — A critical review of recent developments and future trends, *International Journal of Computer Integrated Manufacturing*, 2011, vol. 24, no. 1, pp. 1–31.

9. **Yusof Y., Latif K.** Survey on computer-aided process planning, *International journal of advanced manufacturing technology*, 2014, no. 75, pp. 77–89.

10. **Sanfilippo E. M., Borgo S.** What are Features? An Ontology-Based Review of the Literature, *Computer-Aided Design*, 2016, vol. 80, pp. 9–18.

11. **GOST 3.1109-82.** Unified system of technological documentation (ESTD). Terms and definitions of basic concepts (with Amendment N 1) (in Russian).

12. **Zinnatullin I. A., Goryainov D. S.** Automated feature recognition method, *Izvestia of Samara Scientific Center of the Russian Academy of Sciences*, 2016, vol. 18, no. 4(2), pp. 257–261 (in Russian).

13. **Schekin A. V., Suldin S. P.** Associativity of the Tool Paths in the CAM-Application CNC-Module. Lathe Operation. *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2015, vol. 16, no. 8, pp. 570–575 (in Russian).

14. **Shchekin A. V.** Automation of Obtaining the Detail Parameters for Tasks of Design-Technological Parametrization, *Inzhenernyye tekhnologii i sistemy=Engineering Technologies and Systems*, 2019, vol. 29, no. 3. pp. 345–365, DOI: 10.15507/2658-4123.029.201903.345-365 (in Russian).

15. **Shchekin A. V.** The specifics of information inheritance in CAD/CAM-integration, *Ontology of designing*, 2020, vol. 10, no. 2, pp. 201–217, DOI: 10.18287/2223-9537-2020-10-2-201-217 (in Russian).

16. **Module** Computer numerical control. Turning, available at: <https://kompas.ru/kompas-3d/application/machinery/module-chpu> (in Russian).

17. **Module** Computer numerical control. Milling, available at: <https://kompas.ru/kompas-3d/application/machinery/module-chpu-fo> (in Russian).

18. **Shchekin AV.** Design and technological parametrization as a part of an integrated CAM System, *Informatsionnye Tehnologii*, 2019, vol. 25, no. 7, pp. 34–54 (in Russian).

В. Н. Тарасов, д-р техн. наук, проф., зав. каф., e-mail: veniamin_tarasov@mail.ru,
Поволжский государственный университет телекоммуникаций и информатики, г. Самара

Системы массового обслуживания с запаздыванием во времени

Рассмотрены различные системы массового обслуживания (СМО), образованные четырьмя законами распределений вероятностей: экспоненциальным, гиперэкспоненциальным, эрланговским и гиперэрланговским второго порядка. Эти четыре закона образуют шестнадцать различных СМО. В отличие от классической теории в данной статье рассмотрены СМО со сдвинутыми вправо от нулевой точки законами распределений. Такие СМО относятся к типу $G/G/1$, которые являются системами с общими законами распределений интервалов между требованиями входного потока и времени обслуживания. Как известно, для систем $G/G/1$ нельзя получить решения для основной характеристики СМО — среднего времени ожидания — в общем случае. Поэтому важны исследования таких систем для частных случаев законов распределений. В статье приведен обзор авторских результатов для среднего времени ожидания в очереди в замкнутой форме для систем со сдвинутыми вправо от нулевой точки входными распределениями. Для их получения использован метод спектрального разложения решения интегрального уравнения Линдли. Представлены спектральные разложения решения интегрального уравнения Линдли для восьми систем и с их помощью выведены расчетные формулы для среднего времени ожидания в очереди. Показано, что в системах с запаздыванием среднее время ожидания меньше, чем в обычных системах. Полученные расчетные формулы для среднего времени ожидания расширяют и дополняют известную незавершенную формулу теории массового обслуживания для среднего времени ожидания для систем $G/G/1$. Предложенный подход позволяет рассчитать среднее значение и моменты высших порядков времени ожидания для указанных систем в математических пакетах для широкого диапазона изменения параметров трафика. Учитывая тот факт, что вариация задержки пакетов (джиттер) в стандарте по телекоммуникациям определяется как разброс времени ожидания вокруг его среднего значения, джиттер можно будет определить через дисперсию времени ожидания.

Ключевые слова: системы с запаздыванием во времени, сдвинутые законы распределений, преобразование Лапласа, интегральное уравнение Линдли, метод спектрального разложения

Введение

При рассмотрении систем $G/G/1$ будем опираться на метод спектрального разложения решения интегрального уравнения Линдли (ИУЛ), который сыграл большую роль при их исследовании в теории массового обслуживания. Наиболее доступно этот метод на конкретных примерах изложен в классике теории массового обслуживания [1].

Настоящая статья посвящена анализу систем массового обслуживания (СМО) с запаздыванием во времени, т. е. систем со сдвинутыми вправо от нулевой точки входными распределениями.

В работе [2] впервые приведены результаты по исследованию системы $M/M/1$ с запаздыванием во времени со сдвинутыми вправо от нулевой точки экспоненциальными входными распределениями, которую в отличие от классической системы обозначим $M^-/M^-/1$. Эти результаты

получены методом спектрального разложения. Здесь и далее верхний индекс "—" будет означать операцию сдвига закона распределения.

В статье [2] показано, что среднее время ожидания требования в очереди в системе $M^-/M^-/1$ меньше, чем в классической системе $M/M/1$ при одинаковом коэффициенте загрузки за счет того, что коэффициенты вариации времен поступления c_λ и обслуживания c_μ становятся меньше единицы из-за ввода параметра запаздывания $t_0 > 0$. Таким образом, операция сдвига закона распределения трансформирует классическую систему $M/M/1$ в немарковскую систему.

Результаты работы [2] совместно с [1] позволили развить метод спектрального разложения решения (ИУЛ) на системы $H_2^-/H_2^-/1$, $H_2^-/M^-/1$, $M^-/H_2^-/1$, $E_2^-/E_2^-/1$, $E_2^-/M^-/1$, $M^-/E_2^-/1$, $HE_2^-/HE_2^-/1$ с запаздыванием (см. работы [2–4] и другие работы). Все рассматриваемые в статье СМО, образованные четырьмя

наиболее известными законами распределений — M , E_2 , H_2 , HE_2 со сдвинутыми входными распределениями — относятся к типу $G/G/1$.

В теории массового обслуживания исследования систем $G/G/1$ актуальны в связи с тем, что они активно используются в современной теории телетрафика, к тому же нельзя получить решения для таких систем в конечном виде для общего случая. При использовании метода спектрального разложения решения ИУЛ для определения среднего времени ожидания будем придерживаться подхода и символики автора классики теории массового обслуживания Л. Клейнрока [1]. По методике спектрального разложения необходимо найти закон распределения времени ожидания для каждой рассматриваемой системы через спектральное разложение вида

$$A^*(-s)B^*(s) - 1 = \psi_+(s)/\psi_-(s),$$

где $A^*(s)$, $B^*(s)$ — преобразования Лапласа функций плотности распределения интервалов входного потока и времени обслуживания соответственно; $\psi_+(s)$ и $\psi_-(s)$ — некоторые дробно-рациональные функции от s , которые возможно разложить на множители. Эти функции $\psi_+(s)$ и $\psi_-(s)$ должны удовлетворять следующим условиям согласно работе [1]:

— для $\operatorname{Re}(s) > 0$ функция $\psi_+(s)$ является аналитической без нулей в этой полуплоскости;

— для $\operatorname{Re}(s) < D$ функция $\psi_-(s)$ является аналитической без нулей в этой полуплоскости, где D — некоторая положительная константа, определяемая из условия $\lim_{t \rightarrow \infty} a(t)/e^{-Dt} < \infty$.

Кроме того, функции $\psi_+(s)$ и $\psi_-(s)$ должны удовлетворять следующим условиям:

$$\lim_{|s| \rightarrow \infty, \operatorname{Re}(s) > 0} \frac{\psi_+(s)}{s} = 1; \quad \lim_{|s| \rightarrow \infty, \operatorname{Re}(s) < D} \frac{\psi_-(s)}{s} = -1.$$

Метод спектрального разложения решения ИУЛ впервые подробно представлен в классической работе по теории массового обслуживания [1], а впоследствии применялся во многих работах, включая [5, 6]. Аналогичный подход к решению ИУЛ использован в работе [7]. Здесь вместо термина "спектральное разложение" использована факторизация, а вместо функций $\psi_+(s)$ и $\psi_-(s)$ — компоненты факторизации $\omega_+(z, t)$ и $\omega_-(z, t)$ функции $1 - z \cdot \chi(t)$.

В зарубежной литературе в данной области следует выделить работу [8], где представлены результаты приближения очередей к интернету и мобильным сервисам как очередей с запаздыванием во времени, но безотносительно

к СМО. Аппроксимативные методы относительно законов распределений подробно описаны в работах [6, 9], а современные исследования СМО приведены в работах [10, 11].

Постановка задачи

В данной работе ставится задача нахождения решения для основной характеристики — среднего времени ожидания требований в очереди в СМО, образованных четырьмя сдвинутыми вправо от нулевой точки законами распределений: M , E_2 , H_2 , HE_2 . Эти четыре закона распределения образуют $4 \times 4 = 16$ различных СМО типа $G/G/1$, а в данной работе рассмотрены восемь систем.

Для решения поставленной задачи необходимо вначале построить для указанных систем в каждом случае спектральные разложения вида $A^*(-s)B^*(s) - 1 = \psi_+(s)/\psi_-(s)$ с учетом представленных выше условий.

Решение поставленной задачи

Рассмотрим класс функций плотности $f(t)$, преобразуемых по Лапласу, т. е. функций, для которых существует функция

$$F^*(s) = \int_0^\infty e^{-st} f(t) dt \equiv L[f(t)].$$

Далее воспользуемся теоремой запаздывания преобразования Лапласа: для любого $t_0 > 0$ справедливо равенство

$$L[f(t - t_0)] = e^{-st_0} F^*(s), \quad (1)$$

где $\operatorname{Re}(s) > 0$. Функции плотности из табл. 1 относятся к классу функций, преобразуемых по Лапласу.

В работах [2–5] автором с использованием равенства (1) получены спектральные разложения $A^*(-s)B^*(s) - 1 = \psi_+(s)/\psi_-(s)$ и выведены расчетные формулы для среднего времени ожидания для СМО, образованных экспоненциальными и гиперэкспоненциальными законами распределений, коих всего четыре. На основании этих результатов теперь можно сформулировать теорему для СМО, образованных сдвинутыми вправо законами распределений M , E_2 , H_2 , HE_2 , преобразуемых по Лапласу и наиболее часто используемых в теории массового обслуживания.

Теорема. Спектральные разложения решения ИУЛ $A^*(-s)B^*(s) - 1 = \psi_+(s)/\psi_-(s)$ для всех

систем с запаздыванием полностью совпадают со спектральными разложениями для соответствующих обычных систем, т. е. основное выражение спектрального разложения $A^*(-s)B^*(s) - 1 = \psi_+(s)/\psi_-(s)$ инвариантно к операции сдвига во времени функции плотности.

Доказательство. Рассмотрим СМО с запаздыванием, образованные двумя функциями плотности $a(t)$ и $b(t)$, преобразования Лапласа которых обозначим $A^*(s)$ и $B^*(s)$ соответственно. Таким образом, функция плотности распределения интервалов между требованиями входного потока в этом случае имеет вид

$$a(t) = \begin{cases} a(t - t_0), & t \geq t_0; \\ 0, & 0 \leq t < t_0, \end{cases} \quad (2)$$

а для времени обслуживания — вид

$$b(t) = \begin{cases} b(t - t_0), & t \geq t_0; \\ 0, & 0 \leq t < t_0. \end{cases} \quad (3)$$

Функции (2) и (3) являются сдвинутыми вправо от нулевой точки распределениями.

Запишем преобразования Лапласа функций (2) и (3) с учетом свойства (1):

$$L[a(t - t_0)] = e^{-st_0} A^*(s), \quad L[b(t - t_0)] = e^{-st_0} B^*(s).$$

Тогда выражение $A^*(-s)B^*(s) - 1 = \psi_+(s)/\psi_-(s)$ для спектрального разложения решения ИУЛ для системы примет вид

$$\begin{aligned} \psi_+(s)/\psi_-(s) &= e^{t_0 s} A^*(-s) e^{-t_0 s} B^*(s) - 1 = \\ &= A^*(-s) B^*(s) - 1 \end{aligned} \quad (4)$$

У экспонент показатели степени с противоположными знаками обнуляются, и операция сдвига тем самым нивелируется. Правая часть выражения (4) полностью совпадает со спектральным разложением решения ИУЛ для обычной системы.

Теорема доказана.

Следствие. Расчетные формулы для среднего времени ожидания для всех систем со сдвинутыми распределениями будут иметь точно такой же вид, как у соответствующих систем с обычными распределениями, но с измененными параметрами вследствие проведения операции сдвига во времени [2–6]. Таким образом, среднее время ожидания для систем с запаздыванием фактически зависит от параметра сдвига $t_0 > 0$.

Продemonстрируем сказанное выше на примере системы $M^-/M^-/1$. Функция плотности

распределения интервалов между требованиями входного потока в этом случае имеет вид

$$a(t) = \begin{cases} \lambda e^{-\lambda(t-t_0)}, & t \geq t_0; \\ 0, & 0 \leq t < t_0, \end{cases} \quad (5)$$

а для времени обслуживания — вид

$$b(t) = \begin{cases} \mu e^{-\mu(t-t_0)}, & t \geq t_0; \\ 0, & 0 \leq t < t_0. \end{cases} \quad (6)$$

Функции (5) и (6) являются сдвинутыми вправо от нулевой точки экспоненциальными распределениями [2].

Запишем преобразования Лапласа функций (5) и (6) с учетом свойства (1):

$$A^*(s) = \frac{\lambda e^{-t_0 s}}{s + \lambda}, \quad B^*(s) = \frac{\mu e^{-t_0 s}}{s + \mu}.$$

Тогда выражение $A^*(-s)B^*(s) - 1 = \psi_+(s)/\psi_-(s)$ для спектрального разложения решения ИУЛ для системы $M^-/M^-/1$ примет вид

$$\begin{aligned} \frac{\psi_+(s)}{\psi_-(s)} &= \frac{\lambda e^{t_0 s}}{\lambda - s} \cdot \frac{\mu e^{-t_0 s}}{\mu + s} - 1 = \\ &= \frac{\lambda \mu - (\lambda - s)(\mu + s)}{(\lambda - s)(\mu + s)} = \frac{s(s + \mu - \lambda)}{(\lambda - s)(\mu + s)}. \end{aligned} \quad (7)$$

У экспонент показатели степени с противоположными знаками обнуляются, и операция сдвига тем самым нивелируется. Правая часть выражения (7) полностью совпадает со спектральным разложением решения ИУЛ для классической системы $M/M/1$ [1]. Аналогично обстоит дело и со всеми пятнадцатью оставшимися системами.

Далее с учетом условий, приведенных во введении, строим функции $\psi_+(s)$ и $\psi_-(s)$, а с использованием последних определяем преобразование Лапласа функции плотности времени ожидания. Так, для системы $M^-/M^-/1$ имеем:

$$\psi_+(s) = \frac{s(s + \mu - \lambda)}{(\mu + s)}, \quad \psi_-(s) = \lambda - s.$$

Заметим, что функция $\psi_+(s)$ не имеет нулей и полюсов в области $\text{Re}(s) > 0$, а $\psi_-(s)$ не имеет нулей в области $\text{Re}(s) > \lambda$. Действуя по методике спектрального разложения, определяем постоянную K из условия

$$K = \lim_{s \rightarrow 0} \frac{\psi_+(s)}{s} = \lim_{s \rightarrow 0} \frac{s + \mu - \lambda}{s + \mu} = 1 - \lambda/\mu,$$

где параметры λ и μ определяются следующими выражениями через первые начальные моменты распределений (5) и (6):

$$\bar{\tau}_\lambda = \lambda^{-1} + t_0, \quad (8)$$

$$\bar{\tau}_\mu = \mu^{-1} + t_0. \quad (9)$$

В этом случае отношение λ/μ не определяет коэффициент загрузки, как в случае системы М/М/1, и параметры λ и μ не являются интенсивностями поступления требований и обслуживания соответственно.

Преобразование Лапласа для функции распределения времени ожидания (см. работу [1]), имеет вид

$$\Phi_+(s) = \frac{K}{\psi_+(s)} = \frac{(1 - \lambda/\mu)(\mu + s)}{s(s + \mu - \lambda)},$$

а преобразование Лапласа функции плотности времени ожидания — вид

$$W^*(s) = s\Phi_+(s) = \frac{(1 - \lambda/\mu)(\mu + s)}{(s + \mu - \lambda)}, \quad (10)$$

Значение первой производной функции (10) со знаком минус $-\frac{dW^*(s)}{ds}$ при $s = 0$ даст среднее время ожидания для системы М[−]/М[−]/1:

$$\bar{W} = \frac{\lambda/\mu}{\mu - \lambda},$$

что с учетом выражений (8) и (9) окончательно можно записать в виде

$$\bar{W} = \frac{(\bar{\tau}_\mu - t_0)^2}{\bar{\tau}_\lambda - \bar{\tau}_\mu}. \quad (11)$$

Следовательно, решение для среднего времени ожидания для всех систем с запаздыванием будет функцией параметра сдвига t_0 , где $0 < t_0 < \bar{\tau}_\mu$.

Распределения (5) и (6) содержат по два параметра, для их определения методом моментов уравнения (8) и (9) надо дополнить выражениями для начальных моментов второго порядка. В качестве вторых моментов удобнее использовать коэффициенты вариаций:

$$c_\lambda = (1 + \lambda t_0)^{-1}; \quad (12)$$

$$c_\mu = (1 + \mu t_0)^{-1}. \quad (13)$$

Тогда в качестве входных параметров для расчета СМО М[−]/М[−]/1 задаем значения t_0 , $\bar{\tau}_\lambda$, c_λ , $\bar{\tau}_\mu$, c_μ , а неизвестные параметры λ , μ и t_0 определяем из системы уравнений моментов (8), (9), (12) и (13). Эта система уравнений является переопределенной, и входные параметры будут связаны условием

$$c_\mu = 1 - (1 - c_\lambda)/\rho,$$

где $\rho = \bar{\tau}_\mu/\bar{\tau}_\lambda$ — коэффициент загрузки системы.

Аналогичные рассуждения для других систем с запаздыванием приведут к схожим результатам. Для этого сведем в табл. 1 числовые характеристики рассматриваемых законов распределений, которые использованы в работах [2—4].

Числовые характеристики сдвинутых распределений (табл. 1) явно свидетельствуют о су-

Таблица 1

Числовые характеристики распределений

Распределение	Первый начальный момент $\bar{\tau}_\lambda$	Второй начальный момент $\bar{\tau}_\lambda^2$	Квадрат коэффициента вариации c_λ^2
М	$1/\lambda$	$2/\lambda^2$	1
E ₂	$1/\lambda$	$3/(2\lambda^2)$	1/2
H ₂	$p/\lambda_1 + (1-p)/\lambda_2$	$2[p/\lambda_1^2 + (1-p)/\lambda_2^2]$	$\frac{(1-p^2)\lambda_1^2 - 2\lambda_1\lambda_2p(1-p) + p(2-p)\lambda_2^2}{[(1-p)\lambda_1 + p\lambda_2]^2}$
HE ₂	$p/\lambda_1 + (1-p)/\lambda_2$	$3p/(2\lambda_1^2) + 3(1-p)/(2\lambda_2^2)$	$\frac{\lambda_1^2 - 2p\lambda_2(\lambda_1 - \lambda_2) + p(1-2p)(\lambda_1 - \lambda_2)^2}{2[(1-p)\lambda_1 + p\lambda_2]^2}$
М [−]	$\frac{1}{\lambda} + t_0$	$2\left(\frac{1}{\lambda^2} + \frac{t_0}{\lambda}\right) + t_0^2$	$\frac{1}{(1 + \lambda t_0)^2}$
E ₂ [−]	$\frac{1}{\lambda} + t_0$	$\frac{3}{2\lambda^2} + 2\frac{t_0}{\lambda} + t_0^2$	$\frac{1}{2(1 + \lambda t_0)^2}$
H ₂ [−]	$\frac{p}{\lambda_1} + \frac{(1-p)}{\lambda_2} + t_0$	$t_0^2 + 2t_0\left[\frac{p}{\lambda_1} + \frac{(1-p)}{\lambda_2}\right] + 2\left[\frac{p}{\lambda_1^2} + \frac{(1-p)}{\lambda_2^2}\right]$	$\frac{(1-p^2)\lambda_1^2 - 2\lambda_1\lambda_2p(1-p) + p(2-p)\lambda_2^2}{[t_0\lambda_1\lambda_2 + (1-p)\lambda_1 + p\lambda_2]^2}$
HE ₂ [−]	$\frac{p}{\lambda_1} + \frac{(1-p)}{\lambda_2} + t_0$	$t_0^2 + 2t_0\left[\frac{p}{\lambda_1} + \frac{(1-p)}{\lambda_2}\right] + \frac{3p}{2\lambda_1^2} + \frac{3(1-p)}{2\lambda_2^2}$	$\frac{\lambda_1^2 - 2p\lambda_2(\lambda_1 - \lambda_2) + p(1-2p)(\lambda_1 - \lambda_2)^2}{2[t_0\lambda_1\lambda_2 - p(\lambda_1 - \lambda_2) + \lambda_1]^2}$

щественном влиянии на них параметра сдвига t_0 . Теперь необходимо определить неизвестные параметры данных распределений. Эти параметры также получены в работах [2—4] и для случаев функций плотности распределения интервалов входных потоков $a(t)$ сведены в табл. 2.

Аналогичные параметры для распределений времени обслуживания $b(t)$ будут иметь место путем замены λ на μ .

В табл. 3 приведены полученные в работах [2—4] преобразования Лапласа функции плотности времени ожидания в очереди в рас-

Таблица 2

Параметры сдвинутых распределений, полученные методом моментов

Распределение	Плотность $a(t)$	Параметры $\rho, \lambda, \lambda_1, \lambda_2$		
M^-	$\lambda e^{-\lambda(t-t_0)}$	$\lambda = \frac{1}{\bar{\tau}_\lambda - t_0}$		
E_2^-	$4\lambda^2(t-t_0)e^{-2\lambda(t-t_0)}$	$\lambda = \frac{1}{\bar{\tau}_\lambda - t_0}$		
H_2^-	$\rho\lambda_1 e^{-\lambda_1(t-t_0)} + (1-\rho)\lambda_2 e^{-\lambda_2(t-t_0)}$	$\rho = \frac{1}{2} \pm \sqrt{\frac{1}{4} - \frac{(\bar{\tau}_\lambda - t_0)^2}{2[(\bar{\tau}_\lambda - t_0)^2 + c_\lambda^2 \bar{\tau}_\lambda^2]}}$	$\lambda_1 = \frac{2\rho}{(\bar{\tau}_\lambda - t_0)}$	$\lambda_2 = \frac{2(1-\rho)}{(\bar{\tau}_\lambda - t_0)}$
HE_2^-	$4\rho\lambda_1^2(t-t_0)e^{-2\lambda_1(t-t_0)} + 4(1-\rho)\lambda_2^2(t-t_0)e^{-2\lambda_2(t-t_0)}$	$\rho = \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{3(\bar{\tau}_\lambda - t_0)^2}{8[(\bar{\tau}_\lambda - t_0)^2 + c_\lambda^2 \bar{\tau}_\lambda^2]}}$	$\lambda_1 = \frac{2\rho}{(\bar{\tau}_\lambda - t_0)}$	$\lambda_2 = \frac{2(1-\rho)}{(\bar{\tau}_\lambda - t_0)}$

Таблица 3

Преобразования Лапласа функции плотности времени ожидания, компоненты спектральных разложений решения ИУЛ, расчетные выражения для среднего времени ожидания

СМО	Преобразования Лапласа функции плотности времени ожидания и компоненты спектрального разложения	Расчетные выражения для среднего времени ожидания
$M^-/M^-/1$	$W^*(s) = \frac{(1-\lambda/\mu)(\mu+s)}{(s+\mu-\lambda)}, \quad \psi_+(s) = \frac{s(s+\mu-\lambda)}{(\mu+s)}, \quad \psi_-(s) = \lambda-s$	$\bar{W} = \frac{(\bar{\tau}_\mu - t_0)}{(\bar{\tau}_\lambda - \bar{\tau}_\mu)}$
$M^-/E_2^-/1$	$W^*(s) = \frac{(1-\rho)(2\mu+s)^2}{(s+s_1)(s+s_2)}, \quad \psi_+(s) = \frac{s(s+s_1)(s+s_2)}{(2\mu+s)^2}, \quad \psi_-(s) = \lambda-s$	$\bar{W} = \frac{3\rho(\bar{\tau}_\mu - t_0)}{4(1-\rho)}$
$M^-/H_2^-/1$	$W^*(s) = \frac{s_1 s_2 (s+\mu_1)(s+\mu_2)}{\mu_1 \mu_2 (s+s_1)(s+s_2)}, \quad \psi_+(s) = \frac{s(s+s_1)(s+s_2)}{(s+\mu_1)(s+\mu_2)}, \quad \psi_-(s) = \lambda-s$	$\bar{W} = \frac{s_1+s_2}{s_1 s_2} - \frac{\mu_1+\mu_2}{\mu_1 \mu_2}$, где s_1, s_2 — абсолютные значения отрицательных нулей числителя функции $\psi_+(s)/\psi_-(s)$
$E_2^-/E_2^-/1$	$W^*(s) = \frac{s_1 s_2 (2\mu+s)^2}{4\mu^2 s(s+s_1)(s+s_2)}, \quad \psi_+(s) = \frac{s(s+s_1)(s+s_2)}{(2\mu+s)^2}, \quad \psi_-(s) = -\frac{(2\lambda-s)^2}{(s-s_3)}$	$\bar{W} = \frac{s_1+s_2}{s_1 s_2} - \frac{1}{\mu}$, где $s_1 = (\mu-\lambda) + \sqrt{(\mu-\lambda)^2 + 8\lambda\mu}, \quad s_2 = 2(\mu-\lambda)$
$E_2^-/M^-/1$	$W^*(s) = \frac{s_1(s+\mu)}{\mu(s+s_1)}, \quad \psi_+(s) = \frac{s(s+s_1)}{s+\mu}, \quad \psi_-(s) = -\frac{(2\lambda-s)^2}{s-s_2}$	$\bar{W} = 1/s_1 - 1/\mu$, где $s_1 = (\mu-4\lambda)/2 + \sqrt{[(\mu-4\lambda)/2]^2 + 4\lambda(\mu-\lambda)}$
$H_2^-/H_2^-/1$	$W^*(s) = \frac{s_1 s_2 (s+\mu_1)(s+\mu_2)}{\mu_1 \mu_2 (s+s_1)(s+s_2)}, \quad \psi_+(s) = \frac{s(s+s_1)(s+s_2)}{(s+\mu_1)(s+\mu_2)}, \quad \psi_-(s) = \frac{(s-\lambda_1)(\lambda_2-s)}{s-s_3}$	$\bar{W} = \frac{1}{s_1} + \frac{1}{s_2} - \frac{1}{\mu_1} - \frac{1}{\mu_2}$, где s_1, s_2 — абсолютные значения отрицательных нулей числителя функции $\psi_+(s)/\psi_-(s)$
$H_2^-/M^-/1$	$W^*(s) = \frac{s_1(s+\mu)}{\mu(s+s_1)}, \quad \psi_+(s) = \frac{s(s+s_1)}{s+\mu}, \quad \psi_-(s) = \frac{(s-\lambda_1)(\lambda_2-s)}{s-s_2}$	$\bar{W} = 1/s_1 - 1/\mu$, где s_1 — абсолютное значение отрицательного нуля числителя функции $\psi_+(s)/\psi_-(s)$
$HE_2^-/HE_2^-/1$	$W^*(s) = \frac{s_1 s_2 s_3 s_4 (s+2\mu_1)^2 (s+2\mu_2)^2}{16\mu_1^2 \mu_2^2 (s+s_1)(s+s_2)(s+s_3)(s+s_4)}, \quad \psi_+(s) = \frac{s(s+s_1)(s+s_2)(s+s_3)(s+s_4)}{[(s+2\mu_1)^2 (s+2\mu_2)^2]}, \quad \psi_-(s) = -\frac{(2\lambda_1-s)^2 (2\lambda_2-s)^2}{(s-s_5)(s-s_6)(s-s_7)}$	$\bar{W} = \frac{1}{s_1} + \frac{1}{s_2} + \frac{1}{s_3} + \frac{1}{s_4} - \frac{1}{\mu_1} - \frac{1}{\mu_2}$, где s_1, s_2, s_3, s_4 — абсолютные значения отрицательных нулей числителя функции $\psi_+(s)/\psi_-(s)$

сма триваемых системах, компоненты спектральных разложений решения ИУЛ, а также расчетные формулы для среднего времени ожидания в соответствующих системах.

