

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 25

2019

№ 4

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

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

**«ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ КЭША
ДЛЯ ИСПОЛЬЗОВАНИЯ В САМОАДАПТИРУЮЩИХСЯ КОНТЕЙНЕРАХ ДАННЫХ»**

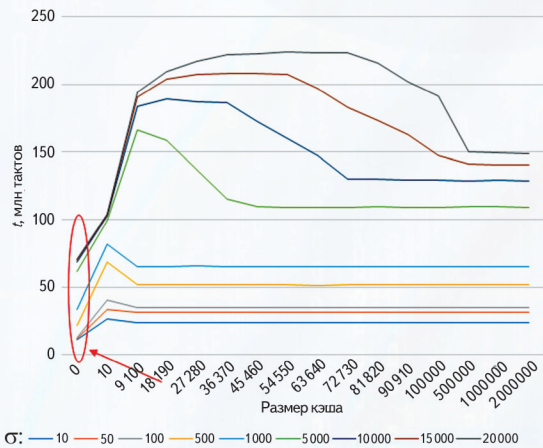


Рис. 2. Зависимость времени выполнения поиска в контейнере от размера кэша для различных σ при $k=5$

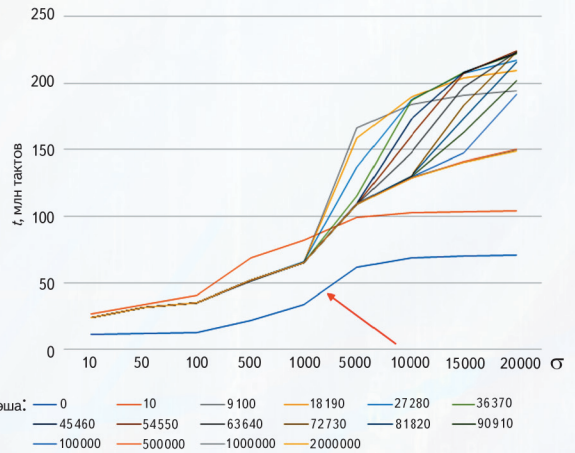


Рис. 3. Зависимость времени выполнения поиска в контейнере от σ для различных размеров кэша при $k=5$

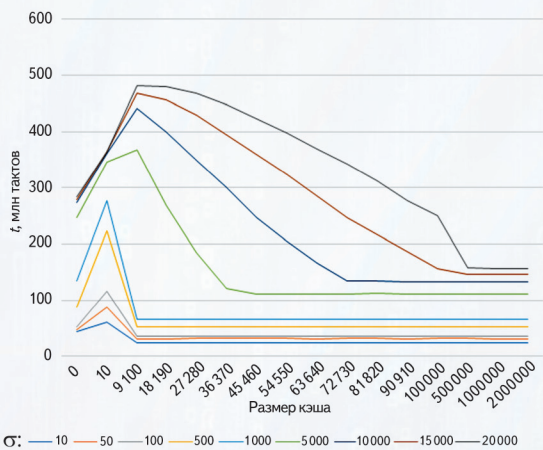


Рис. 4. Зависимость времени выполнения поиска в контейнере от размера кэша для различных σ при $k=20$

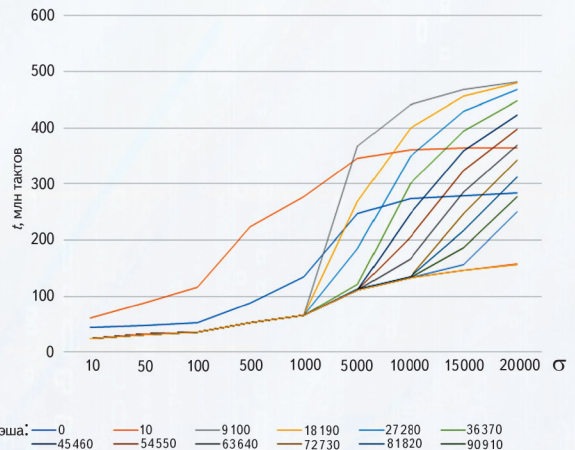


Рис. 5. Зависимость времени выполнения поиска в контейнере от σ для различных размеров кэша при $k=20$

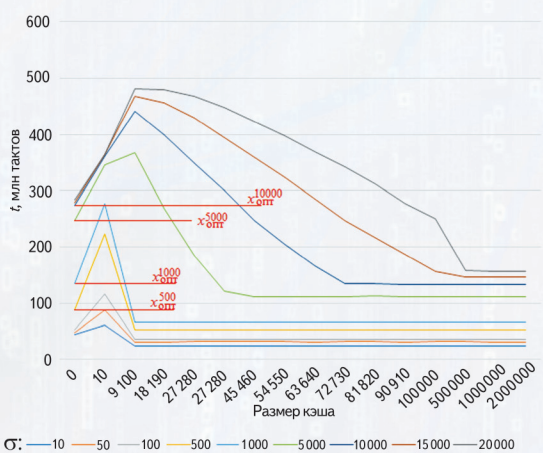


Рис. 6. Оптимальные размеры кэша для различных σ при $k=20$

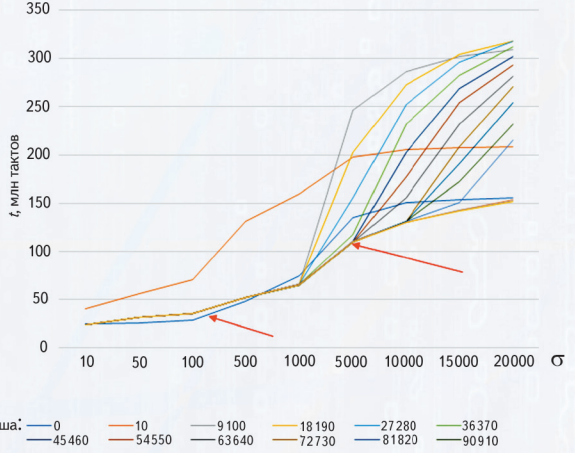


Рис. 7. Зависимость времени выполнения поиска в контейнере от σ для различных размеров кэша при $k=11$

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 25
2019
№ 4

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

МОДЕЛИРОВАНИЕ

- Левин В. И.** Непрерывная логика и анализ надежности сложных систем. Математический аппарат 195
- Тарасов В. Н., Бахарева Н. Ф., Када Отхмане.** Математическая модель теле-трафика на основе системы $HE_2/M/1$ 205
- Антониади Г. Д., Архипов В. О., Цуприков А. А.** Математическая модель задержки автотранспорта на регулируемом перекрестке 210

БАЗЫ ДАННЫХ

- Потапов Д. Р.** Исследование эффективности применения кэша для использования в самоадаптирующихся контейнерах данных 216

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Филиппов И. В., Мелик-Адамян А. Ф., Сухомлинов В. А.** Размещение сетевых функций в облачных инфраструктурах 223
- Пуртов А. М.** Использование образцов для выбора маршрутов в сетях передачи данных 228

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Торчинский В. Е., Торчинская И. В., Файнштейн А. С., Файнштейн С. И.** "Гибридный" алгоритм планирования государственных закупок товарно-материальных ценностей 234
- Полесский С. Н., Востриков А. В., Прокофьева Е. Н., Романова И. И.** Методика оценки надежности вычислительной техники с учетом влияния отказов вспомогательного оборудования 241

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Алдунин Д. А., Федин Г. Г.** Математическая модель для построения оптимальной индивидуальной образовательной траектории обучающегося при изучении массовых открытых онлайн-курсов 250

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.
ПОЛЕЩУК О. М., д. т. н., проф.
САКСОНОВ Е. А., д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
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. 25
2019
No. 4

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

MODELING

- Levin V. I.** Continuous Logic and Analysis of Reliability of Complex Systems. Mathematical Means 195
- Tarasov V. N., Bakhareva N. F., Kada Othmane.** The Mathematical Model of Tele-traffic Based on the HE₂ /M/1 System 205
- Antoniadi G. D., Arkhipov V. O., Tsouprikov A. A.** Mathematical Model of Transport Delay at the Crossroads 210

DATABASE

- Potapov D. R.** Cache Efficiency Research for Using in Adaptive Data Storage 216

COMPUTING SYSTEMS AND NETWORKS

- Philippov I. V., Melik-Adamyan A. F., Sukhomlinov V. A.** Cloud Deployment of Network Functions 223
- Purtov A. M.** Use of Samples for the Choice of Routes in Networks of Data Transmission 228

APPLICATION INFORMATION SYSTEMS

- Torchinskiy V. E., Torchinskaya I. V., Fainshtein A. S., Fainshtein S. I.** The "Hybrid" Algorithm of Planning Municipal Procurement of the Material Assets 234
- Polesskij S. N., Vostrikov A. V., Prokof'eva E. N., Romanova I. I.** Methods of Assessing the Reliability of Computing with Due Account for the Failure Effect of Auxiliary Equipment 241

INFORMATION TECHNOLOGIES IN EDUCATION

- Aldunin D. A., Fedin G. G.** Mathematical Modelling of Building Individual Educational Trajectory for Studying MOOCs 250

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.
Saksonov E. A., Dr. Sci. (Tech.), Prof.
Sokolov B. V., 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"

В. И. Левин, д-р техн. наук, проф., e-mail: vilevin@mail.ru,
Пензенский государственный технологический университет

Непрерывная логика и анализ надежности сложных систем. Математический аппарат

Обсуждается разработка автоматически-логической модели надежности сложных технических систем и соответствующих логических методов оценки надежности таких систем, которые в отличие от известных используют не традиционные вероятностные показатели надежности, а детерминированные логические показатели. Предложено использовать в качестве исходных данных наблюдаемые моменты последовательных отказов и восстановлений элементов технической системы, а в качестве характеристик надежности самой системы — моменты последовательных отказов и восстановлений системы. В этом случае задача оценки надежности системы сводится к построению ее математической модели в виде автоматных логических функций, выражающих моменты ее последовательных отказов и восстановлений через аналогичные моменты всех ее элементов. Данная статья представляет собой первую часть работы, в которой детально разрабатывается автоматически-логическая модель, предназначенная для вычисления логической функции надежности сложных технических систем. Новизна работы заключается в построении адекватной логической модели надежности сложной системы, позволяющей свести оценку надежности сложной технической системы к вычислению ее логических функций надежности. В процессе вычислений впервые используется математический аппарат логических определителей, что и позволяет решить проблему сложности. В статье детально разработаны логическая модель надежности и методы ее исследования, позволяющие вводить новые показатели надежности сложных технических систем, не требующие для своей оценки использования вероятностных методов и исходных статистических данных об отказах элементов. На основе разработанной логической модели надежности и методах ее исследования решена задача построения автоматной модели надежности систем, которая позволит вести практические расчеты сложных технических систем методами теории динамических автоматов с помощью аппарата логических определителей.

Ключевые слова: сложная система, переключательный процесс, надежность процесс, динамический автомат, двоичный оператор, структура оператора, логическая теория надежности

Введение

Имеется огромное число публикаций по классической вероятностной теории надежности [1—4] и значительно меньше — по постклассической, детерминистской теории надежности. Начало последней было положено работами автора [5—7], в которых была открыта возможность моделирования надежности процессов автоматически-логическими методами. Затем последовали различные реализации этой возможности.

В работах [8, 9] были предложены автоматически-логические модели и методы анализа надежности технических систем, основанные на математическом аппарате двузначной (булевой) и бесконечнозначной (непрерывной) логики (НЛ). Было показано, что предложен-

ные модели и методы позволяют анализировать надежность в принципе любых систем в аналитической форме, что имеет большое теоретическое и практическое значение. Затем эти модели и методы были усовершенствованы в работах [10, 11]. Но попытки непосредственного применения предложенного подхода к сложным системам, у которых логические схемы-модели сложны, а их входные процессы имеют большое число последовательных изменений сигнала (что является следствием многократного восстановления блоков системы), наталкиваются на очень большие трудности. Эти трудности обусловлены необозримостью получаемых аналитических выражений и большой сложностью их вычисления. В связи с этим в настоящей работе предложен другой подход к анализу надежности сложных систем,

основанный на математическом аппарате логических определителей (ЛО), вводимых как числовые характеристики некоторых квазиматриц, вычисляемые по соответствующим формулам алгебры НЛ [12, 13]. В целом квазиматрицы и ЛО в рассматриваемой нами области играют ту же роль параметров укрупненного (блочного) описания изучаемых, существенно нелинейных систем, что и обычные матрицы и определители в области линейных систем, т. е. способствуют лучшей обзримости и вычислимости различных характеристик изучаемых технических систем. В нашем случае это характеристики надежности.

Настоящая работа содержит математический аппарат логических определителей и другие математические заготовки, необходимые для исследования надежности сложных систем (типовые модели, методы и т. д.). Этот материал частично публиковался ранее [12, 13]. Для настоящей статьи его изложение усовершенствовано. В отдельной статье будет подробно изложена собственно методика исследования надежности сложных систем с помощью указанной математики. Работа в целом может рассматриваться как минимонография по логической теории надежности сложных систем, подробно описывающая нетрадиционный подход к изучению надежности таких систем.

Предлагаемый в работе подход нацелен на исследование надежности сложных систем и не имеет аналогов как в классической литературе по надежности, так и в современной литературе [1–4, 14–24].

1. Порядковая логика и логические определители

1. Рассмотрим множество $X = \{x_1, \dots, x_n\}$ из n элементов $x_i, x_j \in [A, B]$, и расположим элементы в порядке неубывания:

$$x^{(1)} \leq x^{(2)} \leq \dots \leq x^{(n)}, x^{(r)} \in X. \quad (1.1)$$

Введем над множеством X операцию выделения произвольного порядкового элемента $x^{(r)}$ этого множества (*r-операцию*):

$$y \equiv f^{(r)}(x_1, \dots, x_n) = x^{(r)}, r = 1, \dots, n. \quad (1.2)$$

Здесь r называется *рангом* операции. Легко видеть, что r -операция обобщает операции конъюнкции $\wedge = \min$ и дизъюнкции $\vee = \max$ непрерывной логики (НЛ), переходя в них соответственно при $r = 1$ и $r = n$. Результатом r -операции над элементами множества является один из элементов этого же множества. Назовем

произвольную функцию, аргументы которой x_1, \dots, x_n взяты из множества X и которая представляется в виде суперпозиции r -операций над X с различными значениями ранга r , *функцией порядковой логики*. Простейший пример такой функции — сама r -функция (1.2). Более сложный пример — функция $y = f^{(2)}[f^{(2)}(x_1, x_2, x_3), f^{(3)}(x_1, x_2, x_3, x_4)]$. Любая функция порядковой логики $y = f(x_1, \dots, x_n)$ на любом наборе аргументов (x_1, \dots, x_n) принимает значение одного из аргументов. Это связано с тем, что r -операции, суперпозицией которых является выражение y , всегда имеют своим результатом одну из переменных, участвующих в операциях.

Задать функцию порядковой логики $y = f(x_1, \dots, x_n)$ можно, перечислив все $n!$ вариантов упорядочения аргументов x_1, \dots, x_n и указав для каждого варианта аргумент x_i , значение которого принимает функция. Такое задание функции порядковой логики есть частный случай первичного задания любой функции непрерывной логики. Поэтому от такого первичного задания функции порядковой логики можно перейти к ее аналитическому представлению с помощью суперпозиции операций НЛ — конъюнкции и дизъюнкции (отрицание здесь не участвует, так как r -операция всегда имеет своим результатом значение одной из переменных, но не ее отрицания). Методика перехода та же, что и для функций НЛ.

Пример 1. Функция порядковой логики $y = f^{(2)}(x_1, x_2, x_3)$ — медиана — задана таблицей. Требуется найти ее представление с помощью НЛ.

Задание функции-медианы $y = f^{(2)}(x_1, x_2, x_3)$

Упорядочение аргументов	Значение функции	Упорядочение аргументов	Значение функции
$x_1 \leq x_2 \leq x_3$	x_2	$x_2 \leq x_3 \leq x_1$	x_3
$x_1 \leq x_3 \leq x_2$	x_3	$x_3 \leq x_1 \leq x_2$	x_1
$x_2 \leq x_1 \leq x_3$	x_1	$x_3 \leq x_2 \leq x_1$	x_2

Согласно таблице искомую функцию можно представить так:

$$y = \begin{cases} x_1 & \text{при } x_2 \leq x_1 \leq x_3 \text{ или } x_3 \leq x_1 \leq x_2; \\ x_2 & \text{при } x_1 \leq x_2 \leq x_3 \text{ или } x_3 \leq x_2 \leq x_1; \\ x_3 & \text{при } x_1 \leq x_3 \leq x_2 \text{ или } x_2 \leq x_3 \leq x_1. \end{cases}$$

Объединим с помощью конъюнкции НЛ первую строку при втором условии со второй строкой при втором условии, первую строку при первом условии с третьей строкой при втором условии и вторую строку при первом условии с третьей строкой при первом условии:

$$y = \begin{cases} x_1x_2 & \text{при } x_1x_2 \geq x_3 \\ \text{(т. е. при } x_1x_2 \geq x_1x_3, x_2x_3); \\ x_1x_3 & \text{при } x_1x_3 \geq x_2 \\ \text{(т. е. при } x_1x_3 \geq x_1x_2, x_2x_3); \\ x_2x_3 & \text{при } x_2x_3 \geq x_1 \\ \text{(т. е. при } x_2x_3 \geq x_1x_2, x_1x_3). \end{cases}$$

Объединяя теперь три строки в одну с помощью операции дизъюнкции НЛ, получаем искомое представление: $y = x_1x_2 \vee x_1x_3 \vee x_2x_3$.

Из сказанного следует, что функции порядковой логики — отдельный класс функций НЛ. Поэтому выражения функций порядковой логики можно подвергать эквивалентным преобразованиям (в целях их упрощения) с помощью законов НЛ. Однако некоторые законы присущи лишь порядковой логике:

- *тавтология*

$$f^{(r)}(x, \dots, x) = x; \quad (1.3)$$

- *переместительный*

$$f^{(r)}(x_1, \dots, x_n) = f^{(r)}(x_{i_1}, \dots, x_{i_n}) \quad (1.4)$$

(здесь x_{i_1}, \dots, x_{i_n} — любая перестановка аргументов x_1, \dots, x_n);

- *распределительный*

$$f^{(r)}[\varphi^{(q_1)}(x_1, \dots, x_n), \varphi^{(q_2)}(x_1, \dots, x_n), \dots, \varphi^{(q_p)}(x_1, \dots, x_n)] = \varphi^{(q_r)}(x_1, \dots, x_n) \quad (1.5)$$

(здесь $q_1 < q_2 < \dots < q_p$; $1 \leq r \leq p$)

и его частные случаи

$$\bigwedge_{i=1}^n f^{(r_i)}(x_1, \dots, x_n) = f^{\left(\bigwedge_{i=1}^n r_i\right)}(x_1, \dots, x_n); \quad (1.6)$$

$$\bigvee_{i=1}^n f^{(r_i)}(x_1, \dots, x_n) = f^{\left(\bigvee_{i=1}^n r_i\right)}(x_1, \dots, x_n).$$

С помощью этих законов можно преобразовывать представления функций порядковой логики, не являющиеся выражениями НЛ.

2. Рассмотрим множество X_q , состоящее из q непересекающихся подмножеств $(x_{i1}, \dots, x_{im_i})$, $i = 1, \dots, q$, с элементами $x_{ij} \in [A, B]$, упорядоченными согласно условию

$$x_{i1} \leq x_{i2} \leq \dots \leq x_{im_i}, \quad i = 1, \dots, q. \quad (1.7)$$

Число элементов этого множества $n = \sum_{i=1}^q m_i$. Множество X_q частично упорядоченное; его удобно записывать в виде *квазиматрицы* q -го порядка со строками — упорядоченными подмножествами

$$\mathbf{X}_q = \left\| \begin{array}{c} x_{11} \dots x_{1m_1} \\ \dots \\ x_{q1} \dots x_{qm_q} \end{array} \right\| = \|x_{ij}\|, \quad (1.8)$$

$$i = 1, \dots, q; j = 1, \dots, m_i.$$

Квазиматрица (1.8) отличается от обычной матрицы неодинаковым числом элементов в различных строках и упорядоченностью элементов в строках согласно (1.7). Рассмотренное выше неупорядоченное множество $X = (x_1, \dots, x_n)$ есть частный случай множества (1.8) при n строках из одного элемента каждая. Поэтому неупорядоченное множество X можно записать в следующем виде *матрицы-столбца*:

$$\mathbf{X}_n = \left\| \begin{array}{c} x_1 \\ \dots \\ x_n \end{array} \right\|. \quad (1.9)$$

В другом частном случае, когда множество X_q полностью упорядочено, оно содержит лишь одно упорядоченное подмножество (одну строку в (1.8)). В этом случае матричная запись множества X_q имеет вид *матрицы-строки*:

$$\mathbf{X}_n = \|x_1, \dots, x_n\|. \quad (1.10)$$

Для частично упорядоченного множества X_q , заданного своей квазиматрицей (1.8), как и для упорядоченного множества X , вводится r -операция (1.2) в виде функции

$$y = f^{(r)}(x_{11}, \dots, x_{qm_q}) = x^{(r)}, \quad r = 1, \dots, n, \quad (1.11)$$

выделяющей нужный порядковый элемент $x^{(r)}$ из X_q . Эта функция называется *логическим определителем* (ЛО) r -го ранга q -го порядка от квазиматрицы $X_q = \|x_{ij}\|$ и обозначается

$$X_q^r = \left| \frac{x_{11} \dots x_{1m_1}}{x_{q1} \dots x_{qm_q}} \right|^{(r)} = \|x_{ij}\|^{(r)}, \quad r = 1, \dots, n. \quad (1.12)$$

Специально отметим частные случаи — *опредетель-столбец*

$$X_n^r = \left\| \frac{x_1}{x_n} \right\|^{(r)}, \quad r = 1, \dots, n, \quad (1.13)$$

соответствующий матрице-столбцу (1.9) и совпадающий с обычной r -функцией вида (1.2), и *опредетель-строку*

$$X_1^r = |x_1, \dots, x_n|^{(r)} = x_r, \quad r = 1, \dots, n, \quad (1.14)$$

соответствующий матрице-строке (1.10). Логический определитель X_q^r от квазиматрицы X_q является числовой характеристикой этой квазиматрицы, так же как обычный определитель (детерминант) есть характеристика квадрат-

ной матрицы. Формально ЛО — это обобщение обычной r -функции (1.2) на случай частично упорядоченного множества аргументов, сохраняющее все основные черты r -функции. Так, любой ЛО $X_q^r = |x_{ij}|^{(r)}$ на любом наборе элементов x_{11}, \dots, x_{qm_q} принимает значение одного из элементов. Далее, любая функция, аргументы которой являются элементами x_{11}, \dots, x_{qm_q} квазиматрицы X_q и которая представлена в виде суперпозиции ЛО X_q^r различных рангов r от X_q , есть функция порядковой логики. Так что ЛО и суперпозицию ЛО можно задать, указав для каждого варианта упорядочения элементов x_{11}, \dots, x_{qm_q} элемент x_{ij} , значение которого принимает функция. От такого задания ЛО можно перейти к их аналитическому представлению с помощью операций НЛ (пример 1). Значит, ЛО и их суперпозиции образуют специальный класс функций НЛ. Их можно подвергать эквивалентным преобразованиям с помощью законов НЛ и порядковой логики (1.3)—(1.6).

2. Свойства логических определителей

Свойство 1. ЛО является монотонно неубывающей функцией ранга

$$X_q^r \geq X_q^p, \text{ если } r > p. \quad (2.1)$$

Свойство 2. Перестановка любых двух строк ЛО X_q^r не меняет его значения.

Доказательства свойств 1, 2 вытекают из определения X_q^r .

Свойство 3. Общее для всех элементов определителя слагаемое можно вынести за знак ЛО:

$$|x_{ij} + c|^{(r)} = |x_{ij}|^{(r)} + c. \quad (2.2)$$

Доказательство: прибавление общего слагаемого ко всем элементам x_{ij} не меняет их взаимной упорядоченности согласно (1.7).

Свойство 4. Общий для всех элементов дизъюнктивный (конъюнктивный) член можно вынести за знак ЛО:

$$|x_{ij} \vee c|^{(r)} = |x_{ij}|^{(r)} \vee c; |x_{ij} \wedge c|^{(r)} = |x_{ij}|^{(r)} \wedge c. \quad (2.3)$$

Доказательство следует из того, что введение общего для всех элементов дизъюнктивного (конъюнктивного) члена c не меняет взаимной упорядоченности элементов, а лишь приводит к замене на c тех из них, которые вначале были меньше (больше) c .

Свойство 5. Общий для всех элементов сомножитель c можно вынести за знак ЛО с сохранением первоначального ранга r , если $c > 0$,

и с заменой его на дополнительный ранг $n - r + 1$ при $c < 0$:

$$|x_{ij}c|^{(r)} = \begin{cases} c|x_{ij}|^{(r)}, & c > 0; \\ c|x_{ij}|^{(n-r+1)}, & c < 0. \end{cases} \quad (2.4)$$

Доказательство: в случае $c > 0$ упорядоченность значений $x_{ij}c$ и x_{ij} ($i = 1, \dots, q; j = 1, \dots, m_i$) одинаковая, а в случае $c < 0$ — обратная (максимальному x_{ij} соответствует минимальное $x_{ij}c$ и т. д.).

Свойство 6. Если $c > x_{im_i}$ ($i = 1, \dots, q$), то значение ЛО не меняется при добавлении к нему справа столбца из элементов c :

$$\begin{aligned} \frac{|x_{11} \dots x_{1m_i} c|^{(r)}}{|x_{q1} \dots x_{qm_q} c|^{(r)}} &= \\ &= \begin{cases} \frac{|x_{11} \dots x_{1m_i}|^{(r)}}{|x_{q1} \dots x_{qm_q}|^{(r)}}, & r = 1, \dots, n; \\ c, & r = n + 1, \dots, n + q. \end{cases} \end{aligned} \quad (2.5)$$

Свойство 7. При добавлении к ЛО столбца из c при $c < x_{i1}$ ($i = 1, \dots, q$) значение ЛО не изменится, если его ранг уменьшить на число строк:

$$\begin{aligned} \frac{|c x_{11} \dots x_{1m_i}|^{(r)}}{|c x_{q1} \dots x_{qm_q}|^{(r)}} &= \\ &= \begin{cases} c, & r = 1, \dots, q; \\ \frac{|x_{11} \dots x_{1m_i}|^{(r-q)}}{|x_{q1} \dots x_{qm_q}|^{(r-q)}}, & r = q + 1, \dots, q + n. \end{cases} \end{aligned} \quad (2.6)$$

Свойство 8. Значение ЛО не меняется при исключении элемента ∞ (бесконечность) в конце какой-либо строки:

$$\frac{|x_{11} \dots x_{1m_i} \infty|^{(r)}}{|x_{q1} \dots x_{qm_q}|^{(r)}} = \begin{cases} \frac{|x_{11} \dots x_{1m_i}|^{(r)}}{|x_{q1} \dots x_{qm_q}|^{(r)}}, & r = 1, \dots, n; \\ \infty, & r = n + 1. \end{cases} \quad (2.7)$$

Свойство 9. Значение ЛО не меняется, если из него исключить элемент $-\infty$ в начале какой-либо строки, а ранг уменьшить на единицу:

$$\begin{aligned} \frac{|-\infty x_{11} \dots x_{1m_i}|^{(r)}}{|x_{q1} \dots x_{qm_q}|^{(r)}} &= \\ &= \begin{cases} -\infty, & r = 1, \dots, q; \\ \frac{|x_{11} \dots x_{1m_i}|^{(r-1)}}{|x_{q1} \dots x_{qm_q}|^{(r-1)}}, & r = q + 1, \dots, q + n. \end{cases} \end{aligned} \quad (2.8)$$

Доказательства свойств 6—9 вытекают из определения ЛО.

Свойство 10. Значение ЛО не изменится, если любую совокупность строк заменить ЛО, образованными этой совокупностью и расположенными в одной строке в порядке возрастания ранга:

$$X_q^r = \left| \frac{X_{q \setminus i \dots k}}{X_{i \dots k}^1 X_{i \dots k}^2 \dots X_{i \dots k}^N} \right|^{(r)} \quad (2.9)$$

Здесь $X_{q \setminus i \dots k}$ — квазиматрица, полученная из квазиматрицы X_q исключением строк i, \dots, k ;

$X_{i \dots k}^p = \left| \frac{i}{k} \right|^{(p)}$ — ЛО p -го ранга из строк i, \dots, k .

Доказательство: указанная замена означает совместное упорядочение элементов строк i, \dots, k и не влияет на значение порядкового элемента $x^{(r)}$ множества X_q , а следовательно, и на значение X_q^r .

Свойство 11. ЛО q -го порядка с двумя одинаковыми строками можно представить как ЛО $(q - 1)$ -го порядка с различными строками:

$$\left| \frac{x_{11} \dots x_{1m_1}}{x_{q-1,1} \dots x_{q-1,m_{q-1}}} \right|^{(r)} = \left| \frac{x_{11} \dots x_{1m_1}}{x_{q-1,1} x_{q-1,1} \dots x_{q-1,m_{q-1}} x_{q-1,m_{q-1}}} \right|^{(r)} \quad (2.10)$$

Доказательство: такая перестановка удовлетворяет условию упорядоченности элементов в строках (1.7), т. е. снова дает логический определитель, причем не меняет его значения.

Свойство 12. Конечный ЛО можно представить как бесконечный:

$$\left| \frac{x_{11} \dots x_{1m_1}}{x_{q1} \dots x_{qm_q}} \right|^{(r)} = \left| \frac{x_{11} \dots x_{1m_1} \infty \infty \dots}{x_{q1} \dots x_{qm_q} \infty \infty \dots} \right|^{(r)} \quad (2.11)$$

Доказательство данного свойства получается повторным применением формулы (2.7).

Бесконечности в формуле (2.11) можно заменить конечными элементами x_{ik} , $k > m_i$, такими чтобы сохранилась упорядоченность (1.7) элементов в строках и выполнялись неравенства

$$x_{ik} \geq \bigvee_{i=1}^q x_{im_i}, \quad i = 1, \dots, q.$$

Свойство 13. Значение ЛО r -го ранга не изменится, если в любой i -й строке исключить элементы $x_{i, r+1}; x_{i, r+2} \dots$

$$\left| \frac{x_{11} \dots x_{1m_1}}{x_{q1} \dots x_{qm_q}} \right|^{(r)} = \left| \frac{x_{11} \dots x_{1r_1}}{x_{q1} \dots x_{qr_q}} \right|^{(r)}, \quad (2.12)$$

где $r_i = r \wedge m_i$.

Доказательство: действительно, r -м порядковым элементом $x^{(r)}$ квазиматрицы X_q может быть только один из r первых элементов какой-либо ее строки.

Свойство 14 (закон тавтологии):

$$\left| \frac{x \dots x}{x \dots x} \right|^{(r)} = x, \quad r = 1, \dots, n. \quad (2.13)$$

Доказательство следует из определения ЛО.

Свойство 15 (распределительный закон):

$$\left| \frac{X_q^{p_{11}} \dots X_q^{p_{1m_1}}}{X_q^{p_{s1}} \dots X_q^{p_{sm_s}}} \right|^{(r)} = X_q^{p_s^r}, \quad (2.14)$$

$$\text{где } P_s^r = \left| \frac{p_{11} \dots p_{1m_1}}{p_{s1} \dots p_{sm_s}} \right|^{(r)}.$$

Здесь $p_{i1} < p_{i2} < \dots < p_{im_i}$, $r = 1, 2, \dots, n$, $n = \sum_{i=1}^s m_i$.

Свойство 16 (частный случай распределительного закона):

$$\bigwedge_{i=1}^n X_q^{p_i} = X_q^{\bigwedge_{i=1}^n p_i}. \quad (2.15)$$

Свойство 17 (частный случай распределительного закона):

$$\bigvee_{i=1}^n X_q^{p_i} = X_q^{\bigvee_{i=1}^n p_i}. \quad (2.16)$$

Доказательство свойств 15—17 вытекает из свойства 1. По нему упорядочение множества ЛО X_q^r различных рангов r от одной квазиматрицы X_q можно заменить упорядочением множества рангов.

3. Раскрытие логического определителя

Раскрыть ЛО — значит указать аналитическое представление функции НЛ, выражающей значение ЛО через значения его элементов. В п. 1 был предложен прямой метод раскрытия ЛО. Однако этот метод громоздок и не работает в случае больших ЛО. Удобнее раскрывать ЛО по готовым формулам, дающим аналитическое представление функции НЛ, выражающей значения целого класса ЛО.

1. ЛО-столбец r -го ранга с n элементами выражается дизъюнктивной нормальной формой (ДНФ):

$$X_n^r = \left| \frac{x_1}{x_n} \right|^{(r)} = \bigvee_{i_1 \neq \dots \neq i_{n-r+1}}^n (x_{i_1} \dots x_{i_{n-r+1}}), x_{i_k} \in \{x_1, \dots, x_n\} \quad (3.1)$$

или конъюнктивной нормальной формой (КНФ):

$$X_n^r = \left| \frac{x_1}{x_n} \right|^{(r)} = \bigwedge_{i_1 \neq \dots \neq i_r}^n (x_{i_1} \vee \dots \vee x_{i_r}), x_{i_k} \in \{x_1, \dots, x_n\}. \quad (3.2)$$

Доказательство (3.1). Пусть $x^{(1)}, \dots, x^{(n)}$ — упорядоченные согласно (1.1) элементы x_1, \dots, x_n . Каждая конъюнкция состоит из $n - r + 1$ различных элементов. Одна конъюнкция вида $b = x^{(r)}x^{(r+1)} \dots x^{(n)}$, остальные вида $b_i = x^{(s)}b_i'$, где $s < r$, т. е. $b = x^{(r)}$, $b_i \leq x^{(r)}$ и правая часть выражения (3.1) равна $x^{(r)}$, т. е. левой части. Формула (3.2) доказывается аналогично.

