

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 24  
2018  
№ 10

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Иванников А. Д., Северцев В. Н.** Математическая модель множества входных воздействий цифровых систем при их моделировании на уровне логических сигналов . . . . . 627
- Попков С. И.** Метод внешней оптимизации для идентификации марковских процессов . . . . . 633

### ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Харахинов В. А.** Генетический алгоритм как альтернатива обучения слоя Кохонена . . . . . 642
- Лядова Л. Н., Сухов А. О., Медведева Е. Ю.** Алгоритмы синтаксического разбора для текстовых динамически настраиваемых предметно-ориентированных языков . . . . . 648

### БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Васильев В. И., Вульфин А. М., Гузаиров М. Б., Кириллова А. Д.** Интервальное оценивание информационных рисков с помощью нечетких серых когнитивных карт . . . . . 657

### БАЗЫ ДАННЫХ

- Игнатьев Н. А.** Выбор собственного пространства объекта с использованием нелинейных преобразований признаков . . . . . 665

### ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Май В. П.** Система объемной визуализации синоптических объектов . . . . . 671

### ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИОМЕДИЦИНСКИХ СИСТЕМАХ

- Новик В. П., Матвеев И. А., Леонов В. Ю.** Сравнение информативных признаков изображений радужки глаза методом оптимального пути . . . . . 681

Главный редактор:

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

Зам. главного редактора:

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

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.

ЖУРАВЛЕВ Ю. И.,

акад. РАН, д. ф.-м. н., проф.

КУЛЕШОВ А. П.,

акад. РАН, д. т. н., проф.

ПОПКОВ Ю. С.,

акад. РАН, д. т. н., проф.

РУСАКОВ С. Г.,

чл.-корр. РАН, д. т. н., проф.

РЯБОВ Г. Г.,

чл.-корр. РАН, д. т. н., проф.

СОЙФЕР В. А.,

акад. РАН, д. т. н., проф.

СОКОЛОВ И. А.,

акад. РАН, д. т. н., проф.

СУЕТИН Н. В., д. ф.-м. н., проф.

ЧАПЛЫГИН Ю. А.,

акад. РАН, д. т. н., проф.

ШАХНОВ В. А.,

чл.-корр. РАН, д. т. н., проф.

ШОКИН Ю. И.,

акад. РАН, д. т. н., проф.

ЮСУПОВ Р. М.,

чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.

АНТОНОВ Б. И.

БАРСКИЙ А. Б., д. т. н., проф.

ВАСЕНИН В. А., д. ф.-м. н., проф.

ВАСИЛЬЕВ В. и., д. т. н., проф.

ВИШНЕКОВ А. В., д. т. н., проф.

ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.

ДОМРАЧЕВ В. Г., д. т. н., проф.

ЗАБОРОВСКИЙ В. С., д. т. н., проф.

ЗАРУБИН В. С., д. т. н., проф.

КАРПЕНКО А. П., д. ф.-м. н., проф.

КОЛИН К. К., д. т. н., проф.

КУЛАГИН В. П., д. т. н., проф.

КУРЕЙЧИК В. В., д. т. н., проф.

ЛЬВОВИЧ Я. Е., д. т. н., проф.

МАРТЫНОВ В. В., д. т. н., проф.

МИХАЙЛОВ Б. М., д. т. н., проф.

НЕЧАЕВ В. В., к. т. н., проф.

ПОЛЕШУК О. М., д. т. н., проф.

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.

ЛЫСЕНКО А. В.

ЧУГУНОВА А. В.

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. 24  
2018  
No. 10

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

## CONTENTS

### MODELING AND OPTIMIZATION

- Ivannikov A. D., Severtcev V. N.** Mathematical Model for Digital System Input Interaction Set in the Process of Logical Simulation ..... 627
- Popkov S. I.** The Method of External Optimization for Methods of Markov Processes Identification ..... 633

### INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Kharakhinov V. A.** The Genetic Algorithm as the Alternative Method for Training Kohonen Layer ..... 642
- Lyadova L. N., Sukhov A. O., Medvedeva E. Yu.** Syntactic Parsing Algorithms for Textual Domain-Specific Languages ..... 648

### INFORMATION SECURITY

- Vasilyev V. I., Vulfin A. M., Guzairov M. B., Kirillova A. D.** Interval Estimation of Information Risks with use of Fuzzy Grey Cognitive Maps ..... 657

### DATABASE

- Ignatiev N. A.** Selecting an Object's Own Space Using Nonlinear Transformations of Features ..... 665

### GEOINFORMATION SYSTEMS

- May V. P.** The Volume Visualization System of Synoptic Objects ..... 671

### INFORMATION TECHNOLOGIES IN BIOMEDICAL SYSTEMS

- Novik V. P., Matveev I. A., Leonov V. Yu.** Matching of Informative Features of Iris Images with the Optimal Path Method ..... 681

#### 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.  
Martyonov 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.  
Lysenko A. V.  
Chugunova A. V.

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"

# МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 681.31

DOI: 10.17587/it.24.627-632

А. Д. Иванников, д-р техн. наук, проф., adi@ippm.ru, В. Н. Северцев, д-р техн. наук,  
Институт проблем проектирования в микроэлектронике РАН, Москва

## Математическая модель множества входных воздействий цифровых систем при их моделировании на уровне логических сигналов

*Рассматривается моделирование проектов цифровых систем на уровне логических сигналов в целях отладки проектов. Предполагается, что цифровая система выполняет некоторую последовательность функций из конечного алфавита. При описании методами теории множеств анализируется структура множества входных воздействий цифровых систем и их блоков. Предлагается математическая модель множества входных воздействий для цифровых систем и их блоков для каждой выполняемой функции с учетом временных ограничений в виде помеченного ориентированного графа. Полученную формальную модель предполагается использовать при выборе набора отладочных тестов.*

**Ключевые слова:** моделирование на уровне логических сигналов, отладка проектов цифровых систем, множество входных взаимодействий, графовая модель входных взаимодействий

### Введение

При проектировании цифровых систем важной задачей является выбор конечного числа конечных по времени тестовых примеров, которые подаются на компьютерную модель проектируемой системы для проверки правильности функционирования последней [1–4].

Для того чтобы составить репрезентативный набор входных тестовых примеров, необходимо иметь описание множества допустимых входных воздействий как на разрабатываемую цифровую систему, так и на ее блоки. Целью настоящего исследования является разработка математической модели множества допустимых входных цифровых систем в целом и их блоков при их моделировании на уровне логических сигналов.

### Описание используемой модели

При проектировании сложных цифровых систем разработчик должен обеспечить, прежде всего, требуемое внешнее поведение цифровой системы, т. е. требуемое взаимодействие системы с внешней средой. При этом существенным

является как последовательность выходных сигналов цифровой системы, так и моменты времени появления и изменений этих сигналов, причем временные ограничения обычно задаются интервалами значений [1, 5].

Взаимодействие цифровой системы с объектом управления и внешним миром вообще осуществляется через внешние линии и шины — наборы линий, по которым передается однородная информация, например, адреса или данные. В цифровых системах управления широко используются двунаправленные шины и линии, имеющие также состояние с высоким выходным сопротивлением (отключенное состояние). Будем рассматривать логическую модель сигналов на шинах и линиях цифровых систем, т. е. считать, что значения сигналов представляются как 0 или 1 на линиях и как число из диапазона  $0...2^n$  на шинах системы. Цифровые сигналы внешних шин и линий назовем терминальными переменными, они принадлежат множеству  $\mathbf{P}$ . Переменная  $p \in \mathbf{P}$  всегда имеет одно из значений конечного множества  $Z_p$ , элементы которого определяют как целочисленное значение сигнала, так и направление работы шины или линии.

Событием по переменной  $p$  называется изменение переменной  $p$  со значения  $z_1 \in Z_p$  на значение  $z_2 \in Z_p$  в момент времени  $t$ . Обозначим такое событие  $\chi_{p,z_1,z_2}^t$ . Взаимодействие цифровой системы с внешней средой, включая управляемый объект, есть последовательность переключений сигналов на терминальных шинах и линиях, т. е. последовательность событий. Для каждой проектируемой системы имеется множество  $\Psi$  допустимых взаимодействий с внешней средой, каждое из которых есть отображение  $\psi : [0, t) \rightarrow Q, t \in T, Q = \prod_{p \in P} Z_p$  [4].

В цифровых системах для каждого конечного временного интервала число событий по терминальным переменным, т. е. число изменений их значений, конечно. В связи с этим любое взаимодействие  $\psi$  может быть представлено в виде вектора  $(z_{p_1}^H, \dots, z_{p_k}^H)$  начальных значений переменных  $p_1, \dots, p_k$  ( $k$  — мощность множества  $P$ ) в момент времени  $t = 0$  и последовательности событий по переменным множества  $P$  с конечным числом событий за любой конечный интервал времени:

$$\psi = (z_{p_1}^H, \dots, z_{p_k}^H), \quad (1)$$

$$\chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots,$$

где  $t_1 \leq t_2 \leq t_3 \leq \dots$  — упорядоченная последовательность времен событий;  $p_{i_1}, p_{i_2}, p_{i_3}, \dots$  — переменные, принадлежащие множеству  $P$ ;  $z_{j_1}, z_{j_3}, z_{j_5}, \dots$  — значения переменных непосредственно перед событием;  $z_{j_2}, z_{j_4}, z_{j_6}, \dots$  — значения переменных непосредственно после события.

Если в последовательности (1) выделить только события, являющиеся изменениями входных сигналов, то такую последовательность можно назвать входным воздействием. Однако часто моменты подачи входных сигналов на цифровую систему определяются готовностью системы принять эти сигналы, на что указывают определенные выходные сигналы системы. Выполнение какой-либо операции, например, считывания данных цифровой системой, может инициироваться не сигналами внешней среды, а самой системой. В связи с этим использование в качестве аргументов функционирования цифровой системы входных воздействий не всегда удобно.

Выделим из последовательности событий (1) взаимодействия  $\psi$  последовательность входных событий и выходных событий управления об-

меном, которые по заданному протоколу обмена обуславливают моменты времени входных событий. Назовем эту последовательность входным взаимодействием

$$\mu = (z_{p_1}^H, \dots, z_{p_{n+q}}^H), \quad (2)$$

$$\chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots,$$

где  $\chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}$ ,  $\chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}$ ,  $\chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots$  — входные события и выходные события управления обменом;  $t_1 \leq t_2 \leq t_3 \leq \dots$  — упорядоченная последовательность времен событий входного взаимодействия.

В рассматриваемой модели в качестве аргументов функционирования цифровых систем используются входные взаимодействия, что дает возможность рассматривать режимы работы, инициируемые как внешними входными сигналами, так и самими цифровыми системами [1, 6, 7]. Все вышесказанное относится не только к цифровым системам в целом, но и к блокам, из которых цифровые системы состоят.

### Структура множества допустимых входных взаимодействий

Каждый блок цифровой системы в процессе функционирования выполняет ту или иную последовательность функций (операций) из конечного алфавита функций  $K$ . Выполнение каждой функции вызывается одним из входных взаимодействий определенного класса, причем каждое входное взаимодействие этого класса содержит конечное число событий.

Обозначим  $f$  конечную последовательность функций, а  $F$  — в общем случае счетное множество конечных последовательностей  $f$ . Каждая последовательность функций  $f$ , начинающаяся с момента времени  $t = 0$  (например, включения питания), задается по крайней мере одним входным взаимодействием  $\mu^f \in M$ . Этот факт следует из того, что  $M$  содержит все допустимые входные взаимодействия для любой допустимой последовательности функций цифрового блока.

В большинстве случаев одни и те же функции могут выполняться с различными наборами данных, что обуславливает задание различными  $\mu$  одной и той же последовательности функций  $f$ . В связи с тем, что для различных экземпляров блока цифровой системы задержки выходных событий управления обменом

относительно входных событий различаются в определенных пределах, а также в связи с допустимостью варьирования моментов времени входных событий относительно друг друга и относительно выходных событий управления обменом, множество  $\mathbf{M}$  содержит континуальное подмножество  $\mathbf{M}^f \subset \mathbf{M}$  входных взаимодействий, каждое из которых вызывает выполнение цифровым блоком конечной последовательности функций  $f$ . Множество входных взаимодействий может быть представлено в виде

$$\mathbf{M} = \bigcup_{f \in \mathbf{F}} \mathbf{M}^f; \mathbf{M}^{f'} \cap \mathbf{M}^{f''} = \emptyset \text{ при } f' \neq f''. \quad (3)$$

Входное взаимодействие  $\mu \in \mathbf{M}^f$  содержит конечное множество событий

$$\{\chi_{p_1, z_{j_1}, z_{j_2}}, \dots, \chi_{p_n, z_{j_{2n-1}}, z_{j_{2n}}}\}, \quad (4)$$

где  $n$  — число событий в  $\mu$ .

Множество  $\mathbf{M}^f$  содержит также входные взаимодействия, времена событий в которых различаются в определенных пределах. Ограничения на эти различия могут быть заданы в следующем виде:

$$t_{\min}^{l,m} \leq t_m - t_l \leq t_{\max}^{l,m}, \quad (l, m) \in \mathbf{C}, \quad \mathbf{C} \subset \{1, 2, \dots, n\} \times \{1, 2, \dots, n\}, \quad (5)$$

где  $t_{\min}^{l,m}$ ,  $t_{\max}^{l,m}$  — минимально и максимально допустимые промежутки времени между  $l$ -м и  $m$ -м событиями;  $\mathbf{C}$  — конечное множество пар событий множества (2), для которых заданы временные ограничения.

Выделим все пары  $(l, m)$  в  $\mathbf{C}$ , для которых  $t_m$  есть время выходного события управления обменом в множество  $\mathbf{C}_{\text{вых}}$ , а все пары  $(l, m)$ , для которых  $t_m$  есть время входного события, — в множество  $\mathbf{C}_{\text{вх}}$ . Тогда ограничения на моменты времени выходных событий обмена имеют вид

$$t_{\min}^{l,m} \leq t_m - t_l \leq t_{\max}^{l,m}, \quad (l, m) \in \mathbf{C}_{\text{вых}}, \quad (6)$$

а ограничения на моменты времени входных событий

$$t_{\min}^{l,m} \leq t_m - t_l \leq t_{\max}^{l,m}, \quad (l, m) \in \mathbf{C}_{\text{вх}}. \quad (7)$$

Рассмотрим пространство

$$\mathbf{G} = \prod_{\mathbf{C}_{\text{вых}}} \{t_m - t_l | t_m - t_l \geq 0\}.$$

Каждая точка  $g \in \mathbf{G}$  определяет конкретные значения задержек выходных событий управления обменом. В пространстве  $\mathbf{G}$  выделим область  $\mathbf{G}_f \in \mathbf{G}$ , для всех точек которой выполняются ограничения (7). Область  $\mathbf{G}_f$  определяет допустимые задержки выходных событий управления обменом. При этом  $\mathbf{G}_f \neq \emptyset$ .

Для любой точки  $g \in \mathbf{G}_f$  в связи с допустимостью таких задержек выходных событий управления обменом существует непустое множество входных взаимодействий  $\mathbf{M}_g^f$ , обеспечивающих выполнение цифровым блоком последовательности функций  $f$ :

$$\mathbf{M}^f = \bigcup_{g \in \mathbf{G}_f} \mathbf{M}_g^f, \quad \mathbf{M}_g^f \neq \emptyset,$$

где  $\mathbf{M}_g^f$  — множество входных взаимодействий, обеспечивающих выполнение конечной последовательности функций  $f$  при фиксированных задержках выходных событий управления обменом, определяемых  $g \in \mathbf{G}_f$ .