С подробным изложением алгоритмов расчета среднего времени ожидания для рассматриваемых систем, можно ознакомиться в работах [2—4]. Таким образом, здесь представлены опубликованные результаты по восьми системам из шестнадцати.

Таблица 4

Результаты экспериментов для СМО $M^-/M^-/1$ и $M/M/1$

Входные параметры				Среднее время ожидания	
ρ	t_0	c_λ	c_μ	для системы $M^-/M^-/1$	для системы $M/M/1$
0,1	0,9	0,91	0,1	0,001	0,111
	0,5	0,95	0,5	0,028	
	0,1	0,99	0,9	0,090	
	0,01	0,999	0,99	0,109	
0,5	0,9	0,55	0,1	0,010	1,000
	0,5	0,75	0,5	0,250	
	0,1	0,95	0,9	0,810	
	0,01	0,995	0,99	0,980	
0,9	0,9	0,19	0,1	0,090	9,000
	0,5	0,55	0,5	2,250	
	0,1	0,91	0,9	7,290	
	0,01	0,991	0,99	8,821	

Таблица 5

Результаты экспериментов для СМО $E_2^-/E_2^-/1$ и $E_2/E_2/1$

Входные параметры				Среднее время ожидания	
ρ	c_λ	c_μ	t_0	для системы $E_2^-/E_2^-/1$	для системы $E_2/E_2/1$
0,1	0,643	0,071	0,9	0,000	0,017
	0,672	0,354	0,5	0,002	
	0,700	0,636	0,1	0,013	
	0,706	0,700	0,01	0,016	
0,5	0,389	0,071	0,9	0,001	0,390
	0,530	0,354	0,5	0,081	
	0,672	0,636	0,1	0,309	
	0,704	0,700	0,01	0,382	
0,9	0,134	0,071	0,9	0,034	4,359
	0,389	0,354	0,5	1,057	
	0,643	0,636	0,1	3,519	
	0,701	0,700	0,01	4,271	

Результаты вычислительных экспериментов

В табл. 4—7 приведены данные расчетов в пакете MathCAD для четырех характерных систем $M^-/M^-/1$, $E_2^-/E_2^-/1$, $H_2^-/H_2^-/1$ и $HE_2^-/HE_2^-/1$ для случаев малой, средней и вы-

Таблица 6

Результаты экспериментов для СМО $H_2^-/H_2^-/1$ и $H_2/H_2/1$

Входные параметры		Среднее время ожидания			
ρ	(c_λ, c_μ)	Для системы $H_2^-/H_2^-/1$			Для системы $H_2/H_2/1$
		$t_0 = 0,9$	$t_0 = 0,5$	$t_0 = 0,1$	
0,1	(1,1)	0,06	0,07	0,10	0,11
	(2,2)	0,28	0,36	0,42	0,45
	(4,4)	1,19	1,54	1,73	1,78
	(8,8)	4,81	6,31	6,97	7,11
0,5	(1,1)	0,56	0,75	0,95	1,00
	(2,2)	2,31	3,13	3,87	4,04
	(4,4)	9,29	12,61	15,45	16,13
	(8,8)	37,22	50,50	61,54	64,18
0,9	(1,1)	6,04	8,30	8,91	9,00
	(2,2)	24,14	33,22	35,84	36,20
	(4,4)	96,51	132,30	143,27	144,83
	(8,8)	386,03	527,68	571,47	577,86

Таблица 7

Результаты экспериментов для СМО $HE_2^-/HE_2^-/1$ и $HE_2/HE_2/1$

Входные параметры		Среднее время ожидания			
ρ	(c_λ, c_μ)	Для системы $HE_2^-/HE_2^-/1$			Для системы $HE_2/HE_2/1$
		$t_0 = 0,9$	$t_0 = 0,5$	$t_0 = 0,1$	
0,1	(0,71;0,71)	0,01	0,01	0,02	0,02
	(2;2)	0,29	0,32	0,33	0,34
	(4;4)	1,21	1,55	1,66	1,68
	(8;8)	4,93	6,48	7,05	7,16
0,5	(0,71;0,71)	0,27	0,31	0,39	0,40
	(2;2)	2,32	3,15	3,82	3,98
	(4;4)	9,35	12,94	15,85	16,53
	(8;8)	37,50	52,13	63,96	66,73
0,9	(0,71;0,71)	3,06	4,11	4,39	4,40
	(2;2)	24,42	33,36	35,88	36,21
	(4;4)	97,71	133,3	143,8	145,31
	(8;8)	390,90	532,7	574,5	580,56

сокой нагрузки $\rho = 0,1; 0,5; 0,9$ для широкого диапазона изменения коэффициентов вариаций c_λ , c_μ и параметра сдвига t_0 . Результаты для систем с запаздыванием сравниваются с результатами для обычных систем. Коэффициент загрузки ρ в табл. 4–7 определяется отношением средних интервалов $\rho = \bar{\tau}_\mu / \bar{\tau}_\lambda$. В расчетах для удобства использовано нормированное время обслуживания $\bar{\tau}_\mu = 1$.

Как и следовало ожидать, уменьшение коэффициентов вариации c_λ и c_μ за счет введения параметра сдвига $t_0 > 0$ в законы распределения входного потока и времени обслуживания влечет за собой уменьшение среднего времени ожидания в системах с запаздыванием в несколько раз.

Адекватность представленных результатов, с одной стороны, подтверждается тем, что при стремлении параметра сдвига t_0 к нулю среднее время ожидания в системе с запаздыванием стремится к его значению в обычной системе. С другой стороны, адекватность представленных математических моделей систем с запаздыванием достигается корректным использованием классического метода спектрального разложения ИУЛ для рассмотренных систем.

Заключение

В работе представлены спектральные разложения решения интегрального уравнения Линдли для восьми систем с запаздыванием, с помощью которых выведены расчетные формулы для среднего времени ожидания в очереди для этих систем в замкнутой форме. С использованием данных формул можно рассчитать среднее время ожидания во всех рассмотренных системах.

Операция сдвига во времени для всех систем с запаздыванием, с одной стороны, приводит к увеличению загрузки системы с запаздыванием. Например, для системы с запаздыванием $M^-/M^-/1$ загрузка увеличивается в $(1 + \mu t_0)/(1 + \lambda t_0)$ раз по сравнению с обычной системой $M/M/1$. С другой стороны, операция сдвига во времени уменьшает коэффициенты вариаций интервала между поступлениями и времени обслуживания требований. В связи с тем, что среднее время ожидания в системе $G/G/1$ связано с коэффициентами вариаций интервалов поступления и обслуживания квадратичной зависимостью, среднее время ожидания в системе с запаздыванием будет меньше, чем в обычной системе при одинаковом

коэффициенте загрузки. Например, для системы $M^-/M^-/1$ при загрузке $\rho = 0,9$ и параметре сдвига $t_0 = 0,9$ коэффициент вариации интервалов поступления c_λ уменьшается с 1 для обычной системы до 0,19, коэффициент вариации времени обслуживания c_μ уменьшается с 1 до 0,1, а время ожидания уменьшается с 9 единиц времени для обычной системы почти до 0,09 единиц времени для системы с запаздыванием (табл. 4).

Кроме того, введение параметра сдвига приводит к достаточно широкому диапазону изменения коэффициентов вариаций c_λ и c_μ в отличие от обычных систем, которые применимы только в случае фиксированных значений коэффициентов вариаций. Следовательно, системы с запаздыванием расширяют диапазон их применимости в современной теории телетрафика.

Список литературы

1. Клейнрок Л. Теория массового обслуживания. Пер. с англ. под редакцией В. И. Неймана. М.: Машиностроение, 1979. 432 с.
2. Тарасов В. Н., Бахарева Н. Ф., Блатов И. А. Анализ и расчет системы массового обслуживания с запаздыванием // Автоматика и телемеханика. 2015. № 11. С. 51–59.
3. Тарасов В. Н. Расширение класса систем массового обслуживания с запаздыванием // Автоматика и телемеханика. 2018. № 12. С. 57–70.
4. Тарасов В. Н. Анализ и сравнение двух систем массового обслуживания с гиперэрланговскими входными распределениями // Радиоэлектроника, информатика, управление. 2018. № 4 (47). С. 61–70.
5. Brannstrom N. A Queueing Theory analysis of wireless radio systems // Applied to HS-DSCH. Lulea university of technology, 2004. 79 p.
6. Whitt W. Approximating a point process by a renewal process: two basic methods // Operation Research. 1982. N. 1. P. 125–147.
7. Бочаров П. П., Печинкин А. В. Теория массового обслуживания. М.: Изд-во РУДН, 1995. 529 с.
8. Novitzky S., Pender J., Rand R. H., Wesson E. Nonlinear Dynamics in Queueing Theory: Determining the Size of Oscillations in Queues with Delay // SIAM J. Appl. Dyn. Syst. 2019. Vol. 18, N. 1. P. 279–311.
9. Алиев Т. И. Аппроксимация вероятностных распределений в моделях массового обслуживания // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 2 (84). С. 88–93.
10. Adan I., D'Auria B., Kella O. Special volume on 'Recent Developments in Queueing Theory' of the third ECQT conference // Queueing Systems. 2019. N. 1. P. 1–190. DOI: <https://doi.org/10.1007/s11134-019-09630-1>.
11. Legros B. M/G/1 queue with event-dependent arrival rates // Queueing Systems. 2018. Vol. 89, N. 3. P. 269–301. DOI: <https://doi.org/10.1007/s11134-017-9557-7>.
12. Bazhba M., Blanchet J., Rhee C. H. et al. Queue with heavy-tailed Weibull service times // Queueing Systems. 2019. Vol. 93, N. 11. P. 1–32. DOI: <https://doi.org/10.1007/s11134-019-09640-z>.

Queuing Systems with a Time Lag

The article discusses various queuing systems (QS) formed by four laws of probability distributions: exponential, hyperexponential, Erlang and hyper-Erlang of the second order. These four laws form sixteen different QS. In contrast to the classical theory, this article considers QS with distribution laws shifted to the right from the zero point. Such QS are of type G/G/1 with arbitrary laws of the distribution of intervals between the requirements of the input flow and the service time. As you know, for such systems it is impossible to obtain solutions for the main characteristic of QS – the average waiting time in the general case. Therefore, studies of such systems are important for special cases of distribution laws. The article provides an overview of the author's results for the average waiting time in a queue in a closed form for systems with input distributions shifted to the right from the zero point. To solve this problem, the spectral decomposition method for solving the Lindley integral equation was used. In the course of solving the problem, spectral decompositions of the solution of the Lindley integral equation for eight systems were obtained and with their help calculation formulas were derived for the average waiting time in the queue. It is shown that in systems with delay, the average waiting time is shorter than in conventional systems. The obtained calculation formulas for the average waiting time expand and complement the well-known incomplete formula of the queuing theory for the average waiting time for G/G/1 systems. The proposed approach allows us to calculate the average value and moments of higher orders of waiting time for these systems in mathematical packages for a wide range of changes in traffic parameters. Given the fact that the variation in packet delay (jitter) in the telecommunications standard is defined as the spread of waiting time around its average value, the jitter can be determined through the variance of the waiting time.

Keywords: time delay system, shifted distributions, Laplace transform, Lindley integral equation, spectral decomposition method

DOI: 10.17587/it.27.291-298

References

1. Kleinrock L. Queuing theory, Moscow, Mashinostroenie Publ, 1979, 432 p.
2. Tarasov V. N., Bakhareva N. F., Blatov I. A. Analysis and calculation of queuing system with delay, *Automation and Remote Control*, 2015, vol. 52, no. 11, pp. 1945–1951/ DOI: 10.1134/S0005117915110041.
3. Tarasov V. N. Extension of the Class of Queueing Systems with Delay, *Automation and Remote Control*, 2018, vol. 79, no. 12, pp. 2147–2157. DOI: 10.1134/S0005117918120056.
4. Tarasov V. N. Analysis and comparison of two queueing systems with hypererlangian input distributions, *Radio Electronics Computer Science Control*, 2018, vol. 47, no. 4, pp. 61–70. DOI 10.15588/1607-3274-2018-4-6.
5. Brannstrom N. A. Queueing Theory analysis of wireless radio systems, *Applied to HS-DSCH*, Lulea university of technology, 2004, 79 p.
6. Whitt W. Approximating a point process by a renewal process: two basic methods, *Operation Research*, 1982, vol. 30, no. 1, pp. 125–147.
7. Bocharov P. P., Pechinkin A. V. Teoriya massovogo obsluzhivaniya, Moscow, Publishing House of Peoples' Friendship University, 1995, 529 p.
8. Novitzky S., Pender J., Rand R. H., Wesson E. Nonlinear Dynamics in Queueing Theory: Determining the Size of Oscillations in Queues with Delay, *SIAM J. Appl. Dyn. Syst.*, 18-1 2019, vol. 18, no. 1, pp. 279–311. DOI: <https://doi.org/10.1137/18M1170637>.
9. Aliev T. I. Approximatsiya veroyatnostnykh raspredelenij v modelyakh massovogo obsluzhivaniya, *Nauchno-tekhnicheskij vestnik informacionnykh tekhnologij, mekhaniki i optiki*, 2013, vol. 84, no. 2, pp. 88–93.
10. Adan I., D'Auria B., Kella O. Special volume on 'Recent Developments in Queueing Theory' of the third ECQT conference, *Queueing Systems*, 2019, pp. 1–190. DOI: <https://doi.org/10.1007/s11134-019-09630-1>.
11. Legros B. M/G/1 queue with event-dependent arrival rates, *Queueing Systems*, 2018, vol. 89, no. 3, pp. 269–301. DOI: <https://doi.org/10.1007/s11134-017-9557-7>.
12. Bazhba M., Blanchet J., Rhee CH. et al. Queue with heavy-tailed Weibull service times, *Queueing Systems*, 2019, vol. 93, no. 11, pp. 1–32. DOI: <https://doi.org/10.1007/s11134-019-09640-z>.

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 004.932.2, 681.516.7.015.2

DOI: 10.17587/it.27.299-305

Ш. С. Фахми, д-р техн. наук, проф., e-mail: Shakeebf@mail.ru,
Санкт-Петербургский государственный электротехнический университет
им. В. И. Ульянова (Ленина) "ЛЭТИ",
Институт проблем транспорта им. Н. С. Соломенко Российской академии наук,
А. Г. Давидчук, канд. техн. наук, доц., e-mail: davag@bk.ru,
Санкт-Петербургский государственный электротехнический университет
им. В. И. Ульянова (Ленина) "ЛЭТИ",
Е. В. Костикова, канд. тех. наук, доц., e-mail: kostikova.ev@mail.ru,
Государственный университет морского и речного флота им. адмирала С. О. Макарова

Новые алгоритмы сжатия без потерь транспортных сюжетов

Обсуждаются актуальность развития алгоритмов сжатия и передачи изображений без потерь и применение их для создания транспортных видеотрансмиссионных систем наблюдения. Приведен краткий обзор методов сжатия изображений без потерь. Предлагается новый подход к сжатию транспортных сюжетов на основе пирамидально-рекурсивного метода разбиения исходного изображения на полигоны различной формы и размера. Рассмотрены три новых алгоритма реализации предложенного метода: а) разбиение на квадраты (либо на прямоугольники); б) деление на треугольники; в) итерационный алгоритм триангуляции Делоне (ТД). Приведен анализ результатов тестирования различных известных алгоритмов сжатия без потерь (кодирование длин серии, Хаффмана и арифметическое) и сравнение их с предложенными алгоритмами. Показано, что предложенные алгоритмы эффективнее с точки зрения степени сжатия (в 2...3 раза) по сравнению с известными при этом вычислительная сложность увеличивается более чем в 3...4 раза.

Ключевые слова: сжатие, восстановление, степень сжатия, сложность алгоритмов, полигонально-рекурсивный метод

Введение

Существует множество разнообразных методов сжатия изображений, и каждый из них характеризуется собственным механизмом сжатия для достижения наименьшей средней длины кода, времени кодирования и декодирования и при этом обеспечивает большой коэффициент сжатия и сохранение приемлемого качества восстановленных изображений.

Алгоритмы сжатия изображений широко применяются в медицинской визуализации, компьютерной связи, военной связи с помощью радаров, в телеконференциях, магнитно-резонансной томографии, широкоэкранным телевидении и спутниковых изображениях, описанных в работе [1]. Некоторые приложения требуют высококачественной визуальной информации, а другие требуют меньшего качества [2, 3]. Следовательно, алгоритмы сжатия изображений с точки зрения качества восстановления визу-

альных данных делятся на два типа: без потерь и с потерями. Другими словами, в режиме без потерь все первичные данные восстанавливаются из кодированного набора данных с точностью до нулевого значения ошибки, тогда как методы с потерями частично извлекают данные, устраняя избыточную информацию [4].

Методы сжатия без потерь в основном используются в факсимильных передачах битовых изображений, ZIP-файлов, цифровых медицинских изображений, Интернет-телефонии и передаче высококачественной видеотрансмиссионной информации [5].

1. Постановка задачи исследования

Глобальный мобильный трафик в 2019 г. вырос на 30,6 % относительно 2017 г., а в настоящем сегменте он упал на 3,3 %, в то время как трафик мобильных данных вырос в 18 раз за по-

следние 5 лет [6]. Мобильный трафик изображений и видеоконтента включает в себя большую часть общих мобильных передаваемых данных. Facebook, Instagram, Netflix и YouTube ежедневно генерируют, хранят и передают огромное количество визуального контента. В то же время пользователи все чаще ожидают более высокого качества визуальных данных, что создает значительные проблемы.

Прием, сжатие и передача изображений транспортных сюжетов (ТС), в частности видеопоток с изображением транспортных средств с различной степенью интенсивности движения, представляется наиболее острой проблемой с точки зрения оперативного реагирования и передачи видеoinформации, полученной из различных камер наблюдения, в реальном времени.

Цифровые изображения представляют собой набор отсчетов, характеризующихся амплитудой (яркостью) и геометрическими пространственными координатами. Пиксели организованы в виде двумерного массива (строк и столбцов), который образует двумерное изображение [7]. Доступность и простота использования цифровых камер приобрела массовый характер благодаря совершенствованию технологий и цифровизации процесса захвата, хранения и передачи изображений. Изображения, полученные цифровыми камерами, могут быть различного формата и объема, и поэтому процесс анализа, хранения или передачи этих изображений всегда был серьезным препятствием для осуществления различных операций обработки в реальном времени. Кроме того, в работе [7] приведен обзор известных методов сжатия без потерь: кодирование длин серии, кодирование по Хаффману, арифметическое кодирование, где отмечено, что лучшее сжатие достигается методом арифметического кодирования. При этом сохраняется первоначальное (или приемлемое) качество восстановленных изображений с учетом ограничений пропускной способности канала.

Таким образом, сжатие без потерь — это технология, используемая для уменьшения избыточности видеoinформации в целях минимизации битовой скорости передачи сжатых визуальных данных при заданной точности восстановления и пропускной способности канала [8].

В данной работе ставится задача достижения более высокого коэффициента сжатия без потерь по сравнению с известными с учетом специфики ТС, заключающейся в следующем (рис. 1, см. третью сторону обложки):

- 1) наличие однородных областей с высокой степенью корреляции, представляющих фон;
- 2) переменная интенсивность движения транспортных средств;
- 3) геометрические (анизотропия) свойства объекта — транспортных средств.

2. Краткий обзор методов сжатия без потерь

Методы сжатия изображений делятся на два класса.

2.1. Алгоритмы сжатия с потерями

Сжатие с потерями, как указано в названии, приводят к некоторой потере информации в зависимости от заданной степени сжатия и точности восстановленных изображений. При этом заданная точность определяется целью поставленной прикладной задачи с учетом ограничения пропускной способности канала в условиях шума и помех.

Сжатие с потерями чаще всего используется для передачи реалистических сюжетов мультимедийных данных. Существуют следующие основные методы сжатия с потерями:

- а) сжатие с предсказанием [9];
- б) сжатие с преобразованием [5];
- в) фрактальное сжатие [10];
- г) пространственное сжатие [11, 12].

2.2. Алгоритмы сжатия без потерь

Сжатие без потерь — это сжатие, при котором после распаковки изображение остается таким же, как исходное изображение. Сжатие данных без потерь, скорее всего, использует статистическую избыточность для более точного преобразования данных без потери информации.

Существуют следующие основные методы кодирования без потерь [7]:

- а) кодирование длин серий (англ. run-length encoding, RLE);
- б) кодирование Хаффмана;
- в) арифметическое кодирование и др.

Кодирование длин серий (КДС). Данный метод кодирования является одним из самых простых методов сжатия видеоданных. Принцип RLE заключается в использовании повторяющихся значений яркостей пикселей в исходном изображении.

Алгоритм подсчитывает число последовательных повторений значений яркостей соседних пикселей, и затем исходное изображение

кодируется на основе полученных числа повторений и самих значений следующим образом.

Последовательность повторяющихся значений яркостей пикселей заменяется последовательностью из числа повторов яркости и значения яркости.

Например, строка, состоящая из следующих яркостей "28_28_28_28_28_28", требующая для хранения шесть байтов (при условии, что на хранение одного значения отводится один байт), можно заменить строкой "6_28", состоящей из двух байтов. Очевидно, что этот алгоритм тем эффективнее, чем длиннее серия повторов. Данный способ сжатия изображений является наиболее эффективным для тех видеоданных, где имеется много таких повторов, например, простых графических изображений, линейных рисунков, анимации и т. п. Однако основным недостатком этого алгоритма является его крайне низкая эффективность на последовательностях неповторяющихся яркостей. Например, если рассмотреть последовательность "28_233_28_233_28_233_28_233" (8 байтов), то после применения алгоритма RLE она превратится в "1_28 1_233 1_28 1_233 1_28 1_233 1_28 1_233" (16 байтов).

Кодирование Хаффмана (КХ). КХ было предложено доктором Дэвидом А. Хаффманом в 1952 г. и было названо "методом построения минимального избыточного кода". КХ заключается в том, что появление каждой яркости рассматривается оптимальным способом как двоичная строка. Другими словами, идея, положенная в основу КХ, основана на использовании частоты появления символа в последовательности и замене кодов фиксированной длины кодами переменной длины, присвоении более коротких кодовых слов встречающимся с большей частотой символам и уменьшении таким образом общей длины данных. При использовании кодовых слов переменной длины желательно создать префикс-код, избегая необходимости в разделителе для определения границ кодового слова. В этом состоит отличительная особенность кода Хаффмана [13].

КХ — это форма статистического кодирования, которая уменьшает число битов, необходимых для представления строки символов. Алгоритм Хаффмана прост и может быть описан в терминах создания дерева кода Хаффмана. Алгоритм КХ включает следующие этапы:

1. Прием входного списка свободных узлов — значений яркостей пикселей. Каждый узел имеет вес, который может быть равен

либо вероятности, либо числу вхождений узла в сжимаемое сообщение.

2. Выбираются два свободных узла дерева с минимальными значениями весов.

3. Формируется их родитель с их суммарным весом, далее добавляется в список, и два его потомка удаляются из этого списка.

4. Одной дуге, выходящей из родителя, присваивается бит 1, а другой — бит 0.

5. Переход к п. 2. Далее повторяется цикл до тех пор, пока в списке свободных узлов останется только один свободный узел. Он и будет считаться корнем дерева.

Алгоритм, представленный выше, называется полуадаптивным или полустатическим КХ, поскольку он требует знания частот для каждого пикселя изображения. Вместе со сжатым выводом должно быть сохранено дерево Хаффмана с кодами Хаффмана для символов или просто частоты пикселей, которые используются для создания дерева Хаффмана. Эта информация необходима в процессе декодирования и помещается в заголовок сжатого файла.

Арифметическое кодирование (АК). АК является мощным инструментом для статического кодирования без потерь и получило наибольшее распространение в течение нескольких лет. В АК вместо кодирования каждого пикселя изображения (символа) по отдельности всей последовательности изображений присваивается одно арифметическое кодовое слово. Определяется кодовое слово из интервала от 0 до 1 (0, 1). Результатом АК является одно число, меньшее 1 и большее или равное 0. Это единственное число может быть уникально декодировано, чтобы создать точный поток требуемых символов, которые вошли в его конструкцию. Для построения выходного числа символов задается множество вероятностей.

3. Предлагаемый метод сжатия без потерь

В данной статье предложен новый пирамидально-рекурсивный метод (ПРМ) сжатия на основе пирамидально-рекурсивного разбиения исходного изображения на полигоны различной формы и размера, при этом алгоритм принципиально отличается от известных рассмотренных выше трех методов тем, что применяется векторное квантование. Исходное изображение разбивается на полигоны, и формируется полигональная сетка путем анализа перепадов яркостей соседних точек, а затем осуществляется поиск опорных точек (ОТ) в каждом полигоне для кодирования и передачи по каналу связи.

Для восстановления результирующего изображения используется известный алгоритм триангуляция Делоне (ТД) [14].

Описание предложенного алгоритма. В данной статье предлагаются три алгоритма для реализации предложенного метода:

- 1) разбиение на квадраты (либо на прямоугольники);
- 2) деление на треугольники;
- 3) итерационный алгоритм триангуляции Делоне (ТД).

Суть этих трех вариантов алгоритмов ПРМ заключается в следующем. Исходное изображение подвергается разбиению на полигоны в зависимости от степени различия яркостей пикселей (заданного порога по яркости). Процесс разбиения продолжается до тех пор, пока размер полигона не будет равным одному пикселю.

Разбиение на квадраты или прямоугольники. На этапе сжатия изображений алгоритм включает следующие основные шаги:

- 1) разбиение изображения на полигоны (квадраты или прямоугольники) в зависимости от порога отклонения по яркости и формирование неравномерной сетки полигонов (N_{Pol} — число полигонов);
- 2) нахождение ОТ в пределах каждого полигона путем выделения границ объектов: ОТ могут быть фиксированные (вершины прямоугольников) либо произвольные и имеют произвольные координаты в пределах полигона. Подробное описание алгоритмов нахождения ОТ можно найти в работе авторов [12, 15];
- 3) формирование динамических массивов ОТ;
- 4) передача по каналу связи.

На этапе восстановления изображений алгоритм включает следующие основные шаги:

- 1) прием сжатых динамических массивов ОТ;
- 2) выполнение ТД [14];
- 3) аппроксимация яркостей пикселей в пределах каждого треугольника;
- 4) получение результирующего изображения в соответствии с заданной минимальной ошибкой, обеспечивающей восстановление без потерь.

Алгоритм разбиения на треугольники. На этапе сжатия изображений алгоритм включает следующие основные шаги (рис. 2, см. третью сторону обложки):

- 1) разбиение изображения на треугольники в зависимости от заданного порога отклонения по яркости. Для достижения сжатия без потерь необходимо задать порог равным единице;
- 2) в качестве ОТ принимаются вершины треугольников, полученных в результате разбиения;

3) формирование динамических массивов ОТ, содержащих информацию о структуре пирамиды и координатах ОТ каждого треугольника;

- 4) передача по каналу связи.