2. Общий бесконечный ЛО r -го ранга q -порядка выражается ДНФ:

$$X_q^r = \left| \frac{x_{11} \dots x_{1i_1} \dots}{x_{21} \dots x_{qi_q} \dots} \right|^{(r)} = \bigvee_{\sum_{s=1}^q i_s = r+q-1}^n (x_{1i_1} \dots x_{qi_q}). \quad (3.3)$$

Доказательство. Сначала докажем частный случай (3.3) при $q = 2$:

$$X_2^r = \left| \frac{x_{11} \dots x_{1i_1} \dots}{x_{21} \dots x_{2i_2} \dots} \right|^{(r)} = \bigvee_{k=1}^r (x_{1k} x_{2,r+1-k}). \quad (3.4)$$

Согласно свойству 13 логический определитель X_2^r можно представить как конечный ЛО:

$$X_2^r = \left| \frac{x_{11} \dots x_{1r}}{x_{21} \dots x_{2r}} \right|^{(r)}, \quad (3.5)$$

который, не учитывая упорядоченности элементов в строках, представим в виде ЛО-столбца

$$X_2^r = \left| \frac{x_{11}}{x_{1r}} \right| \dots \left| \frac{x_{21}}{x_{2r}} \right|. \quad (3.6)$$

Раскроем ЛО (3.6) по формуле (3.1). Каждая конъюнкция в (3.1) включает $r + 1$ различных

элементов. Из этих элементов хотя бы один вида x_{1i} и хотя бы один вида x_{2j} . Пусть B_{ks} — s -я из конъюнкций, включающих k элементов вида x_{2j} и $r + 1 - k$ элементов вида x_{1i} . Тогда согласно

(3.1) $X_2^r = \bigvee_{k=1}^r \bigvee_s B_{ks}$. При фиксированном k по условию (1.7) максимальна та конъюнкция B_{ks} ($s = 1, 2, \dots$), в которую входят элементы $x_{1k}, \dots, x_{1r}; x_{2,r+1-k}, \dots, x_{2r}$; она равна $x_{1k}x_{2,r+1-k}$. Отсюда $\bigvee_s B_{ks} = x_{1k}x_{2,r+1-k}$. Подставив это в выражение X_2^r , получим (3.4). Теперь формулу (3.3) докажем индукцией по q . При $q = 1$ (3.3) переходит в равенство $X_1^r = |x_{11} \dots x_{1i_1} \dots|^{(r)} = x_{1r}$, (см. (1.14)), а при $q = 2$ — в доказанное соотношение (3.4). Допустим, что формула (3.3) верна для некоторого $q = p$. Докажем, что тогда она верна и при $q = p + 1$. Представим X_{p+1}^r по правилу (2.9) в виде блочного определителя 2-го порядка:

Раскроем последний по формуле (3.4) $X_{p+1}^r = \bigvee_{k=1}^r X_p^k x_{p+1,r+1-k}$. Согласно вышеуказанному допущению ЛО X_p^k можно выразить в виде (3.3). Отсюда

$$X_{p+1}^r = \left| \frac{x_{11} \dots x_{1i_1} \dots}{x_{p1} \dots x_{pi_p} \dots} \right|^{(r)} = \left| \frac{X_p^1 \dots X_p^i \dots}{X_{p+1,1} \dots X_{p+1,i_{p+1}} \dots} \right|^{(r)}.$$

Раскроем последний по формуле (3.4) $X_{p+1}^r = \bigvee_{k=1}^r X_p^k x_{p+1,r+1-k}$. Согласно вышеуказанному допущению ЛО X_p^k можно выразить в виде (3.3). Отсюда

$$X_{p+1}^r = \bigvee_{k=1}^r \left(\bigvee_{\sum_{s=1}^p i_s = k+p-1} x_{1i_1} \dots x_{pi_p} \right) x_{p+1,r+1-k} = \bigvee_{k=1}^r \bigvee_{\sum_{s=1}^{p+1} i_s = r+p} x_{1i_1} \dots x_{pi_p} x_{p+1,r+1-k} = \bigvee_{\substack{\sum_{s=1}^{p+1} i_s = r+p, \\ 1 \leq i_{p+1} \leq r}} a_{1i_1} \dots a_{p+1,i_{p+1}}.$$

В последнем полученном выражении по свойству 13 можно опустить условие $1 \leq i_{p+1} \leq r$, не изменив значение ЛО. В итоге будет справедливо равенство $X_{p+1}^r = \bigvee_{\sum_{s=1}^{p+1} i_s = r+p} x_{1i_1} \dots x_{p+1,i_{p+1}}$,

что и требовалось доказать.

3. Общий конечный ЛО r -го ранга q -го порядка выражается ДНФ:

$$X_q^r = \left| \begin{array}{c} x_{11} \dots x_{1m_1} \\ \hline x_{q1} \dots x_{qm_q} \end{array} \right|^{(r)} = \bigvee_{s=1}^q \left(\tilde{x}_{1i_1}^{m_1} \dots \tilde{x}_{qi_q}^{m_q} \right). \quad (3.7)$$

Здесь и ниже запись $\tilde{x}_{1i_1}^{m_1}$ означает, что элемент x_{ki_k} исключается из тех конъюнкций, для которых из условия на Σi_s формально получается $i_k > m_k$.

Доказательство формулы (3.7) получается, если в соответствии с соотношением (2.11) представить конечный ЛО X_q^r как бесконечный и применить к последнему правило раскрытия (3.3), учитывая, что $x \wedge \infty = x$.

Пример 2. По формуле (3.1) раскроем ЛО-столбец

$$X_3^r = \left| \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \right|^{(r)} = \begin{cases} x_1 x_2 x_3, & r = 1; \\ x_1 x_2 \vee x_1 x_3 \vee x_2 x_3, & r = 2; \\ x_1 \vee x_2 x_3, & r = 3. \end{cases}$$

Второе из выписанных выражений было получено более сложным путем — с использованием прямого метода в примере 1.

Пример 3. По формуле (3.7) раскроем общий ЛО 2-го порядка:

$$X_3^r = \left| \begin{array}{cc} x_{11} & x_{12} \\ x_{21} & x_{22} \end{array} \right|^{(r)} = \begin{cases} x_{11} x_{21}, & r = 1; \\ x_{11} x_{22} \vee x_{12} x_{21}, & r = 2; \\ x_{12} x_{22} \vee x_{11} x_{21}, & r = 3; \\ x_{12} x_{22}, & r = 4. \end{cases}$$

4. Раскрытие больших логических определителей

1. Раскрытие больших ЛО (т. е. ЛО с большим числом элементов) по явным формулам п. 3 слишком трудоемко. В таких случаях целесообразнее применять *разложения* ЛО на ЛО меньших размеров. Простейшее такое разложение — (2.9).

Пример 4. Раскроем ЛО 4-го порядка

$$X_4^4 = \left| \begin{array}{c} x_{11} x_{12} \\ x_{21} x_{22} \\ x_{31} \\ x_{41} \end{array} \right|^{(4)}.$$

Запишем данный ЛО как блочный ЛО 2-го порядка, объединив первую строку со второй и третью с четвертой:

$$X_2^4 = \left| \begin{array}{ccc} B_2^1 & B_2^2 & B_2^3 \\ C_2^1 & C_2^2 & B_2^4 \end{array} \right|^{(4)},$$

где

$$B_2^r = \left| \begin{array}{cc} x_{11} x_{12} \\ x_{21} x_{22} \end{array} \right|^{(r)}, \quad r = 1, \dots, 4;$$

$$C_2^r = \left| \begin{array}{c} x_{31} \\ x_{41} \end{array} \right|^{(r)}, \quad r = 1, 2.$$

Раскроем теперь ЛО X_2^4 по формуле (3.7):

$$X_2^4 = B_2^2 \vee B_2^3 C_2^2 \vee B_2^4 C_2^1.$$

Остается подставить сюда выражения B_2^r из *примера 3* и значение определителя C_2^r :

$$C_2^r = \begin{cases} x_{31} x_{41}, & r = 1; \\ x_{31} \vee x_{41}, & r = 2. \end{cases}$$

Получаем окончательно выражение ЛО, сложность которого — 13 двухместных операций \vee и \wedge НЛ:

$$X_2^4 = x_{11} x_{22} \vee x_{12} x_{21} \vee (x_{12} x_{22} \vee x_{11} \vee x_{21}) \wedge \wedge (x_{31} \vee x_{41}) \vee (x_{12} \vee x_{22}) x_{31} x_{41}.$$

Раскрытие этого ЛО по формуле (3.7) дает выражение

$$X_2^4 = x_{11} x_{22} \vee x_{12} x_{21} \vee x_{11} x_{31} \vee x_{11} x_{41} \vee \vee x_{21} x_{31} \vee x_{21} x_{41} \vee x_{12} x_{22} x_{31} \vee x_{12} x_{22} x_{41} \vee \vee x_{12} x_{31} x_{41} \vee x_{22} x_{31} x_{41},$$

сложность которого — 23 операции.

Более конкретные правила разложения ЛО, когда однозначно указываются блоки, на которые разлагается ЛО, изложены ниже.

2. Назовем *логическим дополнением* элемента x_{ij} в ЛО X_q^r ЛО, полученный из X_q^r исключением элемента x_{ij} . Обозначим его $X_q^r \setminus x_{ij}$. ЛО-столбец r -го ранга с n элементами можно разложить поэлементно по следующей ДНФ:

$$X_n^r = \left| \frac{x_1}{x_n} \right|^{(r)} = \bigvee_{i=1}^n x_i X_n^r \setminus x_i. \quad (4.1)$$

Доказательство. Раскрыв ЛО $X_n^r \setminus x_i$ в правой части по правилу (3.1), получим раскрытый по этому правилу ЛО левой части X_n^r .

Общий ЛО r -го ранга q -го порядка можно разложить поэлементно по ДНФ:

$$X_q^r = \left| \frac{x_{11} \dots x_{1m_1}}{x_{q1} \dots x_{qm_q}} \right|^{(r)} = \bigvee_{i,j} x_{ij} X_q^r \setminus x_{ij}. \quad (4.2)$$

Доказательство. Рассматривая ЛО X_q^r без учета упорядоченности элементов в строках (т. е. как ЛО-столбец), применяем к нему формулу (4.1).

Формулы (4.1), (4.2) задают разложения ЛО по элементам.

3. Пусть $X_q^r = \left| \frac{x_{11} \dots x_{1i_1} \dots}{x_{q1} \dots x_{qi_q} \dots} \right|^{(r)}$ — общий бесконечный ЛО r -го ранга q -го порядка, а $X_{d,b}^s$ — это блок-ЛО s -го ранга, составленный из строк $d, d+1, \dots, b$ ЛО X_q^r . Справедливо разложение ЛО по блокам:

$$X_q^r = \bigvee_{\sum_{i=1}^p s_i = r+p-1} X_{1,k_1}^{s_1} X_{k_1+1,k_2}^{s_2} \dots X_{k_{p-1}+1,q}^{s_p}. \quad (4.3)$$

Доказательство. Представим X_q^r в блочном виде (2.9):

$$X_q^r = \left| \frac{X_{1,k_1}^1 \dots X_{1,k_1}^{i_1} \dots}{X_{k_1+1,k_2}^1 \dots X_{k_1+1,k_2}^{i_2} \dots}{X_{k_{p-1}+1,q}^1 \dots X_{k_{p-1}+1,q}^{i_p} \dots} \right|^{(r)}.$$

Рассматривая теперь блоки $X_{d,b}^s$ как элементы ЛО X_q^r , раскроем его по формуле (3.3). Получим соотношение (4.3).

Пусть $X_q^r = \left| \frac{x_{11} \dots x_{1m_1}}{x_{q1} \dots x_{qm_q}} \right|^{(r)}$ — общий конечный

ЛО r -го ранга q -го порядка, а $X_{d,b}^s$ — см. выше. Тогда справедливо разложение ЛО по блокам:

$$X_q^r = \bigvee_{\sum_{i=1}^p s_i = r+p-1} \tilde{X}_{1,k_1}^{s_1} \tilde{X}_{k_1+1,k_2}^{s_2} \dots \tilde{X}_{k_{p-1}+1,q}^{s_p}. \quad (4.4)$$

Здесь M_i — это число элементов в соответствующем блок-ЛО $X_{d,b}^{s_i}$, а запись $\tilde{X}_{d,b}^{s_i}$ означает, что ЛО $X_{d,b}^{s_i}$ не входит в те конъюнкции, для которых из условия на Σs_i получается $s_i > M_i$. Доказательство повторяет доказательство разложения (4.3), но с раскрытием ЛО по формуле (3.7).

4. Разложения (2.9), (4.1)—(4.4) составляют основу иерархических процедур раскрытия ЛО. В такой процедуре 1-й шаг — это разложение вычисляемого ЛО $X_q^r = |X_{ij}|_q^r$ по одной из формул (2.9), (4.1)—(4.4) на блоки-ЛО низшего порядка; 2-й шаг — разложение получившихся ЛО на ЛО еще более низкого порядка и т. д., пока не придем к выражению исходного ЛО

через ЛО 1 порядка, т. е. элементы x_{ij} . Иерархическая процедура раскрытия ЛО показана выше в примере 4.

Трудоемкость такой процедуры и сложность получаемого выражения ЛО зависят от формулы разложения ЛО и способа его разделения на блоки. Наибольший эффект достигается при использовании поблочных разложений с делением на каждом шаге имеющихся ЛО на два равновеликих. При этом формулы разложения бесконечного (4.3) и конечного (4.4) ЛО принимают соответственно вид:

$$X_q^r = \bigvee_{i+j=r+1} X_{1,|q/2|}^i X_{|q/2|+1,q}^j; \quad (4.5)$$

$$X_q^r = \bigvee_{i+j=r+1} \tilde{X}_{1,|q/2|}^{M_1} \tilde{X}_{|q/2|+1,q}^{M_2} \quad (|a| \text{ — целая часть } a). \quad (4.6)$$

Получаемые по ним выражения ЛО обладают сложностью

$$\tilde{N}_q^r = r^2(q-1) + 2r - 1; \quad (4.7)$$

$$\tilde{N}_q^r = krn, \quad k \leq 2 \quad (k = \text{const}). \quad (4.8)$$

Оценка (4.8) получена в предположении одинакового числа элементов m во всех q строках ЛО; в ней n — общее число элементов ($n = mq$). Использование дихотомических блочных разложений (4.5), (4.6) обеспечивает раскрытие больших ЛО с приемлемой сложностью вычислений.

5. При раскрытии особенно больших ЛО получаемые с помощью разложений (4.5), (4.6) выражения ЛО могут оказаться недопустимо сложными. В таких случаях целесообразно приближенное раскрытие ЛО, основанное на получении двусторонних аналитических оценок значения ЛО. Эти оценки имеют следующий вид:

для ЛО-столбца

$$(x_1 \dots x_{n-r+1}) \vee (x_{n-r+2} \dots x_{2(n-r+1)}) \vee \dots \vee (x_{(M_1-1)(n-r+1)} \dots x_{M_1(n-r+1)}) \vee \vee (x_r \dots x_n) \leq \left| \frac{x_1}{x_n} \right|^{(r)} \leq \quad (4.9)$$

$$\leq (x_1 \vee \dots \vee x_r) \wedge (x_{r+1} \vee \dots \vee x_{2r}) \wedge \dots \wedge (x_{(M_2-1)r+1} \vee \dots \vee x_{M_2r}) \wedge (x_{n-r+1} \vee \dots \vee x_n),$$

где $M_1 = \lfloor n/(n-r+1) \rfloor$, $M_2 = \lfloor n/r \rfloor$; для общего бесконечного ЛО

$$x_{1l} \wedge \dots \wedge x_{ql} \leq \left| \frac{x_{11} \dots x_{1i_1} \dots}{x_{q1} \dots x_{qi_q} \dots} \right|^{(r)} \leq x_{1l} \vee \dots \vee x_{ql}, \quad (4.10)$$

где $l = [r/q]$ и $[\cdot]$ — символ округления до ближайшего большего целого числа;

для общего конечного ЛО

$$\tilde{x}_{l_1}^{m_1} \wedge \dots \wedge \tilde{x}_{l_q}^{m_q} \leq \left\lfloor \frac{x_{11} \dots x_{1m_1}}{x_{q1} \dots x_{qm_q}} \right\rfloor^{(r)} \leq x_{1l_1} \vee \dots \vee x_{ql_q}, \quad (4.11)$$

где

$$d_p = \left\lfloor \frac{(q+r-1)m_p}{\sum_{i=1}^q m_i} \right\rfloor, \quad l_p = \left\lfloor \frac{rm_p}{\sum_{i=1}^q m_i} \right\rfloor, \quad p = 1, \dots, q.$$

Данные оценки позволяют получать приближенные выражения ЛО со сложностью, пропорциональной размерам ЛО, что делает возможным вычисление ЛО практически неограниченных размеров.

Пример 5. Оценим ЛО X_4^4 из примера 4. Так, $d_1 = d_2 = [7 \cdot 2/6] = 2$, $d_3 = d_4 = [7 \cdot 1/6] = 1$, $l_1 = l_2 = [4 \cdot 2/6] = 2$, $l_3 = l_4 = [4 \cdot 1/6] = 1$, и искомые оценки имеют вид:

$$x_{12}x_{22}x_{31}x_{41} \leq X_4^4 \leq x_{12} \vee x_{22} \vee x_{31} \vee x_{41}.$$

Сложность их совместного вычисления — в наличии шести операций, а точность зависит от численных значений x_{ij} . Например, если $x_{12} = x_{22} = 10$, $x_{31} = x_{41} = 11$, то имеем оценки $10 \leq X_4^4 \leq 11$, погрешность которых 10 %.

Заключение

Настоящая работа содержит математическую часть выполненных исследований по теории надежности систем. В ней описан математический аппарат создаваемой автором логической теории надежности сложных систем, так называемые логические определители. Изложена также общая методика и конкретные методы использования указанного аппарата для вычисления характеристик надежности сложных систем, сопровождаемые содержательными примерами. Далее на основе разработанного математического аппарата логической теории надежности решены задачи построения аналитических формул для вычисления характеристик надежности нескольких классов сложных систем произвольно высокой размерности.

Данная работа продолжает цикл исследований автора, посвященный математическому аппарату логической теории надежности. От опубликованных ранее работ автора [6—13] статья существенно отличается использованием

нового оригинального математического аппарата логических определителей. Это позволило принципиально упростить как вычисление характеристик надежности сложных систем, так и анализ их надежностного поведения, причем эти операции оказались применимы в равной степени как к низкоразмерным, так и к высококоразмерным системам.

Список литературы

1. **Lloyd D. K., Lipov M.** Reliability: Management, Methods and mathematics. Englewood Cliffs, New Jersey, Prentice-Hall, 1962. 528 p.
2. **Barlow R. E., Proshan F.** Mathematical Theory of Reliability. N.-Y.: John Wiley and Sons, 1965.
3. **Reinschke K.** Modelle Zur Zuverlissigkeits- und Empfindlichkeitsanalyse von Systemen. Berlin, Technik, 1979 (in German).
4. **Левин В. П.** Теория надежности радиотехнических систем. М.: Советское радио, 1978. 264 с.
5. **Левин В. И.** Динамика конечных автоматов и надежность сложных систем // Автоматика и вычислительная техника. 1976. № 6. С. 17—24.
6. **Левин В. И.** Динамические автоматы и надежность технических систем. I // Электронное моделирование. 1980. № 4. С. 12—17.
7. **Левин В. И.** Динамические автоматы и надежность технических систем. II // Электронное моделирование. 1980. № 5. С. 63—72.
8. **Левин В. И.** Логические методы в теории надежности. I // Вестник Тамбовского государственного технического университета. 2009. Т. 15, № 4. С. 873—884.
9. **Левин В. И.** Логические методы в теории надежности. II // Вестник Тамбовского государственного технического университета. 2010. Т. 16, № 1. С. 119—132.
10. **Левин В. И.** Логические методы расчета надежности систем. I // Системы управления, связи и безопасности. 2017. № 2. С. 182—195.
11. **Левин В. И.** Логические методы расчета надежности систем. II // Системы управления, связи и безопасности. 2017. № 3. С. 84—97.
12. **Левин В. И.** Логические методы в теории надежности сложных систем. I // Вестник Тамбовского университета. 2011. Т. 16, № 5. С. 15—28.
13. **Левин В. И.** Логические методы в теории надежности сложных систем. II // Вестник Тамбовского университета. 2011. Т. 16, № 6. С. 25—39.
14. **Безопасность и надежность технических систем.** М.: Университетская книга, Логос, 2008. 376 с.
15. **Викторова В. С., Степаняц А. С.** Модели и методы расчета надежности технических систем. М.: Ленанд, 2014. 256 с.
16. **Хан Дж. Дж., Догановский Н., Мир Уи. К.** Анализ надежности с учетом видов отказов // Методы менеджмента качества. 2013. № 6.
17. **Бусленко Н. П.** Математическое моделирование производственных процессов на ЦВМ. М.: Наука, 2010.
18. **Гнеденко Б. В., Беляев Ю. К., Соловьев А. Д.** Математические методы в теории надежности. М.: Наука, 2010. 524 с.
19. **Капур К., Ламберсен Л.** Надежность и проектирование систем. М.: Мир, 2010. 504 с.
20. **Максимов Я. А.** Технология моделирования надежности информационных систем // В мире научных открытий. 2009. № 1.
21. **Половко А. М., Гуров С. В.** Основы теории надежности. СПб.: БХВ-Петербург, 2006. 702 с.
22. **Рябинин И. А.** Надежность и безопасность структурно-сложных систем. СПб.: Изд-во Санкт-Петербургского ун-та, 2007.
23. **Острейковский В. А.** Теория надежности. Учеб. для вузов. М.: Высшая школа, 2008.
24. **O'Connor P., Kleyner A.** Practical Reliability Engineering. N.-Y.: Wiley, 2012.

Continuous Logic and Analysis of Reliability of Complex Systems. Mathematical Means

In recent years the increasing attention of scientists and designers of technical systems has been acquiring the issues of improving methods for assessing the reliability and safety of these systems, in connection with tasks of increasing the values of these characteristics. The purpose of the article is to develop an automata-logical model of reliability of complex technical systems and corresponding logical methods for evaluating the reliability of such systems, which, unlike known ones, use not the traditional probabilistic reliability indicators, but deterministic logical indicators. In order to achieve this goal, the article suggests using the observed moments of successive failures and recovery of the elements of the technical system as initial data, and as the reliability characteristics of the system itself the moments of successive failures and recovery of this system. In this case, the problem of estimating the reliability of a system is reduced to constructing its mathematical model in the form of automata logical functions expressing the moments of its successive failures and reconstructions through analogous moments of all its elements. This article is the first part of the work in which an automata-logical model designed to calculate the logical function of reliability of complex technical systems is developed in detail. The novelty of the work is the construction of an adequate logical model of the reliability of complex system, which makes it possible to reduce the estimation of reliability of a complex technical system to the calculation of its logical reliability functions. In the process of calculation, the mathematical apparatus of logical determinants is used for the first time, which allows us to solve the complexity problem. In the article the logical model of reliability and methods of its investigation are developed in detail, allowing to introduce new indicators of reliability of complex technical systems that do not require for their evaluation the use of probabilistic methods and initial statistical data on element failures. On the basis of the developed logical model of reliability and methods of its investigation, the problem of constructing an automata system for reliability of systems is solved, which will allow to fulfill practical calculations of complex technical systems by methods of the theory of dynamic automata using the apparatus of logical determinants.

Keywords: complex system, switching process, reliability process, dynamical automaton, binary operator, structure of operator, logical theory of reliability

DOI: 10.17587/it.25.195-204

References

1. Lloyd D. K., Lipov M. Reliability: Management, Methods and Mathematics, Englewood Cliffs, New Jersey, Prentice-Hall, 1962, 528 p.
2. Barlow R. E., Proshan F. Mathematical Theory of Reliability, N.-Y., John Wiley and Sons, 1965.
3. Reinschke K. Modelle Zur Zuverlässigkeits- und Empfindlichkeitsanalyse von Systemen, Berlin, Technik, 1979 (in German).
4. Levin B. R. Teoriya nadezhnosti radiotekhnicheskikh sistem (Theory of reliability of radio systems), Moscow, Sovetskoe radio, 1978, 264 p. (in Russian).
5. Levin V. I. Dinamika konechnykh avtomatov i nadezhnost' slozhnykh sistem (Dynamics of finite automata and reliability of complex systems), Avtomatika i Vychislitel'naya Tehnika, 1976, no. 6, pp. 17–24 (in Russian).
6. Levin V. I. Dinamicheskie avtomaty i nadezhnost' tekhnicheskikh sistem. I (Dynamic automata and reliability of technical systems. I), Elektronnoe Modelirovanie, 1980, no. 4, pp. 12–17 (in Russian).
7. Levin V. I. Dinamicheskie avtomaty i nadezhnost' tekhnicheskikh sistem. II (Dynamic automata and reliability of technical systems. II), Elektronnoe Modelirovanie, 1980, no. 5, pp. 63–72 (in Russian).
8. Levin V. I. Logicheskie metody v teorii nadezhnosti. I (Logical methods in the theory of reliability. I), Vestnik Tambovskogo Gosudarstvennogo Tekhnicheskogo Universiteta, 2009, vol. 15, no. 4, pp. 873–884 (in Russian).
9. Levin V. I. Logicheskie metody v teorii nadezhnosti. II (Logical methods in the theory of reliability. II), Vestnik Tambovskogo Gosudarstvennogo Tekhnicheskogo Universiteta, 2010, vol. 16, no. 1, pp. 119–132 (in Russian).
10. Levin V. I. Logicheskie metody rascheta nadezhnosti sistem. I (Logical methods for calculating the reliability of systems. I), Sistemy Upravleniya, Svyazi i Bezopasnosti, 2017, no. 2, pp. 182–195 (in Russian).
11. Levin V. I. Logicheskie metody rascheta nadezhnosti sistem. II (Logical methods for calculating the reliability of systems. II), Sistemy Upravleniya, Svyazi i Bezopasnosti, 2017, no. 3, pp. 84–97 (in Russian).
12. Levin V. I. Logicheskie metody v teorii nadezhnosti slozhnykh sistem. I (Logical methods in the theory of reliability of complex systems. I), Vestnik Tambovskogo Universiteta, 2011, vol. 16, no. 5, pp. 15–28 (in Russian).
13. Levin V. I. Logicheskie metody v teorii nadezhnosti slozhnykh sistem. II (Logical methods in the theory of reliability of complex systems. II), Vestnik Tambovskogo Universiteta, 2011, vol. 16, no. 6, pp. 25–39 (in Russian).
14. Bezopasnost' i nadezhnost' tekhnicheskikh sistem (Safety and reliability of technical systems), Moscow, Universitetska kniga, Logos, 2008, 376 p. (in Russian).
15. Viktorova V. S., Stepanânc A. S. Modeli i metody rascheta nadezhnosti tekhnicheskikh sistem (Models and methods for calculating the reliability of technical systems), Moscow, Lenand, 2014, 256 p. (in Russian).
16. Han Dž. Dž., Doganovskij N., Mir Ui. K. Analiz nadezhnosti s uchetom vidov otkazov (Reliability analysis based on failure modes), Metody Menedzhmenta Kachestva, 2013, no. 6. (in Russian).
17. Buslenko N. P. Matematicheskoe modelirovanie proizvodstvennykh processov na CVM (Mathematical modeling of production processes on digital computers), Moscow, Nauka, 2010. (in Russian).
18. Gnedenko B. V., Belâev Ū. K., Solov'ev A. D. Matematicheskie metody v teorii nadezhnosti (Mathematical methods in the theory of reliability), Moscow, Nauka, 2010, 524 p. (in Russian).
19. Kapur K., Lambersen L. Nadezhnost' i proektirovanie sistem (Reliability and system design), Moscow, Mir, 2010, 504 p. (in Russian).
20. Maksimov A. A. Tehnologiya modelirovaniya nadezhnosti informacionnykh sistem (Information systems reliability modeling technology), V Mire Nauchnykh Otkrytij, 2009, no. 1. (in Russian).
21. Polovko A. M., Gurov S. V. Osnovy teorii nadezhnosti (Fundamentals of the theory of reliability), SPb., BHV-Peterburg, 2006, 702 p. (in Russian).
22. Râbinin I. A. Nadezhnost' i bezopasnost' strukturno-slozhnykh sistem (Reliability and safety of structurally complex systems), SPb., Publishing house of Sankt-Peterburgskii un-t, 2007. (in Russian).
23. Ostrejkovskij V. A. Teoriya nadezhnosti (Theory of Reliability), Moscow, Vysshâ shkola, 2008. (in Russian).
24. O'Connor P., Kleynner A. Practical Reliability Engineering, N.-Y., Wiley, 2012.

В. Н. Тарасов, д-р техн. наук, проф., зав. каф., e-mail: veniamin_tarasov@mail.ru,

Н. Ф. Бахарева, д-р техн. наук, проф., зав. каф., e-mail: nadin1956_04@inbox.ru,

Када Отхмане, аспирант,

Поволжский государственный университет телекоммуникаций и информатики, г. Самара

Математическая модель телетрафика на основе системы $HE_2/M/1$

Статья посвящена исследованию системы массового обслуживания (СМО) $HE_2/M/1$ типа $G/M/1$ с гиперэрланговским входным распределением второго порядка и экспоненциальным законом времени обслуживания в целях получения решения для среднего времени ожидания требований в очереди в случае стационарного режима. Для этого использован классический метод спектрального разложения решения интегрального уравнения Линдли. Для практического применения полученных результатов использован метод моментов. Оказывается, что гиперэрланговский закон распределения HE_2 , как и гиперэкспоненциальный H_2 , являющийся трехпараметрическим, может определяться как двумя первыми моментами, так и тремя первыми моментами. Эту характерную особенность гиперэрланговского закона распределения необходимо использовать в теории массового обслуживания. Выбор такого закона распределения вероятностей обусловлен тем, что он является наиболее общим распределением неотрицательных непрерывных случайных величин, поскольку его коэффициент вариации $c_\tau \geq 1/\sqrt{2}$ и охватывает более широкий диапазон, чем гиперэкспоненциальное распределение, для которого $c_\tau \geq 1$. Определение главной характеристики СМО — среднего времени ожидания — является актуальной задачей в теории телетрафика при анализе трафика, чувствительного к задержкам. Метод спектрального разложения решения интегрального уравнения Линдли для СМО $HE_2/M/1$ позволяет получить решение в замкнутой форме.

Ключевые слова: гиперэрланговский закон распределения, интегральное уравнение Линдли, метод спектрального разложения, преобразование Лапласа

Введение

Настоящая статья посвящена анализу систем массового обслуживания (СМО) $HE_2/M/1$ типа $G/M/1$ с произвольными законами распределений входного потока требований в систему, для которых в общем случае не может быть найдено решение для главной характеристики — среднего времени ожидания требований в очереди. Поэтому системы типа $G/M/1$ могут быть исследованы только при конкретных законах распределений входного потока [2–5].

В теории телетрафика по среднему времени ожидания, например, оценивают задержки пакетов в сетях пакетной коммутации при их моделировании с помощью СМО.

В исследовании систем $G/G/1$ и $G/M/1$ важную роль играет метод спектрального разложения решения интегрального уравнения Линдли, и большинство результатов в теории массового обслуживания получены именно с помощью данного метода. Обозначим

- $W(y)$ — функция распределения вероятностей (ФРВ) времени ожидания требования в очереди,
- $C(u) = P(\tilde{u} < u)$ — ФРВ случайной величины $\tilde{u} = \tilde{x} - \tilde{t}$, где, в свою очередь, \tilde{x} — случайное время обслуживания требования, \tilde{t} — случайная величина — интервал времени между поступлениями требований.

Тогда одна из форм интегрального уравнения Линдли [2–5] имеет вид

$$W(y) = \begin{cases} \int_{-\infty}^y W(y-u)dC(u), & y \geq 0; \\ 0, & y < 0. \end{cases}$$

В научной литературе нет данных по рассматриваемой системе, и, видимо, это связано с достаточной сложностью гиперэрланговского закона распределения. К тому же гиперэрланговский закон распределения порядка R является наиболее общим распределением неотрицательных непрерывных случайных величин, поскольку его коэффициент вариации $c_\tau > 0$ [10, 11].

Постановка задачи

Ставится задача определения среднего времени ожидания в системе $HE_2/M/1$ на основе классического метода спектрального разложения решения интегрального уравнения Линдли (ИУЛ) для данной системы в замкнутой форме.

Решение задачи. Для системы $HE_2/M/1$ законы распределения интервалов входного потока и времени обслуживания задаются функциями плотности вида

$$a(t) = 4p\lambda_1^2 te^{-2\lambda_1 t} + 4(1-p)\lambda_2^2 te^{-2\lambda_2 t}; \quad (1)$$

$$b(t) = \mu e^{-\mu t}. \quad (2)$$

Использование классического метода спектрального разложения решения ИУЛ, как это показано в работах [1, 6, 7], позволит определить не только среднее время ожидания, но и моменты высших порядков времени ожидания.

Согласно методу спектрального разложения нам для нахождения закона распределения времени ожидания необходимо найти следующее спектральное разложение: $A^*(-s)B^*(s) - 1 = \psi_+(s)/\psi_-(s)$, где $\psi_+(s)$ и $\psi_-(s)$ — некоторые рациональные функции от s , а $A^*(s)$, $B^*(s)$ — преобразования Лапласа функций плотности (1) и (2) соответственно.

Преобразования Лапласа функций (1) и (2) имеют следующий вид:

$$A^*(s) = p \left(\frac{2\lambda_1}{s + 2\lambda_1} \right)^2 + (1-p) \left(\frac{2\lambda_2}{s + 2\lambda_2} \right)^2;$$

$$B^*(s) = \frac{\mu}{s + \mu}.$$

Тогда спектральное разложение решения ИУЛ для системы HE₂/M/1 $A^*(-s)B^*(s) - 1 = \psi_+(s)/\psi_-(s)$ примет вид

$$\frac{\psi_+(s)}{\psi_-(s)} = \left[p \left(\frac{2\lambda_1}{2\lambda_1 - s} \right)^2 + (1-p) \left(\frac{2\lambda_2}{2\lambda_2 - s} \right)^2 \right] \left(\frac{\mu}{\mu + s} \right) - 1.$$

Выражение, стоящее в квадратных скобках, представим в виде

$$\begin{aligned} & \left[p \left(\frac{2\lambda_1}{2\lambda_1 - s} \right)^2 + (1-p) \left(\frac{2\lambda_2}{2\lambda_2 - s} \right)^2 \right] = \\ & = \frac{p(16\lambda_1^2\lambda_2^2 - 16\lambda_1^2\lambda_2s + 4\lambda_1^2s^2)}{(2\lambda_1 - s)^2(2\lambda_2 - s)^2} + \\ & + \frac{(1-p)(16\lambda_1^2\lambda_2^2 - 16\lambda_1\lambda_2^2s + 4\lambda_2^2s^2)}{(2\lambda_1 - s)^2(2\lambda_2 - s)^2} = \\ & = \frac{a_0 - a_1s + a_2s^2}{(2\lambda_1 - s)^2(2\lambda_2 - s)^2}, \end{aligned}$$

где промежуточные параметры, введенные для сокращения записи, равны $a_0 = 16\lambda_1^2\lambda_2^2$, $a_1 = 16\lambda_1\lambda_2[p\lambda_1 + (1-p)\lambda_2]$, $a_2 = 4[p\lambda_1^2 + (1-p)\lambda_2^2]$.

Продолжая разложение, получим:

$$\begin{aligned} & \frac{\psi_+(s)}{\psi_-(s)} = \\ & = \frac{\mu(a_0 - a_1s + a_2s^2) - (2\lambda_1 - s)^2(2\lambda_2 - s)^2(\mu + s)}{(2\lambda_1 - s)^2(2\lambda_2 - s)^2(\mu + s)} = \end{aligned}$$

$$\begin{aligned} & = \frac{-s(s^4 + c_3s^3 + c_2s^2 + c_1s + c_0)}{(2\lambda_1 - s)^2(2\lambda_2 - s)^2(\mu + s)} = \\ & = \frac{-s(s + s_1)(s - s_2)(s - s_3)(s - s_4)}{(2\lambda_1 - s)^2(2\lambda_2 - s)^2(\mu + s)}. \end{aligned}$$

Окончательно спектральное разложение решения ИУЛ для системы HE₂/M/1 имеет следующий вид:

$$\frac{\psi_+(s)}{\psi_-(s)} = \frac{-s(s + s_1)(s - s_2)(s - s_3)(s - s_4)}{(2\lambda_1 - s)^2(2\lambda_2 - s)^2(\mu + s)}. \quad (3)$$

Многочлен четвертой степени в числителе разложения

$$s^4 + c_3s^3 + c_2s^2 + c_1s + c_0 \quad (4)$$

с коэффициентами

$$\begin{aligned} c_0 &= a_1\mu + 16\lambda_1\lambda_2[\lambda_1\lambda_2 - \mu(\lambda_1 + \lambda_2)], \\ c_1 &= 4\mu(\lambda_1^2 + 4\lambda_1\lambda_2 + \lambda_2^2) - 16\lambda_1\lambda_2(\lambda_1 + \lambda_2) - a_2\mu, \\ c_2 &= 4(\lambda_1^2 + \lambda_2^2) + 16\lambda_1\lambda_2 - 4\mu(\lambda_1 + \lambda_2), \\ c_3 &= \mu - 4(\lambda_1 + \lambda_2) \end{aligned}$$

имеет один действительный отрицательный корень и три положительных корня (либо вместо последних один действительный положительный и два комплексно сопряженных с положительной вещественной частью). Исследование знака младшего коэффициента c_0 показывает, что $c_0 < 0$ всегда в случае стабильной системы, когда $0 < \rho < 1$. Это также подтверждает предположение о наличии таких корней многочлена.