Таким образом, множество допустимых входных взаимодействий представимо в виде

$$\begin{aligned} \mathbf{M} &= \bigcup_{f \in \mathbf{F}} \mathbf{M}^f; \mathbf{M}^{f'} \cap \mathbf{M}^{f''} \neq \emptyset \text{ при } f' \neq f''; \\ \mathbf{M}^f &= \bigcup_{g \in \mathbf{G}_f} \mathbf{M}_g^f; \mathbf{G}_f \neq \emptyset; \mathbf{M}_g^f \neq \emptyset \text{ при } f \in \mathbf{F}. \end{aligned} \quad (8)$$

Таким образом, мы определяем структуру множества допустимых входных взаимодействий, т. е. структуру возможных аргументов функционирования цифровых систем и их блоков.

### Формальная модель множества входных взаимодействий для каждой функции

Рассмотрим способ задания множества  $\mathbf{M}^k$  входных взаимодействий, обуславливающих выполнение цифровой системой или блоком функции  $k$ .

С учетом (2) и (4) каждое  $\mu$ ,  $\mu \in \mathbf{M}^k$ , может быть задано в виде

$$((z_{p_1}^H, \dots, z_{p_m}^H), \{(t_1, p_{i_1}, z'_{i_1}, z''_{i_1}), (t_2, p_{i_2}, z'_{i_2}, z''_{i_2}), \dots, (t_n, p_{i_n}, z'_{i_n}, z''_{i_n})\}), \quad (9)$$

где  $z_{p_1}^H, \dots, z_{p_m}^H$  — начальные значения переменных;  $t_i, p_{i_i}, z'_{i_i}, z''_{i_i}$  — четверка, описывающая  $i$ -е событие.

Исходя из этого попробуем задать все множество  $\mathbf{M}^k$  как

$$(\tilde{\mathbf{Z}}_{p_1}^H, \dots, \tilde{\mathbf{Z}}_{p_m}^H), \{(\theta_1, p_{i_1}, \tilde{\mathbf{Z}}'_1, \tilde{\mathbf{Z}}''_1), (\theta_2, p_{i_2}, \tilde{\mathbf{Z}}'_2, \tilde{\mathbf{Z}}''_2), \dots, (\theta_n, p_{i_n}, \tilde{\mathbf{Z}}'_n, \tilde{\mathbf{Z}}''_n)\},$$

$$t_{\min}^{l,q} \leq \theta_q - \theta_l \leq t_{\max}^{l,q}, (l, q) \in \mathbf{C},$$

где  $\theta_1, \theta_2, \dots, \theta_n$  — времена событий;  $\tilde{\mathbf{Z}}_{p_i}^H, i = 1, 2, \dots, m$  — подмножество возможных начальных значений переменных множества  $\mathbf{P}' \cup \mathbf{P}^0$ ;  $m$  — мощность множества  $\mathbf{P}' \cup \mathbf{P}^0$ ;  $p_{i_1}, p_{i_2}, \dots, p_{i_n}$  — принадлежат множеству  $\mathbf{P}' \cup \mathbf{P}^0$ ;  $\tilde{\mathbf{Z}}'_1, \tilde{\mathbf{Z}}'_2, \dots, \tilde{\mathbf{Z}}'_n$  — подмножества возможных значений переменных  $p_{i_1}, p_{i_2}, \dots, p_{i_n}$  перед событием;  $\tilde{\mathbf{Z}}''_1, \tilde{\mathbf{Z}}''_2, \dots, \tilde{\mathbf{Z}}''_n$  — подмножества возможных значений переменных  $p_{i_1}, p_{i_2}, \dots, p_{i_n}$  после события;  $\mathbf{C}$  — множество пар событий, для промежутков между которыми заданы временные ограничения.

Входное взаимодействие  $\mu$ , заданное в виде (9), принадлежит  $\mathbf{M}^k$ , определенному в виде (10), если:

- $z_{p_1}^H \in \tilde{\mathbf{Z}}_{p_1}^H, \dots, z_{p_m}^H \in \tilde{\mathbf{Z}}_{p_m}^H$ ;
- существует изоморфизм между множеством событий (9) и множеством четверок в (10), такой, что у соответствующих событий совпадают имена переменных:

$$z'_1 \in \tilde{\mathbf{Z}}'_1, \dots, z'_n \in \tilde{\mathbf{Z}}'_n, z''_1 \in \tilde{\mathbf{Z}}''_1, \dots, z''_n \in \tilde{\mathbf{Z}}''_n,$$

а времена событий  $t_1, \dots, t_n$  в (9) удовлетворяют ограничениям из (10).

Рассматриваемое представление  $\mathbf{M}^k$  и введение подмножеств  $\tilde{\mathbf{Z}}_{p_i}^H, \tilde{\mathbf{Z}}', \tilde{\mathbf{Z}}''$  позволяет пред-

ставить множество входных взаимодействий с различными наборами данных.

Пусть задано множество терминальных переменных  $\mathbf{P}$ , каждая переменная с множеством значений  $\mathbf{Z}_p$ . Для каждой переменной  $p$  рассмотрим некоторый алфавит значений

$$\mathfrak{Z}_p = \mathbf{Z}_p \cup \{z_{ij} | (z_i \in \mathbf{Z}_p) \& (z_j \in \mathbf{Z}_p) \& (z_i \neq z_j)\},$$

где  $z_{ij}$  — переход от  $z_i$  к  $z_j$ . В алфавите  $\mathfrak{Z}_p$  выделим непустые подмножества  $\mathfrak{Z}_l^p \subset \mathfrak{Z}_p$  такие, что  $\mathfrak{Z}_p = \bigcup \mathfrak{Z}_l^p$ . Например, если сигнал на шине может принимать значения  $0..255$ ,  $\$$  ( $\$$  — состояние с высоким выходным сопротивлением), то возможно использование следующих подмножеств  $\mathfrak{Z}_i$ :  $\mathfrak{Z}_1 = \{\$\}$ ,  $\mathfrak{Z}_2 = \{0, 1, \dots, 255\}$ ,  $\mathfrak{Z}_3 = \mathfrak{Z}_2 \cup \{z_{ij} | (z_i \in \mathfrak{Z}_2 \cup \mathfrak{Z}_1) \& (z_j \in \mathfrak{Z}_2 \cup \mathfrak{Z}_1)\}$ ,  $\mathfrak{Z}_4 = \mathfrak{Z}_1 \cup \mathfrak{Z}_2 \cup \mathfrak{Z}_3$ . Здесь  $\mathfrak{Z}_2$  — стабильное значение данных,  $\mathfrak{Z}_4$  — нестабильное значение.

Подмножества  $\mathfrak{Z}_l^p, l = 1, 2, \dots$ , образуют конечный алфавит  $\hat{\mathfrak{Z}}_p$ . Каждое множество  $\mathbf{M}^k$  может быть представлено как конечное множество событий в алфавитах  $\hat{\mathfrak{Z}}_p$ . Так как на множестве событий множества  $\mathbf{M}^k$  определен частичный порядок событий по времени, то естественным представлением  $\mathbf{M}^k$  является ориентированный граф  $\mathcal{G}^k(\mathbf{V}^k, \mathbf{E}^k)$ , где каждая вершина из  $\mathbf{V}^k$  соответствует переходу одной из переменных из одного значения  $\mathfrak{Z}_l^p$  в другое. Каждую вершину  $v, v \in \mathbf{V}^k$ , пометим обозначением переменной  $p$  и множествами  $\mathfrak{Z}_l^p, \mathfrak{Z}_m^p$ , если вершине  $v$  соответствует переход переменной  $p$  из значения  $\mathfrak{Z}_l^p$  в значение  $\mathfrak{Z}_m^p$ .

На множестве вершин  $\mathbf{V}_p^k$ , соответствующих изменениям значений одной и той же перемен-

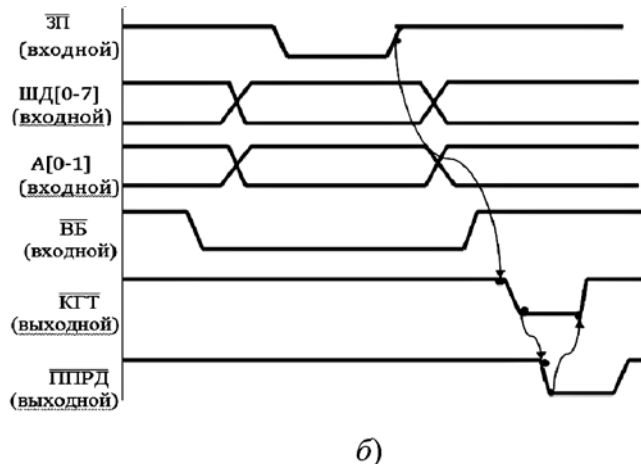
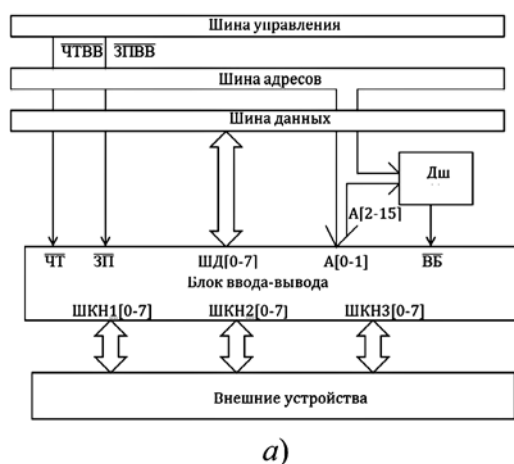


Рис. 1. Блок параллельного интерфейса ввода-вывода:

а — схема включения; б — временная диаграмма работы

ной  $p$ , задано отношение частично-порядка во времени, что определяет множество ребер  $E^k$ . Каждое ребро пометим двумя числами  $t_{\min}$ ,  $t_{\max}$ , причем  $0 \leq t_{\min} \leq t_{\max} \leq \infty$ .

В качестве примера рассмотрим некоторый цифровой блок параллельного интерфейса ввода-вывода (рис. 1, а), временные диаграммы работы которого приведены на рис. 1, б. При функционировании блока используются сигналы: ШД[0-7] — шина данных; А[0-1], А[2-15] — шины адреса соответствующей разрядности; СБР,  $\overline{CT}$ ,  $\overline{3П}$  — сигналы сброса, чтения и записи;  $\overline{ВБ}$ ,  $\overline{КГТ}$ ,  $\overline{ППРД}$  — сигналы выбора блока, канал готов, подтверждения передачи; ШКН1, ШКН2, ШКН3 — шины каналов 1, 2 и 3 с указанием разрядности.

На рис. 2 представлено графовое задание множества  $M^k$  входных взаимодействий, соответствующих записи информации в рассматриваемый цифровой блок. Для этого случая:

$$P = \{A, \text{ШД}, \overline{ВБ}, \overline{3П}, \overline{КГТ}, \overline{ППРД}\};$$

$$Z_A = \{0, 1, 2, 3, \$\}; Z_{\text{ШД}} = \{0, 1, \dots, 255, \$\};$$

$$Z_{\overline{ВБ}} = \{0, 1\}; Z_{\overline{3П}} = \{0, 1\}; Z_{\overline{КГТ}} = \{0, 1\};$$

$$Z_{\overline{ППРД}} = \{0, 1\};$$

$$\hat{Z}_A = \{C, \text{НС}, \text{ВИ}, X\}, C = \{0, 1, 2, 3\};$$

$$\text{НС} = C \cup \{\chi_{i,k} | z_i \neq z_k, z_i \in C, z_k \in C\};$$

$$\text{ВИ} = \{\$, \};$$

$$X = \text{НС} \cup \text{ВИ} \cup \{\chi_{i,\$} | z_i \in C\} \cup \{\chi_{\$,i} | z_i \in C\};$$

$$\hat{Z}_{\text{ШД}} = \{C, \text{НС}, \text{ВИ}, X\}, C = \{0, 1, \dots, 255\};$$

$$\text{НС} = C \cup \{\chi_{i,k} | z_i \neq z_k, z_i \in C, z_k \in C\};$$

$$\text{ВИ} = \{\$, \};$$

$$X = \text{НС} \cup \text{ВИ} \cup \{\chi_{i,\$} | z_i \in C\} \cup \{\chi_{\$,i} | z_i \in C\};$$

$$\hat{Z}_{\overline{ВБ}} = \{0, 1, X\}, X = \{0, 1, \beta_{0,1}, \beta_{1,0}\};$$

$$\hat{Z}_{\overline{3П}} = \{0, 1, X\}, X = \{0, 1, \beta_{0,1}, \beta_{1,0}\};$$

$$\hat{Z}_{\overline{КГТ}} = \{0, 1, X\}, X = \{0, 1, \beta_{0,1}, \beta_{1,0}\};$$

$$\hat{Z}_{\overline{ППРД}} = \{0, 1, X\}, X = \{0, 1, \beta_{0,1}, \beta_{1,0}\}.$$

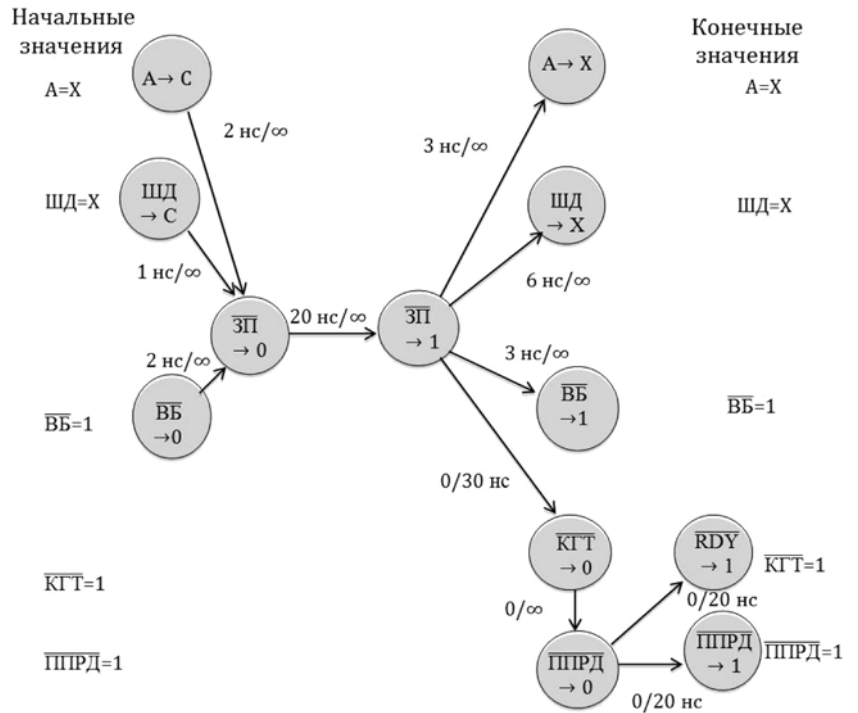


Рис. 2. Задание множества входных взаимодействий для функции записи информации в блок параллельного ввода-вывода

На рис. 2 вершины графа помечены обозначением переменной и новым ее значением. Если заданы начальные значения переменных в алфавите  $\hat{Z}_p$ , этого достаточно. При наличии графа  $G^k(V^k, E^k)$  указанного вида для каждого  $\mu$ , заданного в виде (9), можно определить, принадлежит ли  $\mu$  множеству  $M^k$  или нет.

Граф  $G^k(V^k, E^k)$  более наглядно задает множество  $M^k$ , чем временная диаграмма (см. рис. 1, б), которая обычно используется для представления режимов работы цифровых блоков.

### Заключение

Граф  $G^k(V^k, E^k)$  определяет множество входных взаимодействий для выполнения функции  $k$  цифровым блоком. Используя тот же алгоритм, можно построить аналогичный граф для каждой функции цифровой системы в целом.

Этот граф вместе с выражениями (3) и (8) определяет структуру множества допустимых входных взаимодействий как для каждого цифрового блока в отдельности, так и для цифровой системы в целом. Структура множества допустимых взаимодействий служит исходными данными для выбора набора тестов для отладки проектов цифровых систем методом моделирования.