На этапе восстановления изображений алгоритм включает следующие основные шаги:

- 1) прием сжатых динамических массивов ОТ;
- 2) аппроксимация яркостей пикселей в пределах каждого треугольника;
- 4) получение результирующего изображения в соответствии с заданной минимальной ошибкой, обеспечивающей восстановление без потерь.

Итерационный алгоритм ТД. Данный алгоритм является наиболее сложным с точки зрения вычислительных затрат, но с точки зрения коэффициента сжатия является наиболее эффективным и заключается в следующем.

Вначале угловые точки исходного изображения принимаются в качестве ОТ для построения первых двух начальных треугольников. Далее построение ТД в целом происходит по принципу добавления по одной точке в уже имеющуюся триангуляцию до тех пор, пока не будет получена окончательная треугольная сетка, в результате закраски которой получаем результирующее изображение с заданной точностью (рис. 3, см. третью сторону обложки).

Важно отметить, что поиск ОТ выполняется с применением градиента и фиксации границ объектов итерационным способом с помощью различных способов обхода соседних пикселей [12, 15].

В качестве основы для построения триангуляционной сетки используется известный алгоритм ТД. Следует отметить, что триангуляция набора точек будет ТД, если описанная окружность для каждого треугольника будет свободна от точек, т. е. внутри нее не будет больше ни одной точки из набора. Если внутрь любой области, ограниченной окружностью, попала одна точка другого треугольника, то эта триангуляция не относится к типу Делоне, и следовательно, необходимо флипповать (флипп — это замена одной диагонали на другую).

Преимущество всех алгоритмов АПП заключается в следующем:

- 1) сохранение семантических свойств изображений в виде множества ОТ;
- 2) возможность постепенного восстановления во время передачи сжатых файлов, что позволяет сохранить целостность при возникновении сбоев или обрыва передачи по каналам связи;
- 3) выбор того или иного алгоритма из предложенных в зависимости от цели прикладной задачи.

4. Сравнительный анализ алгоритмов сжатия без потерь

Исследование алгоритмов сжатия без потерь проводилось в Институте проблем транспорта РАН для изображений ТС различных форматов и с разными разрешениями с использованием VC++ (рис. 4). Для оценки эффективности алгоритмов были вычислены следующие показатели:

1) степень сжатия — это отношение объема исходного изображения в байтах к объему сжатого изображения;

2) сложность алгоритмов. Сложность алгоритма определяется числом арифметических

операций на пиксель (т.е. сложение, вычитание, умножение и деление), затрачиваемых на выполнение кодирования и декодирования изображения.

Изображения были выбраны из компактной представительной выборки транспортных сюжетов, полученной в результате классификации транспортных сюжетов [16] по мере неопределенности (M) и мере широкополосности (W), что позволяло разделить изображения на девять классов, из которых были выбраны три изображения:

1) изображение № 1 — изображение с малой широкополосностью и большой неопределенностью;

2) изображение № 2 — изображение с средней широкополосностью и средней неопределенностью;

3) изображение № 3 — изображение с большой широкополосностью и большой мерой неопределенности.

Согласно проведенному сравнительному анализу между алгоритмами сжатия изображений ТС можно сделать следующие выводы (см. таблицу):

1. КДС генерирует выходные данные, которые в два раза больше, чем размер входных данных. Это связано с меньшим числом значений яркостей в исходном файле. Другими словами, кодирование эффективно только в том случае, если существуют последовательности из 4 или более повторяющихся символов, поскольку используются три символа, и поэтому кодирование двух повторяющихся символов приведет даже к увеличению размера файла. Если в исходном файле отсутствуют повторения, то его размер, как минимум, удвоится за счет добавления управляющего символа и информации о числе повторений.

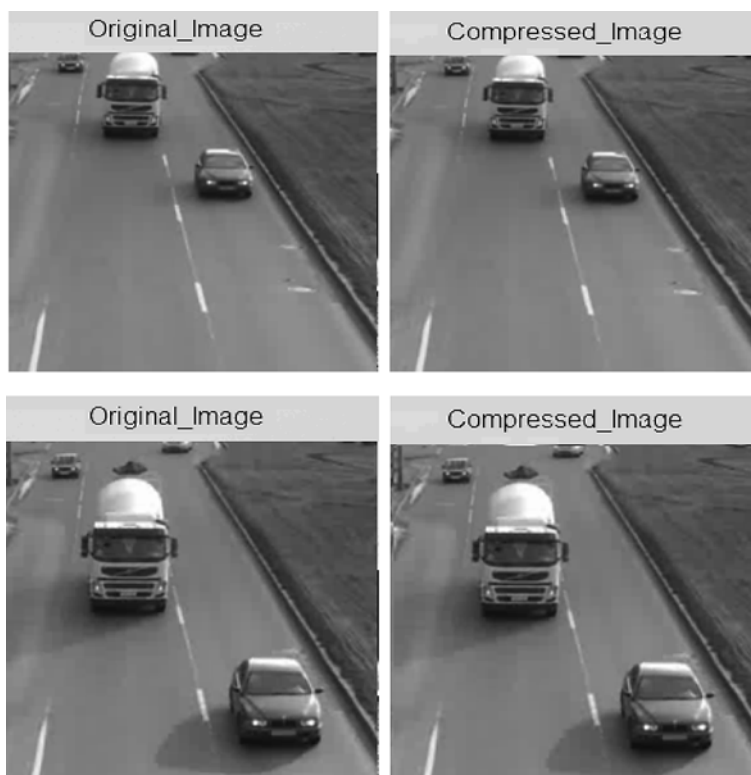


Рис. 4. Оригинальные (слева) и сжатые (справа) изображения

Результаты сравнения алгоритмов сжатия без потерь

Метод сжатия	Изображение № 1	Изображение № 2	Изображение № 3	Область применения
	Степень сжатия*/Сложность* (Операции/ пиксель)			
КДС	1,04/7,4	1,02/6,3	1,02/6,4	Используется в основном для часто встречающихся последовательностей пикселей
КХ	1,57/7,3	1,14/7,6	1,109/7,5	JPEG
АК	1,78/11,3	1,5/10,3	1,58/9,4	JPEG, TIFF и GIF файлы
АКП	1,8/13,2	1,6/11,6	1,37/11,4	Транспортные видеосистемы
Итерационный алгоритм ТД	2,19/24,1	18,6	1,93/15,3	Транспортные видеосистемы
Предложенные алгоритмы	1,83/14,2	1,73/12,9	1,64/13,7	Системы интеллектуального наблюдения

2. КХ по сравнению с АК использует статическую таблицу для всего процесса кодирования, поэтому он быстрее. Однако он не позволяет достичь нужной степени сжатия. Напротив, АК может генерировать высокую степень сжатия, но его скорость сжатия медленнее.

3. ПРМ является наиболее эффективным способом кодирования без потерь по сравнению со всеми алгоритмами по двум причинам:

- имеет улучшенный показатель степени сжатия при сохранении качества изображений;
- имеет возможность управления точностью восстановления сжатых изображений путем задания соответствующего порога яркости (коэффициента корреляции).

Однако преимущество предлагаемого метода сжатия без потерь достигается ценой трехкратного и более увеличения сложности кодера.

4. Алгоритм на основе адаптивного косинусного преобразования (АКП) [5] является наиболее оптимальным с точки зрения вычислительных затрат, в то время как итерационный триангуляционный алгоритм позволяет достичь максимального сжатия при минимальной ошибке восстановления результирующих изображений.

Заключение

Сжатие изображений является важной областью исследований в связи с широким спектром их применения. В этой статье были исследованы и протестированы различные методы сжатия без потерь изображений ТС. Показана эффективность алгоритмов сжатия: кодирование по ОТ без перехода в спектральную область и адаптивное сжатие с переходом в спектральную область с применением косинусного преобразования сигнала изображений.

Проведено сравнение предлагаемых методов с известными методами сжатия без потерь путем оценки степени сжатия и сложности алгоритмов при сохранении качества восстановленных изображений.

Показано, что предлагаемые методы обеспечивают большую степень сжатия ценой увеличения вычислительной сложности, выраженной числом операций на пиксель.

Список литературы

1. Ding J., Furgeson J. C., Sha E. H. Application specific image compression for virtual conferencing // Proceedings of the

International Conference on Information Technology: Coding and Computing (Cat. No.PR00540), Las Vegas, NV, USA, 27–29 March 2000. P. 48–53. doi:10.1109/ITCC.2000.844182.

2. Bhavani S., Thanushkodi K. A survey on coding algorithms in medical image compression // Int. J. Comput. Sci. Eng. 2010. N. 2. P. 1429–1434.

3. Kharate G. K., Patil V. H. Color Image Compression Based on Wavelet Packet Best Tree. arXiv 2010, arXiv:1004.3276.

4. Haque M. R., Ahmed F. Image Data Compression with JPEG and JPEG2000. URL: http://eeweb.poly.edu/~yao/EE3414_S03/Projects/Loginova_Zhan_ImageCompressing_Rep.pdf (accessed on 1 October 2019).

5. Фахми Ш. С., Рыжов Н. Г., Хасан Я. А., Калинина Е. С. Адаптивный алгоритм кодирования и декодирования изображений в транспортных системах наблюдения // Информационные технологии. 2019. Т. 25. № 8. С. 475–481.

6. URL: <https://www.web-canape.ru/business/vsya-statistika-interneta-na-2019-god-v-mire-i-v-rossii/>

7. Chudasama D., Parmar K., Patel D., Dangarwala K. J., Shah S. Survey of Image Compression Method Lossless Approach // International Journal of Engineering Research & Technology, 2015. Vol. 4, Iss. 03.

8. Фахми Ш. С., Селиверстов Я. А., Шаталова Н. В., Крюкова М. С., Хасан Я. А., Исмаил А. М. М., Ковалев К. Е. Методы, алгоритмы кодирования и классификация изображений морских судов // Морские интеллектуальные технологии. 2019. № 3-1 (45). С. 145–155.

9. Xiangxu Yu, Christos G Bampis, Praful Gupta, Alan Bovik. Predicting the Quality of Images Compressed After Distortion in Two Steps // June 2019, IEEE Transactions on Image Processing PP(99):1-1. DOI: 10.1109/TIP.2019.2922850.

10. Уэлстид С. Фракталы и вейвлеты для сжатия изображений в действии. М.: Триумф, 2003. С. 182–186.

11. Фахми Ш. С. Полигональная рекурсивная обработка видеoinформации // Вопросы радиоэлектроники. Серия: Техника телевидения. 2008. № 1. С. 42–51.

12. Бобровский А. И., Еид М. М., Костикова Е. В., Салем А., Алмахрук М. М., Фахми Ш. С. Выделение и хранение опорных точек изображений на основе динамических рекурсивных структур // Вопросы радиоэлектроники. Серия: Техника телевидения. 2016. № 3. С. 99–107.

13. Bandyopadhyay S. K., Paul T. U., Raychoudhury A. Image Compression using Approximate Matching and Run Length // International Journal of Advanced Computer Science and Applications. 2011. Vol. 2, N. 6.

14. Скворцов А. В. Триангуляция Делоне и ее применение. Томск: Изд-во Том. ун-та, 2002. 128 с.

15. Фахми Ш. С., Костикова Е. В., Крюкова М. С., Селиверстов С. А. Видеосистема обнаружения морских судов по триангуляционным решеткам // Морские интеллектуальные технологии. 2018. № 3-1 (41). С. 143–155.

16. Фахми Ш. С., Селиверстов Я. А., Шаталова Н. В., Крюкова М. С., Хасан Я. А., Исмаил А. М. М., Ковалев К. Е. Методы, алгоритмы кодирования и классификация изображений морских судов // Морские интеллектуальные технологии. 2019. № 3-1 (45). С. 145–155.

S. Sh. Fahmi, Professor, e-mail: shakeebf@mail.ru,
St. Petersburg State Electrotechnical University V. I. Ulyanov (Lenin) "LETI",
Solomenko Institute of Transport Problems of the Russian Academy of Sciences,
A. G. Davidchuk, Assistant Professor, e-mail: davag@bk.ru,
St. Petersburg State Electrotechnical University V. I. Ulyanov (Lenin) "LETI"
E. V. Kostikova, Assistant Professor, e-mail: kostikova.ev@mail.ru,
Admiral Makarov State University of Maritime and Inland Shipping

New Lossless Compression Algorithms for Transport Images

The article considers the relevance of the development of lossless image compression and transmission algorithms and their application for creating transport video surveillance systems. A brief overview of lossless transport image compression methods is provided. We propose a method for compressing transport plots based on the pyramid-recursive method of splitting the source image into polygons of various shapes and sizes. We consider two new algorithms for implementing the proposed method that are fundamentally different from each other: with a transition to the spectral region and without a transition to the spectral region of the original signal to ensure lossless compression. The results of testing various well-known lossless compression algorithms are analyzed: series length, Huffman, and arithmetic encoding, and compared with the proposed algorithms. It is shown that the proposed algorithms are more efficient in terms of compression ratio (2–3 times) compared to the known ones, while the computational complexity increases approximately by more than 3–4 times.

Keywords: Compression, recovery, compression ratio, algorithm complexity, polygonal recursive method

DOI: 10.17587/it.27.299-305

References

1. Ding J., Furgeson J. C., Sha E. H. Application specific image compression for virtual conferencing, *In Proceedings of the International Conference on Information Technology: Coding and Computing* (Cat. No. PR00540), Las Vegas, NV, USA, 27–29 March 2000, pp. 48–53, doi:10.1109/ITCC.2000.844182.
2. Bhavani S., Thanushkodi K. A survey on coding algorithms in medical image compression, *Int. J. Comput. Sci. Eng.*, 2010, no. 2, pp. 1429–1434.
3. Kharate G. K., Patil V. H. Color Image Compression Based on Wavelet Packet Best Tree. arXiv 2010, arXiv:1004.3276.
4. Haque M. R., Ahmed F. Image Data Compression with JPEG and JPEG2000, available at: http://eeweb.poly.edu/~yao/EE3414_S03/Projects/Loginova_Zhan_ImageCompressing_Rep.pdf (accessed on 1 October 2019).
5. Fahmi sh. S., Ryzhov N. G., Hasan Ya. A., Kalinina E. S. Adaptive image encoding and decoding algorithm in transport surveillance systems, *Informacionnye Tekhnologii*, 2019, vol. 25, no. 8, pp. 475–481.
6. Available at: <https://www.web-canape.ru/business/vsya-statistika-interneta-na-2019-god-v-mire-i-v-rossii/>
7. Chudasama D., Parmar K., Patel D., Dangarwala K. J., Shah S. Survey of Lossless Image Compression Method Approach, *International Journal of Engineering Research & Technology*, 2015, vol. 4, iss. 03.
8. Fahmy S. S., Seliverstov Y. A., Shatalova N. In., Kryukova M. S., Hassan Y. A., Ismail A. M., Kovalev K. E. Techniques,

coding algorithms and image classification of marine vessels, *Marine Intelligent Technologies*, 2019, no. 3-1 (45), pp. 145–155.

9. Xiangxu Yu, Christos G Bampis, Praful Gupta, Alan Bovik. Predicting the Quality of Compressed Images After Distortion in Two Steps, June 2019, *IEEE Transactions on Image Processing* PP(99):1-1, doi: 10.1109/TIP.2019.2922850.

10. Walsted S. Fractals and wavelets for image compression in action, Moscow, 2003, pp. 182–186.

11. Fahmy S. S. Polygon recursive processing of video information, *Questions of radio electronics. Series: Technique of television*, 2008, no. 1, pp. 42–51.

12. Bobrovsky A. I., Eid M. M., Kostikova E. V., Salem A., Almahruk M. M., Fahmi Sh. S. Selection and storage of image reference points based on dynamic recursive structures, *Questions of Radio Electronics. Series: Technique of Television*, 2016, no. 3, pp. 99–107.

13. Bandyopadhyay S. K., Paul T. U., Raychoudhury A. Image Compression using Approximate Matching and Run Length, *International Journal of Advanced Computer Science and Applications*, 2011, vol. 2, no. 6.

14. Skvortsov A. V. Delaunay Triangulation and its application, Tomsk, Publishing house Vol. UN-TA, 2002, 128 p.

15. Fahmi Sh. S., Kostikova E. V., Kryukova M. S., Seliverstov S. A. Video system for detecting sea vessels using triangulation grids, *Marine Intelligent Technologies*, 2018, no. 3-1 (41), pp. 143–155.

16. Fahmy S. S., Seliverstov Y. A., Shatalova N. In., Kryukova M. S., Hassan Y. A., Ismail A. M., Kovalev K. E. Techniques, coding algorithms and image classification of marine vessels, *Marine Intelligent Technologies*, 2019, no. 3-1 (45), pp. 145–155.

Д. В. Ефанов, д-р техн. наук, доц., e-mail: TrES-4b@yandex.ru,
ООО НТЦ "Комплексные системы мониторинга", г. Санкт-Петербург,
Санкт-Петербургский политехнический университет Петра Великого, г. Санкт-Петербург,
Российский университет транспорта, Москва,
В. В. Сапожников, д-р техн. наук, проф., e-mail: port.at.pgups@gmail.com,
Вл. В. Сапожников, д-р техн. наук, проф., e-mail: at.pgups@gmail.com,
Петербургский государственный университет путей сообщения Императора Александра I,
г. Санкт-Петербург

Организация схем встроенного контроля на основе метода логического дополнения с предварительным преобразованием рабочих функций в контрольные векторы кодов Бергера*

Описан новый способ организации схем встроенного контроля по методу логического дополнения, подразумевающий использование предварительного сжатия сигналов от объекта диагностирования с применением кодеров классических кодов с суммированием (кодов Бергера). Контроль сжатых сигналов осуществляется с помощью равновесного кода "1 из 4". В отличие от известных способов организации схем встроенного контроля создается возможность реализации самопроверяемого цифрового устройства с помощью одной такой схемы, что существенно уменьшает структурную избыточность. Предложено использовать в качестве схемы сжатия кодеры модифицированных кодов Бергера, обладающих улучшенными характеристиками обнаружения ошибок.

Ключевые слова: контроль вычислений, контроль комбинационных устройств, метод логического дополнения, сжатие сигналов, равновесный код, код Бергера, обнаружение ошибок

Введение

При разработке надежных и безопасных устройств и систем управления ответственными технологическими процессами широко применяются методы контроля корректности вычислений, защиты и кодирования данных [1–4]. Одним из вариантов организации контроля вычислений является использование схем встроенного контроля (СВК) [5]. Их применение позволяет не только фиксировать некорректные результаты вычислений, но и косвенно обнаруживать возникновение неисправностей в устройствах [6].

При синтезе СВК часто используются методы избыточного кодирования [7–11]. Выбранный на этапе проектирования код определяет ключевые характеристики реализуемого цифрового устройства: возможность обнаружения ошибок, структурную избыточность, энергопотребление, контролепригодность и пр.

Одним из подходов к организации СВК с применением избыточного кодирования является применение метода логического дополнения, предложенного в работе [12] и исследуемого в большом числе работ, например, в [13–19].

Данная работа посвящена развитию метода логического дополнения и содержит описание нового способа организации СВК с предварительным сжатием сигналов от объекта диагностирования. Применение сжатия сигналов позволяет уменьшить структурную избыточность реализуемого цифрового устройства.

1. Метод логического дополнения

Метод логического дополнения подразумевает организацию СВК таким образом, чтобы формируемый на выходе объекта диагностирования $F(x)$ информационный вектор $\langle f_1 f_2 \dots f_{m-1} f_m \rangle$ преобразовывался путем коррекции сигналов в кодовое слово $\langle h_1 h_2 \dots h_{m-1} h_m \rangle$, принадлежащее заранее выбранному избыточному коду (рис. 1). Такое преобразование осуществляется с использованием специальной схемы коррекции сигналов, образующей блок логического дополнения (БЛД). В блоке логического дополнения для преобразования сигналов применены двухвходовые элементы сложения по модулю $M = 2$ (элементы XOR). На входы элементов преобразования подаются сигналы от объекта диагностирования $F(x)$ (сигналы f_i) и сигналы от блока контрольной логики $G(x)$ (сигналы g_i), что позволяет формировать значения разрядов $h_1, h_2, \dots, h_m$ по правилу: $h_i = f_i \oplus g_i, i = 1, m$. Контроль принадлежности

*Работа выполнена при финансовой поддержке гранта № МД-2533.2021.4 Президента Российской Федерации.

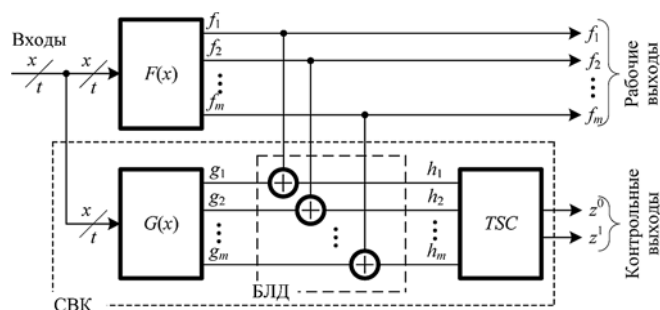


Рис. 1. Общая структура организации контроля комбинационных устройств по методу логического дополнения

формируемого кодового слова заранее выбранному избыточному коду осуществляется с использованием схемы тестера (*TSC*).

Организация СВК в соответствии со структурой, приведенной на рис. 1, позволяет во многих случаях синтезировать более простые цифровые устройства, наделенные свойствами обнаружения неисправностей в процессе функционирования, чем при использовании метода дублирования [17]. Это возможно за счет выбора такого способа преобразования, при котором функции, вычисляемые блоком $G(x)$, будут иметь наиболее простые логические выражения (более простые, чем функции, вычисляемые объектом диагностирования), а тестер в СВК будет проще схемы сравнения, используемой при дублировании.

При синтезе СВК для наделения ее свойством полной самопроверяемости относительно заданной модели неисправностей накладываются ряд ограничений [18]:

- на входы всех элементов блока логического дополнения в процессе функционирования устройства должны поступать хотя бы единожды все проверяющие комбинации из множества {00, 01, 10, 11};

- схема тестера должна быть полностью самопроверяемой, а на ее входы должны поступать хотя бы единожды все проверяющие комбинации (что определяется выбранным для контроля кодом и способом реализации тестера);

- не должно происходить маскировки ошибок на входах элементов преобразования;

- схема блока контрольной логики $G(x)$ должна быть проверяемой (любая неисправность должна проявляться хотя бы на одном входном наборе на выходах устройства).

При организации СВК по методу логического дополнения может быть использован контроль принадлежности формируемого на входах тестера кодового слова какому-либо избыточному коду, например, равновесному коду [19] или коду с суммированием [20]. Другим подходом является организация контроля по признаку самодвойственности формируемых функций.

Кроме того, диагностические признаки можно комбинировать. Такой подход, к примеру, описан в работе [21], где предполагается организация самодвойственного устройства с контролем не только по данному признаку, но и с контролем вычислений по равновесным кодам "2 из 4".

В ряде публикаций, например, в работе [22], отмечаются преимущества использования равновесных кодов с малой длиной кодовых слов, связанные с простотой их тестеров и малым числом требуемых для полной проверки тестовых комбинаций. Одним из таковых кодов является код "1 из 4" (1/4-код). Данный код имеет четыре рабочих кодовых слова {0001, 0010, 0100, 1000} и не обнаруживает всего 12 ошибок, каждая из которых является двукратной разнонаправленной (симметричной) ошибкой. На рис. 2 приведена базовая структура организации СВК по 1/4-коду. При ее использовании выходы объекта диагностирования разбиваются на группы по четыре выхода, для каждой такой группы строятся отдельные схемы контроля, а их выходы объединяются на входах самопроверяемой схемы сжатия парафазных сигналов [23], что позволяет свести контроль вычислений к контролю только одного парафазного сигнала.

Для преобразования любого кодового вектора в кодовое слово 1/4-кода требуется не более трех элементов *XOR*, что позволяет уменьшить число контрольных функций до трех. Это же упрощает процедуру построения полностью самопроверяемой СВК.

В работе [24] предложено использовать равновесный 1/4-код совместно со специальной схемой сжатия сигналов, позволяющей организовывать контроль многовыходного устройства ($m > 4$) по данному коду с реализацией одной СВК. Структура организации СВК с предварительным сжатием сигналов представлена на рис. 3. Данная структура является базовой и строится для устройства с $m = 8$ выходами. В ней, в отличие от структуры, приведенной на рис. 2, блок логического дополнения включает в себя не

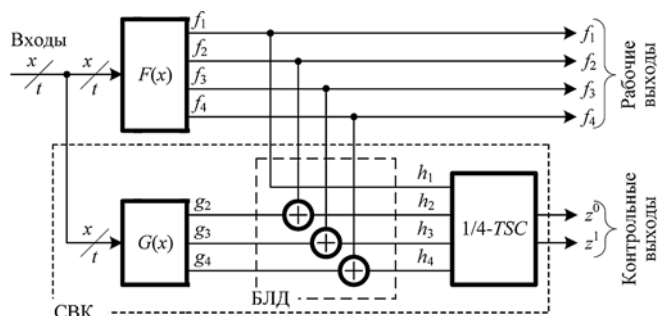


Рис. 2. Структура организации контроля комбинационных устройств по методу логического дополнения до равновесного кода "1 из 4"

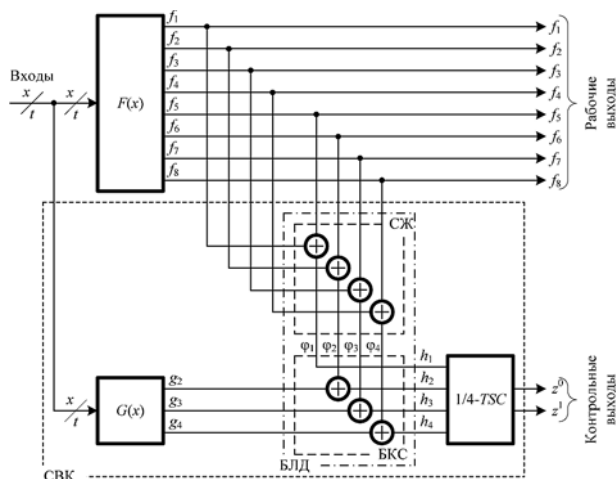


Рис. 3. Структура организации контроля комбинационных устройств по методу логического дополнения до равновесного кода "1 из 4" с предварительным сжатием сигналов

только блок коррекции сигналов (БКС), но еще и схему сжатия (СЖ). Схема сжатия позволяет сжать сигналы от всех восьми выходов в четыре сигнала $\phi_1, \phi_2, \phi_3, \phi_4$. Блок коррекции сигналов позволяет осуществить последующее преобразование значений функций $\phi_1, \phi_2, \phi_3, \phi_4$ в h_1, h_2, h_3, h_4 . Схема сжатия устроена таким образом, что реализуется процедура сжатия сигналов для четырех пар информационных выходов блока $F(x)$.

Особенности обнаружения ошибок в СВК, реализованной в соответствии с представленной структурой, обсуждаются в работах [25, 26]. Исследования показывают, что можно сжимать сигналы не только попарно, а объединяя сразу же несколько выходов. Однако при этом меняются характеристики обнаружения ошибок на выходах объекта диагностирования. Одним из способов предварительного сжатия группы сигналов является сжатие с помощью кодеров каких-либо избыточных кодов. В данной работе предлагается организация схемы сжатия на основе кодеров классических кодов Бергера, описанных в работе [20], свойства которых достаточно полно описаны в монографии [10].