Согласно методу спектрального разложения функции $\psi_+(s)$ и $\psi_-(s)$ должны удовлетворять следующим условиям [1]:

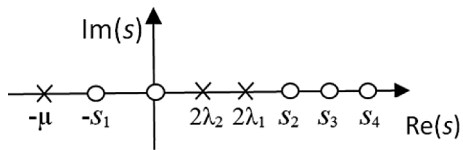
- 1) для $\text{Re}(s) > 0$ функция $\psi_+(s)$ является аналитической без нулей в этой полуплоскости;
- 2) для $\text{Re}(s) < D$ функция $\psi_-(s)$ является аналитической без нулей в этой полуплоскости, где D — некоторая положительная константа, определяемая из условия:

$$\lim_{t \rightarrow \infty} \frac{a(t)}{e^{-Dt}} < \infty. \quad (5)$$

Кроме того, функции $\psi_+(s)$ и $\psi_-(s)$ должны обладать следующими свойствами:

$$\lim_{|s| \rightarrow \infty, \text{Re}(s) > 0} \frac{\psi_+(s)}{s} = 1; \quad \lim_{|s| \rightarrow \infty, \text{Re}(s) < D} \frac{\psi_-(s)}{s} = -1. \quad (6)$$

Теперь с учетом условий (5) и (6) строим рациональные функции $\psi_+(s)$ и $\psi_-(s)$ для рассматриваемого случая:



Нули и полюсы функции $\psi_+(s)/\psi_-(s)$ для системы HE₂/M/1

- $\psi_+(s) = s(s + s_1)/(\mu + s)$, так как нули многочлена (4) $s = 0, s = -s_1$ и полюс $s = -\mu$ лежат в области $\text{Re}(s) \leq 0$;

- $\psi_-(s) = -\frac{(2\lambda_1 - s)^2 (2\lambda_2 - s)^2}{(s - s_2)(s - s_3)(s - s_4)}$, так как ее нули и полюсы лежат в области $\text{Re}(s) < D$, определенной условием (5). Выполнение условий (5) и (6) спектрального разложения для построенных функций $\psi_+(s)$ и $\psi_-(s)$ также подтверждается рисунком.

При построении этих функций удобнее нули и полюса отношения $\psi_+(s)/\psi_-(s)$ отметить на комплексной s -плоскости для исключения ошибок построения функций $\psi_+(s)$ и $\psi_-(s)$. На рисунке полюсы отмечены крестиками, а нули — кружками.

Далее по методике спектрального разложения найдем константу K :

$$K = \lim_{s \rightarrow 0} \frac{\psi_+(s)}{s} = \lim_{s \rightarrow 0} \frac{(s + s_1)}{(s + \mu)} = \frac{s_1}{\mu},$$

где s_1 — абсолютное значение отрицательного корня $-s_1$. Постоянная K определяет вероятность того, что поступающее в систему требование застанет ее свободной.

Для нахождения преобразования Лапласа функции плотности времени ожидания построим функцию

$$\Phi_+(s) = \frac{K}{\psi_+(s)} = \frac{s_1(s + \mu)}{s\mu(s + s_1)}.$$

Отсюда преобразование Лапласа функции плотности времени ожидания $W^*(s) = s\Phi_+(s)$ будет равно

$$W^*(s) = \frac{s_1(s + \mu)}{\mu(s + s_1)}. \quad (7)$$

Для нахождения среднего времени ожидания найдем производную от функции (7) $W^*(s)$ со знаком минус в точке $s = 0$:

$$\bar{W} = -\left. \frac{dW^*(s)}{ds} \right|_{s=0} = \frac{-\mu s_1^2 + \mu^2 s_1}{\mu^2 s_1^2} = \frac{1}{s_1} - \frac{1}{\mu}.$$

Окончательно среднее время ожидания для системы HE₂/M/1 составляет

$$\bar{W} = 1/s_1 - 1/\mu. \quad (8)$$

Из выражения (7) также можно определить дисперсию времени ожидания. Вторая производная от преобразования (7) в точке $s = 0$ дает второй начальный момент времени ожидания, что позволяет определить дисперсию времени ожидания. Учитывая определение джиттера в телекоммуникациях как разброс времени ожидания от среднего значения [8], тем самым получим возможность его определения через дисперсию. Это является важным результатом для анализа трафика, чувствительного к задержкам.

Для практического применения выражения (8) необходимо определить числовые характеристики распределения (1) HE₂. Для распределения (2) эти характеристики известны из курса теории вероятностей. Для этого воспользуемся свойством преобразования Лапласа воспроизведения моментов и запишем начальные моменты до второго порядка для распределения (1):

$$\bar{\tau}_\lambda = \frac{p}{\lambda_1} + \frac{(1-p)}{\lambda_2}; \quad (9)$$

$$\bar{\tau}_\lambda^2 = \frac{3}{2} \left[\frac{p}{\lambda_1^2} + \frac{(1-p)}{\lambda_2^2} \right]. \quad (10)$$

Аппроксимация закона распределения HE₂ с использованием двух первых моментов

Рассматривая равенства (9) и (10) как запись метода моментов, найдем неизвестные параметры распределения (1) λ_1, λ_2, p . Система двух уравнений (9), (10) при этом является недоопределенной, поэтому к ней добавим выражение для квадрата коэффициента вариации

$$c_\lambda^2 = \frac{\bar{\tau}_\lambda^2 - (\bar{\tau}_\lambda)^2}{(\bar{\tau}_\lambda)^2} \quad (11)$$

как связующее условие между (9) и (10). Кроме того, коэффициент вариации будем использовать в расчетах в качестве входного параметра системы. Исходя из вида уравнения (9) положим

$$\lambda_1 = 2p/\bar{\tau}_\lambda, \lambda_2 = 2(1-p)/\bar{\tau}_\lambda \quad (12)$$

и потребуем выполнения условия (11). Подставив выражения (9), (10) и частное решение (12) в (11) и решив полученное уравнение четвертой степени $p(1-p)[8(1+c_\lambda^2)p^2 - 8(1+c_\lambda^2)p + 3] = 0$ относительно параметра p , выберем одно нужное значение с учетом условия $0 < p < 1$:

$$p = \frac{1}{2} + \sqrt{\frac{2(1+c_\lambda^2) - 3}{8(1+c_\lambda^2)}}. \quad (13)$$

Отсюда следует, что коэффициент вариации интервалов входного потока $c_\lambda \geq 1/\sqrt{2}$. Подставив значение параметра p из (13) в соотношение (12), определим значения параметров распределения (1) λ_1, λ_2 . Таким образом, получено частное решение недоопределенной системы уравнений (9) и (10) методом подбора.

Аппроксимация закона распределения HE₂ с использованием трех первых моментов

Учитывая тот факт, что распределение HE₂ является трехпараметрическим, аппроксимацию можно выполнить и на уровне трех первых моментов. Для этого запишем выражения для начального момента 3-го порядка, полученное через преобразование Лапласа $A^*(s)$:

$$\overline{\tau_\lambda^3} = \frac{3p}{\lambda_1^3} + \frac{3(1-p)}{\lambda_2^3}. \quad (14)$$

Присоединив уравнение (14) к уравнениям моментов (9), (10) и решив систему трех нелинейных уравнений с тремя неизвестными при заданных значениях начальных моментов до третьего порядка включительно, в пакете MathCAD находим все три параметра λ_1, λ_2, p . Как показано в работе [1] на примере гиперэкспоненциальных входных распределений, аппроксимация с использованием двух первых моментов в отличие от трех моментов может занижать среднее время ожидания до 10 % в зависимости от значений загрузки и третьего момента.

Такой же подход к аппроксимации законов распределения гиперэкспоненциальным распределением применен в работах автора [1, 6, 7]. Таким образом, гиперэрланговский закон распределения второго порядка может определяться полностью двумя первыми моментами и перекрывать весь диапазон изменения коэффициента вариации от $1/\sqrt{2}$ до ∞ , что шире, чем у гиперэкспоненциального распределения $(1, \infty)$.

Величины $\overline{\tau_\lambda}, \overline{\tau_\mu}, c_\lambda, c_\mu$ будем считать входными параметрами для расчета среднего времени ожидания для системы HE₂/M/1. Тогда алгоритм расчета сведется к последовательному определению параметров распределения (1) из выражений (13), (12) и к нахождению нужного корня многочлена (4), а затем к использованию расчетного выражения (8).

Результаты экспериментов

В таблице приведены данные расчетов в пакете MathCAD для системы HE₂/M/1 для слу-

Результаты экспериментов для СМО HE₂/M/1 в сравнении с H₂/M/1

Входные параметры		Среднее время ожидания	
ρ	c_λ	для системы HE ₂ /M/1	для системы H ₂ /M/1
0,1	0,71	0,03	—
	2	0,08	0,19
	4	0,10	0,23
	8	0,11	0,25
0,5	0,71	0,62	—
	2	2,00	2,16
	4	4,62	4,83
	8	10,15	10,40
0,9	0,71	6,61	—
	2	22,59	22,41
	4	77,28	75,79
	8	295,96	289,13

чаев малой, средней и высокой нагрузки $\rho = 0,1; 0,5; 0,9$. Для сравнения в правой колонке приведены данные для системы H₂/M/1, для которой $c_\lambda \geq 1$ [1].

Значения среднего времени ожидания в системах HE₂/M/1 и H₂/M/1 достаточно близки при средней и высокой нагрузках систем, хотя начальные моменты распределений (начиная со второго) HE₂ и H₂ разнятся. Заметим, что система H₂/M/1 справедлива только при $c_\lambda \geq 1$ и $c_\mu = 1$, потому в таблице для $c_\lambda < 1$ стоят прочерки. Коэффициент загрузки ρ определяется отношением средних интервалов обслуживания и поступления требований $\rho = \overline{\tau_\mu}/\overline{\tau_\lambda}$. Расчеты, приведенные в таблице, выполнены для нормированного времени обслуживания $\overline{\tau_\mu} = 1$. Полученные расчетные данные хорошо согласуются с результатами работы [12].

Заключение

В работе получено спектральное разложение решения интегрального уравнения Линдли для системы HE₂/M/1, с помощью которого выведено расчетное выражение для среднего времени ожидания в очереди для этой системы в замкнутой форме. Результаты расчетов сравниваются с результатами аналогичной системы H₂/M/1 с гиперэкспоненциальным распределением 2-го порядка для входного потока.

Полученное расчетное выражение для среднего времени ожидания расширяет и дополняет известные формулы теории массового обслуживания для среднего времени ожидания для систем типа G/M/1 и G/G/1 с произволь-

ными законами распределений входного потока и времени обслуживания. При этом диапазон изменения параметров у системы $HE_2/M/1$ шире, чем у системы $H_2/M/1$.

Полученный результат с успехом может быть применен в современной теории телетрафика, где задержки пакетов входящего трафика играют первостепенную роль. Для этого необходимо знать числовые характеристики интервалов входящего трафика и времени обслуживания на уровне двух первых моментов, что не вызывает трудностей при использовании современных анализаторов трафика [6, 7].

Список литературы

1. **Тарасов В. Н.** Исследование систем массового обслуживания с гиперэкспоненциальными входными распределениями // Проблемы передачи информации. 2016. № 1. С. 16–26.
2. **Клейнрок Л.** Теория массового обслуживания. Пер. с англ. М.: Машиностроение, 1979. 432 с.
3. **Brannstrom N.** A Queueing Theory analysis of wireless radio systems — Applied to HS-DSCH. Lulea university of technology, 2004. 79 p.

4. **Whitt W.** Approximating a point process by a renewal process: two basic methods // Operation Research. 1982. N. 1. P. 125–147.

5. **Бочаров П. П., Печинкин А. В.** Теория массового обслуживания. М.: Изд-во РУДН, 1995. 529 с.

6. **Тарасов В. Н., Бахарева Н. Ф., Горелов Г. А., Малыхов С. В.** Анализ входящего трафика на уровне трех моментов распределений временных интервалов // Информационные технологии. 2014. № 9. С. 54–59.

7. **Тарасов В. Н.** Математическая модель трафика с тяжелохвостным распределением на основе системы массового обслуживания $H_2/M/1$ // Инфокоммуникационные технологии. 2014. № 3. С. 36–41.

8. **URL:** <https://tools.ietf.org/html/rfc3393>. RFC 3393 IP Packet Delay Variation Metric for IP Performance Metrics (IPPM) (дата обращения: 26.02.2016).

9. **Myskja A.** An improved heuristic approximation for the GI/GI/1 queue with bursty arrivals // Teletraffic and datatraffic in a Period of Change, ITC-13. Elsevier Science Publishers. 1991. P. 683–688.

10. **Алиев Т. И.** Основы моделирования дискретных систем. СПб.: СПбГУ ИТМО, 2009. 363 с.

11. **Алиев Т. И.** Аппроксимация вероятностных распределений в моделях массового обслуживания // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 2 (84). С. 88–93.

12. **Тарасов В. Н., Бахарева Н. Ф.** Обобщенная двумерная диффузионная модель массового обслуживания типа GI/G/1 // Телекоммуникации. 2009. № 7. С. 2–8.

V. N. Tarasov, D. Sc., Professor, Head of Chair, e-mail: veniamin_tarasov@mal.ru,

N. F. Bakhareva, D. Sc., Professor, Head of Chair, e-mail: nadin1956_04@inbox.ru,

Kada Othmane, Postgraduate,

Povolzhsky State University of Telecommunications and Informatics, Samara, 443010, Russian Federation

The Mathematical Model of Teletraffic Based on the $HE_2/M/1$ System

The article is devoted to the study of the $HE_2/M/1$ queueing system type $G/M/1$ with a second-order hypererlangian input distribution and an exponential service time law with the aim of obtaining a solution for the average waiting time in queue in the case of a stationary mode. For this, the classical method of spectral decomposition of the solution of the Lindley integral equation is used. For practical application of the obtained results, the method of moments is used. It turns out that the hypererlangian distribution law HE_2 , like hyperexponential H_2 , which is three-parameter, can be determined by both the first two moments and the first three moments. This characteristic feature of the hypererlangian distribution law must be used in the theory of mass service. The choice of such a law of probability distribution is due to the fact that it is the most common distribution of non-negative continuous random variables, since its coefficient of variation $c_v > 1/\sqrt{2}$ covers a wider range than the hyperexponential distribution for which $c_v \geq 1$. Determination of the main characteristic of QS — the average waiting time is an important task in the theory of teletraffic when analyzing traffic sensitive to delays. The method of spectral decomposition of the solution of the Lindley integral equation for the $QS HE_2/M/1$ allows one to obtain a solution in closed form.

Keywords: Hypererlangian distribution law, Lindley integral equation, spectral decomposition method, Laplace transform

DOI: 10.17587/it.25.205-210

References

1. **Tarasov V. N.** *Issledovanie sistem massovogo obsluzhivaniya s gipereksponencial'nymi vhodnymi raspredeleniyami* [Analysis of queues with hyperexponential arrival distributions], *Problemy Peredachi Informacii*, 2016, no. 1, pp. 16–26 (in Russian).
2. **Kleinrock L.** *Teoriya massovogo obsluzhivaniya*, Moscow, Mashinostroenie, 1979, 432 p. (in Russian).
3. **Brannstrom N.** *A Queueing Theory analysis of wireless radio systems — Applied to HS-DSCH*, Lulea university of technology, 2004, 79 p.

4. **Whitt W.** Approximating a point process by a renewal process, I: two basic methods, *Operation Research*, 1982, no. 1, pp. 125–147.

5. **Bocharov P. P., Pechinkin A. V.** *Teoriya massovogo obsluzhivaniya*, Moscow, Publishing house of RUDN, 1995, 529 p. (in Russian).

6. **Tarasov V. N., Bakhareva N. F., Gorelov G. A., Malahov S. V.** *Analiz vkhodyashchego trafika na urovne trekh momentov raspredelenij vremennykh intervalov* [Analyzing the Incoming Traffic at the Three Moments Distribution of Time Intervals], *Informacionnye Tekhnologii*, 2014, no. 9, pp. 54–59 (in Russian).

7. Tarasov V. N., Bahareva N. F., Gorelov G. A. *Matematicheskaya model trafika s tyazhelohlostnym raspredeleniem na osnove sistemy massovogo obsluzhivaniya $H_2/M/1$* [Mathematical model of traffic with heavy-tailed distribution based on the queuing system $H_2/M/1$], *Informacionnyye Tekhnologii*, 2014, no. 3 pp. 36–41 (in Russian).

8. Available at: <https://tools.ietf.org/html/rfc3393>. RFC 3393 IP Packet Delay Variation Metric for IP Performance Metrics (IPPM) (accessed: 26.02.2016).

9. Myskja A. An improved heuristic approximation for the GI/GI/1 queue with bursty arrivals, *Teletraffic and datatraffic in a Period of Change, ITC-13*, Elsevier Science Publishers, 1991, pp. 683–688.

10. Aliev T. I. *Osnovy modelirovaniya diskretnykh system* (Fundamentals of discrete systems modeling), SPb., Publishing house of SPbGU ITMO, 2009, 363 p. (in Russian).

11. Aliev T. I. *Approximatsiya veroyatnostnykh raspredelenij v modelyah massovogo obsluzhivaniya* [Approximation of probability distributions in queuing models], *Nauchno-Tekhnicheskij Vestnik Informacionnykh Tekhnologij, Mekhaniki i Optiki*, 2013, no. 2 (84), pp. 88–93 (in Russian).

12. Tarasov V. N., Bahareva N. F. *Obobshchennaya dvumernaya diffuzionnaya model' massovogo obsluzhivaniya tipa GI/G/1* [A generalized two-dimensional diffusion queuing model of the GI/G/1 type], *Telekommunikacii*, 2009, no. 7, pp. 2–8 (in Russian).

УДК 004:656.131

DOI: 10.17587/it.25.210-215

Г. Д. Антониади, аспирант, george@antoniadi.com, В. О. Архипов, аспирант, arvlol@yandex.ru,
А. А. Цуприков, доцент, tsouprikov@mail.ru,

Федеральное государственное бюджетное образовательное учреждение высшего образования
"Кубанский государственный технологический университет"

Математическая модель задержки автотранспорта на регулируемом перекрестке

Проведен сравнительный анализ моделей задержки Бэкманна и Вебстера, выявлены их недостатки для регулирования сигналов светофора. Рассмотрена новая модель задержки автотранспорта на регулируемом перекрестке, проанализирован физический смысл составляющих модели, определена значимость (вес, количественный вклад) параметров в значение задержки транспорта при запрещающем сигнале светофора с образованием очереди.

Ключевые слова: модель задержки проезда, светофор, интенсивность проезда, очередь, перекресток, значимость параметров

Введение

Для описания задержки автотранспорта перед перекрестком при светофорном регулировании движения разработано множество математических моделей, которые можно разделить на две группы:

- "точные" модели, описывающие текущую ситуацию на дороге в данный момент времени (модели М. Дж. Бэкманна, Дж. Н. Дарроча, В. Р. МакНейла и др.);
- "приближенные" модели, описывающие стационарные условия движения с аппроксимацией к реальной ситуации на дороге (модели А. Дж. Миллера, Дж. Ф. Ньюелла и др.).

Представляет интерес проведение сравнительного анализа основных моделей задержки транспорта, проверка их достоверности в реальных дорожных условиях проезда перекрестка и определение количественных значений факторов, влияющих на задержку.

Постановка задачи

Основной задачей исследования является анализ, рассмотрение составных частей и элементов базовых моделей задержки М. Дж. Бэкманна и Ф. В. Вебстера, проверка их достоверности, разработка новой модели времени задержки автотранспорта на регулируемом перекрестке, ее сравнительный анализ с базовыми моделями, а также определение значимости (вклада) каждого параметра в значение задержки.

Методы решения задачи исследования

Анализ основных моделей выполнен на основе вычисления с их помощью транспортных условий реального перекрестка в течение цикла работы светофора, достоверность проверялась по теории размерности. Количественный вклад параметров модели в значение задержки определен методом относительных погрешностей.

Прикладная интерпретация и иллюстрация результатов исследования

Впервые "точная" модель оценки задержки транспортных средств (ТС) при жестком светофорном регулировании была разработана М. Дж. Бэкманном [1]. В ней принят биномиальный характер прибытия ТС к перекрестку (автомобили прибывают к перекрестку с одинаковой вероятностью) и детерминированный характер регулирования сигналов светофора:

$$d = \frac{c-g}{c(1-q/s)} \left[\frac{q_0}{q} + \frac{c-g+1}{2} \right], \quad (1)$$

где d — средняя задержка одного ТС за цикл, с; c — длина цикла регулирования, с; g — эффективная длительность зеленого сигнала, с; q — интенсивность прибытия ТС, число автомобилей/с (далее — авт/с); s — интенсивность разъезда из очереди, число автомобилей/с; q_0 — остаточная очередь, число автомобилей (далее — авт.).

Модель отвечает теории размерности, так как левая и правая части функции (1) имеют одинаковые единицы измерения — [секунда]. Это означает, что физические процессы, описываемые моделью, достоверны.

Результаты моделирования для реальных условий при $c = 120$ с, $g = 0...120$ с, $q = 2$ авт/с, $s = 3$ авт/с, $q_0 = 1...11$ авт. выявили следующие недостатки модели:

1. Для работы модели интенсивность разъезда s всегда должна быть больше интенсивности прибытия q , иначе задержка d становится отрицательной. На практике число прибывающих автомобилей может быть больше разъезжающих во время зеленого сигнала.

2. Для модели не существенны интенсивности прибытия ТС к перекрестку и их разъезда, количественные значения q и s не влияют на d , они могут быть большими (например, 200 и 300) или маленькими (например, 2 и 3) — задержка d почти не меняется, главное — их отношение должно быть меньше 1.

3. Очередь q_0 очень слабо влияет на задержку, например, изменение q_0 с 10 до 10 000 машин, т. е. в 1000 раз, или на 100 000 %, приводит к изменению d с 367,46 с до 358,59 с, т. е. на 9 с, или на 0,025 = 2,5 %.

Наиболее широко распространенная "приближенная" модель задержки автотранспорта перед перекрестком Ф. В. Вебстера [2] основана на сочетании теоретического анализа и численного моделирования:

$$d = \frac{c(1-g/c)^2}{2[1-(g/c)x]} + \frac{x^2}{2q(1-x)} - 0,65 \left(\frac{c}{q^2} \right)^{1/3} x^{2+5\left(\frac{g}{c}\right)}, \quad (2)$$

где x — степень насыщения движения (уровень загрузки подхода), авт.

Модель (2) содержит три составляющих. Первая определяет задержку при условии стационарности прибытия транспортного потока, вторая — так называемая "случайная задержка" — представляет случайную составляющую процесса. Она предполагает пуассоновский процесс прибытия ТС и постоянную интенсивность их разъезда, соответствующую пропускной способности перекрестка. Третья составляющая корректирует время задержки и составляет около 2 % от суммы первых двух, поэтому ее можно принять за коэффициент, равный 2.

Модель в целом отвечает теории размерности, поскольку только корректирующая часть функции (2), которой можно пренебречь, имеет размерность [авт^{4/3}].

Моделирование при реальных параметрах работы перекрестка: $c = 120$ с, $g = 0...120$ с, $q = 3$ авт/с, $x = 0,1...0,999$ авт. выявило следующие недостатки модели:

1. Вторая и третья составляющие модели Вебстера практически не влияют на длительность задержки автотранспорта перед перекрестком, их доля составляет около 1...1,7 % от первой составляющей.

2. Варьирование q и x в широких пределах практически не изменяет значение d .

Общим недостатком как для точных, так и для приближенных моделей, в том числе рассмотренных здесь, является тот факт, что они не учитывают значимость очереди ТС для задержки как основного параметра регулирования — он либо отсутствует в модели, либо оказывает второстепенное влияние на управление светофором. Между тем очередь, которая накапливается перед перекрестком, является основной причиной необходимости управления светофором.

Для разработки новой модели задержки и ее применения в системе управления рассмотрим общие правила регулирования работы перекрестка.

Задержка ТС прямо пропорциональна интенсивности прибытия машин к светофору и обратно пропорциональна интенсивности их разъезда при зеленом сигнале.

Длительность разрешающего сигнала светофора должна определяться двумя правилами.

1. Если интенсивность прибытия ТС меньше или равна пропускной способности перекрестка, то время разрешающего сигнала также должно соответствовать интенсивности прибытия и поддерживаться до конца разъезда очереди. При этом пропускная способность должна соответствовать интенсивности разъезда очереди, что регулируется допустимой скоростью проезда через перекресток.

Согласно работе [3] пропускная способность отдельного перекрестка равна

$$P = Q \frac{V}{D} \rho \text{ [авт/ч]},$$

где V — разрешенная скорость движения на перекрестке, км/ч; D — соответствующий этой скорости интервал между машинами, м; Q — отношение времени разрешенного движения в рассматриваемом направлении к полному периоду работы перекрестка; ρ — число полос движения в одном направлении, шт.

Там же указано, что "при обычном ускорении при трогании с места автомобилей $a = 1,8 \text{ м/с}^2$ для пересечения улицы шириной 40 м необходимо 6,65 с, что дает на границе улицы скорость $V = 1,8 \cdot 6,65 = 12 \text{ м/с} = 43 \text{ км/ч}$ — скорость, совершенно недопустимая на перекрестке по соображениям безопасности пешеходов. Поэтому режим пересечения перекрестков должен быть установлен на скорости порядка 10 км/ч, что соответствует пересечению улицы машиной за 14 с.

Типовыми значениями Q для перекрестка двух улиц являются [3]:

- при однополосном движении — 0,40;
- при двухполосном движении — 0,38;
- при трехполосном движении — 0,37.

На практике при отсутствии на перекрестке препятствий в виде объезда, сужений или сворачивающей машины водитель набирает скорость и проходит дистанцию до конца пересечения и следует далее по маршруту в обычном режиме, т. е. стремится быстро набрать разрешенную скорость 60 км/ч, не заботясь о пешеходах, поскольку для них в это время действует запрещенный сигнал.

Поэтому средняя скорость проезда перекрестка зависит только от длины перекрестка и составляет 25...40 км/ч.

Тогда интенсивность разъезда очереди s (пропускная способность перекрестка P), рассчитанная по формуле (3) для $Q = 0,40$, $D = 2$ и $\rho = 2$:

- при $V = 10 \text{ км/ч}$ составит $s = 1,11 \text{ авт/с}$;
- при $V = 30 \text{ км/ч}$ составит $s = 3,33 \text{ авт/с}$;
- при $V = 60 \text{ км/ч}$ составит $s = 6,67 \text{ авт/с}$.

2. Если интенсивность прибытия превышает пропускные возможности дороги, то нужно переходить на жесткий режим переключения сигналов, т. е. включать сигналы на время, пропорциональное числу полос движения на пересекающихся направлениях, поскольку от полосности дороги зависит число автомобилей на ней. Если на некоторых из полос прибытие еще не достигло предела пропускания (например, полоса с поворотом направо), то время зеленого сигнала (зеленая стрелка вправо) регулируется (уменьшается) только на направлении этих полос (полосы), а на других полосах, где предел превышен, используется жесткий режим.

Очередь возникает, когда пропускная способность перекрестка становится меньше интенсивности прибытия транспорта.

Предлагаемая модель задержки ТС основана на физическом смысле процессов при проезде перекрестка, отвечает теории размерности и имеет вид

$$d = \frac{(c - g) q q_0}{g s^2}. \quad (3)$$

Если рассматривать работу перекрестка в течение 1 с, то интенсивность (число автомобилей в секунду) будет соответствовать просто числу машин, например, q машин для прибывающих и s — для отъезжающих.

Тогда составляющая числителя $(c - g)q$ — это время красного сигнала светофора, за которое к перекрестку прибывает q автомобилей с интенсивностью q , т. е. время задержки d прямо пропорционально времени запрещающего сигнала и интенсивности прибытия ТС.

Выражение sg в знаменателе отражает разъезд s автомобилей с интенсивностью s за время зеленого сигнала g , т. е. задержка d обратно пропорционально интенсивности убытия транспорта и времени разрешающего сигнала.

Кроме того, время задержки напрямую зависит от длины очереди q_0 , поэтому дробь q_0/s отражает число машин в очереди, приведенное к интенсивности разъезда s .

Расчеты задержек ТС по моделям (1)—(3) для одинаковых условий проезда перекрестка при цикле $c = 120$ с реальными значениями s , q и q_0 приведены в табл. 1 (для модели Бэкманна значения s и q приняты равными 3 и 2 соответственно).

Из табл. 1 следует, что задержка ТС имеет степенной характер, причем задержка резко возрастает, когда интенсивность прибытия превышает пропускную способность перекрестка.

Время задержки

d , с	d , с Вебстер	d , с Бэкманн	c , с	g , с	s , авт/с	q , авт/с	q_0 , авт	x , авт
0,00	0,01	0,01	120	120	2	3	1	0,1
0,17	0,74	2,10	120	108	2	3	2	0,2
0,56	3,18	7,80	120	96	2	3	3	0,3
1,29	7,53	17,10	120	84	2	3	4	0,4
2,50	13,75	30,00	120	72	2	3	5	0,5
4,50	21,42	46,50	120	60	2	3	6	0,6
7,88	29,90	66,60	120	48	2	3	7	0,7
14,00	38,51	90,30	120	36	2	3	8	0,8
27,00	47,06	117,60	120	24	2	3	9	0,9
67,50	68,77	148,50	120	12	2	3	10	0,99
189,75	91,47	183,00	120	5	2	3	11	0,995

На рис. 1 приведены графики задержки ТС для моделей (3), Вебстера и Бэкманна, построенные согласно данным табл. 1. Кривые показывают, что все они носят степенной характер, но с разными углами наклона, при этом сложно отдать предпочтение какой-либо модели, так как кривые построены только для заданных в табл. 1, т. е. фиксированных значений s , q , q_0 и x , а на практике подъезд, разъезд ТС, очередь и загрузка подхода (плотность автомобилей на полосе) могут варьироваться, при наличии очереди интенсивность разъезда может быть больше интенсивности подъезда (что для модели Бэкманна не приемлемо) и т. п.

Тем не менее для модели (3) характерно значительное увеличение времени задержки при достижении интенсивностью прибытия ТС уровня пропускной способности перекрестка. Это больше соответствует реальным условиям

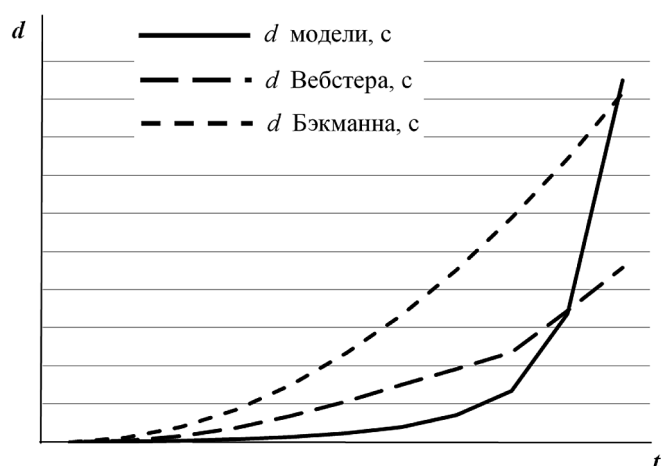


Рис. 1. Зависимость времени задержки ТС от параметров модели

на дороге — интенсивность разъезда уменьшается и образуется очередь ТС.

Для моделей (3) характерно резкое увеличение времени задержки при достижении очередью критического значения — когда интенсивность прибытия становится больше пропускной способности перекрестка, в модели Ф. В. Вебстера эта закономерность выражена гораздо слабее.

Для более полного определения степени влияния аргументов функции $d = f(c, g, q, q_0)$ на ее значение проведем анализ чувствительности функции (3) к изменению аргументов — параметров c , g , q , q_0 .

В работе [4] разработан метод определения количественного вклада (значимости, ценности, веса) каждого параметра дифференцируемой функции $u = f(x, y, z, \dots)$ в значение функции u .

Метод основан на связи относительных погрешностей функции δu и ее параметров δx , δy , δz , ..., которая имеет вид:

$$\delta u = \frac{1}{u} \left(x \left| \frac{\partial u}{\partial x} \right| \delta x + y \left| \frac{\partial u}{\partial y} \right| \delta y + z \left| \frac{\partial u}{\partial z} \right| \delta z + \dots \right)$$

или

$$\delta u = K_x \delta x + K_y \delta y + K_z \delta z + \dots, \quad (4)$$

где $K_x = \frac{x}{u} \left| \frac{\partial u}{\partial x} \right|$; $K_y = \frac{y}{u} \left| \frac{\partial u}{\partial y} \right|$; $K_z = \frac{z}{u} \left| \frac{\partial u}{\partial z} \right|$; ... — ко-

эффициенты влияния (значимости) соответствующих аргументов x , y , z , ... на функцию u .

При задании всем аргументам x , y , z , ... одинаковой единичной погрешности ($\delta_i = 1\%$) можно определить, на сколько процентов изменится величина δu при однопроцентном изменении каждого параметра.

Таблица 2

Вклад аргументов s , g , q и q_0 в значение функции d

t, c	$s, \%$	$c, \%$	$g, \%$	$q, \%$	$q_0, \%$
1	0,37	49,59	49,36	0,41	0,26
12	3,98	44,82	42,33	4,48	4,39
24	7,20	40,53	36,02	8,11	8,14
36	9,88	37,06	30,88	11,12	11,06
48	12,12	34,09	26,52	13,64	13,64
60	14,04	31,58	22,81	15,79	15,79
72	15,69	29,41	19,61	17,65	17,65
84	17,13	27,52	16,82	19,27	19,27
96	18,39	25,86	14,37	20,69	20,69
108	19,51	24,39	12,20	21,95	21,95
120	20,11	23,61	11,04	22,62	22,62
Среднее	12,58	33,50	25,63	14,16	14,13

После определения частных производных функции d по параметрам s , g , q и q_0 вычисляются коэффициенты значимости K_i по каждому параметру в различные моменты времени сигнала светофора и процентный вклад каждого параметра в значение задержки d .

Результаты расчетов слагаемых $K_i \delta_i$ уравнения чувствительности (4) для каждого параметра приведены в табл. 2. Из нее следует, что по вкладу в значение задержки d параметры расположились в следующем порядке:

- время цикла s — 33,5 %;
- длительность разрешающего сигнала светофора g — 25,63 %;
- интенсивность прибытия q — 14,16 %;
- очередь q_0 — 14,13 %;
- интенсивность разъезда s — 12,58 %.