## Список литературы

1. **Иванников А. Д., Стемповский А. Л.** Формализация задачи отладки проектов цифровых систем // Информационные технологии. 2014. № 9. С. 3—10.
2. **Lin Yi-Li, Su Alvin W. Y.** Functional Verification for SoC Software/Hardware Co-Design: From Virtual Platform to Physical Platform // 2011 IEEE International SOC Conference (SOCC). P. 201—206.
3. **Shi Jin, Liu Weichao, Jiang Ming** et al. Software Hardware Co-Simulation and Co-Verification in Safety Critical System Design // 2013 IEEE International Conference on Intelligent Rail Transportation (ICIRT). P. 71—74.
4. **Кашеев Н. И., Пономарев Д. М., Подъяблонский Ф. М.** Построение тестов цифровых схем с использованием обобщенной модели неисправностей и непрерывного подхода к моделированию // Вестник Нижегородского университета им. Н. И. Лобачевского. 2011. № 3 (2). С. 72—77.
5. **Иванников В. П., Камкин А. С., Косачев А. С., Кулямин В. В., Петренко А. К.** Использование контрактных спецификаций для представления требований и функционального тестирования моделей аппаратуры // Программирование. 2007. Т. 33, № 5. С. 47—62.
6. **Иванников А. Д.** Формирование отладочного набора тестов для проверки функций цифровых систем управления объектами // Мехатроника, автоматизация, управление. 2017. Т. 18, № 12. С. 795—801.
7. **Иванников А. Д., Стемповский А. Л.** Математическая модель отладки проектов сложных цифровых систем и микросистем на основе представления последних в виде семейства стационарных динамических систем // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2014. Часть II. С. 123—128.

**A. D. Ivannikov**, Doctor of Technical Sciences, Professor, **V. N. Severtcev**, Doctor of Technical Sciences, Institute for Design Problems in Microelectronics of Russian Academy of Sciences

## Mathematical Model for Digital System Input Interaction Set in the Process of Logical Simulation

*While digital system design debugging by computer simulation the important task is to generate debugging test set, e.g. set of input signals which are applied to a designing system computer model for checking the correctness of its functioning. The generation of complete in some sense debugging test set is possible by some way if the permissible input action set for the system is known. Description forming of such a set is possible if permissible input interaction set for digital system blocks are known. Digital system block model investigation is carried out, first of all, from the point of a set of permissible input interactions. The family of stationary dynamic systems with continuous time and logical signal discrete values are used as models for digital system blocks. In some cases signal exchange between blocks and with outer world is initiated by a block itself. That is why input interactions including input signals and output exchange driving signals are considered as debugging tests. For the description of permissible input interactions of digital system blocks and the system as a whole graph representation is proposed for each fulfilled function.*

**Keywords:** logical simulation, digital system design debugging, input interaction set, input interaction graph model

DOI: 10.17587/it.24.627-632

### References

1. **Ivannikov A. D., Stempkovsky A. L.** Formalizatsiya zadachi otladki projektov cifrovih sistem (Formal Model of Digital System Design Debugging Task), *Informacionnie Tekhnologii*, 2014, no. 9, pp. 3—10 (in Russian).
2. **Lin Yi-Li, Su Alvin W. Y.** Functional Verification for SoC Software/Hardware Co-Design: From Virtual Platform to Physical Platform, *2011 IEEE International SOC Conference (SOCC)*, pp. 201—206.
3. **Shi Jin, Liu Weichao, Jiang Ming** et al. Software Hardware Co-Simulation and Co-Verification in Safety Critical System Design, *2013 IEEE International Conference on Intelligent Rail Transportation (ICIRT)*, pp. 71—74.
4. **Kasheev N. I., Ponomarev D. M., Podyablonsky F. M.** Postroenie testov cifrovih chem. S ispolzovaniem obobshhenoi modeli neispravnostei i neprerivnogo podhoda k modelirovaniyu (Digital Circuits Test Generation Based on Generalized Malfunction Model and Continuous Simulation Approach), *Vestnik Nijegorodskogo Universiteta*, 2011, no. 3(2), pp. 72—77 (in Russian).
5. **Ivannikov V. P., Kamkin A. S., Kossachev A. S., Kulia-min V. V., Petrenko A. K.** Ispol'zovanie kontraktnyh spetsifikacij dlâ predstavleniâ trebovanij i funkcional'nogo testirovaniâ modelej apparatury (The use of contract specifications for representing requirements and for functional testing of hardware models), *Programirovanie*, 2007, v. 33, no. 5, pp. 47—62 (in Russian).
6. **Ivannikov A. D.** Formirovanie otladochnogo nabora testov dlya proverki funkciy cifrovih sistem upravleniya obektami (Debugging Input Set Generation for Testing of Control Digital Systems Functions), *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2017, vol. 18, no.12, pp. 795—801 (in Russian).
7. **Ivannikov A. D., Stempkovsky A. L.** Matematicheskaya model otladki projektov slojnih cifrovih sistem i mikrosistem na osnove predstavleniya poslednih v vide semeistva stacionarnih dinamicheskikh sistem (Design Debugging Mathematical Model for Complex Digital and Microsystems on the Basis of Their Representation as a Family of Stationary Dynamic Systems), *Problemi Rasrabotki Perspektivnih Mikro- i Naneolectronnih System (MES)*, 2014, part II, pp. 123—128 (in Russian).



С. И. Попков, аспирант, e-mail: rslw25@gmail.com, ГБОУ ВПО Московский государственный психолого-педагогический университет, Москва, Россия

## Метод внешней оптимизации для идентификации марковских процессов

*Разработан численный метод для оптимизации идентификации марковских процессов. Метод применяется как вспомогательный в тех случаях, когда необходимо обеспечить высокую эффективность основного метода за ограниченное или реальное время. В качестве примера такого основного метода в данной статье выбран метод перебора значимых параметров. В основу программно-алгоритмической реализации метода внешней оптимизации положен шаблон проектирования "декоратор", позволяющий модифицировать входные и выходные данные преобразуемой функции в целях получения некоторых преобразованных данных, в данном случае — улучшенного значения критерия за ограниченное время. Предложенный подход позволил увеличить эффективность метода перебора значимых параметров более чем на 25 % (по сравнению с эмпирическим подходом к подбору параметров).*

**Ключевые слова:** численный метод, декоратор, марковский процесс, перебор значимых параметров

### Введение

В последние годы возросла актуальность задач, связанных с управлением группами подвижных роботов, которые должны координировать свое поведение для достижения заданного результата. Концепции, базовые понятия, принципы построения, перспективы практического применения и другие аспекты создания подобных систем управления были рассмотрены в ряде работ [1—9]. Однако достаточно развитый математический аппарат, пригодный для практического управления поведением группы агентов, в настоящее время все еще формируется.

Поведение ряда технических, диагностических и других систем удобно описывать, используя специально разработанные параметрические математические модели в форме систем обыкновенных дифференциальных уравнений. Подобная форма представления применялась для ряда моделей, среди которых можно выделить модели усталостного разрушения, адаптивного тестирования и различные модели прикладных многоагентных систем, построенных на базе марковских случайных процессов с дискретными состояниями и непрерывным временем [10—13]. Указанные многоагентные системы, в частности, решают задачу управления группами подвижных объектов, которые должны координировать свое поведение в пространстве и кооперироваться для достижения заданного результата. Область применения таких систем включает управление полетом и

выполнением миссии группы беспилотных летательных аппаратов, групповое управление роботами, а также других мобильных систем.

Важной особенностью практического использования марковских моделей применительно к указанным задачам является тот факт, что их параметры идентифицируются по результатам наблюдений, представленных наборами значений исследуемых функций в контрольных точках.

Марковские модели для описания динамики переходов между состояниями представляются ориентированными графами, в которых вершины соответствуют состояниям, а дуги — переходам, для которых выполняются свойства пуассоновских потоков событий. В этих потоках число событий  $X$ , попадающих в любой временной интервал длины  $\tau$ , начинающийся в момент  $t$ , распределено согласно закону Пуассона:

$$P_{t,\tau}(X = n) = \frac{a(t, \tau)^n}{n!} e^{-a(t, \tau)},$$

где  $P_{t,\tau}(X = n)$  — вероятность появления  $n$  событий в течение рассматриваемого интервала;  $a(t, \tau)$  — среднее число событий, попадающих в интервал длины  $\tau$ , начинающийся в момент времени  $t$ . Рассматриваются только стационарные потоки, в которых  $a(t, \tau) = \eta\tau$ , а  $\eta = \text{const}$  есть интенсивность стационарного потока. Упомянутые выше предположения о свойствах потоков событий обычны для прикладных задач [14], так как эти потоки (или потоки, близкие к ним по свойствам) часто встречаются на

практике благодаря предельным теоремам для потоков событий.

Полагается, что:

- для указанных процессов с  $n + 1$  дискретными состояниями и непрерывным временем заданы начальные распределения вероятностей и наблюдаемые частоты пребывания в состояниях процессов  $\{F_{id}\}_{i=0, \dots, n}$  в моменты времени  $\{t_d\}_{d=0, \dots, D-1}$ , где  $D$  — число моментов времени, в которые фиксировались частоты  $F_{id}$ ;  $0 \leq t_d \leq T$ , где  $T$  — конечный момент времени;
- интенсивности переходов между состояниями полностью или частично являются неизвестными (свободными) параметрами.

Динамика изменения вероятностей пребывания в состояниях процесса определяется системой обыкновенных дифференциальных уравнений Колмогорова в матричной форме:

$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{M}(\lambda)\mathbf{p},$$

где  $0 \leq t \leq T$ ;  $\mathbf{p}(t) = (p_0(t), \dots, p_n(t))^T$  — вероятности пребывания в состояниях процесса;  $\lambda = (\lambda_0, \dots, \lambda_m)^T$  — упорядоченный набор интенсивностей переходов между состояниями,  $n > m$ ;  $\mathbf{M}$  — матрица интенсивностей переходов между состояниями порядка  $n + 1$ . Для указанной системы уравнений ставится задача идентификации набора параметров  $\lambda$ . Значения параметров из этого набора определяются путем сравнения наблюдаемых и прогнозируемых гистограмм, описывающих распределения частот пребывания в состояниях модели, а именно: вычисляются значения, обеспечивающие наилучшее соответствие наблюдаемых и ожидаемых частот попадания в определенное состояние системы в заданные моменты времени.

Для решения этих проблем был предложен численный метод *перебора значимых параметров* [15], предназначенный для идентификации набора параметров  $\lambda$ , определяющих матрицу  $\mathbf{M}$ . Этот метод представлен далее в версии, совместимой с задачей моделирования прикладных многоагентных систем, упомянутых выше.

Несмотря на то что скорость оптимизации с применением описанного алгоритма, как правило, близка к приемлемой, существует целый ряд задач, для которых одним из требований является мгновенное принятие решений. Цель данной работы — разработать и представить математический метод, позволяющий адаптировать параметры основного метода оптимизации

для повышения эффективности его работы, а также продемонстрировать реализацию нового метода в виде алгоритма вычислений и в качестве действующего программного инструмента.

### Метод перебора значимых параметров. Алгоритм вычислений

1. Используя имеющиеся результаты наблюдений, вычислить в качестве начальных оценок параметров  $\lambda_i (i = 0, \dots, m)$ , обозначаемых  $\lambda_i^0 (i = 0, \dots, m)$ , случайно взятые значения интенсивностей переходов между парами соответствующих состояний марковского процесса в единицу времени. Полученные оценки рассматривать как начальные приближения к идентифицируемому параметрам.

2. Положить  $j = 1, k = 1, \beta_1 = 1 + \gamma_1$ , где  $\gamma_1$  — параметр алгоритма.

3. Оценить чувствительность критерия  $X^2$  в малой окрестности каждой из текущих оценок идентифицируемых параметров  $\lambda_i^j (i = 0, \dots, m)$  на  $j$ -й итерации алгоритма, используя для этого разностные аппроксимации абсолютных значений частных производных

$$\delta_i(\varepsilon) = |X^2(\lambda_0^j, \dots, \lambda_i^j + \varepsilon, \dots, \lambda_m^j) - X^2(\lambda_0^j, \dots, \lambda_i^j, \dots, \lambda_m^j)|/\varepsilon,$$

где  $i = 0, \dots, m$ ;  $\varepsilon$  — параметр алгоритма.

4. Выбрать  $l$  текущих оценок идентифицируемых параметров  $\lambda_{i_q}^j$ , где  $i_q \in \{0, \dots, m\}$ ;  $q = 1, \dots, l$ ;  $l$  — параметр алгоритма, соответствующий оценкам, имеющим наибольшие значения оценок чувствительности  $\delta_i(\varepsilon)$ .

5. Для каждой из выбранных на шаге 4 текущих оценок идентифицируемых параметров  $\lambda_{i_q}^j$  на  $j$ -й итерации алгоритма, где  $i_q \in \{0, \dots, m\}$ ,  $q = 1, \dots, l$ , вычислить верхнюю  $\lambda_{i_q}^{j+}$  и нижнюю  $\lambda_{i_q}^{j-}$  границу сдвига оценки по формулам:

$$\lambda_{i_q}^{j+} = \beta_j \lambda_{i_q}^j, \lambda_{i_q}^{j-} = \beta_j^{-1} \lambda_{i_q}^j.$$

6. Выполнив полный перебор всех вариантов текущих оценок набора параметров  $\lambda^j = (\lambda_0^j, \dots, \lambda_m^j)^T$ , в которых каждый из его компонентов  $\lambda_{i_q}^j$ , где  $i_q \in \{0, \dots, m\}$ ,  $q = 1, \dots, l$ , принимает только три возможных значения из множества  $\{\lambda_{i_q}^{j-1}, \lambda_{i_q}^j, \lambda_{i_q}^{j+1}\}$ , а оставшиеся компоненты — значение своей текущей оценки на  $j$ -й итерации алгоритма, выбрать из указанных вариантов одну из оценок  $\lambda^{j*} = (\lambda_0^{j*}, \dots, \lambda_m^{j*})^T$ , обеспечивающих максимальное значение критерия  $X^2$ , равное  $X_{j, \max}^2$ .

7. Если для всех компонентов выбранного набора  $\lambda^{j,*}$  выполняется равенство  $\lambda_i^{j,*} = \lambda_i^{j-1}$  ( $i = 0, \dots, m$ ), то положить  $k = k + 1$ ,  $\gamma_k = \gamma_{k-1}/2$ .

8. Положить  $j = j + 1$ ,  $\beta_j = 1 + \gamma_k$ ,  $\lambda^j = \lambda^{j-1,*}$ .

9. Если  $|X_{j-1, \min}^2 - X_*^2| > \varepsilon_t$ , где  $X_*^2, \varepsilon_t$  — параметры алгоритма, то перейти к шагу 3, иначе завершить вычисления.

### Комментарий к представленному алгоритму вычислений

Для решаемых на практике типовых задач общее время вычислений зависит от параметра  $l$  монотонно, достигая наименьшего значения при  $l = 1$ . Временная сложность одной итерации модифицированного алгоритма при  $l = 1$  есть  $O(m)$ , что свидетельствует об асимптотически линейной зависимости времени вычислений, необходимых для выполнения одной итерации, от числа идентифицируемых параметров (рис. 1).

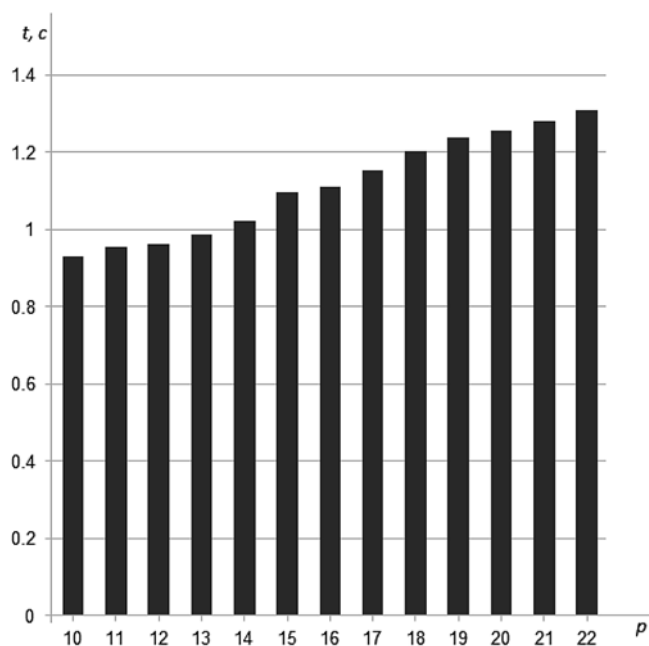


Рис. 1. Эмпирическая зависимость времени выполнения алгоритма  $t$  от числа параметров  $p$

### Введение терминологии для нового математического метода

Приведенный выше метод оптимизации (и соответствующий ему алгоритм) будем условно называть *основным*.