2. Предварительное сжатие сигналов с помощью кодеров кодов Бергера

На рис. 4 приведена обобщенная структура организации СВК с предварительным сжатием сигналов от всех выходов объекта диагностирования с помощью кодера классического кода Бергера. В ней схема сжатия образована кодером $G(f)$, формирующим функции $\phi_1, \phi_2, \dots, \phi_k$, где $k = \lceil \log_2(m+1) \rceil$ (запись $\lceil \dots \rceil$ обозначает целое сверху от вычисляемого значения). К примеру, если устройство $F(x)$ имеет $m = 20$ выходов, то

число выходов кодера кода Бергера будет равно $k = 5$. Число выходов блока контрольной логики $G(x)$ равно числу выходов кодера $G(f)$. Другими словами, использование сжатия сигналов в кодере $G(f)$ позволяет уменьшить число контрольных функций, число элементов преобразования и позволяет выбрать наиболее простой тестер для контроля преобразованных сигналов.

В схеме сжатия требуется обеспечить тестирование кодера $G(f)$, что требует формирования хотя бы по разу всех контрольных слов кода Бергера. Это просто сделать для кодов Бергера, для которых значение $m = 2^q - 1$, $q = 2, 3, \dots$, так как в этом случае формируются все возможные разряды для 2^k контрольных векторов. Максимально просто это сделать для случая $m = 3$. У такого кода Бергера имеется $k = 2$ контрольных разряда, и формируются все возможные сочетания их значений: $\{00, 01, 10, 11\}$.

На рис. 5 изображена базовая структура организации СВК для устройства $F(x)$ с $m = 6$ выходами. В ней предполагается использование

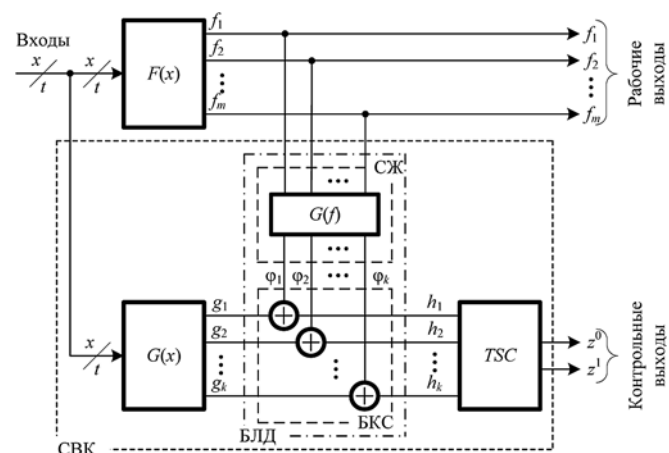


Рис. 4. Обобщенная структура организации СВК на основе преобразования информационного вектора в контрольный вектор кода Бергера

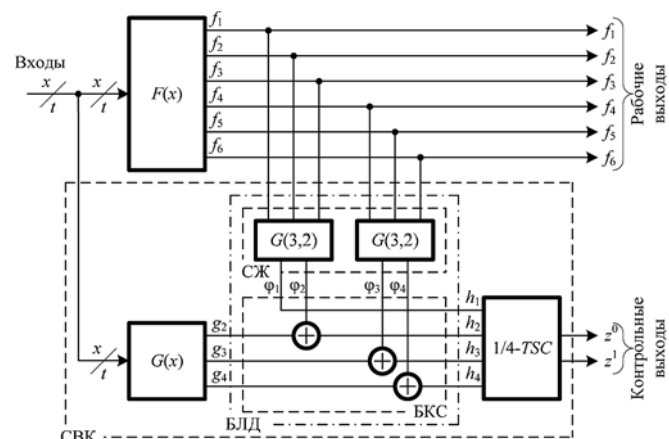


Рис. 5. Структура организации СВК на основе преобразования информационного вектора в контрольный вектор кода Бергера

схемы сжатия, каждая из которых образована двумя кодерами кода Бергера $G(3,2)$. Каждый такой кодер преобразует $m = 3$ входных сигнала в $k = 2$ выходных сигнала и позволяет сформировать четыре функции $\varphi_1, \varphi_2, \varphi_3, \varphi_4$. Далее эти функции преобразуются в кодовое слово, принадлежащее 1/4-коду и контролируруются с помощью тестера данного кода.

Первые три выхода блока $F(x)$ преобразуются в два контрольных выхода φ_1 и φ_2 , на которых формируются контрольные векторы кода Бергера. Для этих целей используется кодер кода Бергера, фактически представляющий собой один полный сумматор (*full-adder*) [7]. Аналогично вторая группа из трех выходов блока $F(x)$ преобразуется в два контрольных выхода φ_3 и φ_4 . Выходы $\varphi_1, \dots, \varphi_4$ поступают на входы схемы коррекции сигналов, предназначенной для формирования на своих выходах кодовых слов равновесного кода "1 из 4" (1/4-кода). Функция φ_1 не корректируется и непосредственно формирует значение разряда h_1 , остальные функции φ_2, φ_3 и φ_4 корректируются.

3. Синтез СВК с предварительным сжатием сигналов с помощью кодеров кодов Бергера

Все элементы в СВК, за исключением блока $G(x)$, являются типовыми. Задачей синтеза СВК является получение выражений, описывающих функции g_2, g_3 и g_4 . Их значения получают

с учетом ряда ограничений: необходимость формирования полных тестов для 1/4-*TSC* и каждого элемента *XOR* в структуре схемы коррекции сигналов [18].

В качестве примера синтезируем СВК для комбинационного устройства, заданного таблицей истинности (см. таблицу). Для получения функций $\varphi_1, \dots, \varphi_4$ выходы исходного устройства разбиваются на два подмножества $\{f_1, f_2, f_3\}$ и $\{f_4, f_5, f_6\}$. Подсчитывается число единичных разрядов в каждом из подмножеств на каждой входной комбинации, а полученное число в двоичном виде записывается в соответствующие столбцы: φ_1, φ_2 для первого подмножества и φ_3, φ_4 — для второго. Далее функция g_1 приравнивается к функции h_1 . Доопределению значений подлежат функции g_2, g_3 и g_4 .

Для доопределения значений функций g_2, g_3 и g_4 применим эвристический подход, описанный в работе [27] для использования 2/4-кода. Суть эвристического подхода состоит в том, что значения функций g_2, g_3 и g_4 на тех входных наборах, на которых $h_1 = 1$, доопределяются с учетом формирования кодового слова $\langle h_1 h_2 h_3 h_4 \rangle = \langle 1000 \rangle$, и фиксируются все тестовые комбинации элементов преобразования (*XOR*). На остальных наборах значения функций g_2, g_3 и g_4 получаются произвольным образом, однако с соблюдением трех условий:

1) на каждой входной комбинации кодовое слово $\langle h_1 h_2 h_3 h_4 \rangle$ должно принадлежать одно-

Описание схемы встроенного контроля

Входные комбинации				Рабочие функции объекта диагностирования						Сжатые функции				Контрольные функции			Кодовые векторы на входе тестера				Тестовые комбинации элементов преобразования		
x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4	f_5	f_6	φ_1	φ_2	φ_3	φ_4	g_2	g_3	g_4	h_1	h_2	h_3	h_4	XOR_2	XOR_3	XOR_4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	00	00	01
0	0	0	1	0	0	1	0	1	1	0	1	1	0	0	1	0	0	1	0	0	10	11	00
0	0	1	0	0	0	1	1	0	1	0	1	1	0	0	1	0	0	1	0	0	10	11	00
0	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	10	00	00
0	1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	0	0	10	11	00
0	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	1	0	0	0	00	00	11
0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	1	0	1	0	0	0	11	11	00
0	1	1	1	1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	11	01	00
1	0	0	0	1	0	0	1	1	0	0	1	1	0	1	0	0	0	0	1	0	11	10	00
1	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	10	00	00
1	0	1	0	0	0	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	10	11	10
1	0	1	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0	0	00	11	11
1	1	0	0	1	0	1	0	0	1	1	0	0	1	0	0	1	1	0	0	0	00	00	11
1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1	0	0	0	00	11	00
1	1	1	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	1	0	0	01	11	00
1	1	1	1	1	0	0	1	1	1	0	1	1	1	1	0	1	0	0	1	0	11	10	11

му из трех оставшихся кодовых слов 1/4-кода (<0001>, <0010>, <0100>);

2) в итоговой таблице истинности должны хотя бы по разу формироваться все кодовые слова 1/4-кода;

3) в итоговой таблице истинности должны быть сформированы полные проверяющие тесты для всех элементов *XOR*.

Если при соблюдении условий № 1 и № 2 не соблюдается условие № 3 (при эвристическом подходе это высоковероятно и на рассматриваемом множестве примеров наблюдалось практически всегда), то проводится поиск наиболее часто встречающейся проверяющей комбинации для того элемента преобразования, для которого не сформирован полный тест, и осуществляется коррекция значений с учетом сформулированных выше условий. Такой подход к получению значений функций g_2 , g_3 и g_4 оказывается трудоемким, однако чем больше число входов устройства $F(x)$, тем проще выполнить условия 1)–3) и тем быстрее можно выполнить процедуры доопределения значений. Альтернативным является подход, при котором между значениями функций g_i и f_i , $i = 1, 4$, изначально устанавливается функциональная зависимость.

Выписывая из таблицы значения данных функций в матричной форме и проводя процедуры по оптимизации методом Карно, получаем:

$$g_2 = x_2 x_3 \vee x_1 \overline{x_2} \overline{x_4};$$

$$g_3 = x_3 \overline{x_4} \vee x_1 \overline{x_2} x_3 \vee \overline{x_1} x_2 \overline{x_4} \vee x_1 x_2 \overline{x_3} \vee \overline{x_1} \overline{x_2} \overline{x_3} x_4 \vee x_1 x_2 \overline{x_3} x_4 = x_3 (\overline{x_4} \vee x_1 \overline{x_2}) \vee \overline{x_1} x_2 (x_3 \vee \overline{x_4}) \vee x_3 x_4 (\overline{x_1} \overline{x_2} \vee x_1 x_2);$$

$$g_4 = x_1 x_3 x_4 \vee \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 \overline{x_3} x_4 \vee x_1 x_2 \overline{x_3} \overline{x_4} = x_1 x_3 x_4 \vee x_3 (\overline{x_1} \overline{x_2} \overline{x_4} \vee x_2 (\overline{x_1} x_4 \vee x_1 \overline{x_4})).$$

Структура, представленная на рис. 5, имеет следующую ключевую особенность.

Утверждение 1. На выходах схемы сжатия обнаруживается ошибка любой кратности и любого вида, за исключением двукратной симметричной ошибки.

Интерес представляет исследование распространения ошибок с входов схемы сжатия на ее выходы. Однако в данной статье этот вопрос не исследуется.

4. Сжатие сигналов на основе модифицированного кода с суммированием

У структуры организации СВК со сжатием с применением кодеров кодов Бергера имеет-

ся следующий недостаток. Кодеры кодов Бергера имеют неравнозначные выходы в смысле формирования и распределения на них единичных и нулевых значений и распространения ошибок с входов на выходы. Так, значение на младшем разряде каждого из кодеров будет искажаться при однократной и трехкратной ошибках, значение же на втором выходе будет искажаться только при двукратной и трехкратной ошибках. Вероятность искажения одного разряда гораздо выше, чем двух и трех.

Улучшения характеристик обнаружения ошибок на выходах схемы сжатия можно добиться путем использования другого кода с суммированием — кода с суммированием взвешенных переходов, или *T*-кода [11].

Правила определения значений контрольных разрядов в кодовом слове кода с суммированием взвешенных переходов:

1. Переходам между разрядами информационного вектора, начиная с младшего разряда, присваиваются весовые коэффициенты $w_{i,i+1}$ из ряда возрастающих степеней числа 2:

$$[w_{i,i+1}] = [w_{m-1,m}, w_{m-2,m-1}, \dots, w_{2,3}, w_{1,2}] = [2^{m-2}, 2^{m-3}, \dots, 2^1, 2^0].$$

2. Подсчитывается суммарный вес активных переходов:

$$W = \sum_{i=1}^{m-1} w_{i,i+1} t_{i,i+1},$$

где $t_{i,i+1} = f_i \oplus f_{i+1}$ — функция активации перехода между разрядами f_i и f_{i+1} .

3. Полученное число представляется в двоичном виде и записывается в разряды контрольного вектора.

Значения контрольных разрядов могут быть определены по следующим формулам:

$$h_1 = f_1 \oplus f_2;$$

$$h_2 = f_2 \oplus f_3;$$

...

$$h_{m-1} = f_{m-1} \oplus f_m.$$

Так как для получения значений контрольных разрядов кода с суммированием взвешенных переходов используются только операции сложения по модулю два, структура кодера данного кода будет стандартной и будет содержать в себе $m - 1$ элемент сложения по модулю два. Структура организации СВК со сжатием сигналов с помощью *T*-кода, представлена на рис. 6. В ней гораздо проще обеспечивается тестируемость элементов *XOR* в схеме сжатия сигналов, сам кодер проще и обнаруживается большее число ошибок.

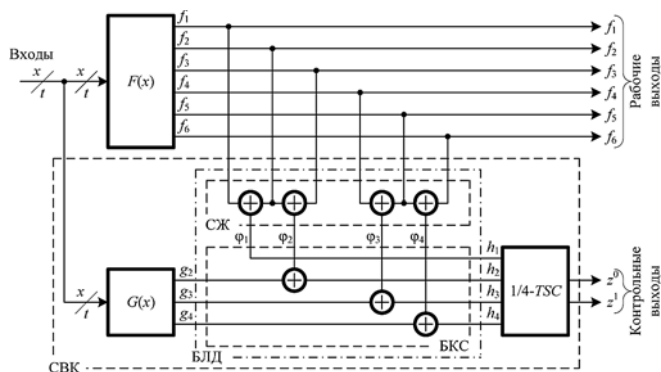


Рис. 6. Структура организации СВК на основе преобразования информационного вектора в контрольный вектор T -кода

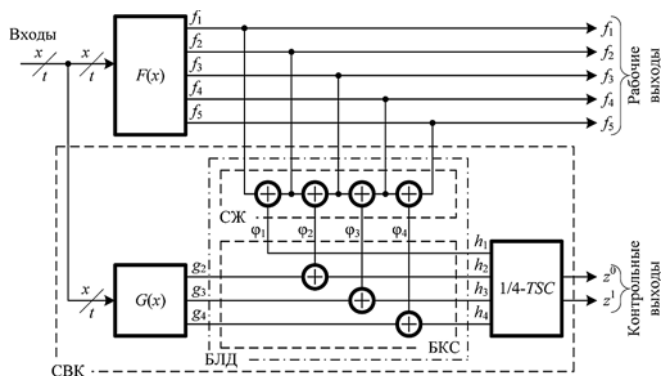


Рис. 7. Базовая структура организации СВК для устройства с пятью выходами на основе преобразования информационного вектора в контрольный вектор T -кода

Из работы [11] известно, что T -кодом обнаруживаются любые ошибки с кратностями $d \leq m - 1$, однако не обнаруживаются любые ошибки с кратностями $d = m$. Поэтому при синтезе СВК по данным кодам согласно структуре, приведенной на рис. 6, требуется осуществлять поиск групп выходов, в которых возможны ошибки только с кратностями $d = 1$ и $d = 2$. Это следует из того факта, что сигналы с трех рабочих выходов устройства $F(x)$ сжимаются в два сигнала (длина информационного вектора T -кода равна трем).

Поиск групп, в которых возможны только искажения с кратностями $d = 1$ и $d = 2$, может быть осуществлен на основании следующего утверждения.

Утверждение 2. В группе из трех выходов невозможно одновременное искажение всех выходов, если для любого логического элемента G_q внутренней структуры объекта диагностирования выполняется условие

$$\frac{\partial f_a}{\partial y_q} \cdot \frac{\partial f_b}{\partial y_q} \cdot \frac{\partial f_c}{\partial y_q} = 0,$$

где f_a, f_b, f_c — выходы, образующие группу, y_q — функция, реализуемая на выходе логического элемента G_q , $q \in Q$, Q — множество логических элементов.

Если выражение в левой части представленной выше формулы равно нулю, то не существует ни одного входного набора, на котором происходит трансляция ошибок на все три выхода выбранной группы.

Учитывая особенности T -кода, можно предложить еще один вариант организации базовой структуры для устройства с пятью выходами (рис. 7). В этой структуре будут обнаруживаться любые ошибки на выходе объекта диагностирования, за исключением ошибок с кратностями $d = 5$.

Утверждение 3. В группе из пяти выходов невозможно одновременное искажение всех выходов, если для любого логического элемента G_q внутренней структуры объекта диагностирования выполняется условие

$$\frac{\partial f_1}{\partial y_q} \cdot \frac{\partial f_2}{\partial y_q} \cdot \frac{\partial f_3}{\partial y_q} \cdot \frac{\partial f_4}{\partial y_q} \cdot \frac{\partial f_5}{\partial y_q} = 0.$$

Заключение

Использование метода логического дополнения на практике дает возможность синтеза большого числа вариантов СВК для одного и того же устройства, что является несомненным его преимуществом перед другими методами. Предложенный в настоящей статье способ организации СВК позволяет уменьшать сложность технической реализации самопроверяемых цифровых устройств для многovyходных исходных объектов. При этом достаточно просто обеспечивается полная самопроверяемость СВК.

Среди недостатков предлагаемого метода необходимо отметить возможность маскировки ошибок на выходах объекта диагностирования при использовании схемы сжатия. Тем не менее, этот недостаток нивелируется при выборе групп сжимаемых выходов, либо же при использовании специального преобразования структуры исходного объекта [10, 11, 13, 28–30].

Исследованиям применения схем сжатия при организации СВК к настоящему времени уделено недостаточно внимания, а в качестве перспектив их использования можно выделить применение кодеров модифицированных кодов с суммированием, обладающих улучшенными характеристиками обнаружения ошибок в информационных векторах по сравнению с классическими кодами Бергера.

Список литературы

1. Fujiwara E. Code Design for Dependable Systems: Theory and Practical Applications. John Wiley & Sons, 2006, 720 p.
2. Ubar R., Raik J., Vierhaus H.-T. Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source). Information Science Reference, Hershey. New York, IGI Global, 2011. 578 p.

3. Дрозд А. В., Харченко В. С., Антошук С. Г., Дрозд Ю. В., Дрозд М. А., Сулима Ю. Ю. Рабочее диагностирование безопасных информационно-управляющих систем. Харьков: Национальный аэрокосмический университет им. Н. Е. Жуковского "ХАИ", 2012. 614 с.
4. Nahapov V. Cyber Physical Computing for IoT-driven Services. New York: Springer International Publishing AG, 2018. 279 p. doi: 10.1007/978-3-319-54825-8.
5. Согомонян Е. С., Слабаков Е. В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
6. Пархоменко П. П., Согомонян Е. С. Основы технической диагностики (оптимизация алгоритмов диагностирования, аппаратные средства). М.: Энергоатомиздат, 1981. 320 с.
7. Piestrak S. J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. — Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
8. Abramovici M., Breuer M. A., Friedman A. D. Digital System Testing and Testable Design. New Jersey: IEEE Press, 1998. 652 p.
9. Lala P. K. Principles of Modern Digital Design. New Jersey, John Wiley & Sons, 2007. 419 p.
10. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Коды с суммированием для систем технического диагностирования. Том 1: Классические коды Бергера и их модификации. М.: Наука, 2020. 383 с.
11. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Коды с суммированием для систем технического диагностирования. Том 2: Взвешенные коды с суммированием. М.: Наука, 2021. 455 с.
12. Saposhnikov V. V., Dmitriev A., Goessel M., Saposhnikov V. V. Self-Dual Parity Checking — a New Method for on Line Testing // Proceedings of 14th IEEE VLSI Test Symposium, USA, Princeton. 1996. P. 162—168.
13. Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science + Business Media B. V., 2008. 184 p.
14. Sen S. K., Roy S. S. An Optimized Concurrent Self-Checker Using Constraint-Don't Cares and 1-out-of-4 Code // National Conference (AECDISC'2008) in Asansol Engineering College. Held during 1-2 August 2008.
15. Sen S. K. A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint Don't Cares // National Conference on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEEERE'2010), Sikkim Manipal Institute of Technology. Sikkim, held during 22—24 December, 2010.
16. Das D. K., Roy S. S., Dmitiriev A., Morozov A., Gössel M. Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes // Proceedings of the 10th International Workshops on Boolean Problems. Freiberg, Germany, September, 2012. P. 33—40.
17. Пивоваров Д. В. Организация систем функционального контроля комбинационных логических схем на основе метода логического дополнения по равновесному коду "1 из 5" // Автоматика на транспорте. 2017. Т. 3. № 4. С. 605—624.
18. Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. V., Pivovarov D. V. The Synthesis Conditions of Completely Self-Testing Embedded-Control Circuits Based on the Boolean Complement Method to the "1-out-of-m" Constant-Weight Code // Automatic Control and Computer Sciences. 2020. Vol. 54, Iss. 2. P. 89—99. DOI: 10.3103/S0146411620020042.
19. Freiman C. V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels // Information and Control. 1962. Vol. 5, Iss. 1. P. 64—71. DOI: 10.1016/S0019-9958(62)90223-1.
20. Berger J. M. A Note on Error Detection Codes for Asymmetric Channels // Information and Control. 1961. Vol. 4, Iss. 1. P. 68—73. DOI: 10.1016/S0019-9958(61)80037-5.
21. Efanov D., Sapozhnikov V., Sapozhnikov V. V., Osadchy G., Pivovarov D. Self-Dual Complement Method up to Constant-Weight Codes for Arrangement of Combinational Logical Circuits Concurrent Error-Detection Systems // Proceedings of 17th IEEE East-West Design & Test Symposium (EWDTS'2019), Batumi, Georgia, September 13—16. 2019. P. 136—143. doi: 10.1109/EWDTS.2019.8884398.
22. Пивоваров Д. В. Построение систем функционального контроля многовыходных комбинационных схем методом логического дополнения по равновесным кодам // Автоматика на транспорте. 2018. Т. 4, № 1. С. 131—149.
23. Nikolos D. Self-Testing Embedded Two-Rail Checkers // Chapter 7 in On-Line Testing for VLSI. 1998. P. 69—79. DOI: 10.1007/978-1-4757-60-69-9_7.
24. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Синтез схемы встроенного контроля для многовыходных комбинационных устройств на основе логического дополнения и сжатия сигналов // Известия вузов. Приборостроение. 2020. Т. 63, № 7. С. 581—597. DOI: 10.17586/0021-3454-2020-63-7-581-597.
25. Ефанов Д. В., Сапожников В. В., Сапожников Вл. В., Осадчий Г. В. Синтез схем встроенного контроля на основе метода логического дополнения с предварительным сжатием сигналов рабочих функций // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2021. № 1. С. 97—115. DOI: 10.17223/19988605/54/12.
26. Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. V. Organization of Testing of Combinational Devices Based on Boolean Complement to Constant-Weight "1-out-of-4" Code with Signal Compression // Automatic Control and Computer Science. 2021. Vol. 55, Iss. 2. P. 113—124. DOI: 10.3103/S014641162102005X.
27. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Метод функционального контроля комбинационных логических устройств на основе кода "2 из 4" // Известия вузов. Приборостроение. 2016. Т. 59, № 7. С. 524—533. DOI: 10.17586/0021-3454-2016-59-7-524-533.
28. Busaba F. Y., Lala P. K. Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors // Journal of Electronic Testing: Theory and Applications. 1994. Vol. 5, Iss. 5. P. 19—28.
29. Matrosova A., Levin I., Ostanin S. Survivable Self-Checking Sequential Circuits // Proceedings of 2001 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'2001), USA, CA, San Francisco, October 24—26. 2001. P. 395—402.
30. Sogomonyan E. S., Gössel M. Design of Self-Testing and On-Line Fault Detection Combinational Circuits with Weakly Independent Outputs // Journal of Electronic Testing: Theory and Applications. 1993. Vol. 4, Iss. 4. P. 267—281. DOI: 10.1007/BF00971975.

D. V. Efanov, D. Sc., Associate Professor, e-mail: TrES-4b@yandex.ru,
 Scientific and Technical Center "Integrated Monitoring Systems" LLC,
 Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation,
 Russian University of Transport, Moscow, Russian Federation,
V. V. Saposhnikov, D. Sc., Professor, e-mail: port.at.pgups@gmail.com,
Vl. V. Saposhnikov, D. Sc., Associate Professor, e-mail: at.pgups@gmail.com,
 Emperor Alexander I St. Petersburg State Transport University, St. Petersburg, Russian Federation

The Concurrent Error-Detection Systems Organization Based on the Boolean Complement Method with the Preliminary Transformation of Operating Functions into Check Vectors of Berger Codes

The article describes a new way of concurrent error-detection (CED) systems organization using the Boolean complement method, which involves the use of pre-compression of signals from the diagnostic object using encoders of classical sum codes (Berger codes). Control of compressed signals is carried out using the constant-weight "1-out-of-4" code. In comparison with the known methods of the CED systems organization, it is possible to implement a self-checking digital device using one such circuit, and this significantly reduces the structural redundancy. The article suggests using the encoders of modified Berger codes with improved error detection characteristics as a compression scheme.

Keywords: calculation checking; combinational devices checking; Boolean complement method; signals compression; constant-weight code; Berger code; error detection

Acknowledgements: This work was supported by the grant no. MD-2533.2021.4 of the President of the Russian Federation.