Полученные значения соответствуют реальной логике работы перекрестка. Время цикла

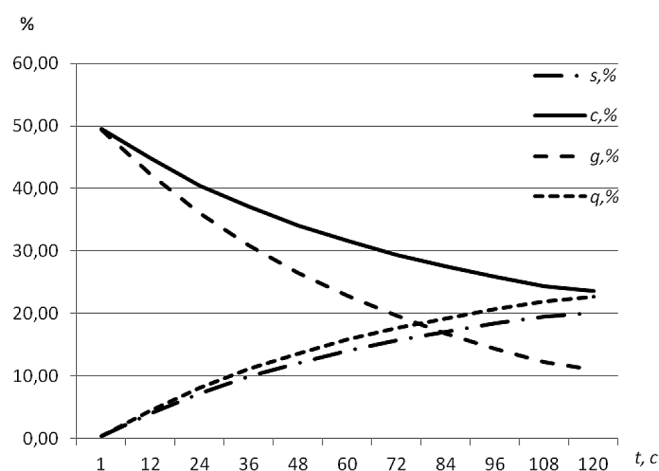


Рис. 2. Значимость параметров модели для задержки ТС

является эквивалентом длительности запрещающего сигнала светофора $r = c - g$, поэтому его влияние на задержку ТС — самое значительное. Длительность зеленого сигнала задает разъезд ТС, поэтому его вклад в задержку также существенен. Это же относится и к другим параметрам. Следует отметить, что значимости параметров имеют один порядок — от 12 до 26, т. е. в принципе являются равнозначными для времени задержки ТС на перекрестке.

По данным табл. 2 построены графики вклада каждого параметра в задержку (рис. 2), который показывает, что с течением времени значимости сигналов светофора уменьшаются, а значимости параметров передвижения (очередь и интенсивности прибытия/убытия) увеличиваются (кривые для интенсивности q и очереди q_0 практически совпадают, поэтому на рис. 2 представлены одной линией q).

Заключение

Проведенное исследование выявило следующие моменты:

1. Недостатком модели Бэкманна является тот факт, что интенсивность разъезда автомобилей должна быть больше интенсивности их прибытия к стоп-линии, что на практике не всегда выполняется.

2. Для моделей Бэкманна и Вебстера количественные значения интенсивностей подъезда, разъезда и очереди (плотности) ТС почти не влияют на задержку d .

3. Вторая и третья составляющие модели Вебстера практически не влияют на длительность задержки автотранспорта перед перекрестком, их доля составляет около 1...1,7 % от первой составляющей.

4. Все модели соответствуют теории размерности, т. е. достоверно с физической точки зрения описывают процессы на перекрестке.

5. Предложена новая модель задержки, для которой наиболее значимым параметром функции задержки является время цикла (длительность запрещающего сигнала светофора), его вклад в задержку составляет 33,5 %, на втором месте идет длительность зеленого сигнала — 25,63 %, далее идут интенсивность прибытия и очередь — 14,16 % и 14,13 %, а также интенсивность разъезда — 12,58 %.

6. Для модели (3) и модели Вебстера характерно значительное увеличение функции задержки при достижении интенсивностью прибытия ТС уровня пропускной способности перекрестка.

7. В течение цикла регулирования влияние на задержку длительности сигналов светофора уменьшается, а влияние очереди и интенсивностей прибытия/разъезда автомобилей увеличивается.

8. Новая модель и изменение ее параметров в ходе цикла работы светофора соответствуют реальным состояниям дорожного движения с большей аппроксимацией, чем базовые модели задержки.

Список литературы

1. Beckmann M. J., McGuire C. B., Winsten C. B. Studies in the Economics of Transportation. Yale University Press, 1956.
2. Webster F. V. Traffic Signal Settings. Road Research Laboratory Technical Paper No. 39, HMSO. London, 1958.
3. Пропускная способность перекрестка. URL: <http://waksman.ru/Russian/Vehi/shel/39.htm> (дата обращения: 2013-04-17).
4. Цуприков А. А. Чувствительность показателей механического бурения к изменению параметров управления // Сборник материалов НТВ СПбГПУ. 2009. № 3 (80). С. 4–7 (in Russian).

G. D. Antoniadis, Graduate Student, george@antoniadis.com,

V. O. Arkhipov, Graduate Student, arvlol@yandex.ru,

A. A. Tsouprikov, Ph. D., Associate Professor, tsouprikov@mail.ru,

Federal State Budgetary Educational Institution of Higher Education "Kuban State Technological University"

Mathematical Model of Transport Delay at the Crossroads

Comparative analysis of delay models Beckmann and Webster, the calculated values of the transport delay for the real traffic conditions at different durations of traffic signals, the intensities of arrival and travel of cars, size of a queue before the stop line, their shortcomings are revealed to control the traffic signals. It is found that for the Beckmann model, the intensity of the separation should be greater than the intensity of the arrival, and the quantitative values of the arrival/departure intensities and the size of the queue slightly affect the delay time.

The basic rules for choosing the duration of the resolving signal of the traffic light depending on the intensity of the transport arrival are considered: if it is less than or equal to the capacity of the intersection, the time of the green signal should correspond to the intensity of arrival and be maintained until the end of the queue, if it exceeds, the traffic light switches the signals in "hard mode" — for a time proportional to the number of bands on the intersecting directions.

A new model of vehicle delay at a controlled intersection is considered, the physical meaning of the model components is analyzed, the significance (weight, quantitative contribution) of the parameters to the value of transport delay at the prohibiting signal of a traffic light with the formation of a queue is determined. The most significant parameter of the function of the delay is the cycle time (the duration of a forbidding signal of a traffic light) with the contribution to the delay of 33.5 %, followed by the duration of green signal of 25.63 %, the intensity of arrival and the queue — 14,16 % and 14,13 %, the intensity of the separation of 12.58 %. The model takes into account the achievement of approaching transport limit — the capacity of the intersection. The verification of the considered models for compliance with the theory of dimension is carried out.

Keywords: model of travel delays, traffic light, intensity of travel, turn intersection, the significance of the parameters

DOI: 10.17587/it.25.210-215

References

1. Beckmann M. J., McGuire C. B., Winsten C. B. Studies in the Economics of Transportation, Yale University Press, 1956.
2. Webster F. V. Traffic Signal Settings. Road Research Laboratory Technical Paper No. 39, HMSO, London, 1958.

3. *Propusknaya sposobnost' perekryostka* (The capacity of the intersection), available at: <http://waksman.ru/Russian/Vehi/shel/39.htm> (date of access: 2013-04-17).

4. Tsouprikov A. A. *Chuvstvitel'nost' pokazatelej mekhanicheskogo bureniya k izmeneniyu parametrov upravleniya* (The sensitivity of the mechanical drilling to the change of control parameters), *Collection of materials of NTV SPbSTU*, 2009, no. 3 (80), pp. 4–7 (in Russian).

Д. Р. Потапов, аспирант, potapovd36@gmail.com,
Воронежский государственный университет, Воронеж

Исследование эффективности применения кэша для использования в самоадаптирующихся контейнерах данных

Проведено исследование зависимости эффективности кэша от среднеквадратичного отклонения (сигмы) и соотношения скоростей хранилищ для нормального распределения. Кроме того, выявлены основные закономерности для определения оптимального размера кэша для реализации самоадаптирующихся контейнеров данных. Исследование проведено для различных структур данных языка с# (Dictionary, Sorted Dictionary и Sorted List) и различных алгоритмов вытеснения кэша (Least Recently Used и Most Recently Used), эффективность оценивалась по времени поиска в самоадаптирующемся контейнере данных.

Ключевые слова: хранение данных, эффективность кэша, оптимальное хранилище данных, нормальное распределение, самоадаптирующийся контейнер данных, оптимальный размер кэша, нагрузка на контейнер данных

Введение

Самоадаптирующийся контейнер данных [1] — это ассоциативная структура данных, которая меняет логику своей работы в зависимости от нагрузки [2–5]. Нагрузка представляет собой различные последовательности операций вставки, выборки и удаления элементов из хранилища данных. Самоадаптирующиеся контейнеры данных могут быть реализованы путем использования различных структур данных в зависимости от нагрузки и условий работы контейнера [6–8]. Однако наиболее эффективно такие контейнеры могут проявить себя в условиях большого объема данных. В таком случае часть данных хранится в более быстрой памяти, а другая часть — в более медленной памяти или на удаленном источнике. Для приложений, где данные статичны или число запросов на получение данных намного превышает число запросов на добавление элементов в хранилище данных, наиболее оптимальным будет использование кэша [9].

Кэш представляет собой набор записей "ключ—значение", расположенный в памяти с большей скоростью доступа, предназначенный для ускорения обращения к данным, расположенным в хранилище данных с меньшей скоростью доступа. Кэширование применяется

как на аппаратном уровне [9] (в процессорах, жестких дисках), так и на программном уровне [9] (в операционных системах, на сетевом уровне (в сетях доставки контента (CDN), системах DNS, сессиях), интернет-приложениях, базах данных, мобильных приложениях, сервисах кэширования в оперативной памяти (Redis [10], Memcached [11, 12])). Помимо выигрыша в производительности при чтении данных кэш может использоваться и для ускорения записи (отложенная запись [13]), однако в данной статье рассматривается использование кэша только для чтения.

Самоадаптирующийся контейнер данных, использующий программный кэш, был реализован с использованием архитектурного шаблона "кэш на стороне" (cache-aside) [14]. При данной архитектуре кэш располагается в оперативной памяти, а основное хранилище — на жестком диске или на удаленном источнике (рис. 1), что позволяет программно изменять размеры кэша в процессе работы, чтобы обеспечить максимальную эффективность. Кэш и основное хранилище являются ассоциативными структурами данных, которые хранят пары "ключ—значение" (<Tkey, Tvalue>).

Важнейшим фактором, который определяет нагрузку на контейнер, является набор запрашиваемых ключей, который в общем случае

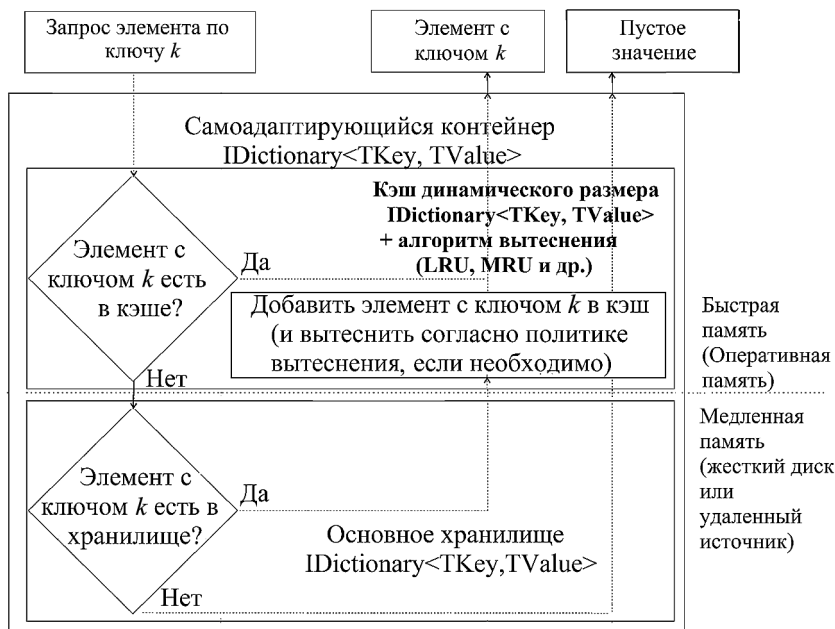


Рис. 1. Общая схема устройства самоадаптирующегося контейнера данных с использованием кэша

может быть описан с помощью нормального распределения. Таким образом, параметр этого распределения — среднее квадратичное отклонение (σ) [15] может влиять на размер кэша. Также необходимо учитывать такой фактор, как соотношение скоростей доступа к кэшу и основному хранилищу.

Постановка задачи

В данной статье проведено исследование зависимости эффективности кэша от среднее квадратичного отклонения и соотношения скоростей хранилищ для нормального распределения. Целью исследования является выявление основных закономерностей, формул и табличных значений для определения оптимального размера кэша в зависимости от перечисленных выше параметров, которые могут быть использованы для автоматического изменения размера кэша в процессе работы самоадаптирующегося контейнера данных.

Описание метода решения задачи

Для решения поставленной задачи было проведено тестирование. Для этого было реализовано консольное приложение языка C#, которое использует:

1) библиотеку Troschuetz.Random для генерации ключей по закону нормального распределения;

2) класс Stopwatch [16] языка C# для подсчета времени выполнения (в тактах процессора) функции поиска в контейнере;

3) программу Excel 2013 [17] для визуализации полученных численных результатов.

Тестирование с помощью данного приложения проводили на следующей программно-аппаратной платформе:

1. Processor: Intel Core i7 6500U @ 2.50 GHz (2 cores).
2. RAM: 16 GB.
3. OS: Windows 10.
4. MS Visual Studio 2015.

В процессе исследования был проведен ряд тестов, параметры которых были заданы следующим образом:

1) элементом кэша и основного хранилища является пара "ключ—значение" ($\langle \text{string}, \text{int} \rangle$), где ключом является строка языка C# (string) длиной в 10 символов, значением выступает случайное число типа int языка C#;

2) основное хранилище содержит 10^5 элементов, которые генерируются случайным образом;

3) запрашиваются 10^7 элементов по ключам, которые генерируются следующим образом:

а) для заданного среднее квадратичного отклонения σ согласно закону нормального распределения $N(0, \sigma^2)$ генерируются целые числа;

б) полученные числа трансформируются в строки длиной в 10 символов путем добавления нулей слева для недостающих символов. Например, для целого числа 1256 строка будет сформирована в виде "0000001256", а для числа 965169 — в виде "0000965169".

Таким образом, параметр σ показывает, насколько большей имеется разброс значений в нагрузке, т. е. чем больше σ , тем меньше вероятность повторного запроса ключа, чем меньше σ , тем больше вероятность повторного запроса;

4) в ходе эксперимента основное хранилище эмулировалось с помощью ассоциативной структуры данных языка C# Dictionary<string, int>, расположенной в оперативной памяти. Таким образом, соотношение скорости кэша и основного хранилища (далее обозначается k) для эксперимента равно единице ($k = 1$);

5) для каждого теста отдельно вычисляется время, затраченное на поиск в кэше ($t_{\text{кэш}}$) и поиск в основном хранилище ($t_{\text{осн. хранилище}}$).

Один тест представляет собой комбинацию из нескольких параметров:

1. Размер кэша. Рассматривались значения 10, 9100, 18 190, 27 280, 36 370, 45 460, 54 550, 63 640, 72 730, 81 820, 90 910, 100 000, 500 000, 1 000 000, 2 000 000 элементов в кэше.

2. Среднеквадратичное отклонение набора запрашиваемых ключей $\sigma = \{10, 50, 100, 500, 1000, 5000, 10\ 000, 15\ 000, 20\ 000\}$ при математическом ожидании $\mu = 0$ для нормального распределения $N(\mu, \sigma^2)$.

3. Используемый алгоритм вытеснения: LRU (Least Recently Used) [18, 19] и MRU (Most Recently Used) [20]. Данное исследование направлено на нахождение общих зависимостей, независимо от конкретных алгоритмов вытеснения. Поэтому для тестирования были выбраны алгоритмы, имеющие принципиально противоположные стратегии вытеснения (LRU вытесняет неиспользованный дольше всех элемент, а MRU — последний использованный), а результаты тестов были усреднены.

4. Ассоциативная структура данных для кэша, реализованная в языке C# (Dictionary<string, int>, SortedDictionary<string, int> и SortedList<string, int>) [21].

Было проведено по 20 тестов для каждой из комбинаций параметров и вычислено среднее арифметическое время поиска в кэше и основном хранилище по всем комбинациям и таким образом получено среднее время $t_{\text{ср. кэш}}$ и $t_{\text{ср. осн. хр}}^{k=1}$ для всех пар (σ , размер кэша) при $k = 1$ (так как во время эксперимента основное хранилище расположено в оперативной памяти, а следовательно, соотношение скоростей равно единице).

Вычислительный эксперимент и анализ

Используя полученные в результате эксперимента данные по среднему времени поиска в кэше и в основном хранилище, расположенном в оперативной памяти, можно вычислить общее время работы самоадаптирующегося контейнера для ситуаций, когда основное хранилище содержит достаточно много элементов и располагается на жестком диске или удаленном источнике. В этом случае соотношение скорости кэша и основного хранилища будет больше единицы ($k > 1$), так как скорость самых современных SSD дисков в разы меньше скорости оперативной памяти, а для HDD дисков и удаленных источников достигает сотен тысяч раз и более (в зависимости от конфигурации компьютера и скорости доступа к уда-

ленному источнику). Таким образом, среднее время поиска в основном хранилище может быть вычислено как $kt_{\text{ср. осн. хр}}^{k=1}$ (так как время поиска на диске или удаленном источнике в k раз больше, чем время поиска в оперативной памяти), а среднее общее время работы контейнера для всех пар (σ , размер кэша) может быть получено по формуле

$$t_{\text{ср. общ}} = t_{\text{ср. кэш}} + kt_{\text{ср. осн. хр}}^{k=1}, \quad (1)$$

где k — необходимое соотношение скорости кэша и основного хранилища; $t_{\text{ср. кэш}}$ — среднее время поиска (в тактах процессора) в кэше, расположенном в оперативной памяти; $t_{\text{ср. осн. хр}}^{k=1}$ — среднее время поиска (в тактах процессора) в основном хранилище, расположенном в оперативной памяти (при $k = 1$).

С использованием формулы (1) и средств Excel были построены графики зависимости времени работы контейнера от сигмы и размера кэша для различных значений k . В результате анализа этих графиков было выявлено, что при указанных выше условиях и входных данных можно выделить три различные по характеру графиков группы в зависимости от k :

1) $k < 8$ (кэш ненамного быстрее основного хранилища). В этом случае наиболее оптимальным решением будет не использовать кэш вообще, поскольку время выполнения без него меньше, чем при любых размерах кэша. Это связано с тем, что при небольших значениях k время, затраченное на поиск в кэше, больше, чем выгода от его использования. На рис. 2 и 3 (см. вторую сторону обложки) наглядно показано что для любой σ минимальные значения достигаются при нулевом размере кэша. В табл. 1 представлены результаты тестов для $k = 5$;

2) $k > 14$ (кэш намного быстрее основного хранилища). В этом случае оптимальным размером кэша является примерно 8σ (согласно математической статистике в диапазоне $(-4\sigma; 4\sigma)$ располагаются 99,997 % всех значений для нормального распределения) [22], т. е. практически все элементы, участвующие в выборке, помещаются в кэш, и затраты на поиск в таком кэше окупаются. Больше этого значения увеличение кэша не дает ощутимого прироста в производительности. На рис. 4 и 5 (см. вторую сторону обложки) можно заметить, что время поиска достигает минимума для каждой σ при достижении размера кэша примерно 8σ , после чего меняется незначительно. В табл. 2 представлены результаты эксперимента для $k = 20$.

Таблица 1

Время поиска в зависимости от σ и размера кэша для $k = 5$

Размер кэша	σ								
	10	50	100	500	1000	5000	10000	15000	20000
0	11008775	11717288	12987867	21848120	33667977	61566760	68428464	69680182	70807284
10	26940705	33890779	40835969	68498983	81871135	99303146	102834053	103707956	103845549
9100	23845447	31277860	35182731	51799736	65341135	166050254	184057853	191102304	194483186
18190	23840746	31314295	35246913	51764485	65095889	158558086	189196597	203905118	209896699
27280	23923557	31357142	35175279	51649246	65459080	136677576	187540742	207563583	217477651
36370	23852812	31345004	35122799	51525828	65150845	115482041	186774839	208546705	221866535
45460	23830789	31370141	35141213	51487477	65000654	109702657	172906803	208553253	222735013
54550	23879114	31352730	35219454	51850471	65122474	109123617	159923849	207654561	224457537
63640	23846348	31325336	35178629	51432401	65344654	109195183	147747251	197333128	223470878
72730	23856182	31365298	35156191	51602573	64936630	109068561	129887982	183386692	223257982
81820	23931219	31407959	35134752	51612698	64890774	109982322	129943863	173610780	215949778
90910	23824473	31292224	35211458	51477154	64830806	109140683	129291463	162604933	201759919
100000	23843379	31376769	35201755	51553538	64965822	108824473	129628560	147536611	191798905
500000	23872934	31377701	35154747	51678465	65384576	109447917	128939390	141179342	150550711
1000000	23840667	31339943	35078915	51574599	65066264	109459951	129133388	140370802	149799327
2000000	23836529	31327811	35159711	51846836	64963718	108802369	128970369	140220465	149080165

Таблица 2

Время поиска в зависимости от σ и размера кэша для $k = 20$

Размер кэша	σ								
	10	50	100	500	1000	5000	10000	15000	20000
0	44035099	46869151	51951467	87392480	134671909	246267041	273713854	278720729	283229134
10	61378441	88280668	115161873	224040782	276603584	345967157	359823183	363293504	364105260
9100	23853405	31312333	35246751	52067457	65825853	367023041	441143427	468451518	481804017
18190	23848658	31348402	35310551	52030195	65582426	268362245	399521701	455330614	479776381
27280	23931456	31391515	35238668	51913748	65941924	184213669	349734798	428521244	468664433
36370	23860900	31379294	35185284	51791098	65633320	120602442	300075423	394017569	448067855
45460	23838958	31404270	35204942	51752677	65486242	111741472	246978712	359526265	422497948
54550	23887135	31386604	35283781	52119815	65607091	111159454	204013413	324071662	397874795
63640	23854226	31359105	35242288	51700179	65836834	111223692	166663594	284596358	369389281
72730	23864142	31399760	35219158	51868719	65421600	111099971	133618742	246133997	342868562
81820	23939240	31441928	35197994	51879990	65371615	112033657	133690887	215972201	312121471
90910	23832615	31325819	35274286	51743793	65314687	111181060	133029939	186473018	277001355
100000	23851499	31410873	35265944	51819655	65451104	110867690	133388158	156539510	249907663
500000	23880920	31411112	35216670	51945440	65872213	111491062	132668858	146516036	157480155

При увеличении k увеличивается разность между временем поиска без использования кэша и временем поиска с использованием оптимального размера кэша, т. е. эффективность использования увеличивается с ростом k .

Однако не всегда есть возможность создать кэш необходимого размера. В этом случае оптимальным решением может оказаться не максимально доступный размер кэша, а отказ от его использования. Это связано с тем, что затраты на поиск в кэше могут быть больше, чем прирост производительности от его использования. На рис. 6 (см. вторую сторону обложки) изображена зависимость времени поиска от размера кэша для различных σ , где наглядно виден рост времени поиска при использовании недостаточного кэша и плавное уменьшение при увеличении доступного размера кэша. На основании проведенных экспериментов можно выделить два случая оптимального размера кэша.

1. Если доступный размер кэша больше определенного значения (но меньше 8σ) (далее $x_{\text{опт}}$), то оптимальный размер кэша равен максимально доступному значению.

2. При доступном размере кэша меньше $x_{\text{опт}}$ оптимальный размер кэша равен 0.

При этом $x_{\text{опт}}$ аппроксимировано может быть вычислено для определенного k и σ (или близких значений из таблиц) по формуле

$$x_{\text{опт}} = \frac{\left(y_0 - y_1 + \frac{(y_2 - y_1)}{(x_2 - x_1)} \cdot x_1 \right)}{\frac{(y_2 - y_1)}{(x_2 - x_1)}},$$

где y_0 — время в тактах при нулевом размере кэша; y_1, y_2 ($y_2 > y_0 > y_1$) — время в тактах на концах отрезка $(x_1; x_2)$; x_1 и x_2 — размеры кэша, следующие друг за другом в таблице.

3) $14 > k > 8$. При данном соотношении скоростей оптимальным может оказаться как решение из пункта 1, так и из пункта 2 (в зависимости от k и σ , табл. 3). На рис. 7 (см. вторую сторону обложки) наглядно показано, что оптимальным размером кэша (минимальным значением времени поиска) может быть как нулевой размер, так и максимально доступный размер кэша.

Таблица 3

Время поиска в зависимости от σ и размера кэша для $k = 11$

Размер кэша	σ								
	10	50	100	500	1000	5000	10000	15000	20000
0	24219304	25778033	28573307	48065864	74069550	135446873	150542620	153296401	155776024
10	40715800	55646735	70566331	130715703	159764115	197968750	205629705	207542175	207949433
9100	23848631	31291649	35208339	51906824	65535022	246439369	286892083	302041989	309411519
18190	23843911	31327938	35272368	51870769	65290504	202479750	273326639	304475316	317848572
27280	23926716	31370891	35200634	51755047	65652218	155692013	252418364	295946648	317952364
36370	23856047	31358720	35147793	51631936	65343835	117530201	232095073	282735051	312347063
45460	23834057	31383792	35166705	51593557	65194889	110518183	202535567	268942458	302640187
54550	23882322	31366279	35245185	51958209	65316321	109937952	177559674	254221401	293824440
63640	23849499	31338844	35204092	51539512	65541526	110006587	155313788	232238420	281838239
72730	23859366	31379083	35181378	51709031	65130618	109881125	131380286	208485614	271102214
81820	23934427	31421546	35160049	51719615	65083111	110802856	131442673	190555348	254418455
90910	23827730	31305662	35236590	51583809	65024358	109956834	130786853	172152167	231856493
100000	23846627	31390410	35227431	51659985	65159935	109641760	131132399	151137771	215042408
500000	23876129	31391065	35179516	51785255	65579630	110265175	130431177	143314019	153322489
1000000	23843821	31353546	35104710	51682270	65259442	110281483	130629389	142494953	152557330
2000000	23839726	31341438	35184410	51954870	65157271	109614881	130472266	142340488	151810443

Заключение

В данной работе представлен результат исследования эффективности применения кэша в зависимости от распределения данных (σ) и соотношения скорости основного хранилища и кэша. В результате анализа полученных данных были выявлены три группы закономерностей в зависимости от соотношения скоростей. В первой группе использование кэша не целесообразно, во второй группе оптимальный размер кэша составил 8σ (или может быть вычислен по указанным в статье формулам в случае недостатка памяти), в третьей группе оптимальное решение может быть получено из таблиц. Полученные закономерности могут быть использованы для динамического изменения размера кэша в зависимости от параметров нагрузки при реализации самоадаптирующегося контейнера данных.

Список литературы

1. **Потапов Д. Р., Артемов М. А., Барановский Е. С.** Обзор условий адаптации самоадаптирующихся ассоциативных контейнеров данных // Вестник ВГУ. Серия "Системный анализ и информационные технологии". Воронеж, 2017. № 1. С. 112–119.
2. **Зобов В. В., Селезнев К. Е.** Инструмент для моделирования нагрузки на контейнеры данных // Матер. 14-й науч.-метод. конф. "Информатика: проблемы, методология, технологии", 10–11 февраля 2011 г. Воронеж, 2014. Т. 3. С. 154–161.
3. **Потапов Д. Р., Селезнев К. Е.** Визуальное представление результатов анализа и сравнения контейнеров данных // Информатика: проблемы, методология, технологии. Матер. XVI Междунар. науч.-метод. конф. 2016. С. 123–128.
4. **Потапов Д. Р.** Визуальное сравнение и анализ контейнеров данных // Математическое и компьютерное моделирование, информационные технологии управления (МКМИТУ-2016). Воронеж. 2016. С. 178–181.
5. **Потапов Д. Р.** Разработка и реализация метода для анализа и сравнения контейнеров данных // Вестник Факультета прикладной математики и механики. 2016. № 5. С. 30–36.
6. **Потапов Д. Р.** Анализ применения многомерных структур данных в самоадаптирующихся контейнерах данных // Междунар. науч.-техн. конф. "Актуальные проблемы прикладной математики, информатики и механики". Воронеж, 18–20 декабря 2017 г. С. 436–442.
7. **Потапов Д. Р.** Обзор методов построения многомерных контейнеров данных "ключ—значение" для использования в самоадаптирующихся контейнерах данных // Прикладная информатика. 2018. № 2 (74). С. 69–82.
8. **Потапов Д. Р., Артемов М. А., Барановский Е. С., Селезнев К. Е.** Обзор методов построения контейнеров данных "ключ—значение" для использования в самоадаптирующихся контейнерах данных // Кибернетика и программирование. 2017. № 5.
9. **Jacob B., Ng S., Wang D.** Memory Systems: Cache, DRAM, Disk. San Francisco: Morgan Kaufmann Publishers, 2007. 900 p.
10. **Carlson J. L.** Redis in Action. NY: Manning Publications, 2013. 320 p.
11. **Sanderson D.** Programming Google App Engine with Python. Sebastopol, CA: O'Reilly Media, 2015. 538 p.
12. **Soliman A.** Getting Started with Memcached. Birmingham: Packt Publishing, 2013. 56 p.
13. **Таненбаум Э.** Архитектура компьютера. СПб.: Питер, 2003. 704 с.
14. **Cache-Aside** pattern. URL: <https://docs.microsoft.com/en-us/azure/architecture/patterns/cache-aside>
15. **Королюк В. С., Портенко Н. И., Скороход А. В., Турбин А. Ф.** Справочник по теории вероятностей и математической статистике. М.: Наука, 1985. 640 с.
16. **Deitel P., Deitel H.** C# 6 for Programmers (6th Edition) (Deitel Developer Series). Prentice Hall, 2016. 768 p.
17. **Walkenbach J.** Microsoft Excel 2016 bible: the comprehensive tutorial resource. John Wiley & Sons, 2015. 1152 p.
18. **Megiddo N., Modha D. S.** Outperforming LRU with an Adaptive Replacement Cache Algorithm // Computer, April 2004. Vol. 37, N. 4. P. 58–65.
19. **O'Neil E. J., O'Neil P. E., Weikum G.** The LRU-K Page Replacement Algorithm for Database Disk Buffering // Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. SIGMOD '93. New York, NY, USA: ACM, 1993. P. 297–306.
20. **Chou H.-T., DeWitt D. J.** An evaluation of buffer management strategies for relational database systems // Proceeding VLDB '85 Proceedings of the 11th international conference on Very Large Data Bases. 1985. Vol. 11. P. 127–141.
21. **Troelsen A., Japikse P.** C# 6.0 and the .NET 4.6 Framework. New York, NY, USA: Apress, 2015. 1832 p.
22. **Casella G., Berger R. L.** Statistical Inference. Duxbury Resource Center, 2001. 688 p.

D. R. Potapov, Postgraduate, e-mail: potapovd36@gmail.com,
Voronezh State University, Voronezh

Cache Efficiency Research for Using in Adaptive Data Storage

There are a lot of applications with a large amount of the static data or data which is using for reading mostly. At this rate cache applying improves performance greatly. In this case an adaptive data storage can change cache size during execution to achieve maximum efficiency. There are two main factors which affect container performance. The first one is a set of requesting data, which in common case can be described as Gaussian distribution. The second is a difference between speed of a main container and the cache. This paper provides a research results of a cache efficiency depending on the standard deviation of normal distribution and storage speed coefficient. The survey takes into consideration combinations of several parameters. One of

such parameters is a data structure. The research is carried out for different programming language C# data structures (Dictionary, SortedDictionary and SortedList). Another important parameter is a cache replacement policy. Various cache algorithms (Least Recently Used and Most Recently Used) are also examined in this research. Adaptive storage performance is evaluated as sum of a search time in the main container and a search time in the cache. The research results are analyzed in the article and the main patterns for calculating the optimal cache size for using in adaptive data storage are identified.

Keywords: store the data, cache efficiency, optimal data storage, normal distribution, adaptive data container, optimal cache size, data structures, container load, cache algorithms, dynamic cache

DOI: 10.17587/it.25.216-222

References

1. **Potapov D. R., Artemov M. A., Baranovskii E. S.** *Obzor uslovii adaptatsii samoadaptiruyushchikhsya assotsiativnykh kontejnerov dannykh* (Review adaptation conditions of adaptive associative data storages), *Vestnik Voronezhskogo gosudarstvennogo universiteta. Seriya: Sistemyi analiz i informatsionnye tekhnologii*, 2017, no. 1, p. 112–119 (in Russian).
2. **Zobov V. V., Seleznev K. E.** *Instrument dlya modelirovaniya nagruzki na kontejnery dannykh* (Tool for modeling the load on data containers), *Materialy chetyrnadtsatoi nauchno-metodicheskoi konferentsii "Informatika: problemy, metodologiya, tekhnologii"*, 10–11.02.2011 g. Voronezh, VGU, 2014, vol. 3, pp. 154–161 (in Russian).
3. **Potapov D. R., Seleznev K. E.** *Vizual'noe predstavlenie rezul'tatov analiza i sravneniya kontejnerov dannykh* (Visual presentation of results of analysis and comparison of data containers), *Informatika: problemy, metodologiya, tekhnologii. Materialy XVI Mezhdunarodnoj nauchno-metodicheskoi konferentsii*, 2016, pp. 123–128 (in Russian).
4. **Potapov D. R.** *Vizual'noe sravnenie i analiz kontejnerov dannykh* (Visual comparison and analysis of data containers), *Matematicheskoe i komp'yuternoe modelirovanie, informacionnye tekhnologii upravlenija (MKMITU-2016)*, Voronezh, 2016, pp. 178–181 (in Russian).
5. **Potapov D. R.** *Razrabotka i realizacija metoda dlja analiza i sravneniya kontejnerov dannykh* (Development and implementation of a method for analyzing and comparing data containers), *Vestnik Fakul'teta prikladnoj matematiki i mehaniki*, 2016, no. 5, pp. 30–36 (in Russian).
6. **Potapov D. R.** *Analiz primenenija mnogomernykh struktur dannykh v samoadaptirujushchihhsja kontejnerah dannykh* (Multidimensional data structures usage in adaptive data storages review), *Sbornik trudov Mezhdunarodnoj nauchno-tehnicheskoi konferentsii "Aktual'nye problemy prikladnoj matematiki, informatiki i mehaniki"*, Voronezh, 18–20 December 2017, pp. 436–442 (in Russian).
7. **Potapov D. R.** Existing methods of multidimensional "key-value" storages construction for using in adaptive data storages review, *Journal of applied informatics*, 2018, 2 (74): 69–82.
8. **Potapov D. R., Artemov M. A., Baranovskii E. S., Seleznev K. E.** *Obzor metodov postroenija kontejnerov dannykh "kljuch-znachenie" dlja ispol'zovanija v samoadaptirujushchihhsja kontejnerah dannykh* (Existing methods of "key–value" storages construction for using in adaptive data storages review), *Kibernetika i Programirovanie*, 2017, no. 5 (in Russian).
9. **Jacob B., Ng S., Wang D.** *Memory Systems: Cache, DRAM, Disk*, San Francisco, Morgan Kaufmann Publishers, 2007, p. 900.
10. **Carlson J. L.** *Redis in Action*, NY, Manning Publications, 2013, 320 p.
11. **Sanderson D.** *Programming Google App Engine with Python*, Sebastopol, CA, O'Reilly Media, 2015, 538 p.
12. **Soliman A.** *Getting Started with Memcached*, Birmingham, Packt Publishing, 2013, 56 p.
13. **Tanenbaum A.** *Arhitektura komp'yutera* (Computer architecture), Saint-Petersburg, Piter, 2003, 704 p. (in Russian).
14. **Cache-Aside** pattern, available at: <https://docs.microsoft.com/en-us/azure/architecture/patterns/cache-aside>
15. **Koroljuk V. S., Portenko N. I., Skorohod A. V., Turbin A. F.** *Spravochnik po teorii verojatnostej i matematicheskoi statistike* (A Handbook on Probability Theory and Mathematical Statistics), Moscow, Nauka, 1985, 640 p. (in Russian).
16. **Deitel P., Deitel H.** *C# 6 for Programmers* (6th Edition) (Deitel Developer Series), Prentice Hall, 2016, 768 p.
17. **Walkenbach J.** *Microsoft Excel 2016 Bible: the comprehensive tutorial resource*, John Wiley & Sons, 2015, 1152 p.
18. **Megiddo N., Modha D. S.** Outperforming LRU with an Adaptive Replacement Cache Algorithm, *Computer*, April 2004, vol. 37, no. 4, p. 58–65.
19. **O'Neil E. J., O'Neil P. E., Weikum G.** The LRU-K Page Replacement Algorithm for Database Disk Buffering, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. SIGMOD '93*, New York, NY, USA, ACM, 1993, p. 297–306.
20. **Chou H.-T., DeWitt D. J.** An evaluation of buffer management strategies for relational database systems, *Proceeding VLDB '85 Proceedings of the 11th international conference on Very Large Data Bases*, 1985, vol. 11, pp. 127–141.
21. **Troelsen A., Japikse P.** *C# 6.0 and the .NET 4.6 Framework*, New York, NY, USA, Apress, 2015, 1832 p.
22. **Casella G., Berger R. L.** *Statistical Inference*, Duxbury Resource Center, 2001, 688 p.