Параметры, существенно влияющие на качество и время работы основного алгоритма, но не относящиеся к числу оптимизируемых основным методом значимых параметров, будем называть *метапараметрами основного алгоритма*.

Эффективность основного алгоритма за одну итерацию можно рассматривать как отношение значения максимизируемого критерия ко времени, затраченному на работу алгоритма, при условии, что исходные значимые параметры остаются неизменными между итерациями (сохраняют свои изначальные значения в качестве входных параметров основного алгоритма). Тогда, адаптивно меняя метапараметры основного алгоритма оптимизации, оказывающие наибольшее влияние на рост его эффективности, можно повысить эффективность основного математического метода в целом.

Эмпирически было определено, что такими метапараметрами являются:

- $M$  — число варьируемых значимых параметров;
- $\Delta$  — начальная поправка для значимых параметров.

Алгоритм-надстройку, осуществляющую модификацию метапараметров основного алгоритма в целях увеличения критерия эффективности, будем называть *внешним алгоритмом оптимизации*.

Для реализации соответствующего численного метода необходимо решить следующие задачи:

1. Каким образом будет осуществлено взаимодействие с основным методом?
2. Как алгоритмически будет представлен процесс адаптивного последовательного изменения метапараметров?

### Взаимодействие между методами: шаблон "декоратор"

Алгоритмический шаблон проектирования "декоратор" описывает программную функцию высшего порядка, которая на основе получаемых значений аргументов, включая указатель на модифицируемую функцию, преобразует процесс работы последней, внося изменения в ее входные параметры и анализируя (с последующим внесением необходимых изменений) выходные данные.

В отношении описанного выше метода оптимизации можно рассматривать алгоритм метода

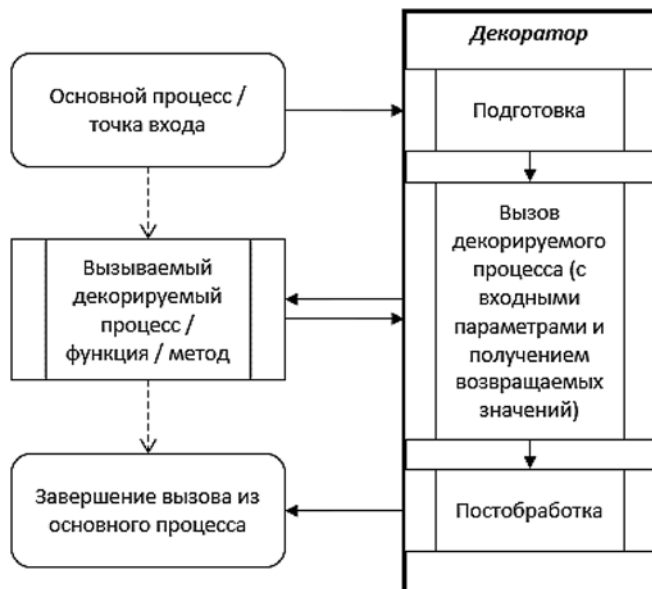


Рис. 2. Общий пример вызова декоратора из произвольного основного процесса. В качестве декорируемого процесса может выступать произвольная функция, метод или внешний процесс, которому временно передается управление из вызывающего процесса

перебора значимых параметров как модифицируемую функцию, для которой в качестве изменяемых входных данных задаются соответствующие значения изменяемых метапараметров.

Общая иллюстрация способа применения шаблона "декоратор" в ходе вызова декорируемой функции представлена на рис. 2.

Условно процесс можно разделить на три части: подготовка к вызову декорируемого процесса, непосредственно вызов и постобработка.

Подготовка заключается в осуществлении действий, предвещающих запуск вызываемого (также называемого декорируемым) процесса. Это может быть вывод сообщения для отладки, предварительное сохранение значимых данных перед осуществлением транзакции, представленной декорируемым процессом, подготовка промежуточных вычислений на основе входных аргументов и параметров, получаемых процессом локально или из глобального контекста, а также другие процедуры, осуществляющие инициализацию модифицированного окружения и контекста вызываемого процесса для достижения конкретной поставленной перед декоратором задачи.

Вызов декорируемого процесса может сопровождаться как изменением входных аргументов и возвращаемых значений, так и их сохранением в случае, если воздействие на

окружение процесса осуществлено каким-то другим образом (например, через вызов другого процесса во время подготовки декоратора).

Постобработка — это процесс модификации выходных данных либо применения их в промежуточных вычислениях для анализа и извлечения результатов, определяющих работу декоратора.

Примером автономной программной реализации такого шаблона можно считать приложение, вызывающее другое приложение, библиотечную функцию или системный вызов.

### Описание процесса адаптивного изменения метапараметров

Надстройка алгоритма является самостоятельной функцией, выполненной в форме шаблона проектирования "декоратор" для объектно-ориентированного метода, представляющего собой реализацию алгоритма оптимизации для заданного марковского процесса либо группы процессов. Среди всех задаваемых аргументов функции последовательно изменяются параметры модели, отвечающие за число изменяемых параметров и значение начального сдвига параметров.

Представим "сетку" — двумерное пространство, в котором по одной оси с фиксированным шагом  $h_1$  задается число одновременно изменяемых параметров от 1 до  $N$ , а по другой также с фиксированным шагом  $h_2$  определяется начальный сдвиг искомых параметров (от  $h_2$  до 1). Полученное пространство позволяет определить значение критерия эффективности метода в каждой ячейке "сетки" — в зависимости от конкретного значения модифицируемых параметров.

Описываемую сетку можно представить в виде, показанном на рис. 3.

Время, затрачиваемое на эмпирический поиск наиболее оптимального решения (например, посредством полного перебора), значительно возрастает с увеличением сложности модели, делая такой поиск непригодным на практике. В связи с этим наблюдением возникает необходимость в численном методе, позволяющем качественно улучшить эффективность основного метода, при этом адаптирующемся под предельно допустимое время, как выделяемое для самого процесса поиска оптимального решения, так и затрачиваемое на каждый шаг алгорит-

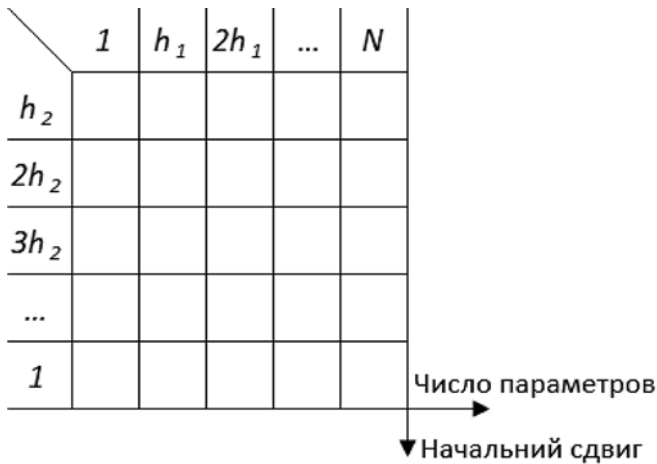


Рис. 3. Двумерное пространство параметров для проведения процедуры оптимизации

ма оптимизации, для которого осуществляется подбор оптимальных параметров запуска.

Такой метод был разработан, его алгоритм представлен ниже.

### Метод внешней оптимизации. Алгоритм вычислений

1. Задается минимальное число изменяемых параметров ( $N \geq 1$ ), максимальное число изменяемых параметров ( $U \leq$  число параметров), шаг сетки по параметрам ( $S$ ), шаг сетки ( $D$ ), предельный срок вычислений ( $H$ ), максимальное время ( $T$ ).

2. Генерируется сетка параметров  $M, d$  (число изменяемых элементов, начальный шаг),  $N \leq M \leq U$ ,  $0 \leq d \leq 1$ , шаг  $M = S$ , шаг  $d = D$ .

3. На сетке выбирается произвольная начальная позиция  $M_0, d_0$  и инициализируется словарь пути  $Z$  и массив оптимального результата  $R$ , хранящий данные о координатах и значениях словаря пути  $Z$ .

4. В окрестностях  $M_0, d_0$  выбирается множество точек  $\{M_x, d_y\}$ , такое, что каждая координата  $M_x$  принадлежит отрезку  $[M_{0-1} \dots M_{0+1}]$ , каждая координата  $d_y$  принадлежит отрезку  $[d_{0-1} \dots d_{0+1}]$ .

5. Инициализируется массив результатов измерений  $P$  для хранения вещественных значений с числом элементов, соответствующим числу точек во множестве  $\{M_x, d_y\}$ .

6. Для каждой точки, принадлежащей множеству  $\{M_x, d_y\}$ , выполняется следующая последовательность действий:

6.1. Если координата выбираемой точки принадлежит диапазону соответствующих зна-

чений, определенных на шаге 1, то перейти к шагу 6.4.

6.2. Если координата выбираемой точки имеет значение меньше нуля, то присвоить такой координате значение, равное максимальному значению из допустимого диапазона; повторить для каждой координаты.

6.3. Если координата выбираемой точки имеет значение больше максимально допустимого для соответствующего диапазона, то присвоить такой координате значение, равное минимальному значению из допустимого диапазона; повторить для каждой координаты.

6.4. Если точка уже была занесена в словарь пути  $Z$  в качестве ключа, то считать значение с соответствующим ключом из словаря и перейти к шагу 6.8.

6.5. Используя имеющийся основной алгоритм оптимизации с параметрами, соответствующими текущему положению сетки в заданной точке, выполнить процедуру оптимизации до критерия.

6.6. Время, потраченное на выполнение алгоритма в пункте 6.2, а также достигнутый уровень значения критерия в ходе выполнения алгоритма фиксируются в словаре пути  $Z$ , где в качестве ключа словаря используется искомая точка.

6.7. Потраченное время прибавить к общему времени  $H_0$ , затраченному на выполнение алгоритма оптимизации во всех искомых точках.

6.8. Занести значение критерия для текущих координат в массив результатов измерений  $P$  для соответствующего значения координат заданной точки.

6.9. Если все точки из множества  $\{M_x, d_y\}$  перебраны, то переход на шаг 7; иначе — переход на шаг 6.1.

7. Осуществляется поиск наибольшего значения критерия в массиве результатов измерений  $P$  и вычисляются соответствующие данному значению координаты  $M_{\max}, d_{\max}$ .

8. Для значения координат  $M_{\max}$  и  $d_{\max}$ , используемых в качестве ключей словаря пути  $Z$ , извлекаются соответствующие значения затраченного времени и уровня критерия; если затраченное время превышает порог максимального времени  $T$ , отводимого на один прогон алгоритма оптимизации с параметрами, то переход на шаг 11.

9. Внешний критерий определяется как функция  $f$  от затраченного времени  $T_0$  на выполнение вложенного алгоритма оптимизации и значение полученного критерия  $C_0$ :  $f(C_0, T_0) = C_0/T_0$ .

10. Если массив оптимального результата  $R$  пуст или значение найденного внешнего критерия в массиве  $R$  не превышает соответствующего значения критерия для словаря пути  $Z$  с ключом  $(M_{\max}, d_{\max})$ , то новый набор значений массива  $R$  задается эквивалентным совокупности ключей словаря пути  $Z$  и соответствующему значению внешнего критерия для координат  $M_{\max}$  и  $d_{\max}$ .

11. Если  $M_{\max} == M_0$  и  $d_{\max} == d_0$ , то переход на шаг 12, иначе переход на шаг 15.

12. Задаются случайные координаты сетки  $M_r$  и  $d_r$ .

13. Если словарь пути  $Z$  содержит в качестве ключа координаты сетки  $M_r$  и  $d_r$ , а размерность пространства ключей словаря не превышает значений соответствующих параметров, заданных на шаге 1, то перейти на шаг 12.

14. Присвоить  $M_0 = M_r$ ,  $d_0 = d_r$ .

15. Если время  $H_0$ , затраченное на выполнение алгоритма оптимизации на всем ансамбле измеренных параметров, превышает пороговое значение предельного срока вычислений  $H$ , то перейти на шаг 16, иначе перейти на шаг 4.

16. Если массив оптимального результата  $R$  пуст, то перейти на шаг 18.

17. Вернуть в качестве результата выполнения алгоритма массив оптимального результата  $R$  и перейти на шаг 19.

18. Вернуть внутренний код сообщения об ошибке "Начальные параметры заданы неверно".

19. Завершение работы алгоритма.

### Комментарий к алгоритму

В терминах шаблона проектирования "декоратор" описанный выше алгоритм можно разбить на три части, подробно описанные выше: подготовка (шаги 1—6.4), вызов декорируемого процесса (шаг 6.5) и постобработка (шаги 6.6—19).

Подготовка перед запуском декорируемого процесса — основного метода оптимизации — заключается в инициализации сетки допустимых значений изменяемых параметров декорируемого процесса перед его вызовом на основе параметров для процесса-декоратора, описывающего "внешний" метод оптимизации. Инициализируются служебные структуры хранения данных для фиксации траектории движения внешнего метода оптимизации по сетке параметров в целях достижения оптимального результата работы вызываемого процесса на основе принципов, представленных ранее. За-

пускается цикл с постусловием, позволяющий осуществлять проход по пространству возможных значений параметров, которое ограничено узлами предварительно сгенерированной сетки. Устанавливается число одновременно изменяемых параметров и начальный сдвиг искомых параметров в соответствии с условиями, определенными для текущей итерации. Также выполняются действия, программная реализация которых варьируется в зависимости от среды выполнения (архитектуры вычислительной системы, операционной системы и т. п.), суть которых сводится к обеспечению работы таймера для проведения измерения временного интервала, который необходим для работы основного метода оптимизации при заданных параметрах.

После проведения этих действий осуществляется вызов декорируемого процесса, которым является основной метод оптимизации. Эта процедура необходима для получения значения возвращаемого критерия при заданных параметрах и возможности провести измерение временного интервала, в ходе которого осуществляется искомый вызов.

Постобработка заключается в фиксации значений, полученных в качестве результата вызова декорируемого процесса в целях последующего отбора наилучших результатов в ходе работы цикла, а также определения критериев останова алгоритма. Также в ходе постобработки фиксируются возможные ошибки при задании начальных параметров работы внешнего алгоритма оптимизации, который сообщает пользователю о недостижимости поставленной цели при текущих условиях выполнения.

Данный метод может быть применен не только для улучшения представленного метода оптимизации, но и для произвольного численного метода, быстроедействие и эффективность которого зависят от подчиняющихся модификации параметров. Метод может быть расширен для  $N$ -мерного пространства параметров за счет увеличения числа шагов, обеспечивающих поиск вокруг текущей точки.

### Пример практического применения

Пример запуска и работы программной реализации метода на практическом примере показаны на рис. 4.

Задается предел времени (в секундах) для всех итераций поиска и для каждого шага. На



тематический метод может повысить эффективность работы произвольного алгоритма оптимизации за счет адаптивного изменения его метапараметров.

### Основные результаты и выводы

1. Исследован метод перебора значимых параметров в реализации, ориентированной на решение задачи моделирования прикладных многоагентных систем. Определены достоинства и недостатки метода, проиллюстрированы эмпирическими результатами.

2. Предложен и разработан новый математический метод, позволяющий адаптировать метапараметры основного метода оптимизации для повышения эффективности его работы.