DOI: 10.17587/it.27.306-313

References

1. Fujiwara E. Code Design for Dependable Systems: Theory and Practical Applications, John Wiley & Sons, 2006, 720 p.
2. Ubar R., Raik J., Vierhaus H.-T. Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source). Information Science Reference, Hershey New York, IGI Global, 2011, 578 p.
3. Drozd A. V., Kharchenko V. S., Antoshchuk S. G., Drozd Yu. V., Drozd M. A., Sulima Yu. Yu. Objects and Methods of On-Line Testing for Safe Instrumentation and Control Systems, Kharkov, National Aerospace University "KhAI, 2012, 614 p. (in Russian).
4. Hahanov V. Cyber Physical Computing for IoT-driven Services, New York, Springer International Publishing AG, 2018, 279 p., doi: 10.1007/978-3-319-54825-8.
5. Sogomonyan E. S., Slabakov E. V. Self-checking devices and fault-tolerant systems, Radio & Svyaz', Moscow, 208 p. (in Russian).
6. Parkhomenko P. P., Sogomonyan E. S. Fundamentals of technical diagnostics (optimization of diagnostic algorithms, hardware, Moscow, Energoatomizdat, 1981, 320 p. (in Russian).
7. Piestrak S. J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 1995, 111 p.
8. Abramovici M., Breuer M. A., Friedman A. D. Digital System Testing and Testable Design, New Jersey, IEEE Press, 1998, 652 p.
9. Lala P. K. Principles of Modern Digital Design, New Jersey, John Wiley & Sons, 2007, 419 p.
10. Sapozhnikov V. V., Sapozhnikov V. V., Efanov D. V. Sum codes for technical diagnostic systems. Volume 1: Classic Berger codes and their modifications, Moscow, Nauka, 2020, 383 p. (in Russian).
11. Sapozhnikov V. V., Sapozhnikov V. V., Efanov D. V. Sum codes for technical diagnostic systems. Volume 2: Weighted codes with summation, Moscow, Nauka, 2021, 455 p. (in Russian).
12. Saposhnikov V. V., Dmitriev A., Goessel M., Saposhnikov V. V. Self-Dual Parity Checking a New Method for on Line Testing, *Proceedings of 14th IEEE VLSI Test Symposium*, USA, Princeton, 1996, pp. 162–168.
13. Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science + Business Media B. V., 2008, 184 p.
14. Sen S. K., Roy S. S. An Optimized Concurrent Self-Checker Using Constraint-Don't Cares and 1-out-of-4 Code, *National Conference (AECDISC'2008) in Asansol Engineering College*, held during 1–2 August 2008.
15. Sen S. K. A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint Don't Cares, *National Conference on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEEERE'2010)*, Sikkim Manipal Institute of Technology, Sikkim, held during 22–24 December, 2010.
16. Das D. K., Roy S. S., Dmitriev A., Morozov A., Gössel M. Constraint-Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes, *Proceedings of the 10th International Workshops on Boolean Problems*, Freiberg, Germany, September, 2012, pp. 33–40.
17. Pivovarov D. V. Organization of concurrent error-detection systems for combinational logic circuits based on the method of Boolean complement according to the constant-weight code "1-out-of-5", *Avtomatika na Transporte*, 2017, vol. 3, iss. 4, pp. 605–624 (in Russian).
18. Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. V., Pivovarov D. V. The Synthesis Conditions of Completely Self-Testing Embedded-Control Circuits Based on the Boolean Complement Method to the "1-out-of-m" Constant-Weight Code, *Automatic Control and Computer Sciences*, 2020, vol. 54, iss. 2, pp. 89–99. DOI: 10.3103/S0146411620020042.
19. Freiman C. V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels, *Information and Control*, 1962, vol. 5, iss. 1, pp. 64–71. DOI: 10.1016/S0019-9958(62)90223-1.
20. Berger J. M. A Note on Error Detection Codes for Asymmetric Channels, *Information and Control*, 1961, vol. 4, iss. 1, pp. 68–73. DOI: 10.1016/S0019-9958(61)80037-5.
21. Efanov D., Sapozhnikov V., Sapozhnikov V. V., Osadchy G., Pivovarov D. Self-Dual Complement Method up to Constant-Weight Codes for Arrangement of Combinational Logical Circuits Concurrent Error-Detection Systems, *Proceedings of 17th IEEE East-West Design & Test Symposium (EWDTS'2019)*, Batumi, Georgia, September 13–16, 2019, pp. 136–143, doi: 10.1109/EWDTS.2019.8884398.
22. Pivovarov D. V. Construction of concurrent error-detection systems for multi-output combinational circuits by the Boolean complement method using constant-weight codes, *Avtomatika na Transporte*, 2018, vol. 4, iss. 1, pp. 131–149 (in Russian).
23. Nikolos D. Self-Testing Embedded Two-Rail Checkers, *Chapter 7 in On-Line Testing for VLSI*, 1998, pp. 69–79. DOI 10.1007/978-1-4757-60-69-9_7.
24. Sapozhnikov V. V., Sapozhnikov V. V., Efanov D. V. Synthesis of a built-in control circuit for multi-output combinational devices based on Boolean complement and signal compression, *Izvestiya vuzov. Priborostroenie*, 2020, vol. 63, iss. 7, pp. 581–597. DOI: 10.17586 / 0021-3454-2020-63-7-581-597 (in Russian).
25. Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. V., Osadchy G. V. Synthesis of built-in control circuits based on the Boolean complement method with preliminary compression of signals of working functions, *Bulletin of the Tomsk State University. Management, computer technology and informatics*, 2021, iss. 1, pp. 97–115. DOI: 10.17223/19988605/54/12 (in Russian).
26. Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. V. Organization of Testing of Combinational Devices Based on Boolean Complement to Constant-Weight "1-out-of-4" Code with Signal Compression, *Automation and Computer Science*, 2021, vol. 55, iss. 2, pp. 113–124. DOI: 10.3103/S014641126102005X.
27. Sapozhnikov V. V., Sapozhnikov V. V., Efanov D. V. Method of concurrent error-detection checking of combinational logical devices based on the code "2-out-of-4", *Izvestiya vuzov. Priborostroenie*, 2016, vol. 59, iss. 7, pp. 524–533, DOI: 10.17586/0021-3454-2016-59-7-524-533 (in Russian).
28. Busaba F. Y., Lala P. K. Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors, *Journal of Electronic Testing: Theory and Applications*, 1994, vol. 5, iss. 5, pp. 19–28.
29. Matrosova A., Levin I., Ostanin S. Survivable Self-Checking Sequential Circuits, *Proceedings of 2001 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'2001)*, USA, CA, San Francisco, October 24–26, 2001, pp. 395–402.
30. Sogomonyan E. S., Gössel M. Design of Self-Testing and On-Line Fault Detection Combinational Circuits with Weakly Independent Outputs, *Journal of Electronic Testing: Theory and Applications*, 1993, vol. 4, iss. 4, pp. 267–281, DOI: 10.1007/BF00971975.

Е. А. Саксонов, главный специалист, д-р техн. наук, проф., e-mail: saksmiem@mail.ru,
Московский технический университет связи и информатики, Москва

Способ обезличивания персональных данных

Предложен метод обезличивания больших объемов персональных данных. Метод сохраняет структуру и семантику данных, позволяет повысить безопасность обезличенных данных и обрабатывать персональные данные без предварительного обезличивания. Разработана математическая модель метода. Получены оценки защищенности обезличенных данных.

Ключевые слова: персональные данные, обезличивание персональных данных, преобразование таблиц персональных данных, защищенность обезличенных персональных данных

Введение

Обработка персональных данных (ПД) применяется для решения различных задач по обслуживанию клиентов, формированию статистических данных, проведению социальных и экономических исследований. Для этого обычно используются большие объемы различных типов ПД, хранение которых должно обеспечивать их защиту от неправомерного использования.

Одним из распространенных подходов к хранению ПД является их обезличивание, представляющее ПД в таком виде, когда без использования дополнительной информации невозможно определить принадлежность ПД конкретному субъекту ПД [1]. Обезличивание обеспечивает возможность многократной обработки ПД для решения различных задач.

Современные информационные технологии позволяют проводить обезличивание ПД различными способами, сохраняя их полноту, структуру и семантическую целостность для проведения обработки.

Способы различаются по применяемым алгоритмам, трудоемкости реализации, возможностям обработки запросов, стойкости к различным видам атак, направленных на деобезличивание [2–6].

Однако наличие большого числа разнообразных способов обезличивания приводит к сложностям в согласовании и обработке данных, обезличенных различными способами.

В данной статье предлагается способ обезличивания ПД, обеспечивающий возможность проведения процедуры деобезличивания, изменения параметров способа и проведения обработки ПД, внесения изменений в ПД без предварительного деобезличивания, повышающий стойкость к атакам, направленным на деобезличивание.

Рассматриваемый способ обеспечивает конфиденциальность, доступность и целостность обезличенных данных. Способ основан на приведении исходной таблицы ПД к такому виду, когда выделение ПД конкретных физических лиц становится практически невозможным из-за очень большой трудоемкости при неизвестных параметрах обезличивания. Способ реализует один из предлагаемых Роскомнадзором методов обезличивания и соответствует установленным требованиям [7, 8].

1. Описание способа обезличивания ПД

В предлагаемом способе обезличивания ПД используется исходная таблица ПД, которая может быть построена из имеющегося множества структурированных ПД, предварительно собранных в удаленных хранилищах в виде множеств файлов, таблиц баз данных. Для построения исходной таблицы требуется привести ПД каждого субъекта к единой структуре, которая будет представлена в строках исходной таблицы.

Таким образом, строка исходной таблицы представляет ПД одного субъекта, а каждый столбец (атрибут) содержит множество однотипных данных, соответствующих элементам структуры, по всем субъектам, занесенным в таблицу.

Допускается отсутствие данных о субъектах в отдельных элементах атрибутов, в этом случае элементом атрибута является специальный набор символов.

Способ основан на применении алгоритма многоэтапного преобразования множества данных к каждому атрибуту исходной таблицы ПД. На каждом этапе множество данных атрибута разбивается по заданному правилу на блоки, которые подвергаются операции перестановки, после чего этап повторяется. Число этапов, правила разбиения на блоки и перестановки блоков на этапе могут меняться для каждого атрибута.

Правила разбиения на блоки, параметры перестановки блоков и число этапов для каждого атрибута составляют ключ преобразования.

В результате преобразования меняется порядок данных во множестве элементов каждого атрибута и, соответственно, содержание строк в таблице, но сохраняется семантика каждого элемента. Предложенный способ позволяет рассеивать элементы атрибута в пределах всего множества элементов, устранить связи между структурными элементами каждой строки исходной таблицы ПД, что значительно усложняет возможность деперсонализации при неизвестном ключе преобразования, даже при наличии множества известных ПД, находящихся в исходной таблице.

Известны способы, применяющие разбиения и перестановки (перемешивание) [9–12]. Общими недостатками этих способов являются:

- сохранение состава подмножеств (блоков) ПД, получаемых после разбиения исходного множества, что приводит к слабому рассеиванию элементов в пределах множества всех элементов атрибута;
- циклические сдвиги, существенно ограничивающие число вариантов перестановок и не увеличивающие рассеивание;
- применение, как правило, одного этапа в преобразовании.

Все это значительно ослабляет защищенность этих способов.

2. Математическая модель преобразования

Пусть исходная таблица ПД $T_0(N, M)$ содержит N записей (строк) и M атрибутов (столб-

цов). Все строки и атрибуты имеют уникальные номера от 1 до N и от 1 до M , соответственно.

Далее будем рассматривать атрибут номер j исходной таблицы, который содержит упорядоченное занумерованное множество из N элементов (данных о субъекте) — $C_{j0} = \{c_{j0}(m)\}$. Здесь $c_{j0}(m)$ — уникальный номер элемента, имеющего порядковый номер m (стоящего на месте m) во множестве C_{j0} . В исходной таблице $T_0(N, M)$ порядковый номер элемента атрибута равен номеру строки, в которой этот элемент находится, и, в данном случае, уникальный номер элемента равен его порядковому номеру в атрибуте исходной таблицы: $c_{j0}(m) = m$, ($1 < c_{j0}(m) \leq N$; $1 \leq m \leq N$). Таким образом, уникальные номера элементов во всех атрибутах, относящихся к одному субъекту (находящихся в одной строке исходной таблицы) совпадают. Уникальные номера являются постоянными и однозначно определяют элементы атрибутов ПД.

Преобразования приводят к несовпадению уникальных и порядковых номеров элементов.

Преобразование множества данных атрибута номер j ($j = 1, 2, \dots, M$) происходит в несколько этапов, число которых R_j ($1 \leq R_j < \infty$).

Каждый этап номер r ($r = 1, 2, \dots, R_j$) содержит два шага.

На первом шаге происходит разбиение множества данных атрибута на непересекающиеся блоки данных. Число блоков равно Z_{jr} , где $1 < Z_{jr} \leq N$. Каждый блок номер i содержит $k_{jr}(i)$ последовательно занумерованных элементов с номерами от $\sum_{n=1}^{i-1} k_{jr}(n) + 1$ до $\sum_{n=1}^i k_{jr}(n)$ ($i = 1, 2,$

$\dots, Z_{jr}$; $0 < k_{jr}(i) < N$). Кроме того, $\sum_{n=1}^{Z_{jr}} k_{jr}(n) = N$,

т. е. блоки разбиения содержат все элементы атрибута.

Число блоков Z_{jr} атрибута j на этапе r и вектор числа элементов в каждом блоке $\mathbf{k}_{jr} = (k_{jr}(1), k_{jr}(2), \dots, k_{jr}(Z_{jr}))$ являются уникальными для каждого атрибута и каждого этапа.

На втором шаге этапа проводится перестановка блоков атрибута, задаваемая матрицей перестановки $\mathbf{X}_{jr} = \|x_{ij}\|$ ($i = 1, 2; j = 1, 2, \dots, Z_{jr}$). Здесь x_{1n} — порядковый номер блока при разбиении ($1 \leq x_{1n} \leq Z_{jr}$ и, как правило, $x_{1n} = n$), а x_{2n} — номер блока разбиения, стоящего на месте номер n после перестановки ($1 \leq x_{2n} \leq Z_{jr}$). В результате получается преобразованное множество атрибута C_{jr}^* , где r — номер этапа ($r = 1, 2, \dots, R_j$). После преобразования на этапе r изменяется расположение элементов атрибута и определяется $c_{jr}(m)$ — уникальный номер

элемента, имеющего после преобразования порядковый номер m ($m = 1, 2, \dots, N$).

Исходным множеством для преобразования на первом этапе является множество данных атрибута номер j исходной таблицы — C_{j0} , исходными множествами для преобразований на этапах 2, 3, ..., R_j являются множества C_{j1}^* , C_{j2}^* , ..., $C_{j(R_j-1)}^*$. Множество $C_{jR_j}^*$ есть окончательный результат преобразования множества данных атрибута номер j . После окончания проведения преобразований всех атрибутов получается таблица обезличенных ПД — $\mathbf{T}^*(N, M)$.

Преобразование исходной таблицы ПД на каждом этапе изменяет порядковые номера элементов всех атрибутов, разрушая связи между элементами строки исходной таблицы.

Вычислим новый порядковый номер n элемента атрибута номер j после выполнения этапа r , если по окончании предыдущего этапа $(r-1)$ элемент имел порядковый номер d . Разбиение и перестановка задаются вектором $\mathbf{k}_{jr} = (k_{jr}(1), k_{jr}(2), \dots, k_{jr}(Z_{jr}))$ и матрицей $\mathbf{X}_{jr} = \|x_{ij}\|$.

Определим сначала h — номер блока, куда после разбиения входит элемент номер d . Этот номер h вычисляется из условия

$$\sum_{i=1}^{h-1} k_{jr}(x_{1i}) \leq d \leq \sum_{i=1}^h k_{jr}(x_{1i}). \quad (1)$$

Номер элемента внутри блока h вычисляется по формуле

$$b = d - \sum_{i=1}^{h-1} k_{jr}(x_{1i}). \quad (2)$$

После перестановки блок номер h будет иметь порядковый номер z , который вычисляется из условия

$$x_{2z} = h. \quad (3)$$

Номер элемента после перестановки вычисляется по формуле

$$n = b + \sum_{i=1}^{z-1} k_{jr}(x_{2i}). \quad (4)$$

В результате определяется новый порядковый номер элемента после проведения этапа. Этот номер становится номером элемента перед следующим этапом преобразования. Уникальный номер элемента не меняется и всегда остается равным порядковому номеру этого элемента в атрибуте исходной таблицы $\mathbf{T}_0(N, M)$ и, таким образом, $c_{j(r-1)}(d) = c_{jr}(n)$.

При проведении деобезличивания для определенного субъекта ПД задается значение элемента атрибута этого субъекта, устанавли-

вается его порядковый номер в соответствующем атрибуте таблицы $\mathbf{T}^*(N, M)$, после чего вычисляется значение уникального номера этого элемента. Затем разыскиваются все элементы других атрибутов, имеющие этот же уникальный номер.

Так, если после этапа r элемент атрибута j имел порядковый номер n , то вычислим порядковый номер этого элемента после этапа $(r-1)$. Используя матрицу перестановки $\mathbf{X}_{jr} = \|x_{ij}\|$, можно вычислить номер h блока разбиения, куда входит элемент с порядковым номером d . Этот номер h вычисляется из условия

$$\sum_{i=1}^{h-1} k_{jr}(x_{2i}) \leq n \leq \sum_{i=1}^h k_{jr}(x_{2i}). \quad (5)$$

Далее номер элемента внутри блока с номером h вычисляется по формуле

$$b = n - \sum_{i=1}^{h-1} k_{jr}(x_{2i}). \quad (6)$$

До перестановки блок имеет порядковый номер h , и порядковый номер элемента равен

$$d = b + \sum_{i=1}^{h-1} k_{jr}(x_{1i}). \quad (7)$$

Таким образом, показано, как вычислить порядковые номера элементов после преобразований и как вычислить порядковые номера для деобезличивания (в обратном порядке).

Параметры преобразования атрибута j на этапе r задаются множеством $\mathbf{K}_{jr} = \{Z_{jr}, \mathbf{k}_{jr}, \mathbf{X}_{jr}\}$, которое является ключом преобразования атрибута на этапе. Вектор $\mathbf{K}_j = (R_j, \mathbf{K}_{j1}, \mathbf{K}_{j2}, \dots, \mathbf{K}_{jR_j})$ задает множество ключей для всех этапов преобразования атрибута j ($j = 1, 2, \dots, M$).

Множество ключей для всех атрибутов и всех этапов преобразования исходной таблицы $\mathbf{K}(\mathbf{T}_0(N, M)) = \{\mathbf{K}_j\}$ ($j = 1, 2, \dots, M$) называется ключом преобразования исходной таблицы.

Эффективность преобразования определяется рассеиванием элементов атрибута по всему множеству элементов. Рассеивание позволяет избежать устойчивых групп элементов, имеющих в исходном множестве, наличие которых упрощает проведение атак на деобезличивание при отсутствии ключей преобразования.

Для оценки качества проводимых преобразований введем меру рассеивания элементов атрибута j на этапе r :

$$R_{jr} = \frac{1}{(N-1)} \sum_{i=1}^{N-1} |c_{jr}(i+1) - c_{jr}(i)|. \quad (8)$$

Здесь величина $|c_{jr(i+1)} - c_{jri}|$ — это "расстояние" между соседними элементами с номерами $(i + 1)$ и i ($i = 1, 2, \dots, N - 1$) в таблице $T_0(N, M)$ после проведения этапа r . Мера рассеивания для всей таблицы $T^*(N, M)$ вычисляется по формуле

$$R(T^*) = \frac{1}{M} \sum_{j=1}^M R_{jR_j}. \tag{9}$$

Предлагаемая мера дает возможность количественно оценить усредненное значение "расстояний" между соседними элементами исходной таблицы после деобезличивания.

Отметим, что циклическая перестановка незначительно влияет на значение меры рассеивания элементов атрибута.

Пример 1.

Пусть задана исходная таблица $T_0(20,1)$, где атрибут номер 1 ($M = 1$) имеет 20 элементов с номерами от 1 до 20 ($N = 20$).

Рассмотрим первый этап преобразования. Число битов $Z_{11} = 5$. Разбиение задается вектором $k_{11} = (k_{11}(1) = 5, k_{11}(2) = 3, k_{11}(3) = 4, k_{11}(4) = 2, k_{11}(5) = 6)$.

Матрица $X_{11} = \|x_{ij}\|$, ($i = 1, 2; j = 1, 2, \dots, 5$) имеет вид: $X_{11} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 1 & 2 & 4 \end{pmatrix}$.

До разбиения исходное множество элементов на этапе 0 C_{10} имеет вид (для простоты значения элементов совпадают с их уникальными номерами):

$$C_{10} = \{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15\ 16\ 17\ 18\ 19\ 20\}.$$

После разбиения на первом шаге первого этапа получены следующие блоки:

$$\{(1\ 2\ 3\ 4\ 5)\ (6\ 7\ 8)\ (9\ 10\ 11\ 12)\ (13\ 14)\ (15\ 16\ 17\ 18\ 19\ 20)\}.$$

После перестановки получено новое множество элементов C_{11}^* (показана перестановка блоков), где изменились порядковые номера элементов, но остались неизменными уникальные номера:

$$C_{11}^* = \{(9\ 10\ 11\ 12)\ (15\ 16\ 17\ 18\ 19\ 20)\ (1\ 2\ 3\ 4\ 5)\ (6\ 7\ 8)\ (13\ 14)\}.$$

Вычислим новый номер n элемента, который до преобразования имел, например, начальный номер $d = 9$. Используя соотношение

$$(1), \text{ получим: } 8 = \sum_{i=1}^2 k_{11}(x_{1i}) \leq 9 \leq \sum_{i=1}^3 k_{11}(x_{1i}) = 12,$$

откуда $h = 3$ и, используя (2), получим: $b = 9 - \sum_{i=1}^2 k_{11}(x_{1i}) = 1$. Далее $z = 1$, поскольку из соотношения (3) будем иметь: $x_{21} = h = 3$.

Новый номер элемента, имевшего до преобразования номер 9, вычисляем, используя формулу (4):

$$n = 1 + \left(\sum_{i=1}^0 k_{11}(x_{1i}) = 0 \right) = 1.$$

Обратная процедура: пусть после этапа 1 порядковый номер элемента $n = 1$. Тогда из соотношения (5) получим

$$0 = \sum_{i=1}^0 k_{11}(x_{2i}) \leq 1 \leq \sum_{i=1}^1 k_{11}(x_{2i}),$$

Таблица 1

Преобразования атрибута

Этап r	Вектор разбиения	Матрица перестановки	Множество значений (уникальных номеров) элементов после преобразования на этапе $r - C_{1r}^*$
0			1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
1	(5,3,4,2,6)	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 1 & 2 & 4 \end{pmatrix}$	9 10 11 12 15 16 17 18 19 20 1 2 3 4 5 6 7 8 13 14
2	(3,4,3,5,2,3)	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 3 & 4 & 2 & 5 & 1 \end{pmatrix}$	8 13 14 18 19 20 1 2 3 4 5 12 15 16 17 6 7 9 10 11
3	(4,6,5,5)	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$	5 12 15 16 17 8 13 14 18 6 7 9 10 11 19 20 1 2 3 4
4	(4,6,5,5)	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$	7 9 10 11 19 5 12 15 16 20 1 2 3 4 17 8 13 14 18 6
5	(7,3,3,4,3)	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 3 & 5 & 2 \end{pmatrix}$	4 17 8 13 7 9 10 11 19 5 12 1 2 3 14 18 6 15 16 20

откуда $x_{21} = 3$ и, следовательно, $h = 3$. Используя формулу (6), получим:

$$b = n - \left(\sum_{i=1}^0 k_{jr}(x_{2i}) = 0 \right) = 1.$$

Далее из формулы (7) имеем

$$d = b + \sum_{i=1}^{h-1} k_{11}(x_{1i}) = 9.$$

В табл. 1 приведены результаты преобразований таблицы $T_0(20,1)$ на пяти этапах. Исходное множество соответствует этапу номер 0.

Из табл. 1 видно, как меняется расстановка элементов атрибута в результате преобразований.

Значения меры рассеивания, вычисленные с применением формулы (5), по результатам преобразований, приведенных в табл. 1, равны: $R_{10} = 1$; $R_{11} = 2,26$; $R_{12} = 2,79$; $R_{13} = 4,16$; $R_{14} = 5,63$; $R_{15} = 6,31$, откуда видно, что рассеивание элементов увеличивается с каждым этапом.

Однако понятно, что мера рассеивания имеет верхнюю границу, после достижения которой, преобразования становятся неэффективными.

Применяя приведенные формулы для каждого элемента исходного множества и для всех этапов, можно определить новые номера элементов каждого атрибута номер j после окончания преобразования, т. е. после проведения R_j этапов ($j = 1, 2, \dots, M$).

3. Оценка защищенности преобразования

Защищенность таблицы $T^*(N, M)$ от атак, направленных на деобезличивание, при отсутствии данных о ключе преобразования, можно оценить числом возможных вариантов преобразования.

Сначала вычислим число вариантов разбиения на блоки. Если имеются N элементов, то между ними будет $(N - 1)$ промежутков, которые имеют порядковые номера от 1 до $(N - 1)$. Число разбиений N элементов на k блоков равно числу возможных сочетаний из $(N - 1)$ по $(k - 1)$, при этом наборы не будут повторяться. Таким образом, число возможных разбиений N элементов на k блоков равно

$$U_0(k, N) = C_{N-1}^{k-1} = \frac{(N-1)!}{(k-1)!(N-k)!}. \quad (10)$$

Общее число возможных разбиений N элементов атрибута с учетом (10) равно

$$\begin{aligned} U_0(N) &= \sum_{k=1}^{N-1} \frac{(N-1)!}{(k-1)!(N-k)!} = \\ &= \sum_{k=1}^{N-1} \left[\prod_{i=1}^{k-1} \left(\frac{N}{(k-i)} - 1 \right) \right]. \end{aligned} \quad (11)$$

Нижняя оценка величины $U_0(N)$ составляет

$$U_0(N) = \sum_{k=1}^{N-1} \left[\prod_{i=1}^{k-1} \left(\frac{N}{(k-i)} - 1 \right) \right] > \sum_{k=1}^{N-1} \left(\frac{N}{k} \right)^{k-1}.$$

Далее на этапе каждое разбиение из k блоков подвергается перестановке, и возможное число перестановок для этого разбиения равно $k!$.

Следовательно, общее число возможных вариантов преобразования атрибута, содержащего N элементов, на одном этапе, с учетом (11) равно

$$\begin{aligned} U(N) &= \sum_{k=1}^{N-1} (U_0(N)k!) = \sum_{k=1}^{N-1} \frac{(N-1)!k!}{(k-1)!(N-k)!} = \\ &= \sum_{k=1}^{N-1} \frac{(N-1)!k}{(N-k)!} = \sum_{k=1}^{N-1} \left[k \prod_{i=1}^{k-1} (N-i) \right]. \end{aligned} \quad (12)$$

Если полное преобразование одного атрибута номер j проводится за R_j этапов, то число вариантов полного преобразования этого атрибута вычисляется с использованием выражения (12) по формуле

$$U_j(N) = \prod_{m=1}^{R_j} U(N) = \prod_{m=1}^{R_j} \left[\sum_{k=1}^{N-1} \left(k \prod_{i=1}^{k-1} (N-i) \right) \right]. \quad (13)$$

Для исходной таблицы $T_0(N, M)$, содержащей M атрибутов, число вариантов преобразований вычисляется с учетом соотношения (13) по формуле

$$T(N, M) = \prod_{j=1}^M U_j(N). \quad (14)$$

Формулы (10)–(14) показывают, что число возможных вариантов преобразования таблицы $T_0(N, M)$ очень велико (особенно с учетом того, что реальные значения N лежат в диапазоне от 10^3 до 10^8). Например, при $N = 100$, $M = 10$, $R_j = 1$, $Z_{j1} = 10$ ($j = 1, 2, \dots, 10$) получим $U_0(10, 100) = 10^5 \cdot 17310308 \geq 10^{12}$. При фиксированном $k = 10$ $U(100) \geq 10^{13}$, откуда $T(100, 10) \geq 10^{130}$.

Для увеличения числа вариантов при больших значениях N можно увеличить число M за счет разбиения элементов одного атрибута на несколько частей, каждая из которых будет входить в состав своего атрибута (создание новых атрибутов). Например, можно телефон, название улицы в адресе, фамилию субъекта

разделить на несколько частей и рассматривать их как отдельные структурные единицы, а их множества рассматривать как атрибуты.

Пример 2.
В табл. 2 первом столбце представлены уникальные номера элементов для каждого атрибута.

Ключи преобразований атрибутов табл. 2:
• **Атрибут 1 (Фамилия)**

$$K_1 = (3, (K_{11} = \{3, (4,6,4), X_{11} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}))),$$
$$(K_{12} = \{3, (6,5,3), X_{12} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}))),$$
$$(K_{13} = \{2, (5,9), X_{13} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix})));$$

• **Атрибут 2 (Имя)**

$$K_2 = (2, (K_{21} = \{4, (3,5,2,4), X_{21} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}))),$$
$$(K_{22} = \{3, (7,3,4), X_{22} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix})));$$

• **Атрибут 3 (Отчество)**

$$K_3 = (3, (K_{31} = \{2, (6,8), X_{31} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}))),$$
$$(K_{32} = \{4, (5,3,2,4), X_{32} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}))),$$
$$(K_{33} = \{4, (2,3,4,5), X_{33} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 \end{pmatrix})));$$

• **Атрибут 4 (Место рождения)**

$$K_4 = (3, (K_{41} = \{4, (4,3,2,5), X_{41} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix}))),$$
$$(K_{42} = \{3, (3,6,5), X_{42} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}))),$$
$$(K_{43} = \{3, (6,1,7), X_{43} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix})));$$

• **Атрибут 5 (Год рождения)**

$$K_5 = (2, (K_{51} = \{2, (5,9), X_{51} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}))),$$
$$(K_{52} = \{3, (3,5,6), X_{52} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}))).$$

В табл. 3 в первом столбце представлены порядковые номера элементов атрибутов после преобразования, уникальные номера нужно вычислять по исходной таблице (см. табл. 2).