И. В. Филиппов, инженер по разработке программного обеспечения, e-mail: ili.filippov@gmail.com,
А. Ф. Мелик-Адамян, канд. техн. наук, архитектор инженерной разработки,
e-mail: areg.melik-adamyam@intel.com,
В. А. Сухомлинов, инженер по разработке программного обеспечения,
e-mail: vadim.sukhomlinov@gmail.com,
Intel Corporation US, США

Размещение сетевых функций в облачных инфраструктурах

Рассматривается задача выполнения сетевых функций (Network Functions, NF) в облачных инфраструктурах. Предлагается концепция представления сетевых функций в виде микросервисов. Для входа и выхода из системы используется граничная функция, преобразующая входящие пакеты к внутреннему представлению и распределяющая их по микросервисам. Обсуждается реализация системы, в которой полная цепочка микросервисов для каждого пакета выполняется на одном сервере по принципу "run-to-completion" с использованием возможности построения графа обработки пакетов системой NFF-GO.

Ключевые слова: сетевые функции, облачная инфраструктура, микросервисы, виртуализация, контейнеризация, NFF-GO

Введение

Компьютерная сеть включает в себя не только оборудование для передачи пакетов, но и многочисленные компоненты, реализующие дополнительную функциональность, связанную с промежуточной обработкой пакетов. В качестве примеров такой функциональности можно привести трансляцию сетевых адресов — NAT, фильтрацию трафика — Firewall, операции инкапсуляции и декапсуляции, шифрования и дешифрования, выставления счетов, проверки следования (Service Level Agreements, SLA) соглашениям на предоставление сервиса и т. д.

Такая функциональность называется сетевыми функциями. Изначально сеть строилась по принципу прозрачности — все пакеты, посланные в сеть, пересылались до адресатов без изменений, сетевые функции были интегрированы в приложения и исполнялись на оконечных устройствах вместе с ними. По мере усложнения сетевых функций они были вынесены в сеть, нарушая принцип прозрачности: помимо доставки пакетов сеть стала выполнять функции обработки пакетов.

Как только сетевые функции были вынесены с оконечных устройств, возникла задача их размещения и развертывания. Требования по производительности для сетевой функции, находящейся внутри сети, многократно возрастают по сравнению с оконечным устройством из-за большего трафика, так как функция обслуживает множество оконечных устройств одновременно.

С течением времени объем сетевого трафика растет за счет большого числа новых приложений. Это приводит к росту требований к пропускной способности и необходимости снижения задержек обработки (например, в связи с введением стандарта связи 5G). Реализация сложной функциональности требует проектирования с возможностью масштабирования сетевой функции как в рамках одного сервера, так и, если он не справляется, в многомашинном окружении. В данной статье будет предложена схема развертывания сетевых функций в облачной инфраструктуре, позволяющая достичь многомашинного масштабирования, эффективного управления, легкости развертывания и поддержания отказоустойчивости.

Существующие подходы

Стандартное (универсальное) оборудование и стандартные операционные системы (ОС) не подходят для высокопроизводительной обработки пакетов. Это связано с особенностями реализации, включающими большое число копирований памяти, с задержками, вносимыми механизмом прерываний, переключениями между пользовательским и системным пространством/режимом работы и другими факторами, делающими возможной универсальную обработку, но снижающими производительность.

По этой причине хронологически первым вариантом размещения сетевых функций является их размещение на специализированном оборудовании. Такое оборудование предназначено для оптимизированного решения одной конкретной задачи. Под каждую сетевую функцию создается отдельное устройство — middle-box.

Со временем появилась возможность создания высокопроизводительных сетевых функций на стандартном оборудовании. Появились специализированные библиотеки, предоставляющие собственные драйверы (например, DPDK), в ОС были добавлены возможности отображения памяти и запуска драйверов в режиме опрашивания, сетевые карты начали предоставлять RSS (Receive Side Scaling) — масштабирование на стороне приема и offloading — исполнение некоторых самых распространенных операций на сетевой карте. Эти и другие изменения сделали возможным второй вариант размещения сетевых

функций — на стандартном оборудовании со специальными библиотеками, драйверами и сетевыми картами.

Оба рассмотренных варианта развертывания объединяет жесткое закрепление каждой функции на исполняющем ее оборудовании (отдельном сервере или middle-box), которое связано физическим проводом с центральным коммутатором системы, — так называемый bump-in-the-wire подход (рис. 1). В сетях интернет-провайдеров находятся десятки специализированных middle-box или стандартных серверов, последовательно обрабатывающих пакеты.

Bump-in-the-wire подход имеет существенные недостатки. Набор функций тяжело изменить, так как необходимо физическое перемещение оборудования. Он тяжело масштабируется, так как резервное оборудование простаивает вне пикового времени. Необходимо физически перенастраивать конфигурацию оборудования для обработки отказов.

Эти проблемы привели к появлению третьего варианта развертывания — концепции виртуализации сетевых функций (NFV, Network Function Virtualization) [1, 2]. Основная идея NFV — исполнение сетевых функций на стандартном оборудовании с использованием механизма виртуализации, позволяющего изолировать функции друг от друга. Предполагается, что на одном сервере будут находиться несколько виртуальных машин, каждая из которых будет исполнять одну сетевую функцию [3]. Виртуальные машины будут связаны виртуальным коммутатором, который будет реализовывать последовательную обработку каждого пакета всеми необходимыми функциями (рис. 2, см. третью сторону обложки).

Похожая схема применяется при четвертом варианте размещения — контейнеризации [4] — технологии виртуализации, которая, однако, использует ядро физической ОС (рис. 3, см. третью сторону обложки). Плюсами контейнеров является снижение накладных расходов (память, реализация виртуализации) и быстрый старт и удаление функций.

В случае виртуализации функции находятся под управлением гипервизора, в случае контейнеризации — под управлением одного из оркестраторов (например, Docker).

Для снижения накладных расходов, вносимых виртуальным коммутатором и разделяемым между функциями ресурсом — сетевой картой, были созданы технологии, повышающие производительность сервера с виртуальными функциями: SR-IOV (Single Root I/O

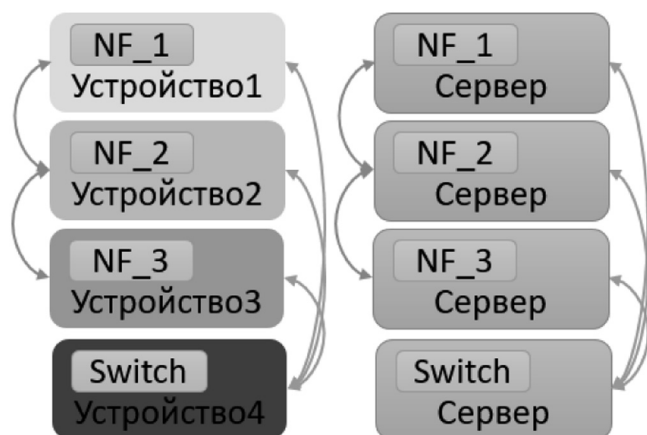


Рис. 1. "Bump-in-the-wire" вариант размещения сетевых функций, выполняющихся на специализированных устройствах или стандартных серверах (NF — сетевая функция, switch — центральный коммутатор)

Virtualization) — использование PCI порта одновременно несколькими виртуальными машинами; PCI passthrough — пересылка пакета в виртуальную машину напрямую из PCI устройства, минуя уровень гипервизора; полная или частичная реализация виртуального коммутатора в сетевой карте — offloading.

Концепция NFV является сложной в реализации. Коммутатор должен знать адреса всех сетевых функций, которые задаются ему через управляющий протокол (например, OpenFlow). Кроме того, он же должен выполнять роль балансировщика нагрузки, запуская дополнительную виртуальную машину, если какая-то виртуальная машина не справляется, и выгружая функциональность на отдельный сервер, если текущий сервер полностью перегружен. При этих действиях необходима автоматическая замена сетевых адресов сетевых функций.

Производительность страдает из-за того, что виртуальные машины не могут общаться напрямую и вынуждены каждый раз передавать пакет в коммутатор при реализации последовательной обработки пакета цепочкой функций, что приводит к лишним разборам каждого пакета.

Предлагаемое решение

◆ *Сетевые функции как микросервисы*

Одним из популярных шаблонов для создания облачных приложений является применение микросервисов [5]. Шаблон подразумевает разбиение функциональности приложения на небольшие отдельные подзадачи, разворачиваемые отдельно и поддерживающие связь друг с другом с помощью удаленных вызовов функций (Remote Procedure Call, RPC).

Преимуществами микросервисного подхода является отсутствие необходимости заново разворачивать все приложение при изменении отдельных частей, высокая обособленность отдельных частей, которые могут быть написаны на разных языках, разными разработчиками.

Рассмотрим обеспечение необходимой производительности в микросервисах. Каждый сервис может быть масштабирован, т. е. скопирован и запущен на свободных ресурсах. Роль доставки пакетов часто выделяется в отдельный сервис, называемый *sidecar* проху, в нем может проходить предобработка пакетов, балансировка нагрузки, динамическое определение маршрута посылки пакета и прочие вспомогательные задачи связи сервиса с сетью.

Задача организации приложения, использующего десятки микросервисов, каждый с десятками клонов, становится достаточно сложной, для ее решения используется специальный уровень инфраструктуры — *service mesh*, отвечающий за балансировку, отказоустойчивость и другие параметры приложения.

Приложения в основном работают на оконечных устройствах и находятся на уровне L7 сетевого стека. Однако сходную модель можно применить и внутри самой сети для организации сетевых функций. Действительно, как и в конечном приложении, в сетевой функции пакет должен пройти последовательно через различные стадии, которые можно назвать микросервисами. Как и в оконечном приложении, микросервисы можно масштабировать.

Представление сетевой функции или цепочки сетевых функций в виде микросервисов позволяет изменить семантику передачи пакетов между функциями от сетевой адресации к транспортному уровню, например, RPC вызовам или адресации точка-точка, что позволяет исключить виртуальный коммутатор и перейти к однократной коммутации. Кроме того, становится возможна более эффективная реализация передачи пакетов, например, с использованием механизмов общей памяти.

Между сетевой функцией, представленной в виде микросервисов, и остальной сетью необходимо использовать структуру, аналогичную *sidecar* проху для отдельного микросервиса. Будем называть такую структуру *граничной функцией*.

◆ *Граничная функция*

Для перехода от сетевого представления передачи пакетов к микросервисному необходима некоторая граница, которая бы преобразовывала входящий и исходящий трафик между пакетным представлением и внутренним представлением системы (рис. 4).



Рис. 4. Предлагаемое размещение сетевых функций: F — различные сетевые функции; C — клоны сетевых функций, выполняющие ту же функциональность

Для каждого входящего пакета граничная функция должна:

- 1) классифицировать пакет;
- 2) в результате классификации определить необходимую цепочку микросервисов для обработки пакета и дополнительные параметры для вызовов микросервисов;
- 3) выполнить вызовы микросервисов согласно определенной цепочке.

Так как сетевые функции обычно работают на уровне ниже L7, классификация пакета должна проводиться на низких уровнях. Например, это может быть разбиение на сетевые потоки по "5-tuple" признакам: двум L3 адресам, типу L4 протокола и двум L4 адресам. По "5-tuple" принципу работает, например, hash функция RSS сетевой карты. Другими примерами может быть классификация на основе MAC-адреса, MPLS-метки, или идентификатора туннеля в GTP. Если входной поток пакетов зашифрован, граничная функция может его расшифровывать или пересылать дальше как единый сетевой поток в зависимости от задачи.

Соответствие пакета и необходимой цепочки микросервисов является внешней настройкой граничной функции. Пакеты, принадлежащие различным сетевым соединениям, могут получать различные сервисы, отличающиеся как последовательностью функций (сервисных цепочек), так и конфигурацией отдельных микросервисов. Администратор сети должен иметь возможность настраивать граничную функцию под различные задачи, задавать правила классификации (например, с помощью протокола OpenFlow) и конфигурировать сервисные цепочки. Предполагается, что сами микросервисы для внешней настройки недоступны, они могут только получать параметры от граничной функции, тем самым реализуя принцип отсутствия внутреннего состояния (stateless), что является основой для масштабирования. Таким образом, реализуются принципы программно-конфигурируемых сетей (SDN — Software Defined Network) [6]. Уровень управления (Control Plane) представлен выбором микросервисов в зависимости от классификации пакета. Уровень данных (Data Plane) представлен микросервисами и отделен от управления.

Помимо управления граничная функция может осуществлять функцию мониторинга созданной системы и предоставлять пользователю различную статистику.

Выполнение найденной цепочки подразумевает несколько действий: инкапсуляцию пакета во внутреннее представление системы,

содержащее контекст для выполнения цепочки и последующей декапсуляции к обычному представлению пакета для дальнейшей отправки по сети; передачу инкапсулированного пакета на исполнение с помощью транспорта конкретной реализации (это могут быть RPC вызовы функций, как в канонических микросервисах, либо передача как IP пакет с внутренней адресацией), использование заданных каналов и т. д.

После выполнения цепочки сервисов пакет должен быть возвращен в граничную функцию, преобразован к обычному сетевому представлению и отправлен далее по сети через вышние порты граничной функции.

◆ *Реализация как run-to-completion*

Ранее мы описали общее представление предлагаемой архитектуры, здесь будет предложена одна из реализаций. Главным недостатком микросервисной архитектуры являются накладные расходы при взаимодействии распределенных сервисов. В случае сетевой функции это пересылка пакетов между микросервисами внутри цепочки.

В качестве эффективного решения предлагается пересылать пакеты по сети только от граничной функции к началу цепочки. Дальнейшее перемещение пакета должно осуществляться внутри одного сервера с помощью разделяемой памяти. Такая архитектура удовлетворяет концепции "run-to-completion": когда пакет попадает на сервер для выполнения, он не выгружается на другие серверы до окончания процесса.

Для реализации всех микросервисов на одном сервере необходим специальный подход к их исполнению. Мы предлагаем использовать нашу систему NFF-GO [7]. Основная идея NFF-GO заключается в построении сетевой функции в виде графа обработки пакетов с помощью зацепления предопределенных, но конфигурируемых блоков обработки пакета, вызываемых для каждого пакета, проходящего по графу. Для реализации шаблона микросервисов достаточно создать граф в виде цепочек, где отдельные микросервисы будут представлены блоками. С помощью нотации загружаемых плагинов можно создать динамический граф, конфигурируемый перед обработкой, а также блоки, загружаемые в граф из сторонних компонентов сторонних производителей.

NFF-GO предоставляет возможность автоматического масштабирования каждого блока в рамках одного сервера на различное число ядер процессора.

Таким образом, общая топология предлагаемого решения сводится к граничной функции, находящейся на отдельном сервере, и множеству серверов, на которых выполняется граф обработки пакетов, созданный с помощью NFF-GO. На каждом сервере осуществляется автоматическое масштабирование в рамках сервера. Если свободных ядер процессора не остается, это означает невозможность обработать на этом сервере входящий поток. В этом случае граничная функция берет на себя функции балансировщика нагрузки, запускает граф обработки на свободном сервере и перераспределяет на него часть сетевых соединений.

Мы нигде не оговаривали, как должны размещаться и администрироваться микросервисы на серверах. В отличие от концепции NFV в нашей модели это не имеет принципиального значения. NFF-GO-граф может быть запущен в виде виртуальной машины, контейнера или другой сущности, администрировать которую предлагает провайдер облачной инфраструктуры. Вопросы запуска, завершения и надежности каждого NFF-GO-графа остаются в ведении оркестратора средств размещения, например, для контейнеров это Kubernetes.

Отметим, что сама граничная функция также представляет собой набор последовательных операций, а значит, может быть реализована с помощью NFF-GO как один из микросервисов, предшествующий попаданию пакетов в остальную систему. Общая модель размещения сетевой функции в облачной инфраструктуре показана на рис. 5 (см. третью сторону обложки).

Результаты

Рассмотрим преимущества предложенного подхода. По сравнению с существующими способами реализации развертывания сетевых функций в многомашинном окружении, а именно, NFV, контейнеризацией, bump-in-the-wire подходом, предлагаемый метод:

- более простой в управлении, так как имеет один объект, с которым осуществляется взаимодействие, — граничную функцию и

однократную коммутацию — требование один раз выбрать цепочку сервисов для каждого сетевого соединения;

- более универсальный, так как не оговаривает конкретные способы размещения сетевых функций на обрабатывающих серверах;
- более производительный, так как gun-to-completion модель не имеет накладных расходов на пересылку пакета по сети, а однократная коммутация не предполагает разбора пакета перед каждым новым сервисом;
- более отказоустойчивый, так как на всех серверах находятся одинаковые графы, поэтому вычислительные мощности являются полностью идентичными;
- легче масштабируемый, так как в рамках одного сервера масштабирование осуществляется системой NFF-GO, что предполагает максимальное использование предоставляемых ресурсов, а на уровне многомашинной системы масштабирование осуществляет граничная функция один раз, а не балансировщик нагрузки, находящийся перед каждым этапом обработки.

Список литературы

1. ETSI // Network Function Virtualization. White paper. 2012.
2. Mijumbi R., Serrat J., Gorricho J. L., Bouten N., De Turck F., Boutaba R. Network function virtualization: State-of-the-art and research challenges // IEEE Communications Surveys & Tutorials. 2016. 18 (1). P. 236–262.
3. Li X., Qian C. A survey of network function placement // In Consumer Communications & Networking Conference (CCNC). 2016. 13th IEEE Annual. P. 948–953.
4. Anderson J. et al. Performance considerations of network functions virtualization using containers. Computing // Networking and Communications (ICNC), 2016 International Conference on IEEE. 2016.
5. Balalaie Armin, Abbas Heydarnoori, Pooyan Jamshidi. Microservices architecture enables devops: Migration to a cloud-native architecture // IEEE Software. 33.3 (2016): 42–52.
6. Xia W., Wen Y., Foh C. H., Niyato D., Xie H. A survey on software-defined networking // IEEE Communications Surveys & Tutorials. 2015. Vol. 17, N. 1. P. 27–51.
7. Philippov I., Melik-Adamyanyan A. Novel approach to network function development // Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia (CEE-SECR'17). ACM, New York, NY, USA, 2017. Article 17. 6 p. DOI: <https://doi.org/10.1145/3166094.3166111>

I. V. Philippov, Software Engineer, ili.filippov@gmail.com,
A. F. Melik-Adamyanyan, Software Architect, areg.melik-adamyanyan@intel.com,
V. A. Sukhomlinov, Software Engineer, vadim.sukhomlinov@gmail.com,
Intel Corporation US, USA

Cloud Deployment of Network Functions

The article discusses the problem of the deployment of network functions to a cloud infrastructure. We propose network function representation as a microservice and introduce a notion of the "Cloud Boundary Node" (CBN) for the boundary between functions system and another network. CBN classifies each incoming packet, determines a service chain for it, encapsulates packet to internal representation and sends it for processing, taking load balancing into account. Service chains are represented as a packet processing graph that should be implemented in a "run-to-completion" methodology — whole service chain is done at one server using shared memory technology. Packet processing graph can be built using NFF-GO system, which provides efficient automatic intra-node scaling. Graph structure is the same for all the nodes; however, the graph can be extended by using plugins, which are determined at the packet classification stage.

As a result, our architecture is easier to manage due to only one single management object — CBN, where the administrator can set the required service chains for every packet. Architecture can be better scaled and effectively handle various failures because all servers have the same processing graph, and CBN can simply rebalance traffic between them. Great performance is achieved by eliminating the costs of data transferring and commutation: only one commutation happens while CBN classification stage and only one data transfer happen between CBN and handling server.

Keywords: Network functions, Cloud infrastructure, Microservices, Virtualization, Containerization, NFF-GO

DOI: 10.17587/it.25.223-228

References

1. ETSI. "Network Function Virtualization" white paper, 2012.
2. Mijumbi R., Serrat J., Gorricho J. L., Bouten N., De Turck F., Boutaba R. Network function virtualization: State-of-the-art and research challenges, *IEEE Communications Surveys & Tutorials*, 2016, 18 (1), pp. 236–262.
3. Li X., Qian C. A survey of network function placement, *In Consumer Communications & Networking Conference (CCNC)*, 2016, 13th IEEE Annual, pp. 948–953.
4. Anderson J. et al. Performance considerations of network functions virtualization using containers. *Computing, Networking*

and Communications (ICNC), 2016 International Conference on IEEE, 2016.

5. Balalaie Armin, Abbas Heydarnoori, Pooyan Jamshidi. Microservices architecture enables devops: Migration to a cloud-native architecture, *IEEE Software*, 33.3 (2016): 42–52.

6. Xia W., Wen Y., Foh C. H., Niyato D., Xie H. A survey on software-defined networking, *IEEE Communications Surveys & Tutorials*, 2015, vol. 17, no. 1, pp. 27–51.

7. Philippov I., Melik-Adamyanyan A. Novel approach to network function development, *Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia (CEE-SECR'17)*, ACM, New York, NY, USA, 2017, Article 17, 6 p., DOI: <https://doi.org/10.1145/3166094.3166111>

УДК 519.876.5

DOI: 10.17587/it.25.228-233

А. М. Пуртов, канд. техн. наук, ст. науч. сотр., e-mail: andr.purtov@yandex.ru,
Омский филиал Института математики им. С. Л. Соболева СО РАН, г. Омск

Использование образцов для выбора маршрутов в сетях передачи данных

Рассмотрена возможность использования образцов для маршрутизации в сетях передачи данных. С этой целью разработан образец, построена на GPSSW имитационная модель фрагмента сети передачи данных, проведены имитационные эксперименты. Результаты экспериментов позволяют сделать вывод о целесообразности применения образцов для маршрутизации.

Ключевые слова: образцы, принятие решений, графы, целенаправленное движение, маршрутизация, сеть передачи данных, имитационная модель, GPSSW, результаты экспериментов

Введение

Разработано много формальных и неформальных методов принятия решений в си-

стемах различного назначения, технических, экономических, социальных и др. Обучение и принятие решений по образцам часто используется в живой природе. Автор статьи разра-

батывает способы использования образцов для организации целенаправленного движения по ориентированным графам. На текущем этапе в качестве объектов для экспериментов используются сети передачи данных (СПД).

В одной из своих последних монографий [1] Н. Г. Загоруйко приводит различные методы анализа данных и классифицирования объектов. Некоторые методы основаны на использовании столпов (эталонов, образцов). В этом случае база данных содержит набор классифицируемых объектов. Считается, что нормированные параметры объектов являются их координатами в n -мерном пространстве. После обработки данных статистическими методами определяются типовые объекты — столпы. Классифицирование заключается в приписывании объектов к тем столпам, расстояния до которых минимально.

Иначе используются образцы в разрабатываемом автором статьи способе принятия решений в сложных системах, например, технических, экономических, социальных. Текущая ситуация в системе представляется набором параметров. Значения некоторых параметров случайны, изменяются в процессе функционирования системы. Поэтому множество ситуаций заранее не определено. Образцы задаются тем же набором параметров, который используется для описания ситуаций. Значения параметров образцов подбираются эвристически и с помощью имитационного моделирования. Решение принимается в зависимости от того, к какому из образцов ближе текущая ситуация в системе.

Первые эксперименты автора статьи по использованию образцов были проведены при сравнительном анализе алгоритмов управления потоками автомобилей на перекрестке [2, 3]. Концептуальная модель отображала четыре пересекающихся потока, 12 направлений, четыре светофора. Сравнились четыре алгоритма управления светофорами:

- светофоры переключаются через заданный интервал времени;
- светофоры могут быть переключены при отсутствии автомобилей в открытом для движения потоке;
- разрешение для движения получает поток с наибольшей очередью автомобилей;
- используются образцы для принятия решений о переключении светофоров.

Критерием для сравнения алгоритмов было среднее время ожидания автомобилей в очередях перед перекрестком.

Для сравнения использовались имитационные модели, запрограммированные на GPSSW.

Текущее состояние перекрестка определялось вектором $S = \{Q_a, Q_c, Q_e, Q_g\}$, где Q_a, Q_c, Q_e, Q_g — суммарные длины очередей автомобилей в направлениях соответствующих потоков (a, c, e, g). Каждому потоку перед началом моделирования задавался образец $ST = \{Q_a, Q_c, Q_e, Q_g\}$, представляющий точку в четырехмерном пространстве. Вычислялись евклидовы расстояния текущего состояния S до каждой из четырех точек. Разрешалось движение тому потоку, для которого расстояние от S до ST минимально.

Результаты имитационных экспериментов показали, что алгоритмы на основе образцов имеют право на существование. Они часто не хуже известных алгоритмов управления автомобилями на перекрестке с точки зрения временных задержек. Использование образцов имеет большие возможности для решения задач распределения ресурса времени между направлениями.

Постановка задачи

Необходимо оценить возможность использования образцов при выборе маршрутов на графах, в частности, в СПД. Для этого надо сравнить время достижения цели пакетами, управляемыми в узлах образцами, с известными методами маршрутизации. Нужно определить набор параметров образца, выбрать фрагмент СПД, построить имитационную модель его функционирования, провести эксперименты. Фрагмент СПД должен быть не очень большим, но при этом должен отражать основные проблемы маршрутизации в разветвленных сетях.

Теоретико-практические предпосылки

Первые сети ЭВМ использовали для маршрутизации протокол X.25. Процедуры X.25 позволяют передавать данные по постоянным виртуальным соединениям и в режиме адаптивной маршрутизации дейтаграмм. Увеличение СПД и повышение скорости каналов привели к вытеснению X.25 более эффективными протоколами, например, RIP (Routing Information Protocol), OSPF (Open Shortest Path First, "кратчайший путь первым"). Когда сети были небольшими, решение принималось на основе минимального времени достижения цели. После роста сетей стало проще использовать

кратчайшие пути с точки зрения числа промежуточных маршрутизаторов. Поэтому, несмотря на большие возможности протоколов RIP и OSPF, маршрутизаторам обычно передается информация о кратчайших путях с точки зрения числа узлов. В последнее время приобрел популярность протокол MPLS (Multiprotocol Label switching) [4]. Протокол находится между сетевым и канальным уровнями. Сверху обычно IP. Снизу могут быть Frame Relay, ATM, PPP, Ethernet. В MPLS могут использоваться фиксированные маршруты. Но главное достоинство заключается в поддержке технологии инжиниринга трафика. MPLS применяется как крупными операторами связи, такими, как Ростелеком, так и мелкими.

В данной статье описано использование для маршрутизации образца со следующими параметрами:

$$ST = \{Q, V, N, T\},$$

где Q — текущая длина очереди к исходящему из узла каналу связи; V — скорость исходящего канала (Мбит/с); N — минимальное расстояние от узла, принимающего решение по направлению передачи пакета, до адресата. Метрика — число участков передачи между узлами; T — текущее время передачи от начального узла до конечного.

При выборе набора параметров образца учитывался теоретический и практический опыт маршрутизации в СПД.

Для анализа возможности использования образцов была построена имитационная модель фрагмента СПД.

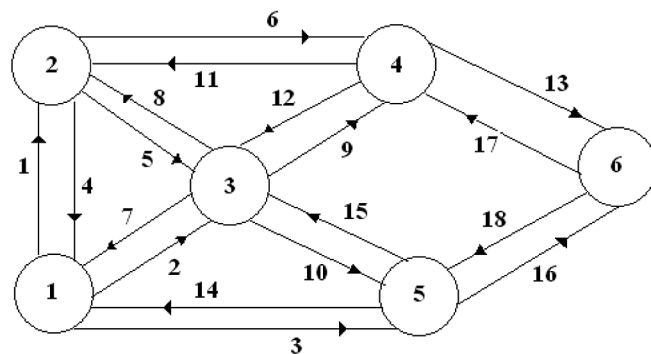
Описание имитационной модели

Имитационная модель разработана в среде бесплатно распространяемой версии GPSSW. В качестве объекта имитации был выбран фрагмент СПД, структура которого приведена на рисунке.

Окружностями с номерами обозначены узлы СПД (маршрутизаторы, коммутаторы). Линии с номерами и стрелками обозначают каналы связи.

При принятии решения в узле СПД о направлении передачи пакета выбирается тот исходящий из узла канал связи, параметры которого наиболее близки к значениям параметров образца.

Минимальные расстояния между узлами приведены в табл. 1.



Структура фрагмента СПД

Таблица 1

Минимальные расстояния между узлами

Номер начального узла	Номер конечного узла	Минимальное расстояние	Номер исходящего канала
1	1	0	
1	2	1	1
1	3	1	2
1	4	2	1
1	5	1	3
1	6	2	3
2	1	1	4
2	2	0	
2	3	1	5
2	4	1	6
2	5	2	5
2	6	2	6
3	1	1	7
3	2	1	8
3	3	0	
3	4	1	9
3	5	1	10
3	6	2	10
4	1	2	12
4	2	1	11
4	3	1	12
4	4	0	
4	5	2	12
4	6	1	13
5	1	1	14
5	2	2	15
5	3	1	15
5	4	2	16
5	5	0	
5	6	1	16
6	1	2	18
6	2	2	17
6	3	2	18
6	4	1	17
6	5	1	18
6	6	0	

В первом и втором столбцах табл. 1 приведены номера начального и конечного узлов. В третьем столбце указано минимальное число участков передачи от начального узла до конечного. В четвертом столбце содержатся номера исходящих каналов, с которых начинаются кратчайшие пути. Структура СПД, приведенная на рисунке, показывает, что между некоторыми парами узлов может существовать несколько путей с минимальными расстояниями. В модели считается, что кратчайшим является тот путь для передачи пакета, который найден первым.

Перед прогоном имитационной модели задаются значения параметров образца $ST = \{Q, V, N, T\}$. Считается, что они являются координатами точки, соответствующей образцу, в четырехмерном пространстве.

При определении направления передачи пакета для каждого исходящего из узла канала связи определяются текущие значения параметров $S = \{Q, V, N, T\}$, которые тоже задают точку в четырехмерном пространстве. Для каждого исходящего канала вычисляются расстояния от S до ST . Пакет передается по тому каналу, для которого расстояние от S до ST минимально.

Описание имитационных экспериментов

Перед прогоном имитационной модели задавались следующие основные параметры:

- среднее время между поступлением пакетов между каждой парой узлов. Время распределено экспоненциально;
- длина пакета 576 байт;
- скорости каналов связи (приведены в табл. 2);
- интервал времени работы СПД, имитируемый в прогоне модели.

Набор и значения параметров образца $ST = \{Q, V, N, T\}$. В имитационных экспериментах использовались разные сочетания параметров образца.

Результаты имитационных экспериментов приведены в табл. 3.

Значения параметров образцов и интенсивностей поступления пакетов выбирались эмпирически с последующей проверкой на имитационной модели. Критерием эффективности набора и значений параметров образцов было среднее время передачи пакетов по СПД. Интенсивности поступления пакетов должны были обеспечить максимальную, но приемлемую загрузку каналов связи.

Таблица 2

Скорости каналов связи

Номер	Узел 1	Узел 2	Скорость канала (Мбит/с)
1	1	2	1
2	1	3	2
3	1	5	1
4	2	1	1
5	2	3	2
6	2	4	1
7	3	1	2
8	3	2	2
9	3	4	2
10	3	5	2
11	4	2	1
12	4	3	2
13	4	6	1
14	5	1	1
15	5	3	2
16	5	6	1
17	6	4	1
18	6	5	1

Таблица 3

Основные данные экспериментов

Номер	Параметры образца	T_{1cp} (мкс)	T_{2cp} (мс)
1	(-, -, 0, -)	300	4,6
2	(0, -, 0, -)	300	3,457
3	(0, 4, 0, -)	300	4,56
4	(0, 3.1, 0, -)	300	3.04
5	(0, 3.0, 0, -)	300	3,46
6	(-, -, 0, 0)	300	3,16
7	(-, -, 0, 1)	300	3,12
8	(-, -, 0, 0)	250	3,77
9	(0, 3.2, 0, 0)	225	3,65

Примечание: T_{1cp} — среднее время между поступлением пакетов между каждой парой узлов; T_{2cp} — среднее время передачи пакетов по СПД.

Прочерк в описании параметров образца означает, что параметр при прогоне имитационной модели не использовался.

В образце в первом эксперименте используется только один параметр, задающий передачу по маршрутам с минимальным числом

участков передачи. Фактически это передача по фиксированным маршрутам, которые пере-страиваются только при изменении структуры сети. Благодаря простоте и высокой скорости обработки пакетов в узле такие методы часто используются в протоколах RIP, OSPF, MPLS.

В образец второго эксперимента добавлен параметр, учитывающий текущую длину очереди пакетов к исходящему из узла каналу. Такой алгоритм можно отнести к классу адаптивных. Этот параметр непосредственно доступен маршрутизатору и позволяет снизить время передачи по сети.

В образцы экспериментов 3, 4, 5 добавлен параметр, учитывающий скорость каналов связи. Видно, что конкретное значение параметра образца может сильно влиять на время передачи пакетов.

В образцах экспериментов 6, 7, 8 используется параметр, учитывающий время передачи пакетов до адресата. Этот критерий позволяет снизить время передачи. Его применение требует дополнительных затрат времени и передачи служебных сообщений. Поэтому в настоящее время он используется редко.

В образце эксперимента 9 используются четыре параметра. Из результатов видно, что увеличение числа используемых параметров не всегда приводит к уменьшению времени передачи.

Результаты экспериментов показывают, что образцы могут использоваться для маршрутизации в СПД.

Заключение

Апробирована идея использования образцов для выбора направлений движения транзакций по графу. Апробация проводилась на процессах выбора маршрутов в СПД. Для этого был выбран набор параметров образца, построена на GPSSW имитационная модель фрагмента СПД, определены значения параметров образца, проведены имитационные эксперименты. Результаты экспериментов показали возможность и целесообразность использования образцов для выбора маршрутов.

Образцы могут использоваться для выбора маршрутов при целенаправленном движении по ориентированным графам в системах различного назначения: технических, экономических, социальных.

В дальнейшем планируется апробация образцов при организации движения в автотранспортных сетях.

Список литературы

1. Загоруйко Н. Г. Когнитивный анализ данных. Новосибирск: Академическое изд-во "Гео", 2013. 186 с.
2. Пуртов А. М. Разработка и анализ имитационной модели перекрестка для системы GisAuto // Омский научный вестник. 2013. № 1 (117). С. 225–229.
3. Пуртов А. М. Имитация систем управления потоками автомобилей на перекрестке // Омский научный вестник. 2016. № 3 (147). С. 92–96.
4. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. СПб.: Питер, 2011. 944 с.