3. Изучен и применен шаблон проектирования "декоратор", позволяющий использовать новый метод в качестве надстройки над произвольным методом оптимизации (в частности — над методом перебора значимых параметров, рассмотренным выше) для увеличения критерия эффективности при минимизации временных затрат на поиск оптимального критерия.

4. Создан и применен подход к адаптивному изменению метапараметров алгоритма оптимизации на основе "сетки" — двумерного пространства для определения значения критерия эффективности метода. Проанализированы проблемы, связанные с эмпирическим поиском оптимального значения критерия эффективности.

5. Продемонстрирована реализация нового метода в виде алгоритма вычислений и результатов работы соответствующей программной реализации.

6. Проанализированы подходы к возможному практическому применению нового метода в качестве действующего программного инструмента.

7. Представлены иллюстрации, отражающие влияние адаптивных параметров на результат работы алгоритма метода перебора значимых параметров с точки зрения эффективности.

8. Реализована поддержка ряда современных редакторов электронных таблиц для проведения детального анализа промежуточных результатов работы метода за счет механизма сериализации в специальный формат XML, совместимый с LibreOffice Calc, Microsoft Excel, WPS Spreadsheets и другими редакторами.

9. Анализ результатов проведенных вычислительных экспериментов показал, что применение разработанного метода в качестве надстройки над методом перебора значимых параметров увеличивает эффективность последнего на 26,32 %.

*Автор выражает благодарность своему научному руководителю, д. т. н. проф. Льву Семеновичу Куравскому.*

### Список литературы

1. **Городецкий В. И., Карсаев О. В., Самойлов В. В., Себряков С. В.** Прикладные многоагентные системы группового управления // Искусственный интеллект и принятие решений. 2009. № 2. С. 3—24.
2. **Осипов Г. С.** Динамические интеллектуальные системы // Искусственный интеллект и принятие решений. № 1. 2008. С. 47—54.
3. **Осипов Г. С.** Методы искусственного интеллекта. М.: Физматлит, 2011. 296 с.
4. **Aras R., Dutech A., Charpillat F.** Cooperation through communication in decentralized Markov games // International Conference on Advances in Intelligent Systems — Theory and Applications — AISTA'2004. Luxembourg-Kirchberg/Luxembourg, 2004.
5. **Boutilier C.** Planning, learning and coordination in multiagent decision processes // Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge, Morgan Kaufmann Publishers Inc. 1996. P. 195—210.
6. **Claus C. & Boutilier C.** The dynamics of reinforcement learning in cooperative multiagent systems // Proceedings of the National Conference on Artificial Intelligence. John Wiley & Sons Ltd. 1998. P. 746—752.
7. **Miikkulainen R.** Creating Intelligent Agents in Games (2006). The Bridge, 5—13, 2006.
8. **Owen G.** Game Theory. Academic Press, 1995.
9. **Read M., Möslinger Ch., Dipper T., Kengyel D., Hilder J., Thenius R., Tyrrell A., Timmis J., Schmickl T.** Profiling Underwater Swarm Robotic Shoaling Performance using Simulation // Proceedings of TAROS 2013. 2013. P. 456—462.
10. **Куравский Л. С., Марголис А. А., Юрьев Г. А., Мармалюк П. А.** Концепция системы поддержки принятия решений для психологического тестирования // Психологическая наука и образование. 2012. № 1. С. 56—65.
11. **Куравский Л. С., Мармалюк П. А., Алхимов В. И., Юрьев Г. А.** Математические основы нового подхода к построению процедур тестирования // Экспериментальная психология. 2012. Т. 5, № 4. С. 75—98.
12. **Куравский Л. С., Марголис А. А., Мармалюк П. А., Юрьев Г. А., Думин П. Н.** Обучаемые марковские модели в задачах оптимизации порядка предъявления психологических тестов // Нейрокомпьютеры: разработка и применение. 2013. № 4. С. 28—38.
13. **Марковские модели в задачах диагностики и прогнозирования: учеб. пособ. / Под ред. Л. С. Куравского.** М.: Изд-во МГППУ, 2017. 203 с.
14. **Овчаров Л. А.** Прикладные задачи теории массового обслуживания. М.: Машиностроение, 1969. 324 с.
15. **Куравский Л. С., Мармалюк П. А., Юрьев Г. А., Думин П. А.** Численные методы идентификации марковских процессов с дискретными состояниями и непрерывным временем // Математическое моделирование. 2017. № 5. С. 133—146.



## The Method of External Optimization for Methods of Markov Processes Identification

*A numerical method for optimization of the identification of Markov processes is developed. The method is used as an additional one, in cases where it is necessary to ensure high efficiency of the primary method for a limited or real time. As an example of such primary method in this article the method of brute force of significant parameters is chosen. The basis of algorithmic and software implementation of the method of external optimization is based on the design pattern "decorator" allowing to modify the input and output of transformed function for the purpose of receiving the transformed data, such as the improved value of the criteria in a given time. The new approach allowed to increase the efficiency of the method of brute force of significant parameters by more than 25 % (compared with the empirical approach to the selection of parameters).*

**Keywords:** numerical method, decorator, Markov process, brute force of significant parameters

DOI: 10.17587/it.24.633-641

### References

1. Gorodeckij V. I., Karsaev O. V., Samojlov V. V., Serebrjakov S. V. Prikladnye mnogoagentnye sistemy gruppovogo upravlenija (Applied multiagent systems with group control), *Iskusstvennyj Intellekt i Prinjatje Reshenij*, 2009, no. 2, pp. 3–24 (in Russian).
2. Osipov G. S. Dinamicheskie intellektual'nye sistemy (Dynamic intelligent systems), *Iskusstvennyj Intellekt i Prinjatje Reshenij*, 2008, no. 1, pp. 47–54 (in Russian).
3. Osipov G. S. *Metody iskusstvennogo intellekta* (Methods of Artificial Intelligence), Moscow, Fizmatlit, 2011, 296 p. (in Russian).
4. Aras R., Dutech A., Charpillet F. Cooperation through communication in decentralized Markov games, *In International Conference on Advances in Intelligent Systems — Theory and Applications — AISTA'2004*, Luxembourg-Kirchberg/Luxembourg, 2004.
5. Boutilier C. Planning, learning and coordination in multiagent decision processes, *In Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, Morgan Kaufmann Publishers Inc., 1996, pp. 195–210.
6. Claus C. & Boutilier C. The dynamics of reinforcement learning in cooperative multiagent systems, *In Proceedings of the National Conference on Artificial Intelligence*, John Wiley & Sons Ltd., 1998, pp. 746–752.
7. Miikkulainen R. Creating Intelligent Agents in Games, *The Bridge*: 5–13, 2006.
8. Owen G. *Game Theory*, Academic Press, 1995.
9. Read M., Möslinger Ch., Dipper T., Kengyel D., Hilder J., Thenius R., Tyrrell A., Timmis J., Schmickl T. Profiling Underwater Swarm Robotic Shoaling Performance using Simulation, *In Proceedings of TAROS 2013*, 2013, pp. 456–462.
10. Kuravskij L. S., Margolis A. A., Jur'ev G. A., Marmaljuk P. A. Koncepcija sistemy podderzhki prinjatija reshenij dlja psihologicheskogo testirovanija (Conception of decision-making support system for psychological testing), *Psihologicheskaja Nauka i Obrazovanie*, 2012, no.1, pp. 56–65 (in Russian).
11. Kuravskij L. S., Marmaljuk P. A., Alhimov V. I., Jur'ev G. A. Matematicheskie osnovy novogo podhoda k postroeniju procedur testirovanija (Mathematical basics of new approach for building of testing procedures), *Jeksperimental'naja Psihologija*, 2012, vol. 5, no. 4, pp. 75–98 (in Russian).
12. Kuravskij L. S., Margolis A. A., Marmaljuk P. A., Jur'ev G. A., Dumin P. N. Obuchaemye markovskie modeli v zadachah optimizacii porjadka pred#javlenija psihologicheskikh testov (Learning Markov models in tasks of order optimization for psychological tests presentation), *Nejrokomp'jutery: Razrabotka i Primenenie*, 2013, no. 4, pp. 28–38 (in Russian).
13. Kuravskii L. S. ed. *Markovskie modeli v zadachah diagnostiki i prognozirovanija: Ucheb. posobie* (Markov models in tasks of diagnostics and prognosis: a tutorial / ed. by L. S. Kuravsky), Moscow, Publishing house of MGPPU, 2017, 203 p. (in Russian).
14. Ovcharov L. A. *Prikladnye zadachi teorii massovogo ob-sluzhivaniya* (Applied tasks in queueing theory), Moscow, Mashinostroenie, 1969, 324 p. (in Russian).
15. Kuravskij L. S., Marmaljuk P. A., Jur'ev G. A., Dumin P. A. Chislennye metody identifikacii markovskih processov s diskretnymi sostojanijami i nepreryvnym vremenem (Numerical methods of identification for Markov processes with discrete states and continuous time), *Matematicheskoe modelirovanie*, 2017, no. 5, pp. 133–146. (in Russian).

**В. А. Харахинов**, аспирант, e-mail: tes4obse@mail.ru,  
Иркутский национальный исследовательский технический университет

## Генетический алгоритм как альтернатива обучения слоя Кохонена

*Предложено применение генетического алгоритма в качестве альтернативного метода обучения слоя Кохонена. Приведено сравнение оценок качества кластерного анализа, полученных следующими методами: K-means; слой Кохонена, обученный стандартным алгоритмом; слой Кохонена, обученный с помощью генетического алгоритма.*

*Качество кластерного анализа определялось с помощью двух индексов: индекса Рэнда и отрегулированного индекса Рэнда.*

*Приведено сравнение временных затрат при обучении слоя Кохонена стандартным алгоритмом и генетическим алгоритмом.*

*Анализ был проведен на наборе экземпляров банковских банкнот, описываемых набором числовых признаков, полученных из образов двух типов карт: реальных и фальшивых.*

**Ключевые слова:** кластерный анализ, K-means, слой Кохонена, генетический алгоритм, индекс Рэнда

### Введение

В настоящее время существует множество алгоритмов, реализующих задачу кластерного анализа. Технику кластеризации можно применять в самых различных прикладных областях, в том числе и для контроля подлинности банковских банкнот.

Существуют классические методы кластерного анализа, давно ставшие популярными в анализе данных — это использование метода K-means или слоя Кохонена. Однако необходимо расширить набор алгоритмов для повышения качества кластеризации и сокращения временных затрат.

Автором было предложено использование генетического алгоритма для обучения сети, а именно настройку весов для слоя Кохонена с возможностью в дальнейшем решать задачу кластерного анализа.

Программная реализация проводилась в среде разработки MATLAB, были использованы такие расширения, как Neural Network Toolbox, Global Optimization Toolbox, а также неофициальный Exploratory Data Analysis Toolbox [1].

### Кластерный анализ

Кластерный анализ (кластеризация) предназначен для разбиения совокупности объ-

ектов на однородные группы (кластеры). В результате применения различных методов кластеризации могут быть получены неодинаковые результаты.

Общая постановка задачи кластеризации выглядит следующим образом.

Пусть  $X$  — множество объектов,  $Y$  — множество номеров кластеров. Задана функция расстояния между объектами  $p(x, x')$ . Имеется конечная обучающая выборка объектов  $X^m = \{x_1, \dots, x_m\} \in X$ . Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике  $p$ , а объекты разных кластеров существенно (по выбранной метрике) отличались. При этом каждому объекту  $x_i \in X^m$  приписывается номер кластера  $y_j$  [2].

### Слой Кохонена и K-means

Алгоритм работы слоя Кохонена схож с известным алгоритмом K-means, их объединяет следующее.

1. Необходимо заранее определить и задать число кластеров.

2. Алгоритмы сходятся, если:

2.1) выполняется один из основных критериев останова алгоритма обучения сети — не

произошло значимого изменения весовых коэффициентов в пределах заданной точности на протяжении последней эпохи обучения;

2.2) в случае K-means алгоритм завершается, когда координаты центров кластеров на текущей итерации не отличаются от соответствующих координат на предыдущей итерации алгоритма.

Отличительной чертой процесса обучения данной сети от процессов обучения многих других видов сетей является то, что необходимо настроить веса синапсов нейронов, а не минимизировать ошибку обучения.

### Генетический алгоритм

Генетический алгоритм (ГА) — это один из методов решения оптимизационных задач. Как известно, оптимизационные задачи заключаются в нахождении минимума (максимума) заданной функции. Такую функцию называют функцией приспособленности (фитнес-функция).

Предложенный Джоном Холландом в 1975 г. генетический алгоритм основан на принципах естественного отбора Ч. Дарвина и является эвристическим; он в ряде случаев дает результаты более эффективно, чем классические методы оптимизации.

В генетическом алгоритме используется как механизм генетического наследования, так и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде и применяются основные понятия линейной алгебры.

Одно пробное решение называется особью, а набор всех пробных решений — популяцией.

Как известно, принцип естественного отбора заключается в том, что в конкурентной борьбе выживает наиболее приспособленный. В нашем случае приспособленность особи определяется фитнес-функцией: чем меньше значение функции, тем более приспособленной является особь, т. е. пробное решение, использовавшееся в качестве аргумента фитнес-функции.

В генетическом алгоритме создание новой популяции реализуется путем применения селекции и генетических операторов (оператора скрещивания и оператора мутации).

Процесс размножения состоит в селекции родительских пар для скрещивания таким образом, чтобы решения (особи) в новой популя-

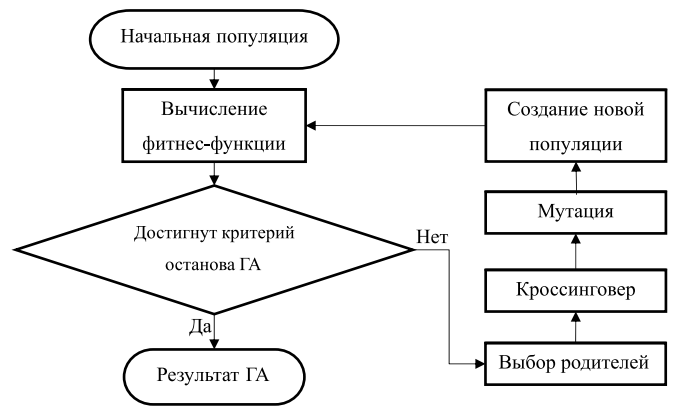


Рис. 1. Блок-схема ГА

ции были ближе к искомому глобальному минимуму функции приспособленности. Следующим шагом в работе генетического алгоритма являются мутации, т. е. случайные изменения полученных в результате скрещивания хромосом. Мутации способны улучшить или ухудшить приспособленность особи-потомка.

Основные принципы работы генетического алгоритма (ГА) заключены в следующей схеме (рис. 1) [3].

Из приведенной блок-схемы видно, что вычисление функции приспособленности выполняется для каждой генерируемой популяции, начиная с начальной. От полученных значений функции приспособленности зависит дальнейшая работа генетического алгоритма: если критерий останова удовлетворен, то алгоритм прекращает работу; если критерий не был достигнут, то на основе значений функции приспособленности создается новая популяция и цикл повторяется.

В данной работе формирование начальной популяции осуществлялось с помощью псевдослучайного выбора заданного числа особей.

В проводимом исследовании стояла задача минимизации функции приспособленности.

Определение критерия останова ГА зависит от его конкретного применения. Основные критерии:

- 1) достижение предельного значения функции приспособленности;
- 2) формирование новой популяции не приводит к значимому улучшению значения функции приспособленности;
- 3) достигнут лимит на время выполнения ГА, либо лимит на число поколений.

В данном исследовании использовалось два критерия — 1 и 3.

Процесс селекции родителей реализован в соответствии с алгоритмом стохастической универсальной селекции Бэкера, так как для достижения высокого уровня стабильности результатов ГА предпочтительнее использовать именно эту селекцию [4].

Скрещивание родительских особей происходило согласно алгоритму *scatter*-скрещивания при  $P_{sc} = 0,8$ , где  $P_{sc}$  — вероятность скрещивания.

Процесс мутации был организован по принципу гауссовской мутации, согласно которому случайно выбранное с помощью распределения Гаусса число добавляется к каждому элементу родительского вектора [4].