Таблица 2

Исходная таблица T_0 (14,5)

№	Фамилия	Имя	Отчество	Место рождения	Год рождения
1	Иванов	Петр	Сергеевич	Москва	1940
2	Петров	Иван	Петрович	Рязань	1954
3	Иванов	Сергей	Андреевич	Москва	1947
4	Сидоров	Петр	Иванович	Тверь	2000
5	Николаев	Сергей	Васильевич	Киев	2002
6	Сергеев	Евгений	Петрович	Москва	1987
7	Митин	Алексей	Дмитриевич	Калуга	1991
8	Петров	Игорь	Николаевич	Ржев	1958
9	Власов	Анатолий	Алексеевич	Мурманск	1954
10	Куракин	Сергей	Александрович	Казань	1936
11	Лебедев	Иван	Николаевич	Уфа	2005
12	Крылов	Иван	Андреевич	Минск	2003
13	Рублев	Антон	Семенович	Москва	1950
14	Пушкин	Александр	Сергеевич	Курск	1928

Таблица 3

Таблица обезличенных данных T^* (14,5),
полученная после преобразования таблицы T_0 (14,5)

№	Фамилия	Имя	Отчество	Место рождения	Год рождения
1	Рублев	Сергей	Васильевич	Москва	1954
2	Пушкин	Евгений	Петрович	Калуга	1936
3	Иванов	Алексей	Андреевич	Рязань	2005
4	Николаев	Игорь	Семенович	Москва	2003
5	Сергеев	Петр	Сергеевич	Тверь	1950
6	Митин	Иван	Александрович	Ржев	1987
7	Петров	Сергей	Николаевич	Мурманск	1991
8	Власов	Иван	Андреевич	Казань	1958
9	Куракин	Иван	Иванович	Уфа	1928
10	Петров	Антон	Сергеевич	Минск	1940
11	Иванов	Александр	Петрович	Москва	1954
12	Сидоров	Анатолий	Дмитриевич	Курск	1947
13	Лебедев	Сергей	Николаевич	Москва	2000
14	Крылов	Петр	Алексеевич	Киев	2002

Заключение

Предлагаемый способ обладает следующими достоинствами:

- исходная таблица ПД может быть получена из множества различных хранилищ ПД, представленных в структурированной форме;
- число атрибутов исходной таблицы может превышать число структурных единиц в начальной форме представления ПД, что позволит увеличить защищенность деобезличенных данных;
- для осуществления преобразования исходной таблицы используются уникальные параметры — ключи преобразования, которые хранятся в защищенной форме отдельно от места хранения персональных данных;
- при обезличивании не затрагиваются сами ПД, что обеспечивает сохранение их целостности;
- имеется возможность проводить поиск и обработку ПД по таблице обезличенных данных без предварительного деобезличивания всей таблицы;
- при проведении процедуры деобезличивания достаточно задавать значения одного или нескольких атрибутов, по которым восстанавливаются записи исходной таблицы, имеющие заданные значения атрибутов.
- высокая защищенность от атак, направленных на деобезличивание, при неизвестных ключах преобразования, обеспечивается очень большим числом возможных вариантов преобразований исходной таблицы;
- имеется возможность динамического изменения ключей преобразования на каждом этапе и для каждого атрибута;
- существует возможность хранения атрибутов таблицы обезличенных данных в различных базах данных и на различных компьютерах (серверах);
- есть возможность добавления новых атрибутов в исходную таблицу, и их преобразования без деобезличивания таблицы обезличенных данных;
- преобразование состоит из повторяющихся простых однотипных процедур, что существенно упрощает его программную реализацию.

Для практической реализации предложенного способа целесообразно использовать следующие рекомендации:

- проводить формирование исходной таблицы ПД на отдельном сервере с передачей та-

блицы на сервер обезличивания по его запросу. Это позволит совместить во времени различные операции;

- при формировании исходной таблицы ПД провести нумерацию строк для формирования уникальных номеров элементов;
- для занесения новых ПД использовать либо предварительно подготовленную избыточную исходную таблицу ПД, содержащую пустые строки, либо создавать несколько исходных таблиц меньшей размерности добавляя в них новые данные.

Большая размерность пространства ключей может быть значительно сокращена, если задавать функциональные связи между ключами. Например, при заданном числе блоков Z_{jr} при разбиении можно задать число элементов в блоке формулой $k_{jr}(i) = f_{jr}(i, Z_{jr})$. Для матриц перестановок также могут быть заданы аналитические зависимости (формальные правила) для их составления.

Предложенный способ обобщает известные способы, основанные на перемешивании данных, увеличивая при этом рассеивание элементов таблицы и защищенность обезличенных данных.

Список литературы

1. **Федеральный закон** от 27 июля 2006 г. N 152-ФЗ "О персональных данных".
2. **Столбов А. П.** Обезличивание персональных данных в здравоохранении // Врач и информационные технологии. 2017. № 3. С. 76—91.
3. **Рябко С. Д.** Об обезличивании персональных данных // Информационная безопасность. 2009. № 5. URL: www.itsec.ru/articles2/bypub/insec-5-2009.
4. **Sweeney L.** k-anonymity: a model for protecting privacy // International Journal on Uncertainty, Fuzziness and Knowledge-based Systems. 2002. Vol. 10, N. 5. P. 557—570.
5. **Трифорова Ю. В., Жаринов Р. Ф.** Возможности обезличивания персональных данных в системах, использующих реляционные базы данных // Доклады ТУСУР. 2014. № 2 (32). С. 188—194.
6. **Бондаренко К. О., Козлов В. А.** Универсальный быстроедействующий алгоритм процедур обезличивания данных // Изв. ЮФУ. Технические науки. 2015. № 11 (172). С. 130—142.
7. **Приказ** Роскомнадзора от 5 сентября 2013 года N 996 "Об утверждении требований и методов по обезличиванию персональных данных".
8. **Методические** рекомендации по применению приказа Роскомнадзора от 5 сентября 2013 г. № 996 "Об утверждении требований и методов по обезличиванию персональных данных". Утверждены руководителем Роскомнадзора 13.12.2013.
9. **Саксонов Е. А., Шередин Р. В.** Процедура обезличивания персональных данных // Наука и образование: элек-

тронное научно-техническое издание. 2011. № 3. Эл № ФС 77 — 30569. Государственная регистрация № 0421100025.

10. **Куракин А. С.** Алгоритм деперсонализации персональных данных // Научно-технический вестник информационных технологий, механики и оптики. СПб: СПбГУ ИТМО, 2012. № 6. С. 130—135.

11. **Карпова И. П.** О реализации метода обезличивания персональных данных // Вестник компьютерных и информационных технологий. 2013. № 6. С. 56—60.

12. **Бондарец Е. А.** Модификация алгоритма перемешивания персональных данных // Ученые заметки ТОГУ: электронное научное издание. 2015. Т. 6, № 2. С. 282—288.

Е. А. Saksonov, Chief Specialist, Professor, e-mail: saksmiem@mail.ru,
Moscow Technical University of Communications and Informatics, Moscow, Russian Federation

Method of Depersonalization of Personal Data

A method of depersonalization of large amounts of personal data is proposed. The method preserves the structure and semantics of data, allows you to increase the security of depersonalized data and process personal data without prior depersonalization. A mathematical model of the method is developed. Estimates of security depersonalized data are obtained.

Keywords: personal data, depersonalization of personal data, transformation of personal data tables, security of depersonalized personal data

DOI: 10.17587/it.27.314-321

References

1. **Federal Law** of July 27, 2006 No. 152-FZ "About personal data".

2. **Stolbov A. P.** Depersonalization of personal data in health-care, *Vrach I informacionnie tehnologii*, 2017, no. 3, pp. 76—91 (in Russian).

3. **Ryabko S. D.** On depersonalization of personal data, *Informacionnaya bezopasnost*, 2009, no. 5, available at: www.itsec.ru/articles2/bypub/insec-5-2009 (in Russian).

4. **Sweeney L.** k-anonymity: a model for protecting privacy, *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002, vol.10, no. 5, pp. 557—570.

5. **Trifonova Yu. V., Zarinov R. F.** Opportunities for depersonalization of personal data in systems using relational databases, *Doklady TUSUR*, 2014, no. 2 (32), pp. 188—194 (in Russian).

6. **Bondarenko K. O., Kozlov V. A.** Universal fast-acting algorithm for the depersonalization of personal data, *Izv. YuFU. Tehnicheskie nauki*, 2015, no. 11 (172), pp. 130—142 (in Russian).

7. **Order** of Roskomnadzor dated September 5, 2013 N 996 "On approval of requirements and methods for anonymization of personal data."

8. **Recommendations** for the application of the Roskomnadzor order from 5.09.2013 № 996 "On approval of requirements and methods for depersonalization of personal data". Utverzdeny rukovoditelem Roskomnadzora 13.12.2013.

9. **Saksonov E. A., Sheredin R. V.** Procedure for depersonalization of personal data, *Nauka I obrazovanie: elektronnoe nauchno-tehnicheskoe izdanie*, 2011, no. 3, EL № FS 77-30569 (in Russian).

10. **Kurakin A. S.** The algorithm of the anonymisation of personal data, *Nauchno-tehnicheskii vestnik informacionnykh tehnologij, mehaniki I optiki*, 2012, no. 6, pp. 130—135 (in Russian).

11. **Karpova I. P.** About the implementation of the method of depersonalization of personal data, *Vestnik Komputernykh i Informacionnykh Tehnologij*, 2013, no. 6, pp. 56—60 (in Russian).

12. **Bondarec E. A.** Modification of the algorithm for mixing personal data, *Uchenie zametki TOGU: elektronnoe nauchnoe izdanie*, 2015, vol. 6, no. 2, pp. 282—288 (in Russian).

Е. А. Гостева, e-mail: kategosteva@yandex.ru, **В. В. Ланин**, ст. преподаватель, e-mail: vlanin@live.com, Национальный исследовательский университет "Высшая школа экономики", г. Пермь

Реализация системы интеллектуального поиска информации в промышленных стандартах

Статья посвящена разработке системы интеллектуального поиска информации по промышленным стандартам. Представлено описание предметной области, проектирование и основные стадии разработки системы: создание модуля извлечения информации из промышленных стандартов, реализация веб-сервиса с помощью фреймворка Flask и клиентского веб-приложения на React JS. Использование данной системы инженерами и разработчиками программного обеспечения позволит эффективно управлять базой определений промышленных стандартов, правильно их интерпретировать и использовать в соответствии с выбранной областью знаний.

Ключевые слова: промышленные стандарты, умное производство, интеллектуальный поиск

Введение

Извлечение информации из текстов и представление ее в виде формальной системы знаний — важная задача в области автоматической обработки текстов на естественном языке. Постоянно растущий объем текстов и документов, которые находятся в свободном доступе, с одной стороны, расширяет границы для изучения различных областей знаний, с другой — затрудняет процесс поиска и выделения значимой информации. Особенно это важно при использовании стандартов в промышленности, где нужно быстро оперировать данными для ускорения процесса производства, выявления новых трендов и имеющихся наработок, уменьшения потери прибыли из-за недостаточности информации. Основная проблема на данный момент заключается в невозможности изучить всю содержащуюся в различных документах информацию за достаточно непродолжительное время. В процессе ее решения появилось направление "information extraction", задача которого заключается в извлечении структурированных данных из слабо-структурированных или неструктурированных документов для облегчения процесса обработки и анализа информации. Задачи из этой области актуальны в производстве, экономике и научной деятельности [1–5].

В рамках данной работы объектом исследования являются промышленные стандарты, предметом — системы интеллектуального поиска в промышленных стандартах. Цель —

разработка системы интеллектуального поиска информации в промышленных стандартах. Для достижения поставленной цели необходимо проанализировать подходы и инструментальные средства для извлечения информации из документов, описать функциональное назначение системы, реализовать модуль извлечения информации, разработать веб-сервис и клиентское приложение для работы с ним.

1. Анализ задачи интеллектуального поиска по стандартам

Современной компании необходимо эффективно использовать имеющуюся информацию, уметь применять ее на практике при разработке продуктов, развитии сферы услуг, создании потребительской ценности. Большая роль в данных процессах отводится управлению знаниями и непрерывному обучению, поэтому лидерами новой экономики стали компании, которые научились быстрее своих конкурентов находить и внедрять существующие знания в свое производство.

"Корпоративные знания рассматриваются менеджментом как один из важнейших активов или как интеллектуальный капитал компании, а управление знаниями — как одно из эффективных направлений управления бизнесом" [6]. Формирование системы управления знаниями позволяет не только поддерживать конкурентоспособность, но и повысить скорость реакции на изменения внешней среды, а также избежать

проблемы недополученной прибыли. Когда все сотрудники говорят "на одном языке", систематизация, получение, классифицирование и передача знаний выходит на более высокий уровень, сокращается время включения сотрудников и принятия решений.

Корпоративные знания имеют несколько категорий — знания о бизнес-процессах (основная производственная информация), корпоративной культуре (взаимодействие с заказчиками, клиентами, коллегами), внешней среде компании (информация о покупателях и конкурентах, знания о продукте), личные знания (компетенции сотрудников). В рамках данной работы рассматриваются источники знаний о бизнес-процессах компании — стандарты. Стандарт — это документ, который содержит требования, правила и руководящие указания для процесса, продукта или услуги, необходим для достижения оптимальной степени согласованности и упорядоченности в заданной области знаний. Для производства продукта или услуги стандарты полезны определением оптимальных параметров, описанием методов оценки соответствия определенному уровню качества и условий их использования [7]. Главным преимуществом использования стандартов является передача технологий. Поскольку стандарты включают результаты достижений в какой-либо области науки, они отражают современное состояние в области технического развития. На данный момент существует несколько сотен промышленных стандартов, среди которых для разработки системы были выбраны стандарты проекта oneM2M в области Интернета вещей [8].

Проект oneM2M — это глобальное партнерство, основанное в 2012 г. ведущими организациями по разработке стандартов в области телекоммуникаций. Цель oneM2M заключается в разработке технических стандартов для обеспечения совместимости в отношении архитектуры, спецификаций API, решений по обеспечению безопасности и регистрации межмашинных технологий и технологий Интернета вещей на основе требований. Опубликованные стандарты позволяют экосистеме поддерживать широкий спектр приложений и услуг, например, Умный город, Умный дом, интеллектуальный транспорт, телематика, общественная безопасность и здравоохранение.

Структура представленных стандартов в большинстве случаев схожа: имеется титульный лист с информацией о документе (номер, название, дата и назначение стандарта), раздел с описанием области применения, раздел со ссылками на другие источники, раздел с основными определениями и сокращениями, да-

oneM2M	
TECHNICAL SPECIFICATION	
Document Number	TS-0001-V1.0.2
Document Name	Security Services
Date	2019-04-05
Abstract	The TS defines security solutions for M2M systems.
The TS defines security solutions for M2M systems.	
This Specification is provided for future development work within oneM2M only. The Partners accept no liability for any use of this Specification.	
The present document has not been subject to any approval process by the oneM2M Partners Type 1. Published oneM2M specifications and reports for implementation should be obtained via the oneM2M Partners Publications Office.	
1	Scope
2	References
2.1	Normative references
2.2	Informative references
3	Definitions, symbols and abbreviations
3.1	Definitions
3.2	Symbols
3.3	Abbreviations
4	Conventions
5	Security Architecture
5.1	Overview
5.1.1	Introduction
5.1.2	Identification and Authentication
5.1.3	Authorization
5.1.4	Privacy Management
5.1.5	Security Layers
5.2	Security Services Layer
5.2.1	Secure Environment Activation Layer
5.2.2	Integration with overall oneM2M architecture
6	Security Services and Interactions
6.1	Security Integration in oneM2M flow of events
6.1.1	Interactions between layers
6.1.2	High level sequence of events
6.1.2.1	Establishment phase
6.1.2.2	Operational phase
6.1.2.3	M2M Service Access
6.1.2.4	Authentication to access M2M resources
6.1.2.5	Authorization to access M2M resources
6.1.2.6	Security for ambient group Event procedures

Рис. 1. Пример структуры стандарта oneM2M

лее следуют разделы, содержащие соответствующую назначению стандарта информацию (рис. 1). Все стандарты представлены на сайте oneM2M в виде "PDF" документов [9].

2. Анализ задачи извлечения информации из текстов

Задача по извлечению знаний из неструктурированных источников на естественном языке относится к информационному поиску, предполагающему нахождение релевантной информации. Как правило, такие источники не обладают ни одним видом идентификации искомой информации, например, разметкой или метаданными, поэтому извлеченные данные в первую очередь записываются в формальном виде (в виде таблицы или базы данных), что позволяет впоследствии более эффективно их обрабатывать и применять, например, строить различные модели или представлять их в виде связанного графа.

К видам извлекаемой информации относятся: именованные сущности, атрибуты объектов, отношения между объектами, факты и события, термины предметной области и их связи, ключевые слова документа, отзывы и мнения о товарах и услугах.

Для решения задачи извлечения информации из тестов используют три подхода: основанный на правилах, основанный на машинном обучении и гибридный, объединяющий два предыдущих.

Первый подход основан на том, что извлекаемая информация имеет определенные языковые конструкции, которые вручную описываются в виде шаблонов и правил их обработки и применяются к тексту, пока не будут найдены все соответствия шаблону. Например, из предложения "Traditionally, the term management refers to the activities (and often the group of people) involved in the four general functions listed below" с помощью правила "ЕСЛИ за словами the term, The term следует существительное ТО извлечь слово с меткой термин" будет извлечен термин "management".

Формальное описание языковой конструкции информации, которую необходимо найти в тексте, называется лексико-синтаксическим шаблоном. Шаблон может быть составлен с помощью регулярных выражений, например, один из вариантов поиска термина будет выглядеть как "[The|the] term NN", где "[The|the]" — варианты написания артикля (в начале или в середине предложения), необходимо наличие одного варианта из квадратных скобок, "term" — обязательное слово, "NN" — указывает на слово-существительное.

Формированием шаблонов и правил занимаются эксперты в данной области знаний, что позволяет достигать высокой точности извлечения информации. Для записи паттернов часто применяются специальные языки и поддерживающие их системы, примеры некоторых из них представлены далее.

Ограничением данного подхода являются тексты общих предметных областей, описание шаблонов для которых становится уже достаточно трудоемким, так как необходимо учесть всевозможные правила и языковые конструкции, а также смену шаблонов при изменении предметной области.

Достоинства данного подхода заключаются в более точных результатах в узких предметных областях (как в данной работе — стандарты в области Интернета вещей) с небольшим разнообразием языковых конструкций (стандарты написаны на английском языке, который имеет строго установленный и соблюдаемый порядок слов в предложении), а также данный подход учитывает особенности заданных слов — регистр букв и их последовательность [10].

Работа второго подхода к извлечению информации может быть основана на методах обучения с учителем, методе обучения без учителя, методе частичного обучения с учителем (бутстрап). Чаще всего проводят обучение с учителем, для этого строят модель, которая сможет отличить искомые данные, а затем обучают ее на выборке, у которой уже имеются вручную расставленные метки для проверки работы модели по критериям точности и полноты [11, 12]. При этом применяют лингвистические ресурсы для классификации сущностей и атрибутов, например, Википедию — проверяют, является ли данное слово определением в данной энциклопедии [13].

В качестве основы модели могут служить деревья принятия решений, логистическая регрессия, скрытые марковские модели, метод опорных векторов, нейронные сети [14, 15]. Подход дает возможность тестировать различные стратегии обучения, учитывается большое число признаков разного вида, а также не требуется привлече-

ние лингвиста для написания шаблонов. Ограничения у данного подхода также имеются: для корректной работы размеченный корпус текстов стандартов (под корпусом понимается множество текстов, соответствующее определенным требованиям) должен иметь достаточно большой объем и высокое качество разметки, при смене метода может потребоваться обучение на новой выборке, необходимо тщательно настраивать выбранную модель для получения более точных результатов, результаты работы обычно плохо объяснимы и исправить возникающие ошибки практически невозможно.

3. Описание шаблонов определений в английском языке

Для написания лексико-синтаксических шаблонов необходимо вначале уточнить понятие "термин", а также сформировать все возможные языковые правила его построения в предложениях.

Термин — слово или словосочетание, обозначающее определенное понятие какой-либо области знаний. Определение — формулировка, разъясняющая содержание и смысл термина. Формальное определение состоит из трех частей: термин — слово или фраза, подлежащая определению; класс объекта или понятия, к которому относится термин; отличительные признаки, отличающие его от всех остальных представителей данного класса.

Самый распространенный способ написать определение — это использовать определительное придаточное предложение, где часть сложного предложения зависит от главного. Ниже представлены примеры конструкций таких предложений, где "(" обозначают необязательную часть, "/" — возможные варианты:

1. (a/an/the) (adjective) noun is/are (a/an/the) (adjective/ adjective and adjective) noun + of + noun/adjective/v-ing.

2. (a/an/the) (adjective) noun is (a/an/the) noun (which/that/where/when/who) + (adverb) past participle + (preposition).

3. (a/an/the) (adjective) noun is/are/ may/can be (adverb) defined/known as / called (a/an/the) noun of noun (which/that where/when/who).

4. (a/an/the) noun + is/are + (adverb) past participle + preposition.

5. (a/an/the) noun/pronoun + can be (adverb) + past participle + preposition.

Согласно Манчестерскому банку академических фраз определения могут быть описаны и с помощью конструкций, представленных на рис. 2 [16].

1. The term 'X' was first used by ...
2. The term 'X' can be traced back to ...
3. Previous studies mostly defined 'X' as ...
4. The term 'X' was introduced by Smith in her ...
5. Historically, the term 'X' has been used to describe ...
6. The term 'X' refers to ...
7. 'X' can broadly be defined as ...
8. 'X' can be loosely described as ...
9. The term 'X' encompasses A), B), and C).
10. 'X' can be defined as ... It encompasses ...

Рис. 2. Примеры конструкций для описания термина

Для проверки работоспособности системы были найдены предложения, построенные по описанным выше конструкциям, например, "Comic books are sequential and narrative publications consisting of illustrations, captions, dialogue balloons, and often focus on super-powered heroes. Astronomy is a branch of scientific study primarily concerned with celestial objects inside and outside of the earth's atmosphere. Most metals are malleable; they can be hammered into flat sheets; non-metals lack this quality" и другие, некоторые из них не содержат терминов, а только соответствуют представленным конструкциям.

4. Инструменты построения систем извлечения информации

Для определения наиболее подходящего инструмента определены следующие критерии и их веса: наличие визуализации (возможность визуального представления присутствует — 10, возможность визуального представления отсутствует — 0), формы визуализации (чем больше, тем лучше), точность извлечения информации (чем больше подготовленных определений было найдено, тем лучше), сложность составления шаблонов (чем меньше среднее время на подготовку каждого шаблона, тем лучше).

Для сравнения предлагается использовать метод вариантных секторов, поэтому для каждого свойства были назначены следующие коэффициенты: наличие визуализации — 6, формы визуализации — 5, точность извлечения информации — 10, сложность составления шаблонов — 8.

Одним из популярных инструментов анализа текстов является программное средство GATE (General Architecture for Text Engineering), которое представляет собой систему обработки естественного языка, включающую в себя различные методы для обработки текстов, например, анализаторы морфологии и синтаксиса, поисковые инструменты и средства извлечения информации. Данную систему можно использовать в процессе извлечения, анализа

и аннотирования информации, для анализа кореферентности, применения методов машинного обучения и онтологий, также имеет возможность добавлять другие плагины.

В ходе применения GATE были использованы два инструмента: система искусственного интеллекта ANNIE (A Nearly-New Information Extraction System) и основанная на машинном обучении библиотека OpenNLP, но ни один из них не показал точных результатов: не все предложения с терминами были найдены, некоторые из которых были не выделены из-за неправильного определения части речи, а также были выделены конструкции, не являющиеся определениями. Еще одним ограничением системы при нахождении определений является то, что определение не найдется, если между хотя бы какими-нибудь частями правила стоит не пробел, а, например, перенос строки [17].

Следующее программное средство, предназначенное для обработки и анализа текстов, — Stanford CoreNLP. Данное программное обеспечение написано на Java, но в результате работы сторонних разработчиков над продуктом появилась возможность работать с Python, Ruby, Perl, Javascript, F# и другими .NET языками. В ядро проекта входит также Stanford Parser (вывод зависимостей и построения дерева), Stanford POS Tagger (определение части речи), Stanford RegexNER (для извлечения именованных сущностей), Stanford EnglishTokenizer (разделение на токены), Stanford TokensRegex (для определения лексико-синтаксических шаблонов) и другие. Имеется также визуальный интерфейс, в котором можно посмотреть работу модулей по определению частей речи (рис. 3), распознаванию именованных сущностей, кореферентности, основные зависимости и другое [18].

Для извлечения информации можно использовать как регулярные выражения, так и именованные переменные. Выводить результаты помимо визуального представления можно

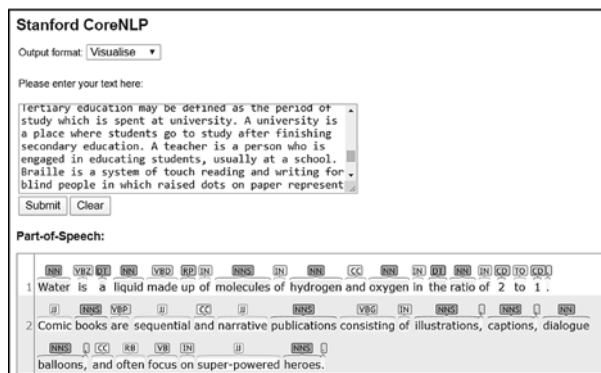


Рис. 3. Результат работы POS Tagging в Stanford CoreNLP

в форматы строки, XML, JSON, CoNLL. К ограничениям данного инструмента можно отнести некоторые неполадки при работе с основным API, однако это можно исправить, используя другую библиотеку на основе Stanford CoreNLP.

NLTK (Natural Language Toolkit) — открытая платформа для создания программ по обработке естественного языка, написанная на языке Python, имеющая легкие в использовании интерфейсы для многих языковых корпусов, а также библиотеки для обработки текстов для классификации, токенизации, стемминга, разметки, фильтрации и семантических рассуждений. Для поиска информации составляется шаблон за счет последовательности символов (регулярное выражение). К преимуществам данного решения можно отнести подробную документацию, простую реализацию, не требуется подключение к серверу, как в предыдущем решении. К ограничениям — иногда возникают ошибки при разделении текста на предложения и определении части речи [19].

Также было проведено сравнение данных инструментов по определенным ранее критериям с помощью метода вариантных секторов, результаты которого представлены в таблице.

5. Требования к функциональным характеристикам

Система должна обеспечивать графический интерфейс в виде веб-страницы, на которой имеется возможность ввода поискового запроса, отображения документов и связей между ними. Также должны иметься возможности построения запросов по содержанию загруженных в систему стандартов, а также визуализации сети связанных стандартов в виде графа. Результаты работы система предоставляет в виде таблицы с информацией о данном термине или стандарте, графа связей стандартов, отображает документ стандарта. Система должна иметь возможность экспорта результатов поиска в формат "PDF".

Функциональным назначением разрабатываемой системы является автоматизация процесса извлечения знаний из документов промышленных стандартов, в том числе повышение соответствия полученной информации информационной потребности пользователя. Для описания функционального назначения системы применяется диаграмма вариантов использования (рис. 4).

Для отображения архитектуры системы используется диаграмма компонентов, представленная на рис. 5.

Разрабатываемая система состоит из трех компонентов: сервера баз данных, сервиса и web-приложения. В качестве сервера баз данных используется SQLite — кроссплатформенная база данных, которую можно создавать в Python без установки дополнительных инструментов. Отличительные особенности данного решения: быстрота, автономность, надежность, обеспечение полной функциональности, отсутствие ограничений на использование.

Сервис реализован с помощью фреймворка Flask (Python), который в отличие от других решений, например, Django, является простым в использовании, позволяет выбирать, с какой именно базой данных работать, и как именно будут взаимодействовать компоненты.