A. M. PurtoV, The Senior Scientific Employee, e-mail: andr.purtov@yandex.ru,
Branch of S. L. Sobolev Institute of Mathematics of Siberian Branch of Russian Academy of Sciences

Use of Samples for the Choice of Routes in Networks of Data Transmission

The author of article elaborate methods of use of samples for the organization of purposeful movement on the orientated graphs. The first experiments on the use of samples were carried out in a comparative analysis of algorithms for controlling the flow of cars at the intersection. The results of simulation showed great possibilities of algorithms based on samples to control traffic lights at the intersection. At the current stage, the efficiency of algorithms based on samples when choosing routes in data networks is investigated. For this purpose, a set of parameters of the sample was determined, a simulation model of a fragment of the data network was built, simulation experiments were conducted. The following set of parameters was selected for the sample: $ST = \{Q, V, N, T\}$. Parameter description: Q — current queue length to outgoing channel from node; V — outgoing channel speed (Mbit/c); N — the minimum number of transmission sections from the current node to the destination; T — current transfer time from the current node to the final node. The simulation model is implemented on GPSSW. Before running the simulation model, the values of the sample parameters $ST = \{Q, V, N, T\}$ are set. They are considered to be the coordinates of the point corresponding to the sample in four-dimensional space. When determining the direction of transmission of the packet for each outgoing communication channel from the node, the current values of the parameters $S = \{Q, V, N, T\}$ are determined, which also specify a point in four-dimensional space. The packet is transmitted through the channel for which

the distance from S to ST is minimal. The criterion for comparing the algorithms was the average time of packet transmission over the data network. Simulation experiments have shown that the average transmission time of data packets can strongly depend on the combinations and values of the sample parameters. The article tested the idea of using samples to select the direction of purposeful movement of transactions on the oriented graph. Testing was carried out on a simulation model of route selection processes in the data network. The results of the experiments showed the possibility and feasibility of using samples for routing. Samples can be used to select routes for targeted traffic on oriented graphs in systems for various purposes: technical, economic, social. In the future, it is planned to test samples in the organization of traffic in road networks.

Keywords: samples, decision making, graphs, targeted movement, routing, data network, simulation model, GPSSW, experimental results

DOI: 10.17587/it.25.228-233

References

1. **Zagorujko N. G.** *Kognitivnyj analiz dannyh* (Cognitive data analysis), Novosibirsk, Akademicheskoe izd-vo "Geo", 2013, 186 p. (in Russian).

2. **Purtov A. M.** *Razrabotka i analiz imitacionnoj modeli perekrestka dlja sistemy GisAuto* (Development and analysis of the intersection simulation model for the GisAuto system),

Omskij Nauchnyj Vestnik, 2013, no. 1 (117), pp. 225–229 (in Russian).

3. **Purtov A. M.** *Imitacija sistem upravlenija potokami avtomobilej na perekrestke* (Simulation of car flow control systems at the intersection), *Omskij Nauchnyj Vestnik*, 2016, no. 3 (147), pp. 92–96 (in Russian).

4. **Olifer V. G., Olifer N. A.** *Komp'yuternye seti. Principy, tehnologii, protokoly* (Computer network. Principles, technologies, protocols), SPb., Piter, 2011, 944 p. (in Russian).



17—19 июня 2019 года в Москве состоится

XIII Всероссийское совещание по проблемам управления,

посвященное 80-летию Института проблем управления
имени В. А. Трапезникова РАН

Сопредседатели программного комитета:

академик РАН **Е. А. Микрин** и член-корреспондент РАН **Д. А. Новиков**

Цели Совещания

- ознакомить участников с последними достижениями науки и практики управления по разным направлениям исследований и практических разработок;
- выявить основные тенденции и связи между различными направлениями науки об управлении, обсудить сценарные прогнозы их развития;
- выявить в процессе дискуссий проблемы и наиболее перспективные направления теории управления;
- содействовать упрочению связей между представителями различных академических и отраслевых научных центров, вузовской науки и реального сектора экономики;
- обсудить проблемы образования в области управления и задачи, которые ставит перед теорией управления современная практика.

Направления работы Совещания:

- ❖ Теория систем управления
- ❖ Управление подвижными объектами и навигация
- ❖ Интеллектуальные системы в управлении
- ❖ Управление в промышленности и логистике
- ❖ Управление системами междисциплинарной природы
- ❖ Средства измерения, вычислений и контроля в управлении
- ❖ Системный анализ и принятие решений в задачах управления
- ❖ Информационные технологии в управлении
- ❖ Проблемы образования в области управления: современное содержание и технологии обучения

Подробная информация о Совещании находится на сайте

<http://vspu2019.ipu.ru>

В. Е. Торчинский, доц., e-mail: vet@magtu.ru,
И. В. Торчинская, вед. инж.-программист, e-mail: asu@magtu.ru,
А. С. Файнштейн, канд. физ.-мат. наук, доц., e-mail: swetlana@mgn.ru,
С. И. Файнштейн, доц., e-mail: sfainshtein@yandex.ru,
Магнитогорский государственный технический университет им. Г. И. Носова

"Гибридный" алгоритм планирования государственных закупок товарно-материальных ценностей

Для планирования госзакупок товарно-материальных ценностей предлагается "гибридный" алгоритм набора заданной суммы из заданных стоимостей заявок (NP-полная задача "Сумма размеров"). Сумма набирается полиномиальным алгоритмом до тех пор, пока размер подзадачи не становится пригодным для решения точным алгоритмом. Такой "гибридный" алгоритм наследует вычислительную сложность и оценку абсолютной погрешности полиномиального алгоритма. В то же время применение на последнем этапе решения точного алгоритма позволяет набирать сумму с большой степенью точности. Данный алгоритм применяется при планировании госзакупок Магнитогорского государственного технического университета им. Г. И. Носова.

Ключевые слова: NP-полнота, задача "Сумма размеров", приближенный алгоритм, "гибридный" алгоритм, планирование, государственные и муниципальные закупки

Введение

Закупочная деятельность в нашей стране регламентируется двумя основными федеральными законами: 44-ФЗ [1] и 223-ФЗ [2]. С увеличением объема торгов существенно повышается трудоемкость работ по подготовке, проведению, учету и контролю проводимых торгов и заключаемых контрактов. Для повышения эффективности управления госзакупками необходима автоматизация планирования госзакупок с помощью современных информационных технологий.

Составление плана закупок товарно-материальных ценностей вуза начинается со сбора заявок, которые ежегодно подают все его подразделения. Далее заявки делятся на категории: канцтовары, оргтехника, мебель и др. Затем для каждой заявки определяется источник финансирования: федеральный бюджет, студенты-контрактники, программа стратегического развития и др. От этого зависит, под действие какого федерального закона попадает заявка и каким способом она должна быть реализована.

Чтобы сформировать план закупок, работнику контрактной службы необходимо сначала проанализировать поданные заявки и вручную проставить способы закупок для тех заявок, для которых важна закупка каким-то определенным способом. Этот момент можно пояснить на примерах.

Например, закупая канцтовары на аукционе, университет обязан купить канцтовары любой торговой марки у продавца, предложившего минимальную цену. Товары, чьи технические характеристики не важны, могут быть закуплены любым способом, в том числе и на аукционе. Для таких товаров изначально способ закупки не определен.

Если же технические характеристики товара важны (например, закупка серверов), то способ закупки — договор с единственным поставщиком. Заметим, что для таких товаров способ закупки проставляется вручную, и они не попадают в массив исходных данных, который формирует предлагаемый алгоритм.

Далее работник отдела выбирает способ закупки, сумму, которую необходимо набрать, и запускает алгоритм. Вначале алгоритм форми-

рует рабочий массив заявок, из которых будет набираться требуемая сумма. Те заявки, у которых способ закупки не определен, и стоимость не превосходит набираемую сумму, автоматически входят в рабочий массив. Затем алгоритм выбирает из рабочего массива заявки, общая стоимость которых близка к заданной сумме.

Все заявки (слагаемые), участвующие в наборе суммы, равноправны и имеют единичные "веса", характеризующие степень их важности. Как уже было отмечено, наиболее важные закупки с единственным поставщиком изначально не попадают в массивы исходных данных, формируемых алгоритмом.

Для остальных товаров еще при подаче заявки указывается, в каком квартале подразделение желает получить товар. Эти сроки находятся в базе данных и учитываются алгоритмом при формировании исходных массивов. Товары, не вошедшие в набранную сумму, тоже будут закуплены в соответствующем квартале, но другим способом. Для них работник контрактной службы должен выбрать и вручную проставить другой способ закупки. После выбора способа закупки для таких товаров также возможно применение алгоритма.

Особенностью нашей задачи является ее сверхбольшая размерность: сумма в рублях имеет порядок $10^6 \dots 10^7$. В то же время число заявок сравнительно не велико, порядка нескольких сотен.

1. Математическая постановка задачи

Задача о наборе суммы

Условие. Заданы конечное множество A , целые положительные веса всех элементов $a \in A$ и положительные целые числа Sum_0 и D .

Вопрос. Существует ли такое подмножество $A' \subseteq A$ такое, что суммарный вес его элементов Sum удовлетворяет условию $Sum_0 - Sum \leq D$?

Заметим, что частным случаем этой задачи при $D = 0$ является модельная NP-полная задача "Сумма размеров" [3]. Таким образом, наша задача является классической NP-полной задачей. В оптимизационной постановке эту задачу можно сформулировать следующим образом: требуется набрать заданную целую положительную сумму из заданных целых положительных слагаемых так, чтобы не превысить заданную сумму и минимизировать отклонение набранной суммы от исходной.

2. Анализ существующих методов решения задачи о наборе суммы

Рассмотрим вначале точные методы решения данной задачи. Наиболее известным методом решения этой задачи является метод динамического программирования [4]. Известно, что с помощью этого метода задача "Сумма размеров" разрешима за псевдополиномиальное время [3]. Для решения задачи составляется таблица размера $n \times Sum$, где n — число элементов множества A , Sum — набираемая сумма, затем с помощью алгоритма вычислительной сложности $O(n \cdot Sum)$ набирается заданная сумма. Если Sum велико, работа алгоритма не завершится за реальное время.

Еще одним точным алгоритмом решения данной задачи является переборный алгоритм с возвратом [5], который в худшем случае является алгоритмом экспоненциальной вычислительной сложности и эффективно работает только при числе слагаемых $n \leq 20$. Так как рассматриваемая в данной работе задача может иметь сверхбольшую размерность, то для исходной задачи оба этих метода неприменимы.

Рассмотрим приближенные методы, позволяющие получать решение задачи "Сумма размеров" для большой размерности исходных данных. В первую очередь, это метод масштабирования для динамического программирования. Метод заключается в уменьшении всех заданных величин в $scale$ раз, где $scale$ — коэффициент масштабирования, и последующем округлении до целого путем отбрасывания дробной части. Погрешность полученного таким образом решения не превосходит $n \cdot scale$. Применим метод масштабирования к нашей задаче. Для того чтобы добиться существенного выигрыша в быстродействии, необходимо взять $scale$ порядка 1000, что даст недопустимо большую погрешность.

В работе [6] для отгрузки готовой продукции со склада листопрокатного цеха предлагается приближенный алгоритм набора сверхбольшой суммы с известной оценкой абсолютной погрешности. Но из-за специфики задачи в математической постановке есть дополнительные ограничения, кроме того, слагаемые определенным образом сгруппированы.

Таким образом, для задачи набора сверхбольшой суммы из заданных слагаемых с заданным допустимым отклонением возникает необходимость разработки нового эффективного приближенного алгоритма, а также оценка его абсолютной погрешности.

3. Описание "гибридного" алгоритма для задачи о наборе суммы

При первом запуске "гибридного" алгоритма исходный рабочий массив слагаемых упорядочивается по невозрастанию, при последующих запусках — перемешивается случайным образом. Подобные приемы "встряхивания" часто используются в эвристических алгоритмах локального поиска [7], которые имеют тенденцию застревать в точках локального оптимума. Число запусков ограничено константой, не зависящей ни от числа слагаемых, ни от набираемой суммы.

При наборе суммы, отклоняющейся от исходной суммы не более чем на D , алгоритм завершает работу. Полученное таким образом решение будем называть допустимым.

Без детализации работу алгоритма на одном запуске можно описать следующим образом. Если число слагаемых не более 20, то сумма набирается точным алгоритмом. Иначе запускается "жадный" FF-алгоритм (first fit, первый подходящий) [3]. Слагаемые берутся в том порядке, как они расположены в исходном массиве. Если слагаемое проходит ограничение по весу, то оно добавляется к текущему решению, иначе отбрасывается. Когда FF-алгоритм заканчивает работу, любое не вошедшее в решение "новое" слагаемое больше остаточной суммы (которую нужно добрать).

Затем оба массива (с набором текущей суммы и с неиспользованными "новыми" слагаемыми) сортируются в порядке невозрастания. Часть слагаемых в порядке неубывания удаляется из текущего решения и возвращается в рабочий массив. Слагаемые возвращаются в рабочий массив до тех пор, пока или остаточная сумма не станет больше либо равна минимальному "новому" слагаемому, или из текущего решения будут удалены все слагаемые. Затем последнее (максимальное) из перемещенных слагаемых безвозвратно удаляется из рабочего массива, после чего для набора остаточной суммы опять запускается либо точный алгоритм, либо FF-алгоритм.

Заметим, что удаление из рабочего массива на каждой итерации одного слагаемого, во-первых, гарантирует завершение одного запуска "гибридного" алгоритма не более чем за n итераций; во-вторых, позволяет получать новые наборы.

Ниже приведены основные шаги данного алгоритма.

Шаг 1. //новый запуск алгоритма на исходном массиве

Шаг 2. //вызов либо FF-алгоритма, либо точного алгоритма

Шаг 3. //FF-алгоритм

Шаг 4. //удаление слагаемых

Шаг 5. //формируем массив без удаленного слагаемого

Шаг 6. //точный алгоритм

4. Трассировка одного запуска алгоритма на модельном примере

Набираем сумму $Ves_0 = 100$ с допустимым отклонением $D = 0$, рабочий массив $A = [11(10), 8(20), 3, 2(10)]$ (в скобках указано число слагаемых данного веса). Ves — текущая набранная сумма, Sol — текущий набор Ves , Ost — текущая остаточная сумма, $MinA$ — минимальное слагаемое в массиве A .

1. FF-алгоритм (набирает сумму $Ost = 100$):
 $Ves = 99$; $Ost = 1$; $Sol = [11(9)]$; $A = [11, 8(20), 3, 2(10)]$; $MinA > Ost$.

2. Возвращаем 11 из решения в массив A :
 $Ves = 88$; $Ost = 12$; $Sol = [11(8)]$; $A = [11(2), 8(20), 3, 2(10)]$; $MinA < Ost$.

Удаляем 11 из массива A : $A = [11(1), 8(20), 3, 2(10)]$.

3. FF-алгоритм (набирает сумму $Ost = 12$):
 $Ves = 11$; $Ost = 1$; $Sol = [11]$; $A = [8(20), 3, 2(10)]$; $MinA > Ost$.

4. Возвращаем 11 из решения в массив A :
 $Ves = 0$; $Ost = 12$; $Sol = []$; $A = [11, 8(20), 3, 2(10)]$; $MinA < Ost$.

Удаляем 11 из массива A : $A = [8(20), 3, 2(10)]$.

5. FF-алгоритм (набирает сумму $Ost = 12$):
 $Ves = 11$; $Ost = 1$; $Sol = [8, 3]$; $A = [8(19), 2(10)]$; $MinA > Ost$.

6. Возвращаем 3 из решения в массив A :
 $Ves = 8$; $Ost = 4$; $Sol = [8]$; $A = [3, 2(10)]$ (в A входят только те слагаемые, которые не превосходят Ost); $MinA < Ost$.

Удаляем 3 из массива A : $A = [2(10)]$, число слагаемых меньше 20.

7. Точный алгоритм (набирает сумму $Ost = 4$):
 $Ves = 4$; $Ost = 0$; $Sol = [2, 2]$;

8. $OptVes = 100$, $OptSol = [11(8), 8, 2, 2]$.

5. Вычислительная сложность "гибридного" алгоритма

Точный алгоритм с возвратом используется для набора суммы только в том случае, если число слагаемых не превосходит 20. Время его работы не увеличивается с ростом числа сла-

гаемых n или с увеличением исходной набираемой суммы, поэтому это время можно считать константой.

Число запусков "гибридного" алгоритма также ограничено константой и не зависит от размерности исходных данных, поэтому за вычислительную сложность примем число шагов "гибридного" алгоритма на одном запуске. Так как на каждой итерации одного запуска происходит удаление хотя бы одного слагаемого, число итераций не превосходит n .

Внутри одной итерации один раз вызывается FF-алгоритм с вычислительной сложностью $O(n)$, один раз сортируются два массива длины не более n , один раз происходит копирование массива длины не более n . Заметим, что в реальной задаче число слагаемых относительно невелико, поэтому сортировка за время $O(n^2)$ будет работать быстрее бинарной сортировки. Окончательно получаем вычислительную сложность $O(n^3)$.

6. Абсолютная и асимптотическая погрешности "гибридного" алгоритма

Будем называть решением полученный FF-алгоритмом набор элементов $A' \subseteq A$, а суммой Sum решения — сумму стоимостей (весов) входящих в него элементов. Обозначим $MaxTerm$ максимальный вес элемента $a \in A$ индивидуальной задачи I .

Определение. Назовем величину δ абсолютным отклонением набранной приближенным алгоритмом суммы от точного решения:

$$\delta = \begin{cases} Sum0 - D - Sum, & \text{если } Sum < Sum0 - D; \\ 0, & \text{если } Sum0 - D \leq Sum \leq Sum0. \end{cases}$$

Утверждение

1. Для всех индивидуальных задач I о наборе FF-алгоритмом заданной суммы $Sum0$ с заданным допустимым отклонением D и максимальным слагаемым $MaxTerm$ имеет место неравенство:

$$Sum0 - Sum \leq \delta, \text{ где } \delta \leq MaxTerm - D - 1.$$

2. Существует бесконечное число индивидуальных задач I , для которых $Sum0$ сколь угодно велико и $Sum = Sum0 - MaxTerm + D + 1$.

3. $R_{FF}^{\infty} = 1$, асимптотическая погрешность FF-алгоритма равна 1.

Доказательство пунктов 1–3 с помощью техники, описанной в работе [6], приведено

в работе [8]. Таким образом, абсолютная погрешность "гибридного" алгоритма фиксирована и не увеличивается ни при возрастании числа слагаемых, ни при возрастании размера набираемой суммы.

7. Пример работы "гибридного" алгоритма на реальных данных

Для примера возьмем следующие входные данные:

- 1) период — 2016 г.;
- 2) категория товарно-материальных ценностей (ТМЦ) — строительные материалы;
- 3) тип финансирования — бюджет;
- 4) число заявок — 104.

Изначально все заявки имеют способ закупки "Не определен" (рис. 1). Общая стоимость всех заявок по категории — 2 002 986,4 руб.

Выставим сумму, которую требуется набрать (999 999 руб) и допустимое отклонение 100 руб. Алгоритм набирает сумму 999 951,6 руб (рис. 2), отклонение от заданной суммы равно 47,4 руб, что составляет 0,0047 %.

Рассмотрим работу алгоритма более подробно. После запуска формируется рабочий массив A , в котором содержится информация о заявках — идентификационный номер ТМЦ и ее стоимость (вес). Для категории "Строительные материалы" было отобрано 125 заявок. Рассмотрим запуск алгоритма на данных, перемешанных случайным образом. Фрагмент массива приведен в табл. 1. Статус заявки "0" означает, что она не была включена в набор суммы.

Затем происходит набор суммы "жадным" алгоритмом. В результате набрана сумма 998 862 руб, в ее набор вошли 72 заявки. Остаточная сумма равна 1137 руб. Минимальное неиспользованное слагаемое $MinA$, которое осталось в исходном рабочем массиве, равно 1500.

Так как остаточная сумма превышает допустимое отклонение в 100 руб, начинаем возвращать в массив A слагаемые из решения до тех пор, пока остаточная сумма не превысит минимальное слагаемое $MinA$. Таким образом, элементы с весом 160, 168, 175 удаляются из набора и возвращаются в рабочий массив A , в табл. 2 им присвоен статус 1. После возврата остаток составляет 1640.

Затем создаем новый массив $NewA$, в который попадут элементы из A , не превышающие остаточную сумму, которую нужно набрать (1640). В массив попадут 8 элементов (табл. 2).

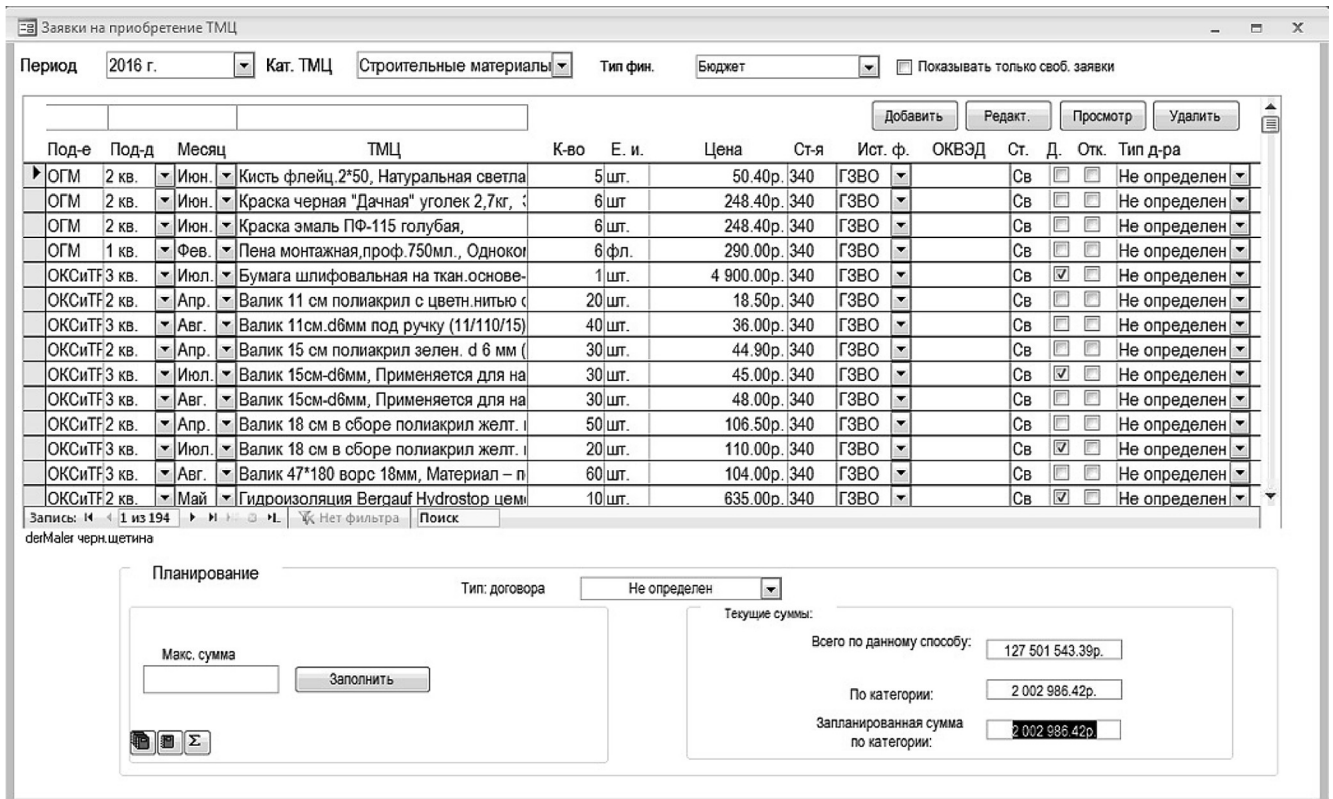


Рис. 1. Входные данные

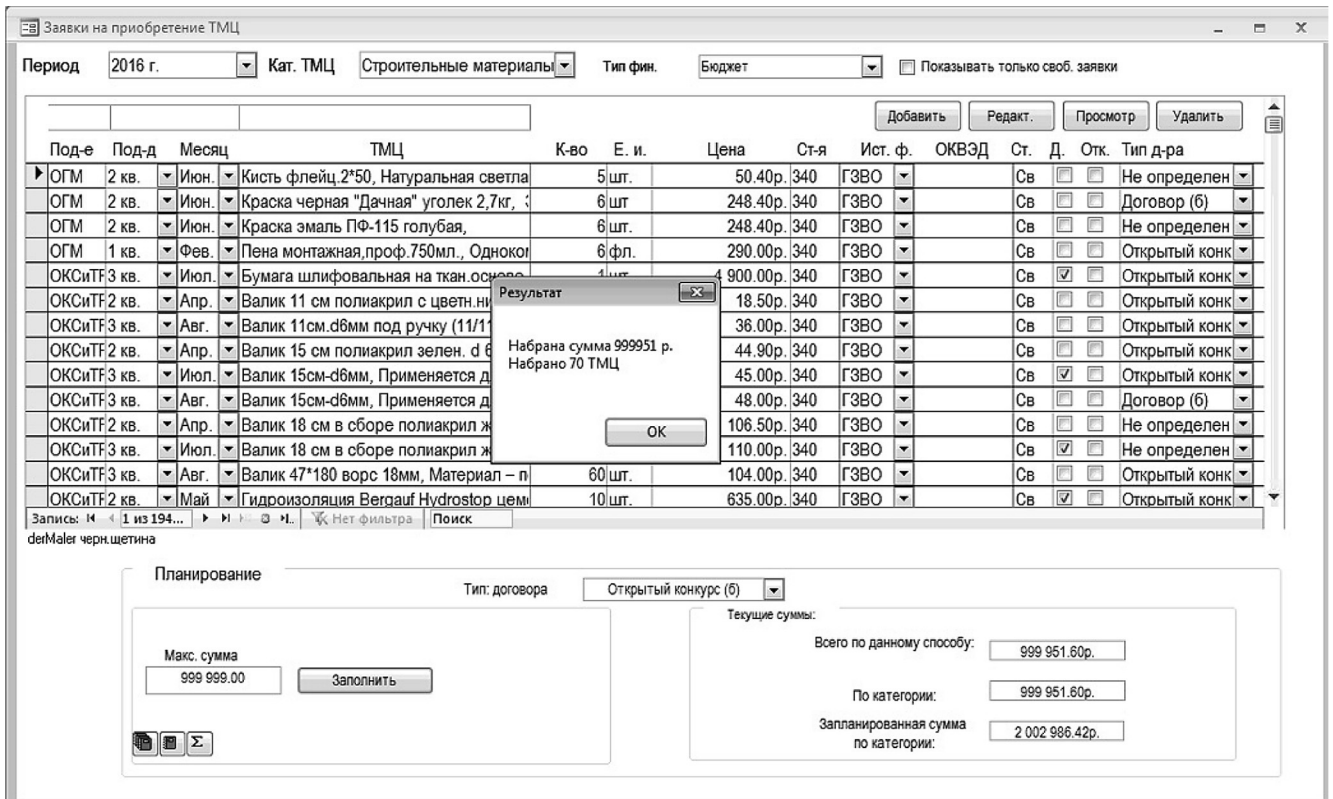


Рис. 2. Результаты работы программы

Таблица 1

Исходный массив *A*

Элемент	Статус	Номер ТМЦ	Индекс	Вес
<i>A</i> [1]	0	31 604	52	1680
<i>A</i> [2]	0	31 601	120	10 3510
<i>A</i> [3]	0	27 900	54	1890
<i>A</i> [4]	0	32 504	43	1500
<i>A</i> [5]	0	31 873	28	1200
...
<i>A</i> [125]	0	32 254	125	316 824

Таблица 2

Массив *NewA*

Элемент	Статус	Номер ТМЦ	Индекс	Вес
<i>NewA</i> [1]	1	31 743	1	160
<i>NewA</i> [2]	1	31 882	2	168
<i>NewA</i> [3]	1	27 886	3	175
<i>NewA</i> [4]	0	32 749	44	1517
<i>NewA</i> [5]	0	28 057	41	1500
<i>NewA</i> [6]	0	31 604	52	1680
<i>NewA</i> [7]	0	27 880	45	1530
<i>NewA</i> [8]	0	32 752	46	1592

Таблица 3

Результаты 10 запусков "гибридного" алгоритма

№	Сумма	Число элементов	Отклонение
1	938 028	118	61971
2	999 910	65	89
3	999 939	98	60
4	999 934	77	65
5	999 908	75	91
6	999 911	67	88
7	999 927	79	72
8	999 941	91	57
9	999 952	70	47
10	999 950	92	49

Так как число слагаемых не превышает 20, для набора остаточной суммы вызывается точный алгоритм. Точный алгоритм набирает сумму 1592 за счет одного элемента *NewA* [8]. Этот элемент добавляется к решению. Таким образом, набрана сумма 999 951 руб., в решение вошли 70 элементов (заявок).

Для получения различных решений и выбора наилучшего варианта запустим "гибридный" алгоритм 10 раз. Возвращаемая на каждом запуске набранная сумма, число слагаемых в наборе и отклонения приведены в табл. 3.

Как видно из табл. 3, на каждом запуске, кроме первого, алгоритм набрал сумму, удовлетворяющую заданному отклонению. Первый запуск алгоритма был проведен в массиве, отсортированном по невозрастанию, остальные — на перемешанных случайным образом.

Заключение

В "самом плохом случае" "гибридный" алгоритм набора заданной суммы наследует абсолютную погрешность "жадного" FF-алгоритма. Но благодаря добору остаточных сумм точным алгоритмом данный алгоритм хорошо работает на реальных данных. Предложенный алгоритм может применяться для решения задач малой, средней и большой размерности.

Список литературы

1. **Федеральный** закон от 05.04.2013 № 44-ФЗ "О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд" // Собрание законодательства РФ. 08.04.2013. № 14. Ст. 1652.
2. **Федеральный** закон от 18.07.2011 № 223-ФЗ "О закупках товаров, работ, услуг отдельными видами юридических лиц" // Собрание законодательства РФ. 25.07.2011. № 30 (ч. 1). Ст. 4571.
3. **Garey M. R., Johnson D. S.** Computers and Intractability: A Guide to the Theory of NP-Completeness. San-Francisco: Freeman, 1979. 347 p. [Гэри М., Джонсон Д. Вычислительные машины и трудноразрешимые задачи. М.: Мир, 1982. 416 с.].
4. **Bellman R.** Dynamic Programming / Princeton, 1957.— 365 p. [Беллман Р. Динамическое программирование. М.: Иностранная литература, 1960. 400 с.].
5. **Lipski W.** Kombinatoryka dla programistów / 2 wyd. Warszawa: WNT, 1989. 188 p. [Липский В. Комбинаторика для программиста. М.: Мир, 1988. 200 с.].
6. **Девятов Д. Х., Файнштейн С. И., Тутарова В. Д., Калитаев А. Н.** Оперативное планирование отгрузки готовой продукции со складов металлургических предприятий // Мехатроника, автоматизация, управление. 2008. № 4. С. 36—40.
7. **Michalewicz Z., Fogel D.** How to Solve It: Modern Heuristics. Springer, Berlin, 2004. 554 p.
8. **Торчинская И. В., Файнштейн С. И.** Приближенный алгоритм набора заданной суммы с заданным отклонением // Межотраслевой институт Наука и образование. 2014. № 1. С. 84—86.

V. E. Torchinskiy, Assistant Professor, e-mail: vet@magtu.ru,
I. V. Torchinskaya, Leading Software Engineer of Department ACS of DIT&ACS., e-mail: asu@magtu.ru,
A. S. Fainshtein, Candidate of physical and mathematical Sciences, Assistant Professor, e-mail: swetlana@mgn.ru,
S. I. Fainshtein, Assistant Professor, e-mail: sfainshtein@yandex.ru,
Magnitogorsk state technical University named after G. I. Nosov

The "Hybrid" Algorithm of Planning Municipal Procurement of the Material Assets

In this work, for planning state procurement of goods-material values, the proposed "hybrid" algorithm sets the given amount of the given values (NP-complete weighted set problem). The sum is collected by a fast "greedy" algorithm until the size of the subproblem becomes suitable for solving by an exact algorithm. This "hybrid" algorithm inherits the computational complexity and the absolute worst-case performance of the "greedy" algorithm. At the same time the use of the exact algorithm at the last stage of the solution allows to gain the sum with a high degree of precision. This algorithm is used in the planning of state procurement of MSTU named after G. I. Nosov.

Keywords: NP-complete, weighted set problem, approximate algorithm, "hybrid" algorithm, planning, state and municipal procurements

DOI: 10.17587/it.25.234-240

References

1. **Federal'nyj zakon** ot 18.07.2011 № 44-FZ "O kontraktnoj sisteme v sfere zakupok tovarov, rabot, uslug dlja obespechenija gosudarstvennykh i munitsepal'nykh nuzhd" (About contract system in the sphere of purchases of goods, works, services for ensuring the state and municipal needs), *Sobranie zakonodatel'stva RF*, 08.04.2013, no. 14, art. 1652. (in Russian).
2. **Federal'nyj zakon** ot 05.04.2013 № 223-FZ "O zakupkakh tovarov, rabot, uslug odel'nymi vidami juridicheskikh lits" (About purchases of goods, works, services by separate types of legal entities), *Sobranie zakonodatel'stva RF*, 25.07.2011, no. 30 (part 1), art. 4571. (in Russian).
3. **Garey M. R., Johnson D. S.** Computers and Intractability: A Guide to the Theory of NP-Completeness, San-Francisco, Freeman, 1979, 347 p.
4. **Bellman R.** Dynamic Programming, Princeton, 1957, 365 p.
5. **Lipski W.** Kombinatoryka dla programistów, 2 wyd., Warszawa, WNT, 1989, 188 p. (in Russian).
6. **Devjatov D. Kh., Fainshtein S. I., Tutarova V. D., Kalitaev A. N.** Operativnoe planirovanie otgruzki gotovoj produktsii so skladov metallurgicheskikh predpriyatij (Operational planning of shipment of finished products from warehouses of metallurgical enterprises), *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2008, no. 4, pp. 36–40 (in Russian).
7. **Michalewicz Z., Fogel D.** How to Solve It: Modern Heuristics, Springer, Berlin, 2004, 554 p.
8. **Torchinskaya I. V., Fainshtein S. I.** Absolute and asymptotic performance for the approximation algorithm finding predetermined sum with a predetermined deviation, *Mezhotraslevoj institut Nauka i obrazovanie*, 2014, no. 1, pp. 84–86 (in Russian).

3–7 июня 2019 года, Санкт-Петербургский политехнический университет Петра Великого (СПбПУ), Санкт-Петербург, Россия

21–25 октября 2019 года, Белорусский национальный технический университет (БНТУ), Минск, Беларусь

30 октября — 1 ноября 2019 года, Саратовский государственный технический университет имени Гагарина Ю. А., Саратов, Россия

XXXII Международная научная конференция

"МАТЕМАТИЧЕСКИЕ МЕТОДЫ В ТЕХНИКЕ И ТЕХНОЛОГИЯХ — ММТТ-32"

1. Качественные и численные методы исследования дифференциальных и интегральных уравнений.
 2. Оптимизация, автоматизация и оптимальное управление технологическими процессами.
 3. Математическое моделирование технологических и социальных процессов.
 4. Математическое моделирование и оптимизация в задачах САПР, аддитивных технологий.
 5. Математические методы в задачах радиотехники, радиоэлектроники и телекоммуникаций, геоинформатики, авионики и космонавтики.
 6. Математические методы и интеллектуальные системы в робототехнике и мехатронике.
 7. Математические методы в медицине, биотехнологии и экологии.
 8. Математические методы в экономике и гуманитарных науках.
 9. Информационные и интеллектуальные технологии в технике и образовании.
 10. Математические и инструментальные методы технологий Индустрии 4.0.
 11. Обсуждение квалификационных работ.
- Школа молодых ученых — ШМУ. Конкурс УМНИК.