### Объединение ГА и нейросетевого подхода при решении задачи кластерного анализа

Объединение генетических алгоритмов и нейронных сетей известно в литературе под аббревиатурой COGANN. Объединение может быть независимым, вспомогательным, либо равноправным. При независимом объединении генетические алгоритмы и нейронные сети используют по отдельности для решения той или иной задачи (никак не взаимодействуют друг с другом). Вспомогательное объединение этих двух методов означает, что они применяются последовательно один за другим, дополняя друг друга. Например, ГА может провести анализ обученной сети, либо выполнить поиск оптимального набора параметров для обучения сети. В случаях равноправного применения ГА можно использовать для выбора топологии сети, обучения сети [5].

Как уже говорилось ранее, отличительной чертой процесса обучения слоя Кохонена является то, что необходимо настроить веса синапсов нейронов, а не минимизировать ошибку обучения. Иными словами, для того чтобы получить обученную сеть необходимо найти центры кластеров, координаты которых, в свою очередь, составят матрицу весов синапсов нейронов, тем самым исключается потребность в проведении обучения сети классическим алгоритмом Кохонена.

Как уже говорилось ранее, алгоритм работы слоя Кохонена схож с известным алгоритмом K-means.

В случае K-means распространен критерий — минимизация суммы квадратов расстояний от точек до центров кластеров, к которым они относятся [6]. То есть задачу кластерного анализа методом K-means можно свести к задаче минимизации функции

$$F = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - c_i)^2,$$

где  $k$  — число кластеров;  $S_i$  — полученные кластеры (подмножества множества всех анализируемых наблюдений);  $x_j$  — наблюдения;  $c_i$  — центры кластеров [7, 8].

В данном случае целью ГА будет минимизация приведенной выше функции  $F$  и нахождение глобальных экстремумов.

На основании приведенного ранее критерия минимизации была сформирована функция приспособленности. Однако сравнение принципов работы алгоритма K-means с принципами работы ГА в качестве алгоритма обучения слоя Кохонена выявляет ряд особенностей (табл. 1).

Таблица 1

Сравнение этапов работы алгоритмов

Особенность	Алгоритм K-means	Сформированная функция приспособленности
Начало работы алгоритма	Определение координат центров кластеров	Формирование начальной популяции (вектор, содержащий координаты центров кластеров)
Проверка останова алгоритма	Координаты центров кластеров на текущей итерации не отличаются от соответствующих координат на предыдущей итерации алгоритма	Формирование новой популяции не приводит к значимому улучшению значения фитнес-функции
Изменение координат центров кластеров	Перерасчет координат происходит по четко определенной формуле	Выполняется ряд операций: селекция, скрещивание, мутация и формирование новой популяции — нового вектора, содержащего координаты центров кластеров

### Оценка качества кластерного анализа

Задача оценки качества кластеризации является более сложной по сравнению с задачей оценки качества классификации. Во-первых, такие оценки не должны зависеть от номеров кластеров, а зависеть только от самого разби-

ения выборки. Во-вторых, не всегда известны истинные номера кластеров объектов, поэтому также нужны оценки, позволяющие оценить качество кластеризации, используя только неразмеченную выборку.

Наиболее популярными метриками качества являются: индекс Рэнда — Rand Index (RI); отрегулированный индекс Рэнда — Adjusted Rand Index (ARI). В обоих случаях предполагается, что известны истинные номера кластеров объектов. ARI не зависит от самих номеров кластеров, а зависит только от разбиения выборки на кластеры.

Пусть  $n$  — число объектов в выборке;  $A$  — вектор истинных номеров кластеров каждого объекта;  $B$  — вектор номеров кластеров, полученный алгоритмом кластеризации. Обозначим  $a$  — число пар объектов, имеющих одинаковые номера кластеров в векторах  $A$  и  $B$  и находящихся в одном кластере;  $b$  — число пар объектов, имеющих различные метки и находящихся в разных кластерах. Тогда Rand Index (RI)

$$RI = \frac{a + b}{\binom{n}{2}} = \frac{2(a + b)}{n(n - 1)}.$$

То есть это доля объектов, для которых эти разбиения (исходное и полученное в результате кластеризации) "согласованы" [9].

В отличие от RI, ARI не зависит от самих значений номеров кластеров и перестановок этих номеров, а зависит только от разбиения выборки на кластеры.

Для вычисления ARI необходимо построить таблицу сопряженности (табл. 2). В таблице  $X = \{X_1, X_2, \dots, X_r\}$  и  $Y = \{Y_1, Y_2, \dots, Y_s\}$  — результаты кластерного анализа, либо результат анализа и целевое разбиение;  $r$  — число кластеров, полученное методом  $X$ ;  $s$  — число кластеров, полученное методом  $Y$  (либо число  $s$  заранее известно, так как известно целевое разбиение).

Таблица 2

Таблица сопряженности

$X \backslash Y$	$Y_1$	$Y_2$	...	$Y_s$	Sums
$X_1$	$n_{11}$	$n_{12}$	...	$n_{1s}$	$a_1$
$X_2$	$n_{21}$	$n_{22}$	...	$n_{2s}$	$a_2$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$X_r$	$n_{r1}$	$n_{r2}$	...	$n_{rs}$	$a_r$
Sums	$b_1$	$b_2$	...	$b_s$	

Значения  $n_{ij}$  — число объектов, пересекающихся во множествах  $X_i$  и  $Y_j$ ;  $n_{ij} = |X_i \cap Y_j|$ . Значение  $a_r = \sum_{i=1}^r n_{ri}$ , значение  $b_s = \sum_{j=1}^s n_{js}$ .

ARI вычисляется по формуле

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[ \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[ \sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[ \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}.$$

Данный индекс является мерой расстояния между различными разбиениями выборки. ARI принимает значения в диапазоне  $[-1, 1]$ . Отрицательные значения соответствуют "независимым" разбиениям на кластеры, значения, близкие к нулю, — случайным разбиениям, и положительные значения говорят о том, что два разбиения схожи (совпадают при  $ARI = 1$ ) [10].

Приведенный ниже пример наглядно демонстрирует вычисление ARI.

Пусть вектора  $X$  и  $Y$  имеют следующие значения:

$X$	1	2	3	3	2	1	1	3	3	1	2	2
$Y$	3	2	3	2	2	1	1	2	3	1	3	1

Все анализируемые объекты разделены на три кластера.

Строится таблица сопряженности (табл. 3)

Имея все значения  $n_{ij}$ ,  $a_i$ ,  $b_j$ , легко рассчитать значение отрегулированного индекса Рэнда.

В данном случае

$$ARI = \frac{6 - [18 \times 18] / \binom{12}{2}}{\frac{1}{2} [18 + 18] - [18 \times 18] / \binom{12}{2}} = 0,08333.$$

Чем больше значения RI и ARI, тем больше соответствие между полученными и истинными кластерами.

Таблица 3

Таблица сопряженности

$X \backslash Y$	$Y_1$	$Y_2$	$Y_3$	Sums
$X_1$	3	0	1	4
$X_2$	1	2	1	4
$X_3$	0	2	2	4
Sums	4	4	4	

## Исходные данные

Анализируемые данные были получены из работы [11]. Исходная выборка содержит 1372 объекта, разделенных на две группы — подлинные и фальшивые купюры. Путем использования различных методов вейвлет-преобразований создатели данной выборки из множества изображений купюр получили матрицу, содержащую информацию об исследуемых банкнотах. В этой матрице каждый объект описывается четырьмя признаками: дисперсия изображения; коэффициент асимметрии изображения; коэффициент эксцесса изображения; энтропия изображения.

Достоинства данной выборки заключаются в следующем.

1. Относительно большой объем анализируемых данных при малом числе кластеров, что делает процесс работы алгоритмов на данной выборке не столь затратным по объему производимых вычислений, сохраняя при этом достаточное число объектов для обучения сети.

2. Наличие целевого вектора (вектора, содержащего истинные номера кластеров объектов).

3. Поскольку создатели выборки получали данные из реальных источников (подлинные и фальшивые купюры), то результаты работы алгоритмов, реализующих кластерный анализ, будут представлять научный интерес.

## Результаты кластерного анализа

Как уже говорилось ранее, генетический алгоритм является эвристическим алгоритмом, исходя из этого более корректно будет резюмировать результаты кластерного анализа из ряда проведенных экспериментов. В данной работе было проведено 25 экспериментов, в каждом из которых кластерный анализ проводился несколькими методами.

На рис. 2 отображены значения индексов Рэнда, вычисленные по результатам экспериментов.

Аналогично, на рис. 3 отображены значения отрегулированных индексов Рэнда.

Чем больше значения индексов, тем лучше качество кластерного анализа. Из рис. 2, 3 можно сделать вывод, что худший результат по перечисленным индексам у слоя Кохонена, обученного стандартным алгоритмом.

Слой Кохонена, обученный с помощью генетического алгоритма, имеет широкий диапазон значений индексов. На приведенных

рис. 2, 3 в 8 из 25 экспериментов значения обоих индексов были выше (в трех экспериментах индексы имеют значения, значительно превосходящие средние значения индексов, полученных другими методами), чем в остальных рассматриваемых методах. В четырех экспериментах индексы были ниже, чем индексы во всех других методах.

В табл. 4 подведены итоги полученных результатов, отображенных графически на рис. 2, 3.

Среднее значение индексов Рэнда, вычисленных по результатам, полученным от слоя

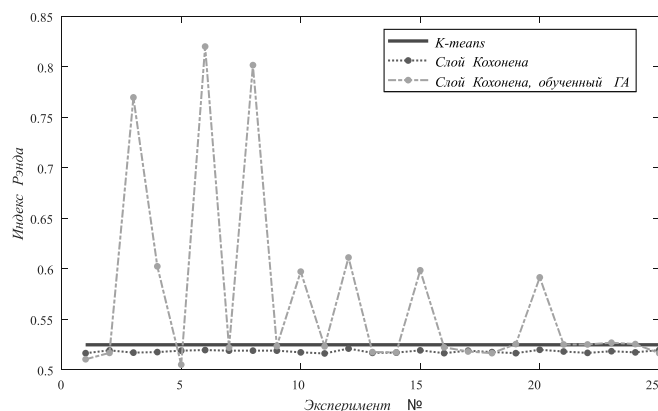


Рис. 2. Индексы Рэнда

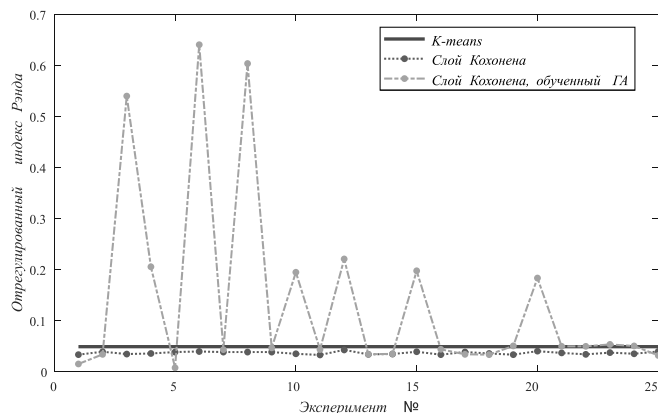


Рис. 3. Отрегулированные индексы Рэнда

Таблица 4

Индексы Рэнда

Метод кластерного анализа	Среднее значение RI по экспериментам	Среднее значение ARI по экспериментам
K-means	0,5249	0,0485
Слой Кохонена (стандартный алгоритм обучения)	0,5182	0,0362
Слой Кохонена (использован ГА для обучения)	<b>0,5692</b>	<b>0,1372</b>

Кохонена (ГА) за 25 экспериментов, выше соответствующих значений: K-means (RI на 0,0444; ARI на 0,0887), слой Кохонена обученный стандартным правилом Кохонена (RI на 0,0511; ARI на 0,1009).

Для процессов обучения нейронных сетей важным аспектом является время, затраченное на обучение. На рис. 4 приведены затраты времени на обучение сети Кохонена двумя методами — стандартным алгоритмом и генетическим алгоритмом. Табл. 5 содержит полученные результаты, отображенные графически на рис. 4.

На рис. 5 отображены значения функции приспособленности в каждом проведенном эксперименте.

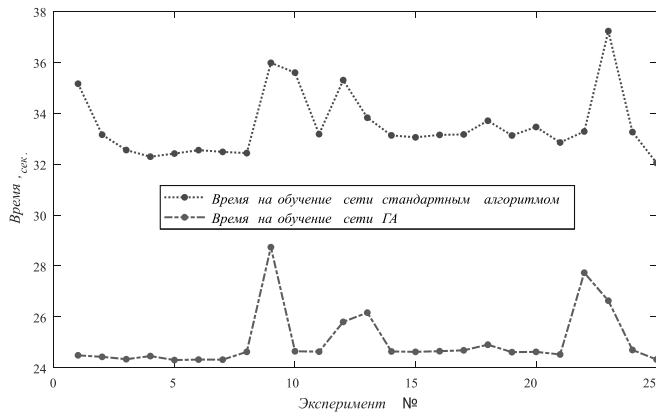


Рис. 4. График временных затрат на обучение

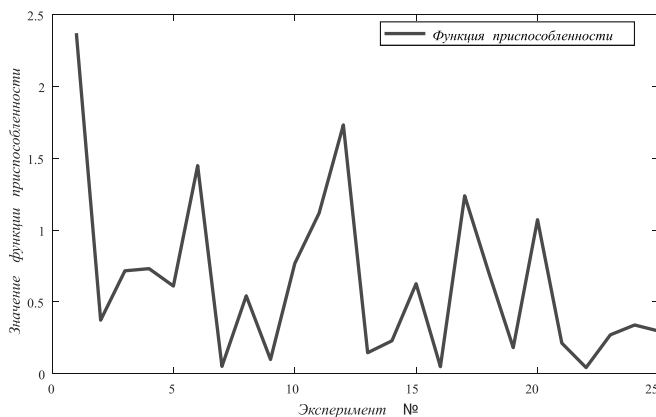


Рис. 5. Значения функции приспособленности

Таблица 5

Временные затраты на обучение сети

Метод кластеризации	Среднее значение, с
Стандартный алгоритм	33,5323
ГА	25,0388

## Заключение

Обзор результатов исследований, представленных в данной работе, показывает, что классические алгоритмы кластерного анализа не всегда эффективнее (с точки зрения затрат времени на анализ) и не всегда дают более высокие оценки качества проведенного анализа. Качество кластеризации сильно варьируется от анализируемых данных.

Предложенный автором метод кластерного анализа, в котором применяется генетический алгоритм в качестве алгоритма настройки матрицы весов слоя Кохонена, по сравнению с другими, описанными в этой статье алгоритмами, на использованной выборке показал более высокое качество анализа. Однако он имел очень широкий диапазон значений индексов Рэнда и значений функции приспособленности, что говорит о необходимости поиска более оптимального набора параметров генетического алгоритма, в особенности таких, как размер популяции, число поколений, вероятность скрещивания и мутации, а также критерии останова генетического алгоритма.

Тем не менее данный подход представляется интересным и заслуживает дальнейшей разработки.

## Список литературы

1. **Exploratory Data Analysis with MATLAB**, 2nd edition. URL: <http://pi-sigma.info/EDA.htm>.
2. **Мандель И. Д.** Кластерный анализ. М.: Финансы и статистика, 1988. 176 с.
3. **Гладков Л. А., Курейчик В. В., Курейчик В. М.** Генетические алгоритмы. М.: Физматлит, 2006. 320 с.
4. **Goldberg D. E.** Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc., 1989. 412 p.
5. **Рутковская Д., Пилиньский М., Рутковский Л.** Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия-Телеком, 2006. 452 с.
6. **Wierzchon S. T., Klopotek M. A.** Modern Algorithms of Cluster Analysis. Springer, 2018. 421 p.
7. **Миркин Б. Г.** Методы кластер-анализа для поддержки принятия решений: обзор. М.: Высшая школа экономики, 2011. 88 с.
8. **Ujjwal Maulik, Sanghamitra Bandyopadhyay, Anirban Mukhopadhyay.** Multiobjective Genetic Algorithms for Clustering. Berlin—Heidelberg: Springer-Verlag, 2011. 281 p.
9. **Rand W. M.** Objective criteria for the evaluation of clustering methods // Journal of the American Statistical Association. 1971. Vol. 66. P. 846—850.
10. **Santos J. M., Embrechts M.** On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification. Berlin—Heidelberg: Springer-Verlag, 2009. Vol. 2. P. 175—184.
11. **Репозиторий** реальных и модельных задач машинного обучения. URL: <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>.