Рис. 4. Диаграмма прецедентов системы интеллектуального поиска

Результаты сравнения инструментальных средств

Инструменты	Свойство (вес)				
	Наличие визуализации (6)	Формы визуализации (5)	Точность извлечения (10)	Сложность составления шаблонов (8)	Общий вес инструмента
GATE (ANNIE)	10	5	6	5	185
GATE (OpenNLP)	10	5	7	5	195
Stanford CoreNLP	10	10	8	8	254
NLTK	0	0	7	7	126

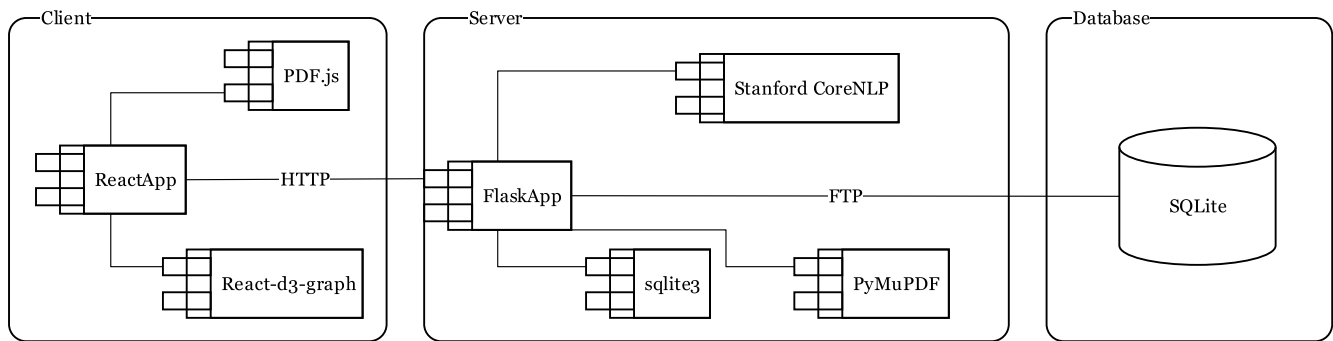


Рис. 5. Диаграмма компонентов разрабатываемой системы

Клиентское web-приложение написано на React JS — библиотека JavaScript для создания пользовательских интерфейсов, отличительными особенностями которой является разработка с помощью небольших и изолированных частей кода — компонентов, упрощенная отладка приложений за счет использования односторонней привязки данных, использование чистых функций, а не сложных и требующих настройки шаблонных классов для формирования интерфейса.

6. Разработка модуля извлечения информации из стандартов

В качестве входных данных на вход модулю поступают стандарты в формате "PDF", поэтому для работы с документами была выбрана библиотека "PyMuPDF", которая является облегченным средством просмотра PDF.

Из каждого стандарта должна быть извлечена следующая информация: название стандарта, номер документа, определения, отсылки к другим стандартам. Информация о документе находится на первой странице и ее можно найти с помощью слов "Document Number" and "Document Name".

Что касается определений, то здесь есть два варианта: это определения, которые вынесены в главу "Definitions", и определения, которые необходимо найти с помощью составленных ранее шаблонов из текста стандарта.

Для работы с уже выделенными в отдельный параграф (пример контента параграфа представлен на рис. 6) терминами была создана функция "extract_definitions_paragraph", которая получает на вход документ стандарта. Для того чтобы извлечь данные определения, необходимо загрузить только интересующие нас страницы, для чего понадобится оглавление ("Contents") — здесь необходимо найти номера страниц третьей главы. Затем данные страницы объединяются, и из них нужно удалить лиш-

нюю информацию, например, знак копирайта, возможные комментарии, номер страницы. Необходимая нам информация представлена в формате "термин: определение". Однако под такой же шаблон попадают некоторые вводные предложения, например, строка "For the purposes of the present document, the terms and definitions given in oneM2M TS-0011 [2] and the following apply: additional authenticated data", поэтому необходимо их удалить, а затем разбить текст по абзацам.

С помощью выбранной библиотеки Stanford CoreNLP и описанных ранее шаблонов начинаем извлекать определения из оставшейся части документа. Для того чтобы найти определения, необходимо разбить весь документ на предложения и определить части речи каждого слова.

Следующий шаг — описание шаблонов с помощью регулярных выражений и поиск определений в тексте с их помощью. В Python для работы с регулярными выражениями есть модуль "re". Метод, который используется для решения данной задачи, называется "re.search(pattern, string)", он осуществляет поиск паттерна по всей строке

3	Definitions, symbols and abbreviations
3.1	Definitions
For the purposes of the present document, the terms and definitions given in oneM2M TS-0011 [2] and the following apply:	
additional authenticated data [14]: data that is authenticated, but not encrypted by an authenticated encryption with associated data algorithm	
AE-ID Certificate : certificate with a certificate chain to a trust anchor certificate and containing an AE-ID in the subjectAltName extension	
NOTE: An AE_ID certificate can be used to verify that an entity has been assigned the AE-ID in the certificate.	
association configuration : phase of a Security Association Establishment Framework in which the entity establishing the Security Association (and the Central Key Distribution Server, in the case of Centralized Security Frameworks), are provided with identities (and any other relevant credentials) to ensure that the security association is established between the intended entities	
association security handshake : phase of a Security Association Establishment Framework in which the security association endpoints perform mutual authentication	
authenticated encryption with associated data [14]: algorithm providing confidentiality for the plaintext and a way to check its integrity and authenticity while providing the ability to check the integrity and authenticity of some additional authenticated data. In this context plaintext refers to data that is authenticated and encrypted	
bootstrap credential : pre-provisioned credential enabling mutual authentication of the Enrollee and the M2M Enrolment function	
bootstrap credential configuration : phase of a Remote Security Provisioning Framework in which the Bootstrap Credentials are pre-provisioned to the Enrollee and the M2M Enrolment function	
bootstrap enrolment handshake : phase of a Remote Security Provisioning Framework in which the Enrollee and M2M Enrolment Function perform mutual authentication	
© oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TTA, TSDSI, TTA, TTC) Page 16 of 268	
This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1.	

Рис. 6. Глава стандарта с определениями

(не только в начале). В качестве паттерна будут задаваться закодированные с помощью специальных символов определенные ранее шаблоны, пару из которых рассмотрим для примера.

Рассмотрим шаблон "(a/an/the) (adjective) noun is/are (a/an/the) (adjective/ adjective and adjective) noun + of + noun/adjective/v-ing". Здесь в начале может идти артикль, но не обязательно, на языке регулярных выражений это выглядит следующим образом: "(A|An|The|a|an|the)_DT\s)". Далее, также необязательно, может идти прилагательное "(\w + _(JJ|JR|JS)\s)", затем идут существительное и один из глаголов "is/are": "\w + _(NN|NNP|NNS)\s(is|are)_VBZ\s". Потом снова добавляем вариативную часть с артиклем: "((a|an|the)_DT\s)". Далее идет прилагательное, прилагательное и прилагательное, либо ничего, описываем это так: "((\w + _(JJ|JR|JS)\s)|((\w + _(JJ|JR|JS)\s)sand_CC\s(\w + _(JJ|JR|JS)\s)))s)". Далее идет существительное с предлогом "of" и один из вариантов: существительное, прилагательное или герундий, после чего могут быть еще слова, но не обязательно: "\w + _(NN|NNP|NNS)\sof_IN\s(\w + _(NN|NNP|NNS|JJ|JR|JS|VBG)\w*)". Таким образом описываются и другие выделенные шаблоны определений и в результате получаем таблицу, как показано на рис. 7.

Еще одним видом извлекаемой информации являются отсылки к другим стандартам. В данном случае не нужно определять части речи,

doc_num	doc_name	term	definition
0 TS-0001-V3.15.1	Functional Architecture	access control attributes	set of parameters of the Originator, target r...
1 TS-0001-V3.15.1	Functional Architecture	access decision	authorization reached when an entity's Privil...
2 TS-0001-V3.15.1	Functional Architecture	application layer	comprises oneM2M Applications and related bus...
3 TS-0001-V3.15.1	Functional Architecture	attribute	stores information pertaining to the resource...
4 TS-0001-V3.15.1	Functional Architecture	child resource	sub-resource of another resource that is its ...

Рис. 7. Часть таблицы с определениями

doc_num	docs_link
0 TS-0001	[TS-0011, TS-0003, TS-0004, TS-0012, TS-0021, ...]
1 TS-0002	[TS-0011, TR-0008]
2 TS-0003	[TR-0008, TR-0019, TR-0013, TS-0001, TS-0011, ...]
3 TS-0004	[TS-0001, TS-0003, TS-0008, TS-0009, TS-0010, ...]

Рис. 8. Часть таблицы с отсылками на другие стандарты

5.1.1 Identification and Authentication

The Identification and Authentication function is in charge of identification and mutual authentication of CSEs and AEs.

Identification is the process of checking if the identity provided for authentication is valid. How to perform an identification process will depend on the purpose of authentication. For example, in the case of resource access, the authentication function can require the identification to check if the AE or CSE has registered with the local CSE; in the case of AE or CSE registration, the authentication function can require the identification to check if the identity provided by an AE or CSE fits a certificate. Once passing this checking process, the AE or CSE is identified, and the identified identity will be supplied to authentication process.

Authentication is the process of validating if the identity supplied in the identification step is associated with a trustworthy credential. How to perform an authentication process will depend on using which mutual authentication mechanism. For example, in the case of using certificate based authentication mechanism, the authentication function can require the authentication to verify a digital signature; in the case of using symmetric key based authentication mechanism, the authentication function can require the authentication to verify a Message Integrity Code (MIC). When this validating process has been completed, the AE or CSE is authenticated.

Рис. 9. Маркировка определений

достаточно задать шаблон для определения номера стандарта, который в общем виде выглядит следующим образом: "\bT(S|R)-\d\d\d\d\b", где "\b" — граница слова, "T" — сокращение от слова "Technical", далее может идти либо "S", что обозначает "Specifications", либо "R" — "Reports", далее обязательно пишется дефис("-") и 4 цифры ("\d"). Используя модуль "re", находим все упоминания других стандартов oneM2M и получаем таблицу, показанную на рис. 8.

Последним шагом в модуле извлечения информации стало выделение найденных определений в документах стандартов с помощью параметра "page.addHighlightAnnot()" (рис. 9).

Данный модуль возвращает две таблицы — с терминами и определениями и с отсылками к другим стандартам, которые в дальнейшем будут записаны в базу данных, а также сохраняет документы с выделенными в них терминами и определениями, которые впоследствии выводятся на экран.

7. Демонстрация результатов работы системы

Для отображения файла в формате PDF используется библиотека "PDF.js", которая преобразовывает файлы из формата PDF в код HTML5 используя метод рендеринга на клиенте, его преимуществами являются малая нагрузка на сервер и рендер PDF в разных расширениях без потери качества

Также необходимо добавить отображение связей между стандартами. Для создания графа применяется библиотека "react-d3-graph", которая позволяет управлять элементами и устанавливать настройки отображения [20]. На рис. 10 представлен результат поиска стандарта "TS-0003".

Search in Standards

Search in Standards

Update Standards: TS-0003

Search Term: Search Standard: Export Report

Standard

Standard

Number

Name

Term

Definition

1 TS-0003- Security additional data that is authenticated, but not encrypted by an authenticated encryption with associated data algorithm

2 TS-0003- Security AE-ID certificate with a certificate chain to a trust anchor certificate and containing an AE-ID in the subjectAltName extension NOTE: An AE-ID certificate can be used to verify that an entity has been assigned the AE-ID in the certificate.

3 TS-0003- Security association phase of a Security Association

Full Screen

Unstick nodes

Nodes: 10 Links: 10

Configurations

automaticRearrangeAfterOnClipboard

documentation

collapse

documentation

directed

documentation

Рис. 10. Результаты поиска стандарта "TS-0003"

Заключение

В рамках данной работы была рассмотрена предметная область, связанная с извлечением информации, что является ключевым этапом в построении сложных систем информационного поиска. Как показал анализ предметной области, это направление довольно неплохо развито и предлагает большой выбор различных методов и инструментальных средств для построения систем извлечения информации. Одним из таких инструментов является Stanford CoreNLP, позволяющий эффективно решать базовые задачи обработки естественного языка, на основе которого и был разработан модуль извлечения информации.

Результатом работы стала реализация системы извлечения информации из промышленных стандартов с помощью языков Python (веб-сервис на фреймворке Flask) и React JS (клиентское веб-приложение на основе React), которая отображает контент стандартов в виде таблицы с терминами и определениями, графа со связями между стандартами и сам стандарт в виде файла формата PDF. Применение данной системы позволит ускорить обработку извлеченной информации, что является важной частью в процессе накопления и применения новых знаний. Данное решение является универсальным и может быть использовано в различных областях.

В рамках дальнейшей работы планируется построение сложных поисковых запросов, взаимодействие с другими форматами документов, выгрузка графа связей в документ, а также его доработка: добавление связей терминов и обеспечение drill-down свойств при выборе конкретного термина.

Список литературы

1. Mannai M., Abdessalem W., Chezala H. Information Extraction Approaches: A Survey // Information and Communication

Technology. Advances in Intelligent Systems and Computing. 2018. Vol. 625. P. 289–297.

2. Muawia E., Ali A., Mubarak H. Information Extraction Methods and Extraction Techniques in the Chemical Document's Contents: Survey // ARPN Journal of Engineering and Applied Sciences. 2015. Vol. 10. P. 1068–1073.

3. Milosevic N., Gregson C., Hernandez R., Nenadic G. A. Framework for Information Extraction from Tables in Biomedical Literature // International Journal on Document Analysis and Recognition. 2019. Vol. 22. P. 55–78.

4. Issertial L., Tsuji H. Information Extraction for Call for Paper // Natural Language Processing: Concepts, Methodologies, Tools, and Applications. 2020. P. 394–409.

5. Suominen H., Zhou L. Information Extraction to Improve Standard Compliance // AI 2015: Advances in Artificial Intelligence. 2015. Vol. 9457. P. 644–649.

6. Lee J. The Method for Calculating Lost Profit Damages Caused by Patent Infringement in the U. S. Patent Law — Focused on Lost Profit Damages of Lost Sales and Price Erosion // Yonsei Law Rev. 2018. Vol. 28. P. 223–270.

7. Hannah M., Leiva C., Noller D. The Importance of Standards in Smart Manufacturing // MESA International White Paper. 2018. Vol. 58.

8. Gezer C., Taskin E. An Overview of oneM2M Standard // 2016 24th Signal Processing and Communication Application Conference (SIU). 2016. P. 1705–1708.

9. Standards for M2M and the Internet of Things. URL: <http://www.onem2m.org/> (дата обращения: 13.02.2020).

10. Большакова Е. И., Воронцов К. В., Ефремова Н. Э., Клышинский Э. С., Лукашевич Н. В., Сапин А. С. Автоматическая обработка текстов на естественном языке и анализ данных: Учеб. пособ. М.: Изд-во НИУ ВШЭ, 2017. С. 83–122.

11. Bovi C. D., Telesca L., Navigli R. Large-scale Information Extraction from Textual Definitions through Deep Syntactic and Semantic Analysis // Transactions of the Association for Computational Linguistics. 2015. Vol. 3. P. 529–543.

12. Espinosa-Anke L., Saggion H., Ronzano F. Hypernym Extraction: Combining Machine Learning and Dependency Grammar // Lecture Notes in Computer Science. 2015. Vol. 1. P. 372–383.

13. Астраханцев Н. А. Автоматическое извлечение терминов из коллекции текстов предметной области с помощью Википедии // Труды ИСП РАН. 2014. № 26 (4). С. 7–20.

14. Bird S., Klein E., Loper E. Natural Language Processing with Python. California: O'Reilly Media. 2016. P. 221–257.

15. Espinosa-Anke L., Carlini R., Saggion H., Ronzano F. DefExt: A Semi Supervised Definition Extraction Tool // GLOBALEX 2016: Lexicographic Resources for Human language Technology Workshop. 2016. P. 24–28.

16. Defining Terms — Academic Phrasebank. URL: <http://www.phrasebank.manchester.ac.uk/writing-definitions/> (дата обращения: 18.02.2020).

17. General Architecture for Text Engineering (GATE). URL: <https://gate.ac.uk/> (дата обращения: 18.02.2020).

18. Manning C. D., Surdeanu M., Bauer J. The Stanford CoreNLP Natural Language Processing Toolkit // 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2014. P. 55–60.

19. Natural Language Toolkit. URL: <https://www.nltk.org/> (дата обращения: 20.02.2020).

20. React-d3-graph. URL: <https://www.npmjs.com/package/react-d3-graph> (дата обращения: 21.04.2020).

E. A. Gosteva, e-mail: kategosteva@yandex.ru, V. V. Lanin, Senior Lecture, e-mail: vlanin@live.com, National Research University Higher School of Economics, Perm, Russian Federation

System Development for Intelligent Search in Industrial Standards

The article is devoted to the description of intelligent information retrieval system development according to industry standards based on the Stanford CoreNLP tool. The description of the subject area, design, and main stages of system development are presented: development of extracting information module from industrial standards, implementation of a web service using the Flask framework, and a client web application on React JS. The use of the developed system by engineers and software developers will make it possible to effectively manage the definition base of industrial standards, understand them correctly and observe them in accordance with the chosen field of knowledge.

Keywords: industrial standards, smart manufacturing, intelligent search

DOI: 10.17587/it.27.322-330

References

1. Mannai M., Abdesslem W., Chezala H. Information Extraction Approaches: A Survey, *Information and Communication Technology. Advances in Intelligent Systems and Computing*, 2018, vol. 625, pp. 289–297.
2. Muawia E., Ali A., Mubarak H. Information Extraction Methods and Extraction Techniques in the Chemical Document's Contents: Survey, *ARPJ Journal of Engineering and Applied Sciences*, 2015, vol. 10, pp. 1068–1073.
3. Milosevic N., Gregson C., Hernandez R., Nenadic G. A. Framework for Information Extraction from Tables in Biomedical Literature, *International Journal on Document Analysis and Recognition*, 2019, vol. 22, pp. 55–78.
4. Issertial L., Tsuji H. Information Extraction for Call for Paper, *Natural Language Processing: Concepts, Methodologies, Tools, and Applications*, 2020, pp. 394–409.
5. Suominen H., Zhou L. Information Extraction to Improve Standard Compliance, *AI 2015: Advances in Artificial Intelligence*, 2015, vol. 9457, pp. 644–649.
6. Lee J. The Method for Calculating Lost Profit Damages Caused by Patent Infringement in the U. S. Patent Law — Focused on Lost Profit Damages of Lost Sales and Price Erosion, *Yonsei Law Reviv.*, 2018, vol. 28, pp. 223–270.
7. Hannah M., Leiva C., Noller D. The Importance of Standards in Smart Manufacturing, *MESA International White Paper*, 2018, vol. 58.
8. Gezer C., Taskin E. An Overview of oneM2M Standard, *2016 24th Signal Processing and Communication Application Conference (SIU)*, 2016, pp. 1705–1708.
9. Standards for M2M and the Internet of Things, available at: <http://www.onem2m.org/> (date of access: 13.02.2020).
10. Bolshakova E. I., Vorontsov K. V., Efremova N. E., Klyshinsky E. S., Lukashovich N. V., Sapin A. S. Automatic processing of text in natural language and data analysis, Moscow, The HSE Publishing House., 2017, pp. 83–122 (in Russian).
11. Bovi C. D., Telesca L., Navigli R. Large-scale Information Extraction from Textual Definitions through Deep Syntactic and Semantic Analysis, *Transactions of the Association for Computational Linguistics*, 2015, vol. 3, pp. 529–543.
12. Espinosa-Anke L., Saggion H., Ronzano F. Hypernym Extraction: Combining Machine Learning and Dependency Grammar, *Lecture Notes in Computer Science*, 2015, vol. 1, pp. 372–383.
13. Astrachantsev N. A. Automatic term acquisition from domain-specific text collection by using Wikipedia, *Proceedings of the Institute for System Programming of RAS*, 2014, no. 26 (4), pp. 7–20 (in Russian).
14. Bird S., Klein E., Loper E. Natural Language Processing with Python, California, O'Reilly Media, 2016, pp. 221–257.
15. Espinosa-Anke L., Carlini R., Saggion H., Ronzano F. DefExt: A Semi Supervised Definition Extraction Tool, *GLOBAL-EX 2016: Lexicographic Resources for Human language Technology Workshop*, 2016, pp. 24–28.
16. Defining Terms — Academic Phrasebank, available at: <http://www.phrasebank.manchester.ac.uk/writing-definitions/> (date of access: 18.02.2020).
17. General Architecture for Text Engineering (GATE), available at: <https://gate.ac.uk/> (date of access: 18.02.2020).
18. Manning C. D., Surdeanu M., Bauer J. The Stanford CoreNLP Natural Language Processing Toolkit, *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55–60.
19. Natural Language Toolkit, available at: <https://www.nltk.org/> (date of access: 20.02.2020).
20. React-d3-graph, available at: <https://www.npmjs.com/package/react-d3-graph> (date of access: 21.04.2020).

УДК 004.5

DOI: 10.17587/it.27.330-336

К. Е. Израйлов, канд. техн. наук, e-mail: konstantin.izrailov@mail.ru,
Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича,
Санкт-Петербургский Федеральный исследовательский центр Российской академии наук

Обобщенная классификация уязвимостей интерфейсов транспортной инфраструктуры Умного города*

Рассматривается задача классификации уязвимостей, присутствующих в устройствах интерфейсов, которые применяются при взаимодействии людей и составляющих транспортной инфраструктуры Умного города. Вводится концептуальная модель терминологической базы предметной области, на основе которой синтезируется единая классификация уязвимостей. Дается формальная запись классов уязвимостей, а также их примеры.

Ключевые слова: умный город, транспортная инфраструктура, классификация, уязвимости, информационная безопасность

Введение

Одной из тенденций развития современного мира является улучшение жизнедеятельности во всех сферах бизнеса, общества и государства. Этой же цели придерживается и концепция

"Умного города", стремящаяся связать информационными потоками физические системы с искусственным интеллектом [1, 2]. К достаточно значимой для данной концепции области можно отнести транспортную инфраструктуру, которая обеспечивает комфортное перемещение людей на транспортных средствах — именно она должна обеспечивать высокий уровень интеллектуализации [3].

*Работа выполнена при финансовой поддержке РФФИ (проект № 19-29-06099).

Как и практически любое внедрение новых информационных технологий, помимо очевидных преимуществ оно несет и определенные недостатки, как правило выражающиеся в возникновении угроз информационной безопасности [4]. И если безопасности отдельных подсистем уделено достаточное число исследований [5–8], то вопросы реализации угроз через интерфейсы оставлены практически без внимания. Под последними понимаются вполне определенный набор устройств, обеспечивающий обмен информацией как между частями подсистемы, так и с их окружением (например, с человеком). Следуя классической точке зрения, реализация угроз возникает вследствие наличия уязвимостей в интерфейсах. Следовательно, их изучение с позиции информационной безопасности является актуальной задачей.

Обзор существующих классификаций

Существует достаточно обширное число публикаций, посвященных общим уязвимостям веб-интерфейсов: обобщенное деление на виды [9], выделение классов для веб-сервисов в автоматизированных системах управления [10], изучение проблем обнаружения [11].

В ряде случаев проводится выявление целых уязвимых критических зон инфраструктуры в Умных городах [12]. В работе [13] описываются основы безопасности для Умных городов в целях недопущения неправомерного использования их уязвимостей.

Отдельные работы посвящены именно уязвимостям различных элементов Умного города. Так, в работе [14] выделяются три зоны уязвимостей беспилотного транспортного средства: система управления движением, техническая инфраструктура и информационное взаимодействие субъектов Умного города. Аналогичному вопросу обнаружения уязвимостей беспилотных транспортных средств посвящена монография [15].

Однако научные работы, непосредственно описывающие классификацию уязвимостей в интерфейсах транспортной инфраструктуры Умного города, отсутствуют, хотя отдельные аспекты таких уязвимостей и были изучены. И, следовательно, классификация, сочетающая как общие информационные свойства уязвимостей, так и специфику конкретной инфраструктуры, будет обладать безусловной актуальностью и новизной.

Новый подход к классификации

Для всестороннего исследования безопасности транспортной инфраструктуры Умного города в качестве первого шага может быть востребована задача однозначной классификации (обладающая необходимостью и достаточностью) уязвимостей ее интерфейсов, т. е. тех уязвимостей, которые расположены в средствах обеспечения обмена информацией между подсистемами инфраструктуры, а также ее внешним окружением.

Для классификации уязвимостей необходимо обратиться к современному пониманию данного термина, которое наиболее близко к определению в виде "недостатка в системе, способного привести к реализации угроз". Однако данная трактовка уязвимостей достаточно сложна для создания полноценной их классификации по причине классического научного противоречия: потребности и возможности. Так, с одной стороны, следуя практической точке зрения, уязвимость должна отражать некоторую негативную особенность конкретного компонента системы (например, уязвимость в том, что конкретный модуль не обеспечивает защиты шифрованием) — *потребность*. С другой стороны, понятие *недостатка* на уровне целой системы является крайне абстрактным, а точнее усредненным по всем компонентам системы, которые при этом могут быть различным и зависеть от реализации (например, уязвимость в том, что все компоненты не реализуют механизмов шифрования) — *возможность*. Для частичного сглаживания данного противоречия и в интересах классификации уязвимостей, несущей практическую пользу и применимой для любых реализаций транспортных инфраструктур Умного города (и, соответственно, их интерфейсов, искусственных интеллектов, способов взаимодействия с ними человека), воспользуемся подходом, строящимся на следующих предпосылках.

Во-первых, для обозначения процесса взаимодействия в рамках Умного города целесообразно ввести качественно отличные и разноцелевые сущности, обменивающиеся информацией посредством интерфейсов. В случае транспортной инфраструктуры Умного города удачным делением на такие сущности (далее — Сущность) является следующее: 1) инфраструктура — сама реализация концепции (например, умные информационные знаки, пункты оплаты, система мониторинга и парковок и др.); 2) беспилотное транспортное средство (далее — БТС) — средство для перевозки пас-

сажиров и товаров (например, автомобили, автобусы и троллейбусы); 3) окружение — внешние элементы инфраструктуры, не являющиеся частью сети Умного города, но необходимые для выполнения функций последнего (погода, пассажиры, здания, дорожное полотно и др.). В ряде случаев, удобно делить окружение на два элемента: Человек (пассажир, пешеход, водитель) и Пассивный объект (погода, знаки, не беспилотный автомобиль), поскольку первые являются "заказчиками" услуг концепции.

Также будем считать, что каждый из интерфейсов между Сущностями может обеспечиваться целым набором средств (далее — Средство), такими как лидары, радары, камеры, пункты оплаты, микрофоны, светофоры, метеодатчики и т. п.

Во-вторых, деление на Сущности позволяет выделить множество классов интерфейсов, каждый из которых ответственен за обмен информацией между парой Сущностей (в том числе между одной и той же). Это приводит к наличию девяти классов следующих интерфейсов (с учетом направления информационного обмена): 1) Инфраструктура ↔ Инфраструктура; 2) Инфраструктура → БТС; 3) Инфраструктура → Окружение; 4) БТС → Инфраструктура; 5) БТС → БТС; 6) БТС → Окружение; 7) Окружение → Инфраструктура; 8) Окружение → БТС; 9) Окружение → Окружение.

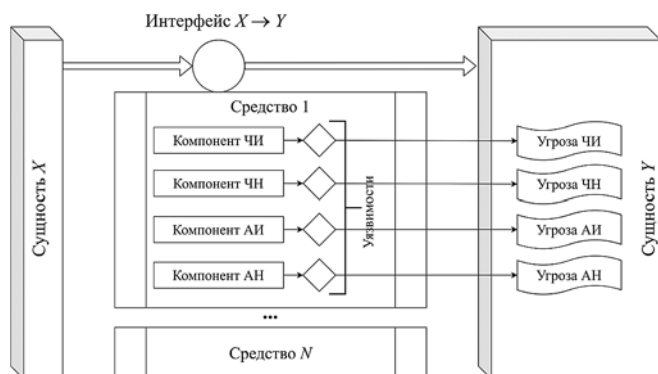
В-третьих, каждое из Средств за счет наличия в нем уязвимостей как раз и выводит транспортную инфраструктуру из такого "идеального" состояния. Это можно обосновать тем, что Средство состоит из компонентов, имеющих некоторый функционал по обслуживанию информационных потоков или их логических групп различных типов; нарушение же работы компонента соответственно приведет и к нарушению числа информационных потоков (увеличению или уменьшению).