Подробная информация о конференции и условиях участия в ней размещена на сайте <http://mmtt.sstu.ru/>

С. Н. Полесский, канд. техн. наук, доц., e-mail: spolessky@hse.ru,
А. В. Востриков, канд. техн. наук, доц., e-mail: avostrikov@hse.ru,
Е. Н. Прокофьева, канд. техн. наук, доц., e-mail: eprokofyeva@hse.ru,
И. И. Романова, ассистент, e-mail: iromanova@hse.ru,

Национальный исследовательский университет "Высшая школа экономики"

Методика оценки надежности вычислительной техники с учетом влияния отказов вспомогательного оборудования

Традиционно для учета влияния вспомогательных элементов на функционирование основных элементов объекта используется один из двух подходов: либо отказ вспомогательных элементов приводит к отказу всего объекта, либо отказ вспомогательных элементов не учитывается. Первая методика значительно занижает показатели надежности, а вторая, соответственно, их завышает. Методика, предложенная в данной статье, учитывает влияние, оказываемое отказом вспомогательных элементов на условия функционирования основных, что позволяет более точно оценить показатели надежности. В качестве объекта исследования выбран типовой модуль оборудования серверного помещения среднестатистической организации, на основе рассмотрения состава которого были сформированы критерии отказа и построены структурные схемы надежности, демонстрирующие преимущества предложенного подхода.

Ключевые слова: серверное помещение, заземление, вспомогательные элементы, основные элементы, интенсивность отказов, критерии отказов, коэффициент готовности, средняя наработка на отказ

Введение

Развитие современной электроники и информационных технологий позволило обеспечить доступность высокоскоростной вычислительной техники практически во всех уголках земного шара. Основной целью создания серверного помещения является удобное и эффективное расположение всех компонентов в одном месте, чтобы у администратора был централизованный, комфортный и быстрый доступ ко всем комплектующим оборудования для быстрого устранения неполадок в случае возникновения проблем. Также важна защита оборудования от сбоев питания и неблагоприятных климатических условий. Для обеспечения этих условий используется комплекс вспомогательных элементов (ВЭ) [1–3]:

— для поддержания в помещении комфортных климатических условий используется система кондиционирования — сплит-система, которая при активном функционировании серверного оборудования позволяет как охладить, так и нагревать воздух до нужной температуры;

— для отвода накапливаемого статического электричества, которое может привести к сбоям в работе аппаратуры, а также нанести вред человеку при прикосновении к ней, используются комплекты заземлений;

— для освещения в помещении используются галогенные лампы, благодаря которым снижается нагрузка в сети питания и уменьшаются электрические помехи.

Традиционно при проведении расчета показателей надежности технических средств ВЭ принимаются эквивалентными основным элементам (ОЭ) [1, 4, 5], таким образом, критерии отказов ВЭ принимаются за отказ всей системы, что явно значительно занижает результаты расчета [5, 6]. В других случаях ВЭ вовсе не учитываются, что, в свою очередь, наоборот завышает прогнозируемые показатели [7, 8]. Оба этих подхода могут приводить к возрастанию погрешности более чем на 20 % [5]. В связи с этим появляется необходимость разработки методики, которая позволит учитывать условное (нелинейное) влияние ВЭ на функционирование системы с точки зрения надежности и значительно уменьшать погрешность расчетов.

Описание методики проведения расчета показателей надежности

Сформулируем методику проведения расчета показателей надежности вычислительной техники серверного помещения с учетом влияния отказов сервисного оборудования:

1. Формулировка требований к объекту исследования (размеры помещения, наличие розеток, нагрузка на пол, освещение помещения, температура помещения, давление воздуха и т. д.).

2. Выделение основных компонентов объекта исследования (серверное оборудование, сплит-система, розетка, лампа освещения и т. д.).

3. Описание состава комплектов оборудования компонентов объекта исследования.

4. Учет основных требований к компонентам объекта исследования, критически важных для функционирования системы (например, тепловой мощности), и формирование итогового состава оборудования.

5. Определение нормируемых показателей надежности, которые задаются для объекта исследования (коэффициент готовности, средняя наработка на отказ, среднее время восстановления и др.).

6. Формирование окончательных критериев отказа системы.

7. Анализ возможных причин отказа системы и разработка окончательной логической схемы функционирования объекта исследования и RBD-схемы (Reliability Block Diagram) на ее основе.

8. Построение структурных схем надежности для прогнозирования заданных показателей надежности объекта исследования и его составных частей.

9. Выбор математических моделей расчета показателей надежности.

10. Расчет показателей надежности объекта исследования.

Как уже было сказано выше, для повышения точности расчетов на этапе 7 разработанной методики следует учитывать условное влияние ВЭ на функционирование системы. Для этого нами предлагается использовать диаграммы Исикавы, что позволит более точно сформировать критерии отказа системы и, в итоге, повысить точность расчетов.

Рассмотрим применение данной методики на примере типового серверного помещения.

Описание объекта исследования

В качестве объекта исследования рассмотрим наиболее распространенный вариант построения серверного помещения. Серверная — это помещение специального назначения, в котором находится телекоммуникационная аппаратура, которое соединяется с магистралями и считается средством обслуживания

здания. В соответствии со стандартом [1—3, 9] серверное помещение должно отвечать следующим требованиям:

1. Наличие не менее двух электрических розеток с заземлением на каждые три погонных метра любой стены. Серверное помещение имеет размеры 3×4 м (что в соответствии со стандартом является минимальными 12 м^2 , минимальная высота потолка $2,44$ м) и в соответствии со стандартом в ней должно иметься четыре розетки с заземлением, в нашем примере их 6 шт.

2. Максимальная распределенная нагрузка на пол должна составлять 12 кН/м^2 , максимальная сосредоточенная нагрузка — $4,4 \text{ кН/м}^2$.

3. При освещении помещения для снижения электрических помех рекомендуется использовать галогенные лампы, так как они снижают скачки напряжения в электрической сети.

4. Серверное помещение рекомендуется располагать вдали от источников электромагнитного излучения (в местах, где возможно разместить крупногабаритное оборудование, функциональные передатчики, радиооборудование, оборудование для спутниковой связи, теле- и радиостанции).

5. Сплит-система должна поддерживать температуру окружающей среды в диапазоне от $+18$ до $+24$ °C и относительной влажности $30...55$ %.

6. Давление воздуха в серверной должно превышать давление воздуха в соседних помещениях не менее чем на $14,7$ Па, за счет чего в комнату не попадает пыль.

На рис. 1 цифрами отмечены следующие компоненты серверного помещения:

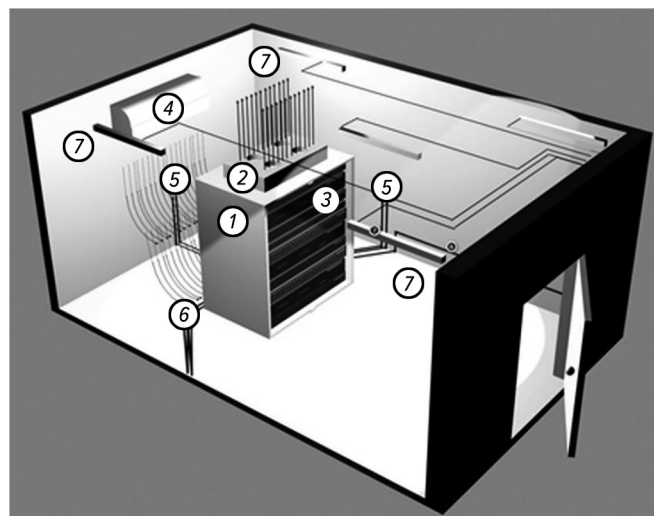


Рис. 1. Схематическое расположение компонентов в помещении

1 — телекоммуникационный шкаф; 2 — вентиляторный блок в крышу телекоммуникационного шкафа; 3 — серверное оборудование (подробное описание приведено ниже); 4 — сплит-система; 5 — двойная розетка; 6 — кабель защитного заземления; 7 — люминесцентная лампа.

С учетом требований стандартов [1–3] телекоммуникационные шкафы подлежат заземлению как металлические конструкции, которые при пробое изоляции токоведущих частей могут оказаться под напряжением. Для безопасного заземления используется медная шина заземления 19" дюймов и комплект проводов заземления для телекоммуникационного шкафа.

Состав заземления серверной стойки согласно [2]:

1. Комплект проводов заземления для телекоммуникационного шкафа [9], состоящий из набора соединительных проводников с крепежом для электрического заземления элементов конструкции шкафа. Каждый конструктивный элемент шкафа (каркас, боковые панели, дверь) имеет специальный заземляющий контакт.

2. Шина заземления предназначена для централизованного подключения 19" заземляющих проводников оборудования, устанавливаемого в шкафах и стойках. Шина выполнена из медной никелированной полосы толщиной 3,2 мм и шириной 25,4 мм и имеет 11 крепежных штырей, укомплектованных шайбами, гроверами и гайками.

Провод соединяется посредством болтового соединения с существующей в здании телекоммуникационной шиной заземления. Категорически запрещается соединение нескольких шкафов одним заземляющим проводником последовательно, так как это чревато разрывом линии заземления и опасно для жизни человека.

В качестве примера рассмотрим вариант размещения в телекоммуникационном шкафу трех одинаковых серверов (рис. 2). Сервер № 1 используется в качестве веб-сервера, на котором расположена серверная компонента приложения с веб-интерфейсом, база данных приложения и каталог с файлами, поступившими на обработку. Данный сервер имеет два жестких диска, объединенных с использованием технологии RAID 1 для высокой производительности и надежности, так как в случае утери информация, поступающая с многочисленных клиентских АРМ, будет трудно восстановима. Сервер № 2 является архивным, в него попада-

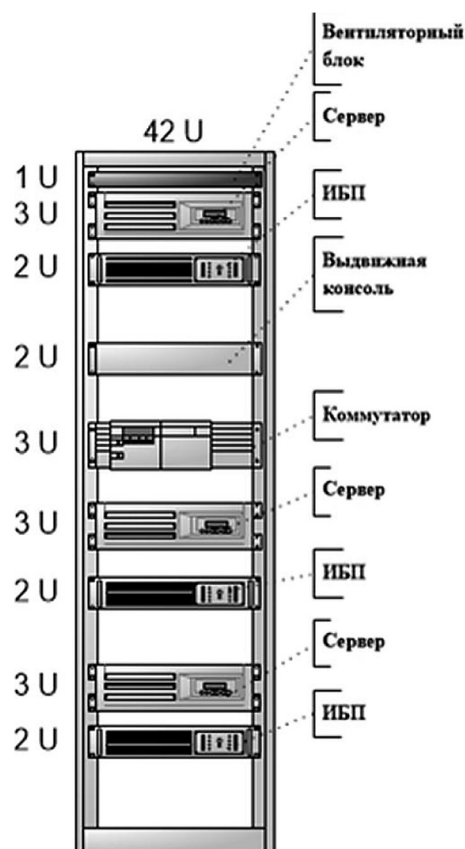


Рис. 2. Состав серверной стойки

ет информация, обработанная сервером № 1. Он также имеет RAID 1, состоящий из двух жестких дисков. Сервер № 3 является сетевым хранилищем компании, на нем находятся виртуальные стенды, дистрибутивы и документация компании, серверное веб-приложение для управления проектами и задачами, данный сервер также защищен RAID 1.

Для защиты серверного оборудования от сбоев питания используется источник бесперебойного питания (ИБП) мощностью 3 кВа.

Для организации сети используется управляемый коммутатор.

Для управления и мониторинга функционирования компонентов серверной стойки используется выдвижная KVM-консоль в стойку 19", 2U.

Для поддержания длительной бесперебойной работы телекоммуникационного оборудования применяются средства обеспечения благоприятных климатических условий. Рекомендуемая температура окружающей среды для работы компонентов сервера выбирается из диапазона + 20...+ 25 °С. При выборе сплит-системы, которая будет обеспечивать подобные условия,

в первую очередь необходимо учитывать общую тепловую нагрузку в серверном помещении.

Тепловую мощность традиционно измеряют в ваттах (Вт), но мы будем использовать БТЕ (Британская термическая единица), поскольку в паспортах на покупные изделия иностранного производства тепловая мощность указана в БТЕ. При приближенном расчете тепловая мощность в 1 Вт будет составлять 3,41 БТЕ /ч [9]. Таким образом, чтобы рассчитать общую тепловую нагрузку всего серверного помещения, необходимо знать тепловую энергию каждой составляющей (табл. 1).

Таблица 1

Тепловая мощность комплектующих серверного помещения

№ п/п	Наименование элемента	Тепловая мощность, БТЕ	Примечания
1	Сервер	3410	на 1 шт.;
2	Коммутатор	46,1714	на 1 шт.;
3	Лампа галогенная	341	на 1 шт.;
4	Системный администратор	300,08	на 1 чел.

Таблица 2

Состав типовой стойки серверного помещения

№ п/п	Наименование элемента	Количество
1	Розетка электрическая	6 шт.
2	ИБП	3 шт.
3	Коммутатор	1 шт.
4	Вентиляторный блок	1 шт.
5	Материнская плата	3 шт.
6	Процессор	3 шт.
7	Карта памяти	3 шт.
8	Жесткий диск	6 шт.
9	DVD-RW привод	3 шт.
10	ИВЭП*	3 шт.
11	Сетевой кабель ftp5e	5 м
12	Кабель питания	9 м
13	Выдвижная консоль	1 шт.
14	Лампа галогенная	5 шт.
15	Сплит-система	1 шт.
16	Комплект заземления	1 шт.

*ИВЭП — источник вторичного электропитания.

Так как используемый телекоммуникационный шкаф имеет вентиляторный блок (рис. 2), то при расчете тепловой нагрузки мощностью тепловыделения оборудования в шкафу можно пренебречь.

Серверное помещение не имеет окон, находится на теневой стороне здания на первом этаже, поэтому прочей тепловой нагрузкой также можно пренебречь. В итоге рассчитанная общая тепловая нагрузка в соответствии с составом серверной будет вычисляться по формуле

$$\begin{aligned}
 P_{\text{общ}} &= 3P_1 + P_2 + 5P_3 + 2P_4 = \\
 &= 3 \cdot 3410 + 46,1714 + 5 \cdot 341 + 2 \cdot 300,08 = \\
 &= 12\,581,3314 \text{ БТЕ},
 \end{aligned}$$

где P_i — тепловая мощность i -й составляющей согласно табл. 1. При пересчете общей тепловой мощности в Вт полученное значение составит около 3,7 кВт. Рекомендуемая мощность охлаждения сплит-системы при данной нагрузке составляет 3,5 кВт. На основании данных работы [1] идеальная влажность для серверного помещения 40...50 % (при низкой влажности возникают проблемы с электростатическими зарядами, при высокой влажности — окисление контактов и замыкание). Для регулирования влажности в серверной сплит-система должна иметь функцию осушения воздуха. Исходя из полученных показателей рекомендуется использовать сплит-систему, например компании "Daikin", которая имеет инвертор, что обеспечивает ее надежность, охлаждение 3,8 кВт, обогрев 4,8 кВт, а также имеется программная осушка воздуха, что позволяет бороться как с избыточной, так и с недостаточной влажностью воздуха.

Таким образом, на основании вышеприведенного анализа получаем следующий итоговый состав оборудования (табл. 2).

Показатели надежности и критерии отказов объекта

В первую очередь необходимо определить, какие из нормируемых показателей надежности будут задаваться для таких систем. На основании описаний из нормативной документации [10—12] регламентируется порядок выбора показателей надежности. Исходя из того, что оборудование серверного помещения относится к изделиям общего назначения, циклического применения, восстанавливаемым и обслуживаемым, задаются следующие показатели:

- коэффициент готовности (K_T);
- средняя наработка на отказ (T_0) или среднее время восстановления (τ_B).

Рассмотрим три возможных варианта для проведения расчета показателей надежности (табл. 3):

- первый случай — традиционный, когда ВЭ не учитывается (принимается абсолютно надежным);
- второй случай — традиционный, в котором, в отличие от первого случая, ВЭ оказывает влияние на показатели надежности всей системы, как и ОЭ;
- третий случай — инновационный, при котором учитывается условное влияние ВЭ на функционирование ОЭ.

На основе итогового состава (табл. 2), полного описания функционирования, а также диаграммы Исикавы формируем критерии отказа системы, основываясь на работе [14].

К отказу системы приводит отказ ОЭ, в том числе: отказ резервной группы (отказ четырех

и более розеток); отказ любого из ИБП; отказ коммутатора; отказ вентиляционного блока; отказ любого из серверов; отказ выдвижной консоли; отказ любого из кабелей питания.

Критерии отказа сервера: отказ материнской платы; отказ процессора; отказ обоих жестких дисков; отказ DVD-RW привода; отказ ИВЭП; отказ сетевого кабеля; отказ кабеля питания.

Таким образом, на основании критериев отказа формируются общие схемы функционирования стойки серверного помещения для всех трех случаев, описанных выше, где в момент времени $t = \tau$ наступает отказ ВЭ. Пункты 1 и 2 (табл. 3) иллюстрируют традиционный подход к решению данной задачи. При применении описываемой методики ОЭ и ВЭ не являются соединенными последовательно, и отказ ВЭ не приводит к отказу ОЭ. Однако при отказе ВЭ ухудшатся условия эксплуатации ОЭ, что впоследствии приводит к возрастанию вероятности его отказа.

Таблица 3

RBD-схемы, составленные на основе логических моделей

№	Логические модели	RBD-схемы
1		
2		
3		

Примечание: RBD-схемы для всех случаев построены на основании работ [11, 15–17]. В логических моделях черным цветом отмечены блоки, которые находятся в состоянии отказа, серым цветом — в состоянии ухудшения условий эксплуатации (экстремальные условия), белым цветом — в состоянии работоспособности.

Структурные схемы надежности серверной стойки

Для прогнозирования заданных показателей надежности вышеприведенного объекта необходимо построение структурной схемы надежности (ССН). Далее на основании [13] формируется ССН всей системы и ее составных частей (СЧ).

На основании критериев отказа, а также итогового состава стойки серверного помещения, приведенного в табл. 2, строится ССН ОЭ, приведенная на рис. 3, состоящая из последовательно соединенных группы нагруженного резерва из трех основных и трех резервных элементов, двух групп, содержащих по три последовательно соединенных элемента, одной группы из двух последовательно соединенных элементов, а также двух нерезервированных элементов [11, 15, 17–20].

На рис. 4 подробно представлена ССН сервера, состоящая из последовательно соединенных семи элементов и однократного нагруженного резерва.

На рис. 5 более подробно показана ССН ВЭ, в состав которой входят последовательно соединенные два элемента, одна группа нагруженного резерва из одного основного и четырех резервных элементов и одна группа нагруженного резерва из двух основных и двух резервных элементов.

Математические модели расчета показателей надежности

В качестве требуемых показателей надежности серверного помещения в нашем случае заданы коэффициент готовности и средняя

наработка на отказ [10, 12]. В табл. 4 представлены математические модели расчета показателей всего объекта для каждого случая.

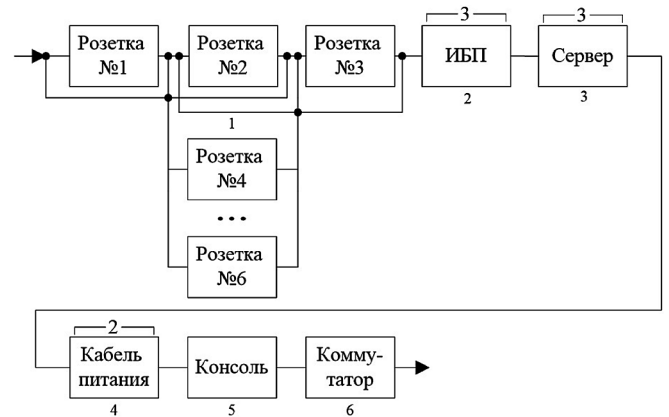


Рис. 3. ССН СЧ ОЭ

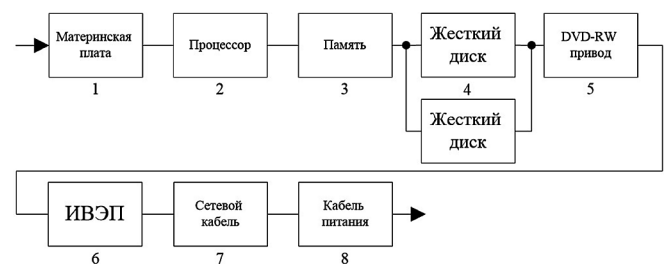


Рис. 4. ССН СЧ сервера

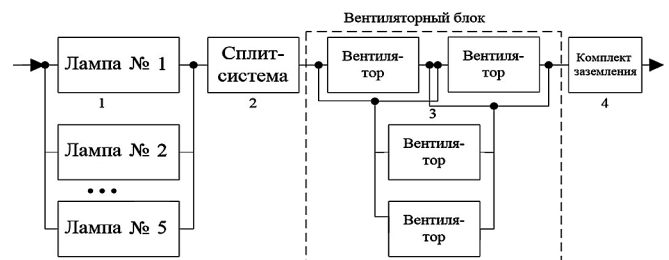


Рис. 5. ССН СЧ ВЭ

Таблица 4

Математические модели расчета ПН

Номер схемы	Математические модели (источник)	
	K_r	T_0
1	$K_r = K_{rOЭ}$ [5]	$T_0 = \int_0^{\infty} e^{-\frac{t}{T_{0OЭ}}} dt$ [19]
2	$K_r = K_{rOЭ} K_{rBЭ}$ [14, 21]	$T_0 = \int_0^{\infty} e^{-t \left(\frac{1}{T_{0OЭ}} + \frac{1}{T_{0BЭ}} \right)} dt$ [4, 11]
3	$K_r = \frac{T_{0OЭ}}{T_{0OЭ} + \tau_{BЭ}}$ [7, 16]	$T_0 = \int_0^{\infty} \left[e^{-t \left(\frac{1}{T_{0OЭ}} + \frac{1}{T_{0BЭ}} \right)} + \frac{t}{T_{0OЭ}} \int_0^t e^{-\tau \left(\frac{1}{T_{0BЭ}} + \frac{1}{T_{0OЭ}} + \frac{1}{T_{0OЭ2}} \right)} d\tau \right] dt$ [7, 18]

В табл. 4 используются следующие обозначения:

- $K_{ГОЭ}$ — коэффициент готовности ОЭ;
- $K_{ГВЭ}$ — коэффициент готовности ВЭ;
- $T_{0ОЭ}$ — средняя наработка на отказ ОЭ при $t_{окр} = 25\text{ }^\circ\text{C}$;
- $T_{0ВЭ}$ — средняя наработка на отказ ВЭ;
- $\tau_{вОЭ}$ — среднее время восстановления ОЭ;
- $T_{0ОЭ2}$ — средняя наработка на отказ ОЭ при $t_{окр} = 50\text{ }^\circ\text{C}$.

Согласно спецификации и ССН оборудования серверной стойки были идентифицированы такие параметры надежности элементов, как интенсивность отказа (ИО) СЧ ОЭ при $t_{окр} = 25\text{ }^\circ\text{C}$, полученные с сайтов производителей элементов или их аналогов, а также пересчитаны ИО СЧ ОЭ при $t_{окр} = 40\text{ }^\circ\text{C}$ по формуле

$$\lambda_{50\text{ }^\circ\text{C}} = \lambda_{25\text{ }^\circ\text{C}} e^{\left[\frac{E_a}{k} \left(\frac{1}{t_{окр2}} - \frac{1}{t_{окр1}} \right) \right]},$$

где E_a — энергия активации (эВ);

k — постоянная Больцмана, равная $8,61733035 \cdot 10^{-5}$ (эВ·К);

$t_{окр1}$ — температура окружающей среды в серверном помещении до отказа ВЭ ($25\text{ }^\circ\text{C}$);

$t_{окр2}$ — температура окружающей среды в серверном помещении после отказа ВЭ ($40\text{ }^\circ\text{C}$).

Кроме того, с использованием экспертного метода, подробное описание которого приведено в работах [4, 7, 11, 21, 22], было определено среднее время восстановления всех СЧ.

Результаты расчета

При использовании исходных данных (табл. 5) и математических моделей (см. табл. 4) были рассчитаны коэффициент готовности и средняя наработка на отказ рассматриваемого объекта для всех случаев трех случаев (см.

Таблица 5

Значения ИО и времени восстановления СЧ объекта

№ п/п	Наименование элемента расчета ССН	$\lambda \cdot 10^6, 1/\text{ч}$		$\tau_{в}, \text{ч}$	Примечания
		$t_{окр1}$	$t_{окр2}$		
ОЭ					
1	Розетка электрическая	5,0	55,0	0,5	на 1 шт.
2	ИБП	12,5	137,5	0,5	на 1 шт.
3	Коммутатор	1,3535682767	14,889251044	0,5	на 1 шт.
4	Вентилятор	19,02587519	209,28462709	0,5	на 1 шт.
5	Материнская плата	9,2082727122	101,290999834	1,0	на 1 шт.
6	Процессор	2,2357735519	24,593509071	0,5	на 1 шт.
7	Карта памяти	0,3234399521	3,557839473	0,5	на 1 шт.
8	Жесткий диск	0,5	5,5	0,5	на 1 шт.
9	DVD-RW привод	28,571428571	314,285714281	0,5	на 1 шт.
10	ИВЭП	10,0	110,0	0,5	на 1 шт.
11	Сетевой кабель	0,7610350076	8,371385084	0,2	на 1 м
12	Кабель питания	38,0	418,0	0,2	на 1,8 м
ВЭ					
13	Выдвижная консоль	11,4155251	—	0,5	на 1 шт.
14	Лампа галогенная	500,0	—	0,1	на 1 шт.
15	Сплит-система	100,0	—	1,0	на 1 шт.
18	Комплект заземления	5,96	—	0,5	на 1 шт.

Примечание: $\tau_{в}$ взято из опыта эксплуатации и ремонта аналогичных изделий.

Таблица 6

Результаты расчетов

Номер схемы	K_r	T_0	δ , %	Примечания
1	0,9998471461	2541	19,802	Погрешности самих математических моделей не учтены
2	0,9997441422	2000	5,705	
3	0,9998169439	2121	—	

табл. 3), а также была проведена оценка относительной погрешности расчетов (табл. 6).

Заключение

Проанализировав полученные численные значения показателей надежности стойки серверного помещения (табл. 6), можно сделать вывод о том, что случай, учитывающий влияние отказа ВЭ на отказ ОЭ, позволяет наиболее прецизионно подходить к оценке коэффициента готовности и средней наработки на отказ. Причиной этого является, во-первых, отсутствие учета влияния ВЭ при рассмотрении первого случая (RBD-схема № 1 (см. табл. 3), относительная погрешность — 19,802 %), что приводит к сильно завышенному значению наработки на отказ, и, во-вторых, достаточно грубое предположение, что отказ ВЭ ведет к непосредственному отказу всей системы (RBD-схема № 2 (см. табл. 3), относительная погрешность — 5,705 %). На деле отказ ВЭ приводит только к ухудшению условий функционирования ОЭ, что хотя и сокращает его наработку, но не является причиной отказа. Третий случай (RBD-схема № 3 (табл. 3)) позволяет преодолеть недостатки двух предыдущих рассмотренных случаев, т. е. показать зависимость отказа ОЭ от отказа ВЭ. Проведенный расчет наработки на отказ для третьего случая показал, что ее значение попадет в границы, заданные двумя предыдущими случаями: верхней границей является первый случай (RBD-схема № 1), а нижней — второй (RBD-схема № 2). Причины этого вполне очевидны и описаны ранее в статье. Это свидетельствует о том, что предлагаемая методика учета влияния отказа ВЭ позволяет дать наиболее точную оценку заданным показателям надежности исследуемого объекта. Оцененные погрешности показывают, насколько полученные значения наработки на отказ первых двух случаев (RBD-схемы № 1 и № 2) отличаются от значения для третьего случая. Исходя из этого

можно сделать вывод, что необходимо учитывать ВЭ при оценке показателей надежности объектов (иначе это приводит к завышенным результатам), но излишне прямолинейный учет приводит к занижению результатов.

Список литературы

1. Стандарт ANSI/TIA/EIA-569-A. Телекоммуникационные трассы и помещения коммерческих зданий, 1998.
2. **Электроустановок** правила устройства, изд. 7. М.: НГС "ЭНАС", 2003. 168 с.
3. Стандарт ANSI/TIA/EIA-607. Требования к телекоммуникационной системе выравнивания потенциалов и заземления коммерческих зданий, 1994.
4. **Beichelt F.** Reliability and Maintenance: Networks and Systems. Chapman and Hall / CRC, 2012. 344 p.
5. **Жаднов В. В., Полесский С. Н.** Определение показателей надежности систем, обладающих вспомогательными элементами. Сборник научных трудов "Проектирование телекоммуникационных и информационных средств и систем". М.: МИЭМ, 2006. С. 151–158.
6. **Garcia A. C., Prijevor L. P.** Reliability Modeling Prediction. DQM Research Centre, 2016. 186 p.
7. **Черкесов Г. Н.** Надежность аппаратно-программных комплексов. Учебное пособие. СПб.: Питер, 2005. 479 с.
8. **Жаднов В. В., Тихменев А. Н.** Имитационное моделирование в задачах оценки надежности отказоустойчивых электронных средств // Надежность. № 1. 2013. С. 32–54.
9. Стандарт JIS Z 8101–99. Статистика. Словарь и условные обозначения, 1999.
10. ГОСТ 27.003–2016 Надежность в технике (ССНТ). Состав и общие правила задания требований по надежности. 2016.
11. **MIL-HDBK-338B.** Electronic reliability design handbook, 1998.
12. ГОСТ 27.002–2015 Надежность в технике (ССНТ). Термины и определения. 2015
13. **Метод "Диаграмма Исикавы"** [Электронный ресурс]. — Режим доступа: <http://www.inventech.ru/pub/methods/metod-0019/> (дата обращения: 29.09.2018).
14. **Похабов Ю. П.** Обеспечение надежности уникальных высокоответственных систем. Надежность. № 24 (3). 2017. С. 17–23.
15. ГОСТ Р 51901.14–2007 (МЭК 61078:2006). Менеджмент риска. Структурная схема надежности и булевы методы, 2007.
16. **Băjenescu T., Băzu M.** Component Reliability for Electronic Systems. London: Artech House, 2010. 354 p.
17. **Swingler J.** Reliability Characterisation of Electrical and Electronic Systems, 1st Edition. UK: Woodhead Publishing, 2015. 274 p.
18. **Половко А. М., Гуров С. В.** Основы теории надежности. СПб.: БХВ—Петербург, 2006. 702 с.
19. **Ушаков И. А.** Курс теории надежности систем. Учебное пособие. М.: Дрофа, 2008. 239 с.
20. **James Li.** Reliability calculation of a parallel redundant system with different failure rate & repair rate using Markov modeling // Journal of Reliability and Statistical Studies. Vol. 9. Issue 1. 2016. P. 1–10.
21. **O'Connor P., Kleyner A.** Practical Reliability Engineering, 5th Edition. Wiley, 2012. 512 p.
22. **Diveev A. I., Sofronova E. A.** The Network Operator Method for Search of the Most Suitable Mathematical Equation // Bio-Inspired Computational Algorithms and Their Applications. Croatia: Intech, 2012. P. 19–42.

S. N. Polesskij, Ph. D., associate professor, e-mail: spolessky@hse.ru,
A. V. Vostrikov, Ph. D., associate professor, e-mail: avostrikov@hse.ru,
E. N. Prokof'eva, Ph. D., associate professor, e-mail: eprokofyeva@hse.ru,
I. I. Romanova, assistant professor, e-mail: iromanova@hse.ru,
National Research University "Higher School of Economics"

Methods of Assessing the Reliability of Computing with Due Account for the Failure Effect of Auxiliary Equipment

Traditionally, when taking into account the influence of auxiliary elements on the operation of main elements of an object, one of two approaches is used: either the failure of auxiliary elements leads to the failure of the entire object, or the failure of auxiliary elements is not taken into account. The first method significantly underestimates the reliability indicators, and the second, respectively, overestimates them. The method, proposed in the article, takes into account the failure effect of the auxiliary elements on the operating conditions of the main ones, which allows us to more accurately assess the reliability indicators. As a research object, a typical module of equipment of the server room of an average organization was chosen; based on consideration of its structure, failure criteria were formed, and structural reliability schemes, demonstrating the advantages of the proposed approach, were constructed.

Keywords: server room, grounding, auxiliary elements, basic elements, failure rate, failure criteria, availability factor, mean time between failures.

DOI: 10.17587/it.25.241-249

References

1. ANSI/TIA/EIA-569-A standard. *Telekommunikacionnyye trassy i pomescheniya kommercheskih zdaniy* (Telecommunication routes and premises of commercial buildings), 1998. (in Russian).
2. *Jelektroustanovok, pravila ustrojstva, izd. 7* (Electrical installations, device rules, ed. 7), Moscow, NGS "JeNAS", 2003, 168 p. (in Russian).
3. ANSI/TIA/EIA-607 standard. *Trebovaniya k telekommunikacionnoj sisteme vyvazhivaniya potencialov i zazemleniya kommercheskih zdaniy* (Requirements for telecommunication potential equalization and grounding of commercial buildings), 1994 (in Russian).
4. Beichelt F. *Reliability and Maintenance: Networks and Systems*, Chapman and Hall, 2012. 344 p.
5. Zhadnov V. V., Polessky S. N. *Opreделение pokazatelej nadjozhnosti sistem, obladajushih vspomogatel'nyimi jelementami* (Determination of reliability indicators of systems with auxiliary elements), *Sbornik nauchnyh trudov "Proektirovanie telekommunikacionnyh i informacionnyh sredstv i sistem"*, M.: MIEM, 2006, pp. 151–158 (in Russian).
6. Garcia A. C., Prijedor L. P. *Reliability Modeling Prediction*, DQM Research Centre, 2016. 186 p.
7. Cherkosov G. N. *Nadezhnost' apparatno-programmnyh kompleksov. Uchebnoe posobie* (Reliability of hardware and software systems. Tutorial) / SPb.: Piter, 2005. 479 p. (in Russian).
8. Zhadnov V. V., Tihmenev A. N. *Imitacionnoe modelirovanie v zadachah ocenki nadezhnosti otkazoustojchivyh jelektronnyh sredstv* (Simulation modeling in problems of reliability assessment of fault-tolerant electronic devices), *Nadezhnost'*, no. 1, 2013, pp. 32–54 (in Russian).
9. JIS Z 8101–99 standard. *Statistika. Slovar' i uslovnye oboznamenija* (Statistics. Vocabulary and Legend), 1999 (in Russian).
10. GOST 27.003–2016. *Nadezhnost' v tehnike (SSNT). Sostav i obshhie pravila zadaniya trebovanij po nadezhnosti* (Reliability in engineering (VAST). Composition and general rules for setting reliability requirements), 2016 (in Russian).
11. MIL-HDBK-338B. *Electronic reliability design handbook*, 1998.
12. GOST 27.002–2015. *Nadezhnost' v tehnike (SSNT). Terminy i opredelenija* (Reliability in engineering (VAST). Terms and Definitions.), 2015 (in Russian).
13. *Metod "Diagramma Isikavy"* (Ishikawa Chart Method), available at: <http://www.inventech.ru/pub/methods/metod-0019/> (date of access: 29.09.2018).
14. Pohabov Ju. P. *Obespechenie nadezhnosti unikal'nyh vysokootvetstvennyh sistem* (Ensuring the reliability of unique highly responsible systems. Reliability). *Nadezhnost'*, vol. 24 (3), 2017, pp. 17–23 (in Russian).
15. GOST R 51901.14–2007 (MJeK 61078:2006). *Menedzhment riska. Strukturnaja shema nadezhnosti i bulevy metody* (Risk management. Reliability block diagram and boolean methods), 2007 (in Russian).
16. Băjenescu T., Băzu M. *Component Reliability for Electronic Systems*, London, Artech House, 2010, 354 p.
17. Swingler J. *Reliability Characterisation of Electrical and Electronic Systems, 1st Edition*, UK, Woodhead Publishing, 2015, 274 p.
18. Polovko A. M., Gurov S. V. *Osnovy teorii nadezhnosti* (Fundamentals of the theory of reliability), SPb., BHV Peterburg, 2006, 702 p. (in Russian).
19. Ushakov I. A. *Kurs teorii nadezhnosti sistem. Uchebnoe posobie* (Course theory of system reliability. Tutorial), Moscow, Drofa, 2008, 239 p. (in Russian).
20. James Li. Reliability calculation of a parallel redundant system with different failure rate&repair rate using Markov modeling, *Journal of Reliability and Statistical Studies*, vol. 9, iss. 1, 2016, pp. 1–10.
21. O'Connor P., Kleyner A. *Practical Reliability Engineering, 5th Edition* / Wiley, 2012. 512 p.
22. Diveev A. I., Sofronova E. A. *The Network Operator Method for Search of the Most Suitable Mathematical Equation, Bio-Inspired Computational Algorithms and Their Applications*, Croatia: Intech, 2012, pp. 19–42.