## The Genetic Algorithm as the Alternative Method for Training Kohonen Layer

*In this article genetic algorithm suggested as the alternative method for training Kohonen layer. This paper considers evaluations of cluster analysis which were produced by follow methods: K-means; Kohonen layer trained with standard kohonen learning algorithm; Kohonen layer trained with genetic algorithm.*

*The evaluations of clustering were determined using two indices: the Rand index and Adjusted Rand index.*

*In addition to this, the article contains the graphical display of the training time for standard kohonen learning algorithm and realized genetic algorithm.*

*The cluster analysis was accomplished using banknote authentication dataset that described by a set of numerical feature and considers two output flag: authentic and counterfeit.*

**Keywords:** Cluster analysis, K-means, Kohonen layer, genetic algorithm, Rand index

DOI: 10.17587/it.24.642-648

### References

1. **Exploratory Data Analysis with MATLAB**, 2nd Edition. URL: <http://pi-sigma.info/EDA.htm>.
2. **Mandel' I. D.** *Klasternyj analiz*, Moscow, Finansy i statistika, 1988, 176 p. (in Russian).
3. **Gladkov L. A., Kurejchik V. V., Kurejchik V. M.** *Geneticheskie algoritmy*, Moscow, Fizmatlit, 2006, 320 p. (in Russian).
4. **Goldberg D. E.** *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, Inc., 1989, 412 p.
5. **Rutkovskaja D., Pilin'skij M., Rutkovskij L.** *Nejronnye seti, geneticheskie algoritmy i nechetkie sistemy*, Moscow, Gorjachaja linija-Telekom, 2006, 452 p. (in Russian).
6. **Wierzchon S. T., Klopotek M. A.** *Modern Algorithms of Cluster Analysis*, Springer, 2018. 421 p.
7. **Mirkin B. G.** *Metody klaster-analiza dlja podderzhki pri-njatija reshenij: obzor*, Moscow, Vysshaja shkola jekonomiki, 2011. 88 p. (in Russian).
8. **Maulik U., Bandyopadhyay S., Mukhopadhyay A.** *Multiobjective Genetic Algorithms for Clustering*, Berlin—Heidelberg, Springer-Verlag, 2011, 281 p.
9. **Rand W. M.** Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association*, 1971, vol. 66, pp. 846—850.
10. **Santos J. M., Embrechts M.** *On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification*. Berlin—Heidelberg. Springer-Verlag, 2009. Vol. 2, pp. 175—184.
11. **Repozitorij** real'nyh i model'nyh zadach mashinnogo obuchenija. URL: <https://archive.ics.uci.edu/ml/datasets/banknote> + authentication (in Russian).

УДК 004.434

DOI: 10.17587/it.24.648-656

**Л. Н. Лядова**, канд. физ.-мат. наук, доц., доц. кафедры информационных технологий в бизнесе, e-mail: LLyadova@hse.ru,

**А. О. Сухов**, канд. физ.-мат. наук, доц. кафедры информационных технологий в бизнесе, e-mail: ASuhov@hse.ru,

**Е. Ю. Медведева**, магистрант, e-mail: medvedevaeyu@mail.ru,  
Национальный исследовательский университет "Высшая школа экономики", г. Пермь

## Алгоритмы синтаксического разбора для текстовых динамически настраиваемых предметно-ориентированных языков

*Предложены алгоритмы разбора для текстовых динамически настраиваемых предметно-ориентированных языков и проверки синтаксической корректности написанных с их помощью программ. В процессе своей работы на основе описания расширенной грамматики языка анализатор строит псевдодерево разбора, которое в дальнейшем используется при проверке синтаксической корректности программ пользователя. В основе алгоритма проверки синтаксиса лежит метод леворекурсивного спуска с возвратом.*

**Ключевые слова:** разбор грамматики, проверка синтаксиса, предметно-ориентированные языки, текстовые языки, языковой инструментарий, метод леворекурсивного спуска, формальные грамматики, дерево разбора

### Введение

В настоящее время все большее число инструментальных программных систем предоставля-

ют в распоряжение пользователей встроенные текстовые языки программирования, позволяющие выполнять создание моделей предметной



области решаемой задачи, настраивать систему на предметную область, меняющиеся условия эксплуатации, потребности пользователей. Примерами таких языков являются:

- встроенный язык *IC: Предприятие*, предназначенный для настройки системы на предметную область деятельности предприятия [1];
- языки, интегрированные в системы имитационного моделирования (*Plant Simulation, AnyLogic* и др.), предназначенные для построения имитационных моделей [2–3];
- языки, встроенные в средства автоматизации проектирования и графические пакеты (*Autodesk 3ds Max, Autodesk AutoCAD, CorelDraw* и др.), используемые для автоматизации рутинных операций и программного создания мелких фрагментов моделей [4].

Большинство таких систем предоставляет в распоряжение пользователей предметно-ориентированные языки (Domain-Specific Language, DSL) — языки, разработанные для решения определенного круга задач в конкретной предметной области. Алфавит и конструкции таких языков близки понятиям предметной области, ее терминологии, отражают операции предметной области, поэтому работать с ними могут не только профессиональные ИТ-специалисты, но и будущие пользователи системы, обладающие навыками алгоритмизации, построения моделей предметной области. При этом в процессе использования таких языков часто возникает потребность не просто построить или модифицировать описание решения задачи, но и настроить язык на потребности пользователей или предметную область. Языки, описание которых может быть модифицировано в процессе эксплуатации системы, в которую они интегрированы, без регенерации ее исходного кода, называются *динамически настраиваемыми языками*.

В работе [5] описан подход к разработке инструментальных средств создания текстовых динамически настраиваемых DSL. Данная статья является продолжением работы [5].

*Целью данного исследования* является разработка алгоритмов синтаксического разбора грамматик текстовых динамически настраиваемых DSL, проверки синтаксической корректности написанных с их помощью программ.

## 1. Языковые инструментарии

Для разработки новых DSL используется специальный класс программного обеспече-

ния — языковые инструментарии (language workbench).

На сегодняшний день существует несколько языковых инструментариев, которые позволяют создавать текстовые DSL: OpenArchitectureWare [6], Meta Programming System [7], IDE Meta-Tooling Platform [8], MontiCore [9], Spoofox Language Workbench [10] и др. Каждый из инструментов имеет как достоинства, так и ограничения в использовании. Подробный анализ языковых инструментариев приведен в работе [5].

Устранить ограничения существующих инструментариев решено при разработке собственного языкового инструментария, подход к созданию которого описан в работе [5].

Основным преимуществом разрабатываемого языкового инструментария является то, что он позволяет выполнять динамическое изменение описания языка. Данная возможность позволяет модифицировать описания языка и разработанных с его помощью моделей во время работы системы без необходимости регенерации кода транслятора и редактора языка. Например, в процессе функционирования системы имитационного моделирования ИТ-инфраструктуры появилась потребность создания узлов нового типа. Для этого требуется добавить в описание DSL новую языковую конструкцию. Традиционный подход к созданию DSL предполагает, что разработчику необходимо с помощью языкового инструментария изменить описание DSL, модифицировать редактор языка, далее сгенерировать исходный код транслятора и редактора, и только после этого пользователь может вернуться к построению новых моделей. Данный процесс является длительным. Было бы намного удобнее внести все изменения во время работы системы и приступить к построению новых моделей без необходимости регенерации исходного кода транслятора и редактора языка.

Кроме того, разрабатываемый языковой инструментарий предоставляет возможность выполнять многоуровневое моделирование. Наличие такой возможности упростит процесс создания новых специализированных DSL, поскольку, определив один DSL общего назначения, например, для описания систем массового обслуживания, разработчик может создать на его основе несколько специализированных DSL имитационного моделирования для конкретных предметных областей, например, языки моделирования ИТ-инфраструктуры, цепочек поставок, движения автотранспорта и др.

## 2. Описание подхода к построению предметно-ориентированных языков

В процессе создания программы на DSL с помощью разрабатываемого языкового инструментария строится целая иерархия моделей:

- *метаязык* — язык, с помощью которого выполняется описание грамматики создаваемых DSL. Перед началом работы языковой инструментарий уже включает встроенный метаязык, который в случае необходимости может быть изменен пользователем;
- *метамодель* — модель создаваемого предметно-ориентированного языка, т. е. описание его грамматики;
- *модель предметной области* — программа, написанная с помощью DSL. В дальнейшем на ее основе может быть сгенерирован код на одном из целевых языков, например, языке программирования высокого уровня.

Каждая нижестоящая модель является экземпляром вышестоящей и строится путем создания экземпляров конструкций вышестоящей модели.

Таким образом, процесс создания программы на DSL состоит из следующих этапов:

1. Настройка метаязыка на особенности предметной области за счет добавления в его описание новых конструкций или изменения существующих. Данный этап является необязательным и выполняется только в том случае, если требуется настройка метаязыка на специфику предметной области, например, при создании специализированных DSL для конкретных предметных областей на основе языка общего назначения.

2. Построение метамодели, т. е. описание грамматики разрабатываемого DSL с помощью конструкций метаязыка.

3. Описание правил генерации кода. На данном этапе на уровне метамодели определяются правила преобразования конструкций DSL в конструкции одного из целевых языков программирования.

4. Построение модели предметной области (написание программы на созданном DSL).

5. Валидация построенной модели предметной области. На данном этапе выполняется проверка синтаксической корректности написанной программы.

6. Генерация кода. На данном этапе выполняется преобразование программы, написанной на DSL, в программу на целевом языке на

основе определенных пользователем правил генерации кода.

В рамках данного исследования будут разработаны алгоритмы, используемые на этапах 1, 2, 4, 5.

В качестве встроенного метаязыка языкового инструментария используются формы Бэкуса—Наура, имеющие следующие расширения:

- символы "{" , "}" (открывающая и закрывающая фигурные скобки) указывают на возможность присутствия конструкций языка, заключенных в фигурные скобки, ноль или более раз;
- символы "[", "]" (открывающая и закрывающая квадратные скобки) указывают на необязательность присутствия конструкций языка, заключенных в квадратные скобки;
- символ "|" (вертикальная черта) обозначает альтернативность использования конструкций языка, находящихся по разные стороны от данного символа;
- символ "\" (обратный слеш) экранирует символы алфавита описываемого языка, которые совпадают с метасимволами, позволяя применять их в качестве символов алфавита.

Описание грамматики метаязыка с помощью расширенных форм Бэкуса—Наура имеет следующий вид:

```
синтаксис_языка ::= правило {правило}
правило ::= нетерминальный_символ "::<=" правая_часть
правая_часть ::= элемент {элемент}
элемент ::= терминальный_символ [элемент] | нетерминальный_символ [элемент] | элемент "|" элемент | "[" элемент "]" | "{" элемент "}"
нетерминальный_символ ::= буква {буква}
терминальный_символ ::= "\" буква {буква} "\"
```

Начальным символом данной грамматики является нетерминальный символ "синтаксис\_языка".

Символ "\" (обратный слеш) используется для экранирования символов алфавита описываемого языка, которые совпадают с метасимволами, позволяя применять их в качестве символов алфавита.

Для упрощения создания новых DSL в описание их грамматик автоматически встраиваются правила для таких конструкций, как "Цифра", "Буква", "Идентификатор".

В случае необходимости создания многоуровневой модели и настройки встроенного метаязыка на предметную область пользователь имеет возможность изменить описание встроенного метаязыка.

Как видно из описания грамматики метаязыка, создаваемые с его помощью DSL будут иметь контекстно-свободную грамматику.

### 3. Алгоритмы разбора описания грамматики DSL

После того как пользователь с помощью встроенного метаязыка описал синтаксис DSL, необходимо выполнить разбор описания грамматики языка. Промежуточное представление, полученное в результате разбора описания грамматики, будет использовано при проверке синтаксической корректности модели. Кроме того, поскольку языковой инструментарий предоставляет возможность динамического изменения описания языка, то необходимо выполнять отслеживание всех модификаций грамматики DSL и оперативно вносить изменения в промежуточное представление, являющееся результатом работы алгоритма разбора описания грамматики. Таким образом, в качестве промежуточного представления следует выбрать такую структуру данных, которая позволяла бы эффективно выполнять поиск требуемых символов языка в процессе синтаксического разбора модели.

Представление метамодели в виде дерева разбора является наиболее приемлемым вариантом, так как данная структура позволяет быстро проводить поиск нужной вершины, содержащей символ языка, реконфигурацию дерева, а также эффективно применяется при разработке трансляторов для контекстно-свободных грамматик [11, 12].

Поскольку один и тот же символ языка может встретиться в описании правил грамматики более одного раза, необходимо либо дублировать его описание в дереве разбора, либо позволить вершине дерева иметь более одной родительской вершины. Дублирование информации усложнит процесс отслеживания динамических изменений описания языка, поскольку при модификации описания некоторой конструкции необходимо выполнить обход дерева для поиска всех ее вхождений. Именно поэтому в предлагаемом подходе решено использовать второй вариант и отступить от классического определения дерева разбора. Введем определение дерева разбора описания контекстно-свободной грамматики DSL.

*Псевдодеревом* назовем дерево, имеющее кратные ребра и петли.

Упорядоченное ориентированное псевдодерево  $G = (V, E)$  назовем *псевдодеревом разбора* описания для расширенной контекстно-свободной грамматики  $Gr = (T, N, P, S, M)$ , где  $T$  — множество терминальных символов,  $N$  — множество нетерминальных символов,  $P$  — множество правил вывода,  $S$  — начальный символ грамматики,  $M$  — множество метасимволов метаязыка, которые использовались для описания грамматики, при выполнении следующих условий:

- каждая вершина псевдодерева принадлежит одному из множеств  $T, N, M$ , т. е.  $\forall v \in V: v \in T \cup N \cup M$ ; корнем псевдодерева является символ  $S$ , листья псевдодерева принадлежат одному из множеств  $T, M$ ;
- если множество правил вывода расширенной грамматики  $Gr$  содержит правило  $A ::= \overline{A_1 A_2 \dots A_n}$ , где  $\forall A \in N, A_i \in T \cup N \cup M, i = \overline{1, n}$ , то вершина  $A$  псевдодерева имеет в качестве потомков вершины  $A_1, A_2, \dots, A_n$ .

На основе построенного псевдодерева система выполняет анализ корректности написанной пользователем программы.

Алгоритм проверки программы в соответствии с правилами описания расширенной контекстно-свободной грамматики имеет следующий вид:

```
global Grammar Gr;
global Tree Tr;
function CheckProgram()
{
    Gr ← InitializeGrammar();
    Tr.V ← ∅;
    Tr.E ← ∅;
    RulesParse();
    l ← NextLexeme();
    NonTerminalNodeParse(Tr.Root);
}
```

Функция *InitializeGrammar* выполняет инициализацию расширенной грамматики на основе заданных пользователем правил: определяет набор терминальных, нетерминальных символов, метасимволов, начальный символ грамматики. Функция *NextLexeme* возвращает очередную лексему исходной программы.