В-четвертых, поскольку наличие уязвимости так или иначе приводит к некоторой угрозе, то характеристики уязвимостей и угроз должны некоторым образом коррелировать. Для деления последних на классы можно воспользоваться подходом категориального деления, когда сложный объект разбивается на так называемые категориальные пары, элементы которых представляют собой противоположности с некоторой точки зрения [16]. Так, в качестве одной пары удобно выбрать пару *Человек vs Автомат*: первый элемент обладает волей, для него предназначается концепция Умного города, второй же — субъект без воли, реализующий необходимый функционал концепции в интересах первого. В качестве

другой пары можно выбрать отражающую некоторый эффект нарушения информационной безопасности с позиции наличия информационных потоков, а именно пару *Избыточность vs Недостаточность*: первый элемент соответствует появлению нового потока (аналог нарушения конфиденциальности, когда информация "утекает" третьим лицам), а второй — исчезновению потока (нарушение доступности, когда информация не достигает законного получателя). Комбинация же элементов пар дает четыре класса, ответственных за нарушения информационных потоков, т. е. классификацию угроз: 1) избыточность информационных потоков, ориентированных на Человека (угроза ЧИ); 2) избыточность информационных потоков, ориентированных на Автомат (угроза АИ); 3) недостаточность информационных потоков, ориентированных на Человека (угроза ЧН); 4) недостаточность информационных потоков, ориентированных на Автомат (угроза АН).

В-пятых, существует состояние транспортной инфраструктуры, условно называемое "идеальным" (как правило, достижимое лишь теоретически), в котором Средства не имеют уязвимостей (и, следовательно, отсутствует возможность реализации угроз), а все информационные потоки считаются корректными. Иными словами, такая система находится в некотором локальном максимуме (с позиции безопасности), а появление любой уязвимости приводит к отклонению от этого экстремума вследствие появления новых или исчезновения существующих информационных потоков (что соответствует элементам категориальной пары).

В-шестых, исходя из связи функционала уязвимостей с угрозами, поделенными ранее на четыре подкласса, можно предположить и наличие в Средстве такого же числа групп компонентов (или меньшего для Средств без соответствующего функционала), каждый из которых несет соответствующий информационно-деструктивный эффект. Для простоты и без потери корректности будущей классификации будем считать, что в каждой группе находится лишь один компонент, т. е. Средство состоит из четырех компонентов, наличие уязвимости в каждом из которых приводит к одному из нарушений информации в Сущности (выбираемой согласно направлению информационного обмена через интерфейс, реализуемый Средством). Можно дать следующую, более встречаемую в практике, интерпретацию эффекта от наличия уязвимостей в каждом из таких компонент Средства: в первом произойдет утечка пользовательских



Концептуальная модель терминологической базы предметной области

данных или несанкционированный доступ (появится новый поток Человека); во втором пользователь не получит запрашиваемую информацию или не сможет ее отправить получающей стороне (исчезнет старый поток Человека); в третьем будут передаваться излишние или некорректные служебные данные (появится новый поток Автомата); в четвертом система не сможет корректно функционировать из-за отсутствия или нарушения обмена служебной информацией (исчезнет старый поток Автомата).

Концептуальная модель описанной выше терминологической базы предметной области — Сущностей, интерфейсов, Средств и их компонентов, уязвимостей, угроз — представлена на рисунке.

Так, на рисунке показано взаимодействие Сущности X с Сущностью Y через Интерфейс $X \rightarrow Y$, который может реализовываться множеством Средств (Средство 1, ..., Средство N). При этом в Средстве 1 присутствуют четыре Компонента с функционалом, нарушения в работе которого (при наличии уязвимости) приводит к нарушению информационных потоков, ведущих к соответствующим Угрозам в Сущности Y .

Таким образом, с одной стороны, уязвимости расположены в достаточно конкретных компонентах средств, реализующих интерфейсы между Сущностями, а с другой стороны, уязвимости обобщены негативным эффектом влияния на компоненты, приводящим к угрозам определенного класса. Это позволяет перейти к классификации уязвимостей интерфейсов, которая сглаживает выявленное ранее противоречие.

Формализация классов уязвимостей

Исходя из предложенной концептуальной модели терминологической базы предметной

области (см. рисунок) синтезируем единую классификацию уязвимостей, используя выделенные ранее Сущности, а также классификации интерфейсов и угроз.

Каждому из классов уязвимостей может быть дано следующее достаточно формальное, хотя и хорошо понятное человеку, определение:

«Уязвимость класса расположена в Компоненте ZW Средства M , которое реализует Интерфейс $X \rightarrow Y$ (т. е. для передачи информации от Сущности X к Сущности Y)», при этом

$$\begin{cases} Z \in \{\text{Человек, Автомат}\}; \\ W \in \{\text{Избыточность, Недостаточность}\}; \\ M \in \text{Implement Interface } (X \rightarrow Y); \\ X, Y \in \{\text{Инфраструктура, БТС, Окружение}\}; \\ \text{Компонент} \in \text{Composition}(M), \end{cases}$$

где Z , W — элементы категориальных пар, определяющие также функционал компонентов Средства; M — одно из средств, реализующих интерфейс (ImplementInterface) $X \rightarrow Y$; X и Y — ранее введенные Сущности транспортной инфраструктуры Умного города; Компонент — компонент из состава (Composition) реализующего средства M .

Важной особенностью такой классификации является ее гибкость: она не является однозначной для любой модели транспортной инфраструктуры и учитывает ее конкретные особенности, поскольку зависит от состава средств, реализующих интерфейс, которые могут отличаться как для разных интерфейсов в рамках одной модели инфраструктуры, так и для разных моделей.

Гипотетический пример классификации

Приведем простейший пример предложенной классификации для модели следующей транспортной инфраструктуры. Предположим, что Окружение состоит только из пассажиров, пользующихся услугами такси в виде БТС. Инфраструктура же предоставляет возможность пассажирам заказывать такси непосредственно на остановках путем голосового задания конечного пункта маршрута; при этом для идентификации и авторизации пассажира используется сам голос. Также логично ввести в Инфраструктуру оператора, представляющего собой службу поддержки пассажиров БТС, для чего в средства интерфейсов Инфраструктуры внедрен соответ-

ствующий функционал (запрос помощи, сигнализация о чрезвычайных ситуациях с БТС и т. п.). Таким образом, существует интерфейс для связи Окружения (и его элемента — Человека) и Инфраструктуры в виде системы распознавания голоса (аппаратный микрофон + программное обеспечение) и кнопки экстренного вызова оператора. Как следствие, будет существовать интерфейс между Инфраструктурой и БТС, обеспечивающий выдачу задания последнему на доставку пассажира, его контроля и т. п.; однако для простоты классификации уязвимостей ограничимся лишь первым.

После идентификации пассажира и доставки его на указанное место Инфраструктура автоматически проводит расчет путем взимания с пользователя необходимой денежной суммы.

Теперь исходя из предложенной классификации опишем каждую из возможных уязвимостей. Согласно формальному определению классов между Инфраструктурой и БТС возможно по одной уязвимости в каждом из четырех компонентов средства, реализующего интерфейс системы распознавания голоса и экстренного вызова оператора:

1) компонент ЧИ, уязвимость в котором приводит к новому информационному потоку данных человека; например, недостатки в алгоритмах работы системы взаимодействия со службой поддержки БТС может привести к тому, что конфиденциальная информация, передаваемая от пользователя оператору, окажется у третьих лиц (например, пассажир на другой остановке узнает о местоположении первого);

2) компонент ЧН, уязвимость в котором приводит к пропаже существующего информационного потока; например, из-за неэффективной балансировки голосового трафика в час пик пассажир не сможет экстренно сообщить оператору Инфраструктуры, о том, что БТС неисправно или попало в ДТП;

3) компонент АИ, уязвимость в котором приводит к новому информационному потоку данных автомата; например, система распознавания голоса некорректно распознала неизвестного пользователя, приняв его за существующего пассажира, что привело к бесплатному проезду первого за счет второго;

4) компонент АН, уязвимость в котором приводит к пропаже существующего информационного потока данных автомата; например, система избавления от шумов привела к тому, что голос пользователя не был распознан из-за низкого качества, что привело к невозможности пассажира доехать до пункта назначения.

Приведенная классификация позволяет отнести любую уязвимость в интерфейсах транспортной инфраструктуры Умного города, использование которой прямо или косвенно приводит к соответствующим угрозам, к одному из классов, зависящих от конкретной модели инфраструктуры — числа Сущностей и интерфейсов их взаимодействий, средств реализации интерфейсов, функционала их компонентов. Отличительными особенностями по сравнению с множеством других классификаций являются следующие. Во-первых, в основу каждого класса уязвимостей заложена как ее "заточенность" под конкретный интерфейс (за счет его реализации средствами), так и обобщающие факторы (влияние на абстрактные информационные потоки). Во-вторых, любая из уязвимостей будет обязательно относиться к одному из классов, при этом невозможна ситуация отнесения уязвимости сразу к двум классам.

Если же какая-либо уязвимость не поддается корректной классификации, то значит, что она расположена не в рассматриваемом интерфейсе или ведет не к информационным угрозам, и, следовательно, ее выявление рассматривается в соответствующей отдельной предметной области, выходящей за рамки данной области уязвимостей в интерфейсах транспортной инфраструктуры Умного города.

Направлением дальнейшего исследования должна стать разработка методов обнаружения уязвимостей интерфейсов, что частично является продолжением предыдущих авторских работ, посвященных поиску уязвимостей в программном коде телекоммуникационных устройств, в особенности при отсутствии исходного кода [17–25].

Список литературы

1. Глебова И. С., Ясницкая Я. С. Возможности реализации концепции "умного города": практика российских городов // Экономика и предпринимательство. 2014. № 1-3 (42). С. 232–235.
2. Ярош Н. Н. Городское хозяйство: от "города солнца" к умному городу // Экономический журнал. 2013. № 2 (30). С. 72–88.
3. Buinevich M., Izrailov K., Stolyarova E., Vladko A. Combine method of forecasting VANET cybersecurity for application of high priority way // 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon, South Korea. 2018. P. 266–271.
4. Buinevich M., Fabrikantov P., Stolyarova E., Izrailov K., Vladko A. Software defined internet of things: cyber antifragility and vulnerability forecast // Application of information and communication technologies (AICT). 2017. P. 293–297.

5. Буйневич М. В., Щербаков О. В., Владыко А. Г., Израйлов К. Е. Архитектурные уязвимости моделей телекоммуникационных сетей // Научно-аналитический журнал Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России. 2015. № 4. С. 86—93.
6. Вебб С. Безопасность умных городов // Технологии и средства связи. 2015. № 3 (108). С. 30—32.
7. Красильников А. А. Угрозы информационной безопасности в развитии умных городов // Форум молодых ученых. 2019. № 2 (30). С. 839—844.
8. Ефременко И. П., Колодезная Г. В. Безопасность и конфиденциальность "умного города" в сетях 5G // Научно-техническое и экономическое сотрудничество стран АТР в XXI веке. 2020. Т. 1. С. 114—118.
9. Бессольцев В. Е., Марков П. Н., Сазонов К. В. Уязвимости интерфейса пользователя веб-приложения // Труды Военно-космической академии имени А. Ф. Можайского. 2018. № 660. С. 100—110.
10. Бессольцев В. Е., Марков П. Н. Уязвимости веб-сервисов, используемых в автоматизированных системах управления // I-methods. 2018. Т. 10, № 3. С. 23—35.
11. Раздобаров А. В., Петухов А. А., Гамаюнов Д. Ю. Проблемы обнаружения уязвимостей в современных Веб-приложениях // Проблемы информационной безопасности. Компьютерные системы. 2015. № 4. С. 64—69.
12. Alqahtani A., Tipper D., Kelly-Pitou K., Babay A. Identifying Vulnerable Critical Infrastructure Zones in Smart Cities // 16th International Conference on the Design of Reliable Communication Networks DRCN, Milano, Italy. 2020. P. 1—7.
13. Tubaishat A., Jouhi M. A. Building a Security Framework for Smart Cities: A Case Study from UAE // 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China. 2020. P. 477—481.
14. Моисеев Д. В., Брюховецкий А. А., Скاتков А. В. Адаптивное обнаружение уязвимостей интерфейсов БТС // Modern Science. 2020. № 8-1. С. 375—378.
15. Скатков А. В., Брюховецкий А. А., Доронина Ю. В., Моисеев Д. В., Скатков И. А., Ченгарь О. В. Адаптивное обнаружение уязвимостей интерфейсов беспилотных транспортных средств. Симферополь: Издательство Типография "Ариал", 2020. 352 с.
16. Буйневич М. В., Израйлов К. Е. Категориальный синтез и технологический анализ вариантов безопасного импортозамещения программного обеспечения телекомму-

никационных устройств // Информационные технологии и телекоммуникации. 2016. Т. 4, № 3. С. 95—106.

17. Израйлов К. Е., Покусов В. В. Утилита для поиска уязвимостей в программном обеспечении телекоммуникационных устройств методом алгоритмизации машинного кода. Часть 3. Модульно-алгоритмическая архитектура [Электронный ресурс] // Информационные технологии и телекоммуникации. 2016. Т. 4, № 4. С. 104—121.

18. Буйневич М. В., Израйлов К. Е. Утилита для поиска уязвимостей в программном обеспечении телекоммуникационных устройств методом алгоритмизации машинного кода. Часть 1. Функциональная архитектура [Электронный ресурс] // Информационные технологии и телекоммуникации. 2016. Т. 4, № 1. С. 115—130.

19. Израйлов К. Е. Утилита для поиска уязвимостей в программном обеспечении телекоммуникационных устройств методом алгоритмизации машинного кода. Часть 2. Информационная архитектура [Электронный ресурс] // Информационные технологии и телекоммуникации. 2016. Т. 4, № 2. С. 86—104.

20. Израйлов К. Е. Метод алгоритмизации машинного кода для поиска уязвимостей в телекоммуникационных устройствах: дис.... канд. техн. наук: 05.13.19. СПб., 2017. 261 с.

21. Buinevich M., Izrailov K. Method and utility for recovering code algorithms of telecommunication devices for vulnerability search // 16th International Conference on Advanced Communication Technology (ICTACT). 2014. P. 172—176.

22. Buinevich M., Izrailov K., Vladyko A. Method for partial recovering source code of telecommunication devices for vulnerability search // 17th International Conference On Advanced Communications Technology (ICTACT). 2015. P. 76—80.

23. Buinevich M., Izrailov K., Vladyko A. Method and prototype of utility for partial recovering source code for low-level and medium-level vulnerability search // 18th International Conference on Advanced Communication Technology (ICTACT). 2016. P. 700—707.

24. Buinevich M., Izrailov K., Vladyko A. Testing of Utilities for Finding Vulnerabilities in the Machine Code of Telecommunication Devices // 19th International Conference on Advanced Communication Technology (ICTACT). 2017. P. 408—414.

25. Buinevich M., Izrailov K., Vladyko A. Metric of vulnerability at the base of the life cycle of software representations // 20th International Conference on Advanced Communication Technology (ICTACT). 2018. P. 1—8.

K. E. Izrailov, PhD, e-mail: konstantin.izrailov@mail.ru,
The Bonch-Bruевич Saint-Petersburg State University of Telecommunications,
St. Petersburg Federal Research Center of the Russian Academy of Sciences

Generalized Classification of Vulnerabilities of Interfaces to the Transport Infrastructure of the Smart City

We consider the problem of classifying the vulnerabilities present in interface devices that are used in the interaction of people and components of the transport infrastructure of the Smart City. A conceptual model of the terminological base of the subject area is introduced, on the basis of which a unified classification of vulnerabilities is synthesized. A formal record of vulnerability classes is given, as well as their examples.

Keywords: smart city, transport infrastructure, classification, vulnerabilities, information security

Acknowledgements: This work was financially supported by the Russian Foundation for Basic Research (project No. 19-29-06099).

DOI: 10.17587/it.27.330-336

References

1. Glebova I. S., Yasnitskaya Ya. S. Possibilities of implementing the concept of "smart city": the practice of Russian cities, *Ekonomika i predprinimatel'stvo*, 2014, no. 1–3 (42), pp. 232–235 (in Russian).
2. Yarosh N. N. Urban economy: from the "city of the sun" to the smart city, *Ekonomicheskij zhurnal*, 2013, no. 2 (30), pp. 72–88 (in Russian).
3. Buinevich M., Izrailov K., Stolyarova E., Vladko A. Combine method of forecasting VANET cybersecurity for application of high priority way, *20th International Conference on Advanced Communication Technology (ICACT)*, Chuncheon, South Korea, 2018, pp. 266–271.
4. Buinevich M., Izrailov K., Stolyarova E., Vladko A. Software defined internet of things: cyber antifragility and vulnerability forecast, *Application of information and communication technologies (AICT)*, 2017, pp. 293–297.
5. Buinevich M. V., Shcherbakov O. V., Vladko A. G., Izrailov K. Ye. Architectural vulnerabilities of telecommunication network models, *Nauchno-analiticheskij zhurnal Vestnik Sankt-Peterburgskogo universiteta Gosudarstvennoy protivopozharnoy sluzhby MCHS Rossii*, 2015, no. 4, pp. 86–93 (in Russian).
6. Vebb S. Smart city safety, *Tekhnologii i sredstva svyaz*, 2015, no. 3 (108), pp. 30–32 (in Russian).
7. Krasilnikov A. A. Information security threats in the development of smart cities, *Forum molodykh uchenykh*, 2019, no. 2 (30), pp. 839–844 (in Russian).
8. Yefremenko I. P., Kolodeznaya G. V. 5G Smart City Security and Privacy, *Nauchno-tehnicheskoye i ekonomicheskoye sotrudnichestvo stran ATR v XXI veke*, 2020, vol. 1, pp. 114–118 (in Russian).
9. Bessol'tsev V. Ye., Markov P. N., Sazonov K. V. Web Application User Interface Vulnerabilities, *Trudy Voenno-kosmicheskoy akademii imeni A. F. Mozhayskogo*, 2018, no. 660, pp. 100–110 (in Russian).
10. Bessol'tsev V. Ye., Markov P. N. Vulnerabilities of web services used in automated control systems, *I-methods*, 2018, vol. 10, no. 3, pp. 23–35 (in Russian).
11. Razdobarov A. V., Petukhov A. A., Gamayunov D. Yu. Vulnerability detection issues in modern web applications, *Problemy informatsionnoy bezopasnosti. Komp'yuternyye sistemy*, 2015, no. 4, pp. 64–69 (in Russian).
12. Alqahtani A., Tipper D., Kelly-Pitou K., Babay A. Identifying Vulnerable Critical Infrastructure Zones in Smart Cities, *16th International Conference on the Design of Reliable Communication Networks DRCN*, Milano, Italy, 2020, pp. 1–7.
13. Tubaishat A., Jouhi M. A. Building a Security Framework for Smart Cities: A Case Study from UAE, *5th International Conference on Computer and Communication Systems (ICCCS)*, Shanghai, China, 2020, pp. 477–481.
14. Moiseyev D. V., Bryukhovetskiy A. A., Skatkov A. V. Adaptive Vulnerability Detection for BTS Interfaces, *Modern Science*, 2020, no. 8-1, pp. 375–378 (in Russian).
15. Skatkov A. V., Bryukhovetskiy A. A., Doronina Yu. V., Moiseyev D. V., Skatkov I. A., Chengar' O. V. Adaptive Vulnerability Discovery of Unmanned Vehicle Interfaces: A Monograph, Simferopol', Publishing house Tipografiya "Arial", 2020, 352 p. (in Russian).
16. Buinevich M. V., Izrailov K. Ye. Categorical synthesis and technological analysis of options for safe import substitution of software for telecommunication devices, *Informatsionnyye tekhnologii i telekommunikatsii*, 2016, vol. 4, no. 3, pp. 95–106 (in Russian).
17. Izrailov K. Ye., Pokusov V. V. A utility for searching for vulnerabilities in the software of telecommunication devices using machine code algorithms. Part 3. Modular-algorithmic architecture [Electronic resource], *Informatsionnyye tekhnologii i telekommunikatsii*, 2016, vol. 4, no. 4, pp. 104–121 (in Russian).
18. Buinevich M. V., Izrailov K. Ye. A utility for searching for vulnerabilities in the software of telecommunication devices using machine code algorithms. Part 1. Functional architecture [Electronic resource], *Informatsionnyye tekhnologii i telekommunikatsii*, 2016, vol. 4, no. 1, pp. 115–130 (in Russian).
19. Izrailov K. Ye. A utility for searching for vulnerabilities in the software of telecommunication devices using machine code algorithms. Part 2. Information architecture [Electronic resource], *Informatsionnyye tekhnologii i telekommunikatsii*, 2016, vol. 4, no. 2, pp. 86–104 (in Russian).
20. Izrailov K. Ye. Algorithmization method for machine code to find vulnerabilities in telecommunication devices: dis. ... Cand. tech. sciences: 05.13.19, SPb., 2017, 261 p (in Russian).
21. Buinevich M., Izrailov K. Method and utility for recovering code algorithms of telecommunication devices for vulnerability search, *16th International Conference on Advanced Communication Technology (ICACT)*, 2014, pp. 172–176.
22. Buinevich M., Izrailov K., Vladko A. Method for partial recovering source code of telecommunication devices for vulnerability search, *17th International Conference On Advanced Communications Technology (ICACT)*, 2015, pp. 76–80.
23. Buinevich M., Izrailov K., Vladko A. Method and prototype of utility for partial recovering source code for low-level and medium-level vulnerability search, *18th International Conference on Advanced Communication Technology (ICACT)*, 2016, pp. 700–707.
24. Buinevich M., Izrailov K., Vladko A. Testing of Utilities for Finding Vulnerabilities in the Machine Code of Telecommunication Devices, *19th International Conference on Advanced Communication Technology (ICACT)*, 2017, pp. 408–414.
25. Buinevich M., Izrailov K., Vladko A. M. Metric of vulnerability at the base of the life cycle of software representations, *20th International Conference on Advanced Communication Technology (ICACT)*, 2018, pp. 1–8.

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор Е. В. Конова.

Корректор М. Ю. Безменова.

Сдано в набор 10.04.2021. Подписано в печать 25.05.2021. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT621. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати,

телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансд солюшнз". Отпечатано в ООО "Авансд солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru

Рисунки к статье Ш. С. Фахми, А. Г. Давидчука, Е. В. Костиковой
«НОВЫЕ АЛГОРИТМЫ СЖАТИЯ БЕЗ ПОТЕРЬ ТРАНСПОРТНЫХ СЮЖЕТОВ»



Рис. 1. Пример видеопотока ТС с камеры наблюдения на мосту

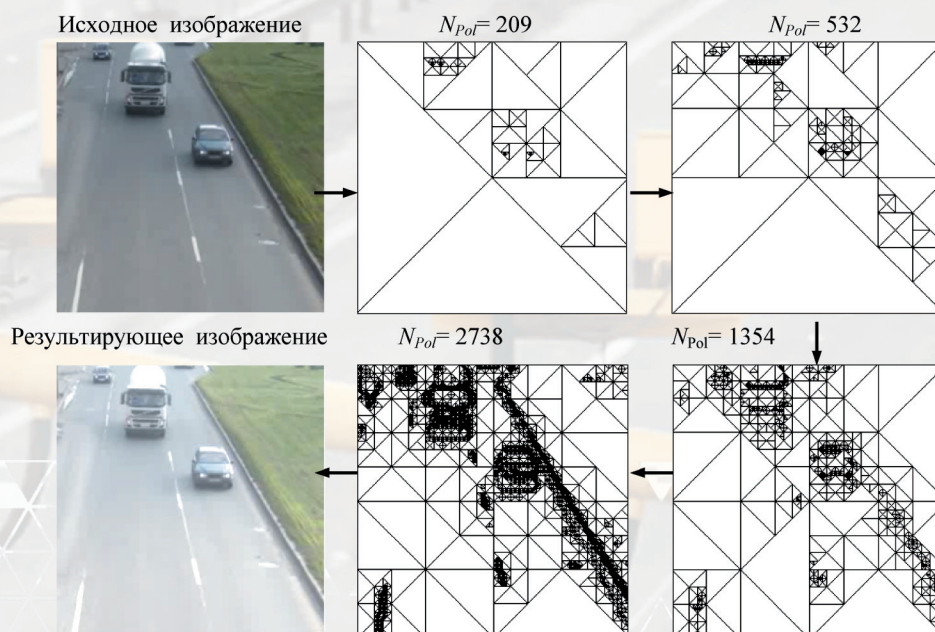


Рис. 2. Процесс сжатия и восстановления изображения алгоритмом деления на треугольники

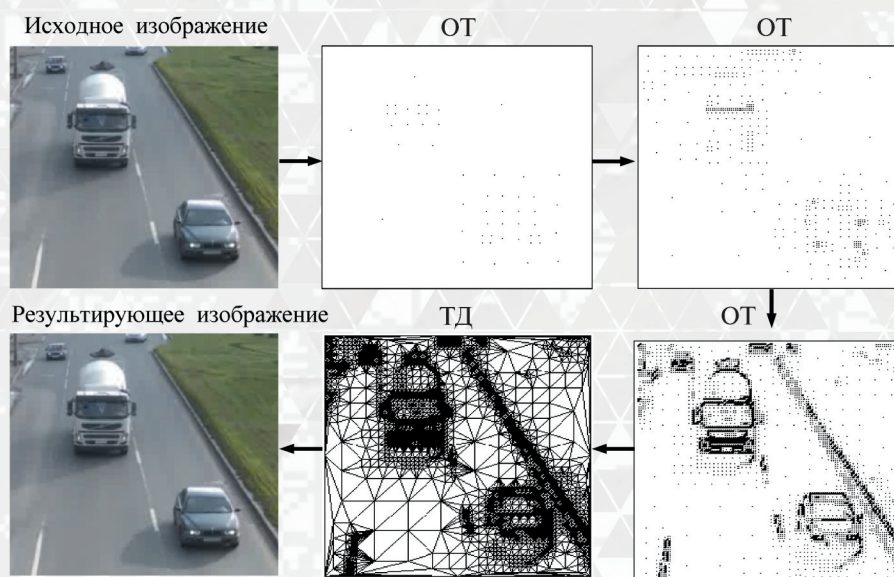


Рис. 3. Итерационный алгоритм ТД восстановления по ОТ



Профессору МГУ им. М. В. Ломоносова, главному
научному сотруднику ИПУ им. В. А. Трапезникова РАН,
заместителю главного редактора журнала
«Информационные технологии»,
доктору технических наук
Николаю Борисовичу ФИЛИМОНОВУ
исполнилось 70 лет!

Н. Б. Филимонов – известный специалист в области автоматизации процессов управления, представитель отечественной научной школы технической кибернетики В. В. Солодовникова, видный ученый и преподаватель. Н. Б. Филимоновым сформировано новое перспективное научное направление – теория полиэдральной оптимизации дискретных динамических процессов управления и наблюдения. Он внес заметный вклад в развитие ряда ключевых направлений современной теории автоматического управления: проблемы динамического качества процессов управления, принципа динамической развязки многоконтурных систем регулирования, концепции многорежимного управления сложными динамическими объектами, методов анализа и синтеза систем терминального управления.

Отметим, что связь Николая Борисовича с издательством «Новые технологии» не ограничивается работой только в одном журнале. Он – ответственный секретарь объединенной редакции Издательства, главный редактор журнала «Мехатроника, автоматизация, управление». Сотрудничество Николая Борисовича с нашим коллективом, сложившимся на базе одной из редакций издательства «Машиностроение», началось еще в конце 70-х годов прошлого века.

Высокий профессионализм, широчайшая эрудиция, чуткость и отзывчивость снискали большое уважение учеников и коллег!

Уважаемый Николай Борисович!
Редакционная коллегия и редакция журнала поздравляют
Вас с юбилеем и желают Вам крепкого здоровья,
большого счастья и новых творческих успехов
в Вашей многогранной деятельности!