Д. А. Алдунин¹, аспирант, e-mail: daldunin@hse.ru,
Г. Г. Федин^{1, 2}, аспирант, стажер-исследователь, e-mail: gfedin@hse.ru

¹Аспирантская школа по компьютерным наукам НИУ ВШЭ,

²Международная научно-учебная лаборатория анализа и выбора решений, НИУ ВШЭ

Математическая модель для построения оптимальной индивидуальной образовательной траектории обучающегося при изучении массовых открытых онлайн-курсов

Рассматривается задача формирования оптимальной индивидуальной образовательной траектории обучающегося при обучении на площадках массовых открытых онлайн-курсов на основании имеющихся знаний и навыков обучающегося и знаний и навыков, которые он хочет приобрести. Предложена математическая модель и сформулированы задачи целочисленного программирования, позволяющие найти оптимальную индивидуальную траекторию при различных предпочтениях обучающегося.

Ключевые слова: электронное обучение, математическое моделирование, целочисленное программирование, индивидуальная образовательная траектория, оптимизация

Введение

Электронное образование дает доступ к лучшим образовательным курсам и программам для тех, кто не имеет возможности учиться в лучших школах и университетах (или университетах вообще). Вместе с тем необходимо учитывать ограничения, вызванные цифровой трансформацией процесса обучения. Дистанционное обучение имеет недостатки, связанные с отсутствием тьютора и аутодидактичностью процесса, что может вызвать разочарование обучающегося в непонятных ситуациях и привести к отказу от выбранных курсов. Однако цифровизация образовательного процесса несет и возможность сделать электронное образование во многом более привлекательным, чем традиционное. По словам доктора Барбары Оукли, автора самого популярного в мире массового открытого онлайн-курса (МООК) "Learning How to Learn" [1], "МООК могут быть художественно и технически увлекательными и иметь потрясающие педагогические преимущества" [2]. Примером такого преимущества может являться построение индивидуальной образовательной траектории (ИОТ). Действительно, сложно представить, чтобы реальная учебная часть планировалась ИОТ для каждого обучающегося с учетом

его или ее предыдущего опыта и предпочтений. Это возможно для частного обучения, но не для массового образования. В то же время в процессе электронного обучения формирование ИОТ может быть автоматизировано с использованием математического моделирования и информационных технологий.

Целью данной статьи является построение инструмента для автоматического формирования оптимальной ИОТ на основе: а) знаний и навыков обучающегося; б) информации о том, какие знания и навыки он или она желает приобрести в ходе обучения; в) информации о том, какие знания и навыки требуются для успешного изучения предлагаемых курсов; г) информации о том, какие знания и навыки обучающийся может получить в процессе изучения предлагаемых курсов.

Представленная модель позволит поставщикам МООК выстраивать свои бизнес-процессы в рамках клиентоцентричного подхода, который является ключом к зрелости бизнеса согласно концепции Modeling Maturity Levels [3], что обуславливает ее актуальность в условиях быстрого роста рынка онлайн-образования, где зрелость сервиса может являться решающим конкурентным преимуществом. Кроме того, прозрачность процесса получения желаемой

компетенции очень важна для обучающегося как участника рынка труда, где его набор компетенций связан с успешной карьерой и, как следствие, развитием компании-работодателя [4].

Представленный подход опирается на следующие преимущества, обеспечиваемые цифровой трансформацией процесса обучения:

- возможность оценки времени, необходимого для успешного освоения курса обучающимся, за счет анализа исторических данных о прежнем образовательном опыте обучающихся методами машинного обучения;
- возможность составления индивидуального расписания для каждого обучающегося;
- доступ к обширной библиотеке курсов.

Предлагаемая модель может стать основой для разработки полноценного рекомендательного сервиса для формирования оптимальных ИОТ на основе МООК, агрегированных с различных платформ, для реальных обучающихся.

Статья организована следующим образом. В первом разделе представлен обзор литературы, в котором приведены ссылки и дано описание наиболее релевантных работ, как относящихся к изучению явления МООК, так и связанных с применением математического моделирования процессов обучения в общем и онлайн-обучения в частности. Во втором разделе формулируется математическая модель задачи формирования оптимальной ИОТ. В заключении отражены основные результаты работы и описаны направления дальнейшего исследования.

Хотя построение ИОТ является очень перспективным направлением исследований, на эту тему была опубликована только одна значимая зарубежная работа с применением математического моделирования [5]. Однако авторы в основном сконцентрировались на использовании теории очередей (теории массового обслуживания) для описания процесса обслуживания множества учеников одним настоящим тьютором и их взаимодействия в течение одного курса.

Необходимо также отметить работу [6], предлагающую алгоритмическое решение для задачи Top-N рекомендаций курсов для обучающихся. Авторы предлагают использовать алгоритм Sparse Linear Method (SLIM), эффективный для работы с разреженными матрицами, возникающими в рекомендательных системах.

Наиболее близкой среди отечественных работ является работа [7], также использующая аппарат математического моделирования. Однако моделирование проводится на уровне отдельных обучающих модулей (тем) и тестирования знаний по ним, т. е. на уровне содержа-

ния курса. Отличается также математический аппарат: авторы предлагают использовать семантическую сеть в виде И/ИЛИ-графа.

В работе [8] рассматривается применение для адаптации контента двух технологий: semantic web (семантическая сеть) и intelligent agents (умные агенты). Использование онтологии с семантическими сервисами позволяет быстрее и удобнее подбирать образовательные материалы. Умные агенты позволяют предоставлять обучающимся виртуальных персональных ассистентов для поддержки в образовательном процессе с учетом стиля обучения и уровня знаний. В этой статье представлена онтология предметной области, которая подходит для адаптивных электронных образовательных сред. Онтология описывает объекты обучения, которые составляют курс, а также учащихся и их стили обучения.

Согласно статье [9] электронное образование сталкивается не только с ограничениями, обусловленными культурными различиями и отсутствием живого тьютора, но и с техническими ограничениями, например, мобильных устройств. В статье представлен проект многоагентной системы для адаптации учебного контента к компетенциям обучающегося в контексте его/ее индивидуальных особенностей и используемого мобильного устройства. В статье также подробно описывается прототип, разработанный для тестирования предложенного проекта.

Как мы видим, тема моделирования процесса электронного образования активно исследуется, и предлагаются решения, использующие самые разные подходы: от семантических сетей до многоагентных систем. Однако авторам не удалось найти работ, в которых задача формирования оптимальной индивидуальной образовательной траектории формулировалась бы в виде задачи математического программирования. Кроме того, работы, связанные с рекомендательными сервисами, посвящены только ранжированию доступных курсов согласно предпочтениям обучающегося. Таким образом, существует необходимость в разработке инструмента формирования оптимальной ИОТ для обучающихся МООК.

1. Математическая модель

Представленная ниже модель позволяет найти оптимальную ИОТ для слушателя МООК. Под оптимальной ИОТ понимается набор курсов, оптимизирующий одну из целевых функций, рассмотренных далее, который

будет предложен слушателю для прохождения. Кроме самой ИОТ, модель позволяет получить набор компетенций, которыми будет обладать обучающийся в результате освоения предложенного набора курсов.

Модель описывает набор знаний и навыков (в данной работе данные понятия не разделяются и предполагается, что в процессе обучения обучающийся может получить как знания, так и навыки), которыми обладает обучающийся до начала прохождения ИОТ, и набор новых знаний и навыков, которые обучающийся хочет получить в результате прохождения ИОТ как подмножества множества всех знаний и навыков (примером такого множества могут служить профессиональные стандарты из Реестра профессиональных стандартов, разработанного Министерством труда и социальной защиты Российской Федерации¹). Также модель включает в себя множество курсов, доступных для включения в ИОТ. Эти курсы характеризуются: а) набором знаний и навыков пререквизитов, которыми обучающийся должен обладать для успешного освоения курса; б) набором знаний и навыков постреквизитов, которые обучающийся приобретет в результате освоения курса; в) временем, которое обучающийся предположительно должен будет затратить для успешного освоения курса; г) ценой курса, которую обучающийся должен будет заплатить для получения доступа к курсу. Предполагается, что обучающийся может начать изучать курс только тогда, когда он обладает всеми знаниями и навыками, входящими в набор знаний и навыков пререквизитов данного курса.

Время, необходимое для освоения курса, как правило, определяется методистами и авторами курсов при публикации на платформах МООС. В дальнейшем планируется предложить метод более точного индивидуального прогнозирования времени, необходимого для освоения курса, на основании исторических данных о том, как с этим курсом справлялись обучающиеся, обладающие сходным набором знаний и навыков.

Задача определения исходного набора знаний и навыков обучающегося может быть решена следующими способами: опросом (обучающийся сам помечает знания и навыки как имеющиеся), тестированием, анализом данных о предыдущем образовательном опыте (приложения к дипломам, сертификаты о прохождении курсов и т. п.) и т. д.

¹<http://profstandart.rosmintrud.ru/obshchiy-informatsionnyy-blok/natsionalnyy-reestr-professionalnykh-standartov/reestr-professionalnykh-standartov/>

Рассматривается задача отыскания набора курсов, необходимых для получения желаемых знаний и навыков.

Пусть

N — число курсов, доступных для изучения;

Z — общее число знаний и навыков;

K — множество номеров знаний и навыков;

S — число знаний и навыков обучающегося до начала прохождения ИОТ;

K^S — множество номеров знаний и навыков обучающегося до начала прохождения ИОТ;

F — число знаний и навыков, которые обучающийся желает получить в результате прохождения ИОТ;

K^F — множество номеров знаний и навыков, которые обучающийся желает получить в результате прохождения ИОТ;

K_i^{Pre} — множество номеров знаний и навыков, необходимых для успешного прохождения курса i (пререквизиты курса), $i = 1, \dots, N$;

K_i^{Post} — множество номеров знаний и навыков, которые обучающийся получит в результате успешного прохождения курса i (постреквизиты курса), $i = 1, \dots, N$;

C_j^{Post} — множество номеров курсов, в которых знание или навык j входит в множество постреквизитов, $j = 1, \dots, Z$;

P_i — цена курса с номером i для обучающегося, $i = 1, \dots, N$;

T_i — время, необходимое обучающемуся для изучения курса с номером i , $i = 1, \dots, N$;

c_i — булева переменная, принимающая значение 1, если курс с номером i включен в ИОТ, и значение 0 иначе, $i = 1, \dots, N$;

k_j — булева переменная, принимающая значение 1, если обучающийся будет обладать знанием или навыком с номером j в результате прохождения ИОТ, и значение 0 в противном случае, $j = 1, \dots, Z$.

Используя представленные обозначения, можем записать систему ограничений, определяющую множество допустимых ИОТ, следующим образом:

$$k_j = 1, \forall j \in K^S; \quad (1)$$

$$k_j = 1, \forall j \in K^F; \quad (2)$$

$$|K_i^{Pre}| c_i \leq \sum_{j \in K_i^{Pre}} k_j, \forall i = 1, \dots, N; \quad (3)$$

$$k_j \leq \sum_{i \in C_j^{Post}} c_i, \forall j \in K \setminus K^S; \quad (4)$$

$$c_i \in \{0, 1\}, \forall i \in \overline{1, N}; \quad (5)$$

$$k_j \in \{0, 1\}, \forall j \in K, \quad (6)$$

где $|\cdot|$ — мощность множества.

Ограничения в данной системе имеют следующий смысл: (1) — знания и навыки, имеющиеся до начала освоения ИОТ, известны; (2) — желаемые знания и навыки должны быть получены; (3) — курс может быть освоен, только если обучающийся обладает всеми знаниями и навыками, входящими во множество его пререквизитов; (4) — все знания и навыки (кроме имеющихся до начала прохождения ИОТ) считаются известными только после прохождения хотя бы одного курса, в котором данные знания и навыки входят в множество постреквизитов; (5) и (6) — ограничения на булевость переменных модели.

Для выбора конкретной ИОТ из множества доступных, определенного системой ограничений (1)–(4), необходимо определить целевую функцию.

В качестве целевых функций могут быть рассмотрены следующие варианты:

- минимизация числа курсов в ИОТ

$$\sum_{i=1}^N c_i \rightarrow \min; \quad (7)$$

- минимизация общей стоимости обучения

$$\sum_{i=1}^N P_i c_i \rightarrow \min; \quad (8)$$

- минимизация ожидаемого суммарного времени прохождения всех курсов

$$\sum_{i=1}^N T_i c_i \rightarrow \min; \quad (9)$$

- минимизация общей стоимости и ожидаемого суммарного времени прохождения всех курсов

$$\sum_{i=1}^N ((P_i + \alpha T_i) c_i) \rightarrow \min, \quad (10)$$

где α — это коэффициент с размерностью $\frac{\text{деньги}}{\text{время}}$, отражающий степень готовности обучающегося платить за сокращение затрат времени. Данный коэффициент может быть определен на основании предпочтений обучающегося с использованием методов, описанных в работе [10].

Кроме целевой функции (10) могут быть использованы и другие линейные комбинации целевых функций (7), (8) и (9) с соответствующими коэффициентами.

Для практического применения предложенной выше модели удобнее модифицировать систему ограничений (1)–(6), уменьшив число переменных. Так как ограничения (1) и (2) фик-

сируют значения части булевых переменных k_j , эти переменные могут быть исключены из модели, а их вхождения заменены на значения из ограничений (1) и (2). Тогда модифицированная система ограничений будет иметь вид

$$\left| K_i^{Pre} \setminus (K^S \cup K^F) \right| c_i - \sum_{j \in K_i^{Pre} \setminus (K^S \cup K^F)} k_j \leq 0, \forall i = 1, \dots, N; \quad (11)$$

$$k_j - \sum_{i \in C_j^{Post}} c_i \leq 0, \forall j \in K \setminus (K^S \cup K^F); \quad (12)$$

$$- \sum_{i \in C_j^{Post}} c_i \leq -1, \forall j \in K^F; \quad (13)$$

$$c_i \in \{0, 1\}, \forall i \in \overline{1, N}; \quad (14)$$

$$k_j \in \{0, 1\}, \forall j \in K \setminus (K^S \cup K^F), \quad (15)$$

где ограничение (11) соответствует ограничению (3), а ограничения (12) и (13) являются частными случаями ограничения (4).

Пусть $x = (c, k)$, а $Ax \leq b$ описывает систему ограничений (11)–(13). Тогда выпуклый многогранник Π , задаваемый этой системой ограничений, может быть записан как

$$\Pi = \{x \mid Ax \leq b\},$$

а множество V , задаваемое ограничениями (11)–(15), может быть записано как

$$V = \Pi \cap (\bar{C} \times \bar{K}),$$

где

$$\bar{C} = \{0, 1\}^N; \\ \bar{K} = \left\{ \{0, 1\}^{|K \setminus (K^S \cup K^F)|} \right\};$$

$\{0, 1\}^N$ — единичный куб размерности N .

Если ввести обозначения

$$l^* = (1_N, 0_k); \\ p^* = (p, 0_k); \\ t^* = (t, 0_k); \\ s^* = (p + \alpha t, 0_k),$$

где (\cdot, \cdot) — вектор, состоящий из элементов двух векторов; 1_N — единичный вектор размера N ; 0_k — нулевой вектор размера $|K \setminus (K^S \cup K^F)|$, то оптимизационные задачи с целевыми функциями (7), (8), (9), (10) и ограничениями (11)–(15) могут быть записаны в векторно-матричной форме как

$$\min_{x \in V} \langle l^*, x \rangle; \quad (16)$$

$$\min_{x \in V} \langle p^*, x \rangle; \quad (17)$$

$$\min_{x \in V} \langle t^*, x \rangle; \tag{18}$$

$$\min_{x \in V} \langle s^*, x \rangle, \tag{19}$$

где $\langle \cdot \rangle$ — скалярное произведение.

Задачи (16)–(19) являются задачами целочисленного программирования с линейными ограничениями, в которых все целочисленные переменные — булевы. Таким образом, эта задача является задачей булевого программирования, а значит, NP-трудной [11]. Такие задачи достаточно хорошо решаются стандартными пакетами для решения задач оптимизации, например, CPLEX, Gurobi, CBC и др. Сравнение различных пакетов представлено в работе [12].

2. Пример использования различных целевых функций для отыскания оптимальной ИОТ

Для тестирования модели и сравнения различных целевых функций был составлен модельный пример. Данный пример включает пять курсов ($N=5$) и шесть компетенций ($Z=6$). Информация о курсах (список пререквизитов курса, список постреквизитов курса, стоимость курса, время прохождения курса) приведена в табл. 1. Предполагается, что до начала обучения обучающийся владеет знанием или навыком k_1 и хочет овладеть знаниями или навыками k_5 и k_6 в результате обучения.

В векторно-матричной форме данный пример будет иметь вид

$$\langle a, (c_1, c_2, c_3, c_4, c_5, k_2, k_3, k_4)^T \rangle \rightarrow \min, \tag{20}$$

$$\begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$(c_1, c_2, c_3, c_4, c_5, k_2, k_3, k_4)^T \in \{0,1\}^8,$$

где a — это один из векторов l^* , p^* , t^* , s^* , которые в данном примере принимают вид

$$l^* = (1, 1, 1, 1, 1, 0, 0, 0),$$

$$p^* = (10, 20, 15, 5, 25, 0, 0, 0),$$

$$t^* = (20, 10, 15, 25, 20, 0, 0, 0),$$

$$s^* = (30, 30, 30, 30, 45, 0, 0, 0) \text{ при } \alpha = 1.$$

Информация о курсах

Курс (c_i)	Пререквизиты курса (K_i^{Pre})	Постреквизиты курса (K_i^{Post})	Стоимость курса (P_i , \$)	Время прохождения курса (T_i , ч)
c_1	$\{k_1\}$	$\{k_2, k_3\}$	10	20
c_2	$\{k_1, k_2\}$	$\{k_4, k_5\}$	20	10
c_3	$\{k_3\}$	$\{k_5\}$	15	15
c_4	$\{k_3, k_5\}$	$\{k_6\}$	5	25
c_5	$\{k_4\}$	$\{k_5, k_6\}$	25	20

Таблица 2

Значения целевых функций для оптимальных ИОТ

Набор курсов в ИОТ	Число курсов в ИОТ	Суммарная стоимость курсов в ИОТ (\$)	Ожидаемое суммарное время обучения (ч)	Линейная комбинация стоимости и времени ($\alpha = 1$) (\$)
c_1, c_2, c_5	3	55	50	105
c_1, c_3, c_4	3	30	60	95

Система неравенств (20) была получена из системы ограничений (11), (12) и (13). Первые три неравенства системы (20) получены из ограничения (12), четвертое и пятое неравенства получены из ограничения (13), оставшиеся неравенства с шестого по девятое получены из ограничения (11).

Можно заметить, что при фиксированном наборе курсов и знаний и навыков упомянутая выше матрица задачи A зависит исключительно от знаний и навыков обучающегося до начала прохождения ИОТ и знаний и навыков, которые обучающийся желает получить в результате прохождения ИОТ.

Для решения приведенной в качестве примера задачи использовалась функция *intlinprog*², входящая в пакет MATLAB. В табл. 2 приведены оптимальные значения целевых функций (7)–(10) и ИОТ, на которых эти оптимальные значения достигаются в данном примере.

Из табл. 2 видно, что для различных целевых функций оптимальными могут быть различные ИОТ. ИОТ, оптимальные для различных целевых функций, приведены в табл. 3.

Так как предложенная модель является статической, т. е. в ней отсутствует состояние во времени, важным ограничением применимости модели является требование отсутствия ситуаций, в которых знания и навыки связаны между собой циклически через один или несколько курсов, так как такие ситуации могут быть неразрешимы: модель не учитывает то, какие зна-

²<https://www.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html>

Таблица 3

Оптимальные ИОТ для различных целевых функций

Целевая функция	Оптимальная траектория
Минимизация числа курсов, освоение которых требуется (7)	c_1, c_2, c_5 и c_1, c_3, c_4
Минимизация стоимости предлагаемых курсов (8)	c_1, c_3, c_4
Минимизация ожидаемого времени обучения (9)	c_1, c_2, c_5
Минимизация линейной комбинации из времени и стоимости с единичными коэффициентами	c_1, c_3, c_4
Минимизация линейной комбинации из времени, стоимости и числа курсов с единичными коэффициентами	c_1, c_3, c_4

ния и навыки будут изучены раньше или позже, и в такой ситуации может предложить такой набор курсов, для которого невозможно будет построить последовательность изучения курсов (в качестве пререквизита будет требоваться знание или навык, который еще не изучен). Такие циклические связи могут быть выявлены при анализе ориентированного графа, множество вершин которого определено множествами курсов и знаний и навыков, а дуги — отношениями "быть пререквизитом" и "быть постреквизитом", до решения задачи. Задача определения оптимального набора курсов может быть решена, а полученное решение проверено на вхождение таких связей. В этом случае лицо, принимающее решения, может быть извещено о наличии связей, мешающих решению задачи, чтобы определить, какие связи должны быть удалены.

Заключение

Для решения поставленной задачи создания инструмента для оптимального выбора курсов для ИОТ на основании знаний и навыков обучающегося и его образовательных целей была предложена математическая модель, рассматривающая данную задачу как задачу оптимизации. Предложенная модель позволяет оптимальным образом выбрать курсы из агрегированного пула курсов с учетом предпочтений обучающегося в отношении времени, стоимости обучения и числа курсов в ИОТ. Сформулированная задача может быть эффективно решена с помощью стандартных математических пакетов на данных реального размера.

Модель была протестирована на модельных данных. Тестирование показало адекватность получаемых результатов, а также в ходе тестирования был продемонстрирован процесс применения предлагаемого инструмента для фор-

мирования оптимального набора курсов для ИОТ обучающегося.

В качестве направлений дальнейшего исследования предполагаются:

- задача формирования оптимальной последовательности курсов из входящего в ИОТ набора (расписания слушателя) с учетом возможности одновременного изучения заданного числа курсов, а также временных промежутков, в которые доступны курсы;
- задача разработки программной системы, реализующей сбор и анализ данных платформ MOOK для создания актуальной базы курсов и знаний и навыков, и реализации на ее основе с использованием предложенного в данной работе подхода полноценной рекомендательной системы для формирования ИОТ на основе MOOK, агрегированных с различных платформ, для реальных обучающихся.

Список литературы

1. **Markoff J.** The Most Popular Online Course Teaches You to Learn // New York Times. 29.12.2015. URL: <https://bits.blogs.nytimes.com/2015/12/29/the-most-popular-online-course-teaches-you-to-learn/?mcubz=0> (дата обращения 01.08.18).
2. **Oakley B.** Why Virtual Classes Can Be Better Than Real Ones // Nautilus Magazine. 29.10.2015. URL: <http://nautilus.us/issue/29/scaling/why-virtual-classes-can-be-better-than-real-ones> (дата обращения 01.08.18).
3. **The journey toward greater customer centricity** // Ernst & Young LLP. 2013. URL: [http://www.ey.com/Publication/vwLUAssets/The_journey_toward_greater_customer_centricity_-_US/\\$FILE/Customer_Centricity_Paper_29_April_Final_US.pdf](http://www.ey.com/Publication/vwLUAssets/The_journey_toward_greater_customer_centricity_-_US/$FILE/Customer_Centricity_Paper_29_April_Final_US.pdf) (Дата обращения 01.08.18)
4. **Sandberg J.** Understanding of work: the basis for competence development // International Perspectives on Competence in the Workplace, ed. C. R. Velde, Springer, Berlin. 2009. P. 3—20.
5. **Rózewski P., Zaikin O.** Integrated mathematical model of competence-based learning-teaching process // Bulletin of the Polish Academy of Sciences: Technical Sciences. 2015. N. 1 (63). P. 245—259. doi: 10.1515/bpasts-2015-0029
6. **Jinjiao Lin, Haitao Pu, Yibin Li, Jian Lian.** Intelligent Recommendation System for Course Selection in Smart Education // Procedia Computer Science. 2018. Vol. 129. P. 449—453. doi: 10.1016/j.procs.2018.03.023
7. **Норенков И. П., Соколов Н. К.** Синтез индивидуальных маршрутов обучения в онтологических обучающих системах // Информационные технологии. 2009. № 3. С. 74—77.
8. **Dung P. Q., Florea A. M.** Adaptation to learners' learning styles in a multi-agent e-learning system // Internet Learning. 2013. Vol. 1. P. 11—20.
9. **Garcia-Gabot A.** A proposal of a multi-agent system for adapting learning contents to user competencies, context and mobile device // Research Papers Faculty of Materials Science and Technology Slovak University of Technology, Bratislava. 2013.
10. **Подиновский В. В.** Теория важности критериев в многокритериальных задачах принятия решений при неопределенности. II. Задачи с количественной информацией о важности критериев и вероятностях значений неопределенного фактора // Информационные технологии моделирования и управления. 2012. № 2 (74). С. 131—137.
11. **Karp R. M.** Reducibility Among Combinatorial Problems // In R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. New York: Plenum. 1972. P. 85—103.
12. **Latest Benchmarks of Optimization Software** // INFORMS Annual Meeting 2017. 23.10.2017. URL: <http://plato.asu.edu/bench.html> (дата обращения 01.08.18).

D. A. Aldunin, Ph. D. student in Computer Science at Higher School of Economics, Moscow, Russian Federation, e-mail: daldunin@hse.ru,

G. G. Fedin, Ph. D. student in Computer Science at Higher School of Economics, Moscow, Russian Federation, e-mail: gfedin@hse.ru

Mathematical Modelling of Building Individual Educational Trajectory for Studying MOOCs

Distant learning has weaknesses related to missing tutor and kind of auto didacticism of the process, which may cause learner's frustration in uncertain situations and force him or her to drop the learning course. Inasmuch as it is very important to help learner to select a set of needed courses, the article deals with the task of forming the optimal individual educational trajectory of a learner when studying massive open online courses on the basis of knowledge and skills of the learner, knowledge and skills he or she wants to acquire and prerequisite and postrequisite skills and knowledge of available courses. The mathematical model is proposed, and problems of integer programming are formulated, which allow to find the optimal individual educational trajectory for different preferences of the learner.

Keywords: e-learning, mathematical modeling, integer programming, individual educational trajectory, optimization

DOI: 10.17587/it.25.250-256

References

1. **Markoff J.** The Most Popular Online Course Teaches You to Learn, New York Times, 29.12.2015, available at: <https://bits.blogs.nytimes.com/2015/12/29/the-most-popular-online-course-teaches-you-to-learn/>
2. **Oakley B.** Why Virtual Classes Can Be Better Than Real Ones, *Nautilus Magazine*, 29.10.2015, available at: <http://nautilus.us/issue/29/scaling/why-virtual-classes-can-be-better-than-real-ones>
3. **The journey** toward greater customer centricity, Ernst & Young LLP, 2013. available at: [http://www.ey.com/Publication/vwLUAssets/The_journey_toward_greater_customer_centricity_-_US/\\$FILE/Custom_Centricity_Paper_29_April_Final_US.pdf](http://www.ey.com/Publication/vwLUAssets/The_journey_toward_greater_customer_centricity_-_US/$FILE/Custom_Centricity_Paper_29_April_Final_US.pdf)
4. **Sandberg J.** Understanding of work: the basis for competence development, *International Perspectives on Competence in the Workplace*, C. R. Veldeed., Springer, Berlin, 2009, pp. 3–20.
5. **Różewski P., Zaikin O.** Integrated mathematical model of competence-based learning-teaching process, *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 2015, no. 1 (63), pp. 245–259, doi: 10.1515/bpasts-2015-0029
6. **Jinjiao Lin, Haitao Pu, Yibin Li, Jian Lian.** Intelligent Recommendation System for Course Selection in Smart Education, *Procedia Computer Science*, 2018, vol. 129, pp. 449–453, doi: 10.1016/j.procs.2018.03.023
7. **Norenkov I. P., Sokolov N. K.** Creation of Individual Learning Routes in Ontology Education Systems, *Informacionnie Tekhnologii*, 2009, vol. 3, pp. 74–77 (in Russian).
8. **Dung P. Q., Florea A. M.** Adaptation to learners' learning styles in a multi-agent e-learning system, *Internet Learning*, 2013, vol. 1, pp. 11–20.
9. **Garcia-Gabot A. A.** proposal of a multi-agent system for adapting learning contents to user competencies, context and mobile device, *Research Papers Faculty of Materials Science and Technology Slovak University of Technology, Bratislava*, 2013.
10. **Podinovskiy V. V.** Teoriya vazhnosti kriteriev v mnogokriterialnyh zadachah prinyatiya reshenij pri neopredelennosti. II. Zadachi s kolichestvennoji nformaciej o vazhnosti kriteriev i veroyatnostyah znachenijne opredelennogo faktora, *Informacionnye tekhnologii modelirovaniya i upravleniya*, vol. 2 (74), 2012, pp. 131–137 (in Russian).
11. **Karp R. M.** Reducibility Among Combinatorial Problems, In R. E. Miller and J. W. Thatcher (editors), *Complexity of Computer Computations*, New York, Plenum, 1972, pp. 85–103.
12. **Latest** Benchmarks of Optimization Software, *INFORMS Annual Meeting 2017*, 23.10.2017, available at: <http://plato.asu.edu/bench.html>

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 08.02.2019. Подписано в печать 26.03.2019. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ419. 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.

Рисунки к статье И. В. Филиппова, А. Ф. Мелик-Адамяна, В. А. Сухомлинова
 «РАЗМЕЩЕНИЕ СЕТЕВЫХ ФУНКЦИЙ В ОБЛАЧНЫХ ИНФРАСТРУКТУРАХ»

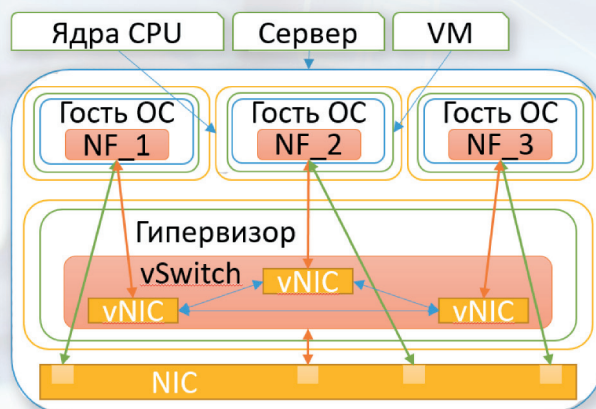


Рис. 2. Размещение сетевых функций в виртуальных машинах (VM):
 NIC – сетевая карта; vNIC – виртуальные сетевые карты; NF_* – сетевые функции

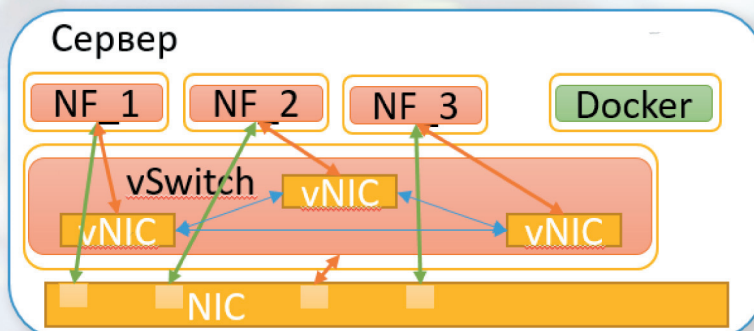


Рис. 3. Размещение сетевых функций в контейнерах:
 NIC – сетевая карта; vNIC – виртуальные сетевые карты; NF_* – сетевые функции

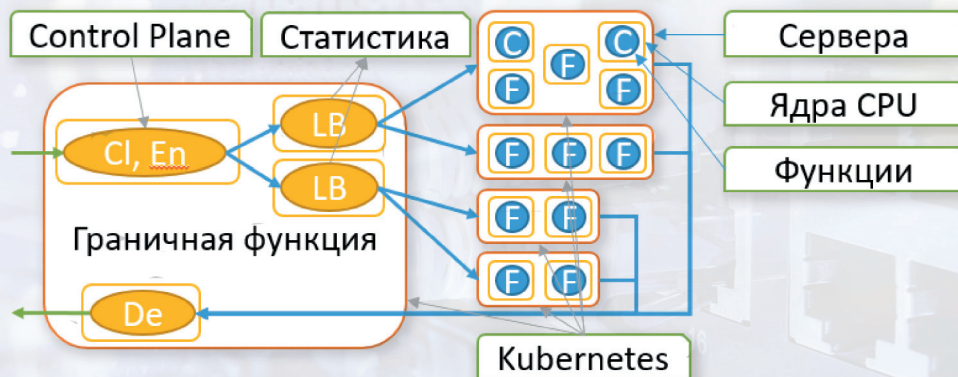


Рис. 5. Предлагаемый вариант размещения сетевых функций:
 Cl – классификация; En – инкапсуляция; De – декапсуляция; LB – балансировка нагрузки;
 F, C – различные сетевые функции и их клоны

Издательство «НОВЫЕ ТЕХНОЛОГИИ»

выпускает научно-технические журналы



Ежемесячный теоретический
и прикладной научно-технический журнал

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В журнале освещаются современное состояние, тенденции и перспективы развития основных направлений в области разработки, производства и применения информационных технологий.

Подписной индекс по Объединенному каталогу
«Пресса России» – 72656



Научно-практический
и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

В журнале освещаются достижения и перспективы в области исследований, обеспечения и совершенствования защиты человека от всех видов опасностей производственной и природной среды, их контроля, мониторинга, предотвращения, ликвидации последствий аварий и катастроф, образования в сфере безопасности жизнедеятельности.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79963

Ежемесячный
междисциплинарный
теоретический и прикладной
научно-технический журнал

НАНО- и МИКРОСИСТЕМНАЯ ТЕХНИКА

В журнале освещаются современное состояние, тенденции и перспективы развития нано- и микросистемной техники, рассматриваются вопросы разработки и внедрения нано микросистем в различные области науки, технологии и производства.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79493



Ежемесячный теоретический
и прикладной
научно-технический журнал

МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ

В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автоматического и автоматизированного управления техническими объектами и технологическими процессами в промышленности, энергетике и на транспорте.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79492

Теоретический
и прикладной
научно-технический журнал

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях человеческой деятельности.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 22765

Адрес редакции журналов для авторов и подписчиков:

107076, Москва, Стромьинский пер., 4. Издательство "НОВЫЕ ТЕХНОЛОГИИ".
Тел.: (499) 269-55-10, 269-53-97. Факс: (499) 269-55-10. E-mail: antonov@novtex.ru