В процессе разбора описания расширенной грамматики DSL языковой инструментарий посредством функции *RulesParse* выполняет последовательный анализ всех правил грамматики в соответствии со следующим алгоритмом:

```

function RulesParse()
{
  foreach (p ∈ Gr.P)
  {
    A ← Tr.FindNode(p.LeftHandSide);
    if (A ≠ null)
    {
      if (A.Children = null)
      {
        p.ParseRightHandSide(A);
      }
      else
      {
        Error(p, "Символ" + A + "описан повторно");
      }
    }
    else
    {
      A ← Tr.CreateNonterminalNode (p.LeftHandSide);
      p.ParseRightHandSide(A);
    }
  }
  foreach (A ∈ Tr.V)
  {
    if (A ∈ Gr.N AND A.Children = null)
    {
      Error(p, "Символ" + A + "не определен");
      PrintErrors();
    }
  }
}

```

Функция *Error* заносит информацию об ошибках, допущенных при описании правил расширенной грамматики, в таблицу ошибок. Функция *PrintErrors* выводит список всех ошибок по окончании анализа правил.

Метод анализа правой части *ParseRightHandSide* правила имеет следующий вид:

```

function ParseRightHandSide(Node A)
{
  foreach (s ∈ p.RightHandSide)
  {
    if (s ∈ Gr.T ∪ Gr.M)
    {
      B ← Tr.CreateTerminalNode(s);
    }
    else
    {
      B ← Tr.FindNode(s);
      if (B = null)
      {
        B ← Tr.CreateNonterminalNode(s);
      }
    }
    Tr.CreateArc(A, B);
  }
}

```

Приведенный алгоритм разбора описания расширенной грамматики DSL имеет полиномиальную сложность.

В процессе разбора описания расширенной грамматики система проверяет корректность описания правил:

- соответствие описания правил грамматики синтаксису метаязыка;
- наличие определения для всех нетерминальных символов;
- отсутствие повторного определение нетерминального символа.

Рассмотрим пример построения псевдодерева разбора для языка

"Записная книга", который имеет следующее описание:

```

Книга ::= {Запись ";"}
Запись ::= Фамилия Имя | Фамилия Имя Город
Фамилия ::= Идентификатор
Имя ::= Идентификатор
Город ::= Идентификатор

```

Начальным символом данной расширенной грамматики является нетерминальный символ "Книга".

Перед началом анализа псевдодерево разбора содержит описание predetermined конструкций языка "Цифра", "Буква", "Идентификатор", которые задаются следующими правилами:

```

Идентификатор ::= Буква {Буква | Цифра}
Буква ::= "А" | "В" | ... | "Z" | "а" | "б" | ... | "z" | "А" | "Б" | ... | "Я" | "а" | "б" | ... | "я"
Цифра ::= "0" | "1" | ... | "9".

```

Если пользователь изменяет правило описания одной из predetermined конструкций языка, то поддерево с ее описанием перестраивается в соответствии с новым определением.

Псевдодерево разбора для predetermined конструкций приведено на рис. 1—3. В псевдодереве разбора прямоугольниками обозначены нетерминальные символы, окружностями — терминальные символы, ромбами — метасимволы, прямоугольником с двойной границей — начальный символ расширенной грамматики.

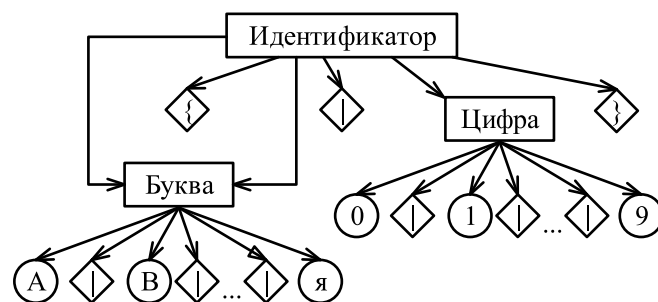


Рис. 1. Построение псевдодерева разбора. Этап I

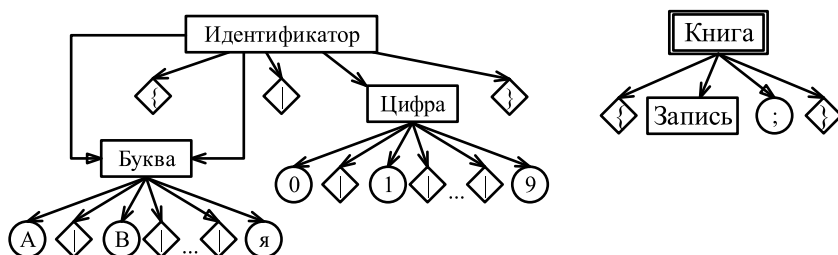


Рис. 2. Построение псевдодерева разбора. Этап II

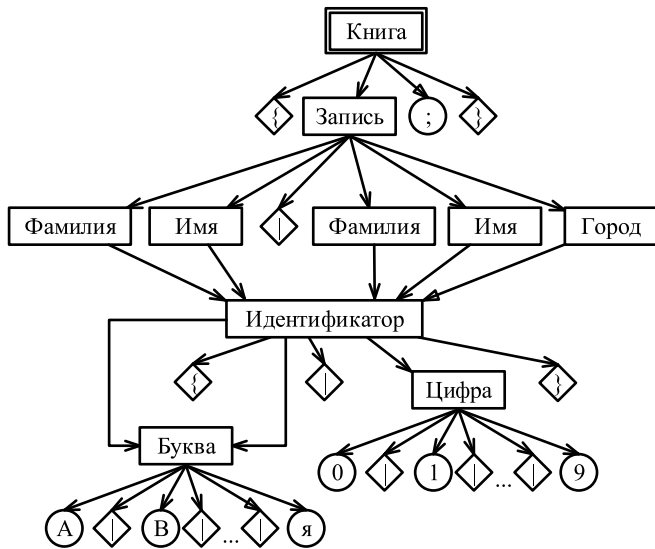


Рис. 3. Построение псевдодерева разбора. Этап III

После анализа первого правила в псевдодерево разбора будет добавлено описание нетерминального символа "Книга" (рис. 2).

Результирующее псевдодерево разбора, полученное после анализа всех правил, приведено на рис. 3.

#### 4. Алгоритм проверки синтаксической корректности модели

После описания расширенной контекстно-свободной грамматики пользователь получает в распоряжение DSL, с помощью которого может разрабатывать программы. Программа считается корректной, если она удовлетворяет правилам описания расширенной грамматики языка.

Поскольку языковой инструментарий позволяет выполнять динамическое изменение описания языка, то проверка корректности программы будет выполняться в реальном времени, во время ее построения. Это обстоятельство заставляет отказаться от использования генераторов анализаторов, таких как ANTLR, Yacc, GNU Bison, Coco/R [13] и др.

В качестве основы для разработки алгоритма синтаксического разбора используется метод леворекурсивного спуска как наиболее простой в реализации. Поскольку пользователь сам определяет расширенную контекстно-свободную грамматику DSL, то нельзя гарантировать выполнение достаточного условия применимости метода рекурсивного спуска, поэтому для реализации используется метод рекурсивного спуска с возвратом.

Одним из ключевых понятий при реализации метода рекурсивного спуска является понятие множества стартовых символов нетерминального символа грамматики.

Множество терминальных символов, с которых может начинаться символ  $A \in N$ , назовем *множеством его стартовых символов*, обозначим его  $starters(A)$  и будем вычислять по следующим правилам:

1. Если правило грамматики имеет вид  $A ::= t\omega$ , где  $t \in T$ ,  $\omega \in (T \cup N)^*$ , то  $starters(A) = \{t\}$ .

2. Если правило грамматики имеет вид  $A ::= B\omega$ , где  $B \in N$ ,  $\omega \in (T \cup N)^*$ , то  $starters(A) = starters(B)$ .

3. Если правило грамматики имеет вид  $A ::= \omega_1 | \omega_2 | \dots | \omega_n$ , где  $\omega_i \in (T \cup N)^*$ ,  $i = \overline{1, n}$ , то  $starters(A) = \bigcup_{i=1}^n starters(\omega_i)$ .

4. Если правило грамматики имеет вид  $A ::= \{\omega_1 | \omega_2 | \dots | \omega_n\} \omega$  или  $A ::= [\omega_1 | \omega_2 | \dots | \omega_n] \omega$ , где  $\omega_i \in (T \cup N)^*$ ,  $i = \overline{1, n}$ , то  $starters(A) = \bigcup_{i=1}^n starters(\omega_i) \cup starters(\omega)$ .

Описание алгоритма разбора последовательности лексем по псевдодереву разбора для расширенной грамматики  $Gr$  на псевдокоде имеет вид:

```
function NonTerminalNodeParse(Node A)
{
  i = 0;
  while (i < A.Children.Count)
  {
    a ← A.Children[i];
    while (l ∉ starters(a))
      a ← NextAlternative();
    if (l ≠ null)
    {
      NodeParse(A, i);
      ++i;
    }
  }
  else
    Error(l, "Ошибка при описании конструкции" + a);
}
}
```

Если правая часть правила расширенной грамматики содержит обязательную группу альтернатив, то в процессе анализа необходимо определить, в соответствии с какой альтернативой следует выполнять разбор. Для этого анализатор последовательно проверяет, принадлежит ли текущая лексема стартовым символам каждой альтернативы, если нет, то пропускает соответствующую данной альтернативе последовательность символов в описании конструкции с помощью функции *NextAlternative*.

Функция разбора терминальных, нетерминальных символов и последовательности альтернатив имеет следующий вид:

```

function NodeParse(Node A, ref int i)
{
  a ← A.Children[i];
  if (a ∈ Gr.T)
    if (a = l)
      l ← NextLexeme();
    else
    {
      Error(l, "Ошибка при описании
      конструкции" + a);
      return;
    }
  else
    if (a ∈ Gr.N)
      NonTerminalNodeParse(a);
    else
      if (a = '[')
        do
          a ← A.Children[+ + i];
          if (l ∈ starters(a))
            {
              while (a ≠ '|' OR a ≠ ']') do
                {
                  NodeParse(A, i);
                  a ← A.Children[+ + i];
                }
              while (a ≠ ']') do
                a ← A.Children[+ + i];
            }
          else
            while (a ≠ '|' OR a ≠ ']') do
              a ← A.Children[+ + i];
          while (a ≠ ']');
        else
          if (a = '{')
            {
              StartAlternative ← a;
              do
                continue ← false;
                a ← A.Children[+ + i];
                if (l ∈ starters(a))
                  {
                    continue ← true;
                    while (a ≠ '|' OR a ≠ '}') do
                      {
                        NodeParse(A, i);
                        a ← A.Children[+ + i];
                      }
                  }
                else
                  while (a ≠ '|' OR a ≠ '}') do
                    a ← A.Children[+ + i];
                if (a = '}' AND continue)
                  a ← StartAlternative;
                while (a ≠ '}');
            }
      }
}

```

Данный алгоритм имеет полиномиальную сложность.

Поскольку структура языка заранее неизвестна, в отличие от языков с неизменным синтаксисом, а также не всегда удается однозначно сопоставить анализируемую лексему с конструкцией метамодели, важно сохранять историю разбора программы для наличия возможности возврата к последнему успешно разобранным символу. Возврат на предыдущие этапы анализа выполняется при сопоставлении последовательности лексем с правилами, содержащими группы альтернатив.

Если на некотором этапе разбора возникает ситуация, при которой анализатор не может сопоставить текущую лексему ни с одним ожидаемым символом языка, то необходимо выполнить возврат к предыдущему этапу анализа в соответствии со следующим алгоритмом:

1. Если правило, в соответствии с которым выполняется разбор, имеет вид

$$A ::= \alpha_1[\omega_1|\omega_2|\dots|\omega_n]\alpha_2,$$

где  $\alpha_1, \alpha_2 \in (T \cup N)^+$ ,  $\omega_i \in (T \cup N)^*$ ,  $i = \overline{1, n}$ , и на очередном шаге был выполнен разбор в соответствии с альтернативой  $\omega_k$ , то следует вернуться к лексеме  $l$ , следующей сразу за последней корректно разобранным конструкцией  $\alpha_1$ , и попытаться выполнить разбор в соответствии с описанием ранее нерассмотренных альтернатив  $\{\omega_j\}$ ,  $i = \overline{k+1, n}$ . При этом возможны два варианта:

1.1. Анализируемая последовательность лексем не может быть разбрана в соответствии с описанием ни одной из альтернатив, это значит, что группа альтернатив была опущена в силу своей необязательности, необходимо выполнить переход к следующему за группой символу  $\alpha_2$  и провести разбор в соответствии с его описанием.

1.2. Лексема  $l$  соответствует стартовому символу одной из альтернатив, т. е.  $l \in starters(\omega_i)$ ,  $i = \overline{k+1, n}$ , тогда следует выполнить разбор последовательности лексем в соответствии с описанием  $i$ -й альтернативы. Если в процессе анализа окажется, что очередная лексема не может быть разбрана в соответствии с описанием данной альтернативы, то следует вновь вернуться к лексеме  $l$  и найти другую подходящую альтернативу в рассматриваемой группе  $\{\omega_j\}$ ,  $j = \overline{i+1, n}$ . Это продолжается до тех пор, пока все альтернативы не будут рассмотрены, либо анализ альтернативы не завершится успехом.

После успешного завершения анализа одной из альтернатив следует продолжить разбор

## Заключение

в соответствии со следующим за группой символом  $\alpha_2$ . Если в какой-то момент лексема не может быть сопоставлена с текущим символом правила, то анализатор вновь возвращается к лексеме  $l$ , предпринимает попытку найти следующую альтернативу и выполнить разбор в соответствии с ее описанием.

2. Если правило, в соответствии с которым выполняется разбор, имеет вид

$$A ::= \alpha_1\{\omega_1|\omega_2|\dots|\omega_n\}\alpha_2,$$

где  $\alpha_1, \alpha_2 \in (T \cup N)^+$ ,  $\omega_i \in (T \cup N)^*$ ,  $i = \overline{1, n}$ , необходимо выполнить разбор лексемы в соответствии с пунктом 1 с той лишь разницей, что после успешного завершения анализа одной из альтернатив, необходимо вернуться к началу группы и попытаться выполнить разбор очередной лексемы в соответствии с описанием одной из альтернатив.

Если на каком-либо этапе анализа конструкций, следующих за группой альтернатив, невозможно сопоставить текущую лексему ни с одним из символов правила, то необходимо вернуться к лексеме, следующей за  $m - 1$  успешно разобранных альтернатив группой, где  $m$  — общее число успешно разобранных альтернатив, и попытаться выполнить анализ в соответствии с описанием другой альтернативы, начиная с  $m + 1$ .

3. Если правило, в соответствии с которым выполняется разбор, имеет вид

$$A \rightarrow \omega_1|\omega_2|\dots|\omega_n,$$

где  $\omega_i \in (T \cup N)^*$ ,  $i = \overline{1, n}$ , то оно интерпретируется как группа альтернатив, и его анализ выполняется в соответствии с пунктом 1.

Разбор считается успешным при одновременном завершении анализа текста исходной программы и обхода псевдодерева разбора.

Последняя анализируемая строка считается ошибочной, если

- рассмотрены все возможные варианты разбора текущей строки, и ни один из них не позволяет корректно проанализировать программу;
- завершен обход псевдодерева разбора, но программа содержит еще неразобранные строки;
- завершен разбор всех строк исходной программы, но обход псевдодерева разбора не был завершен.

Описанный алгоритм возврата имеет экспоненциальную сложность.

Разработанные алгоритмы синтаксического разбора позволяют реализовать в языковом инструментарии возможности многоуровневого моделирования и динамического изменения описания разрабатываемых DSL.

В качестве промежуточного представления для хранения результатов анализа расширенной грамматики DSL используется псевдодерево разбора, что позволяет повысить эффективность поиска конструкций языка на этапе проверки синтаксической корректности программы.

Функции подсветки синтаксиса языка и автодополнения кода текстового редактора DSL также используют в своей работе псевдодерево разбора. Эти функции значительно упрощают процесс построения и чтения модели, а также сокращают число ошибок в ней. Подсветка синтаксиса выполняется на основе списков нетерминальных символов и метасимволов. Составление списков проводится на этапе построения псевдодерева разбора для того, чтобы в дальнейшем каждый раз не выполнять поиск по псевдодереву. Для автодополнения кода необходимо для каждого нетерминального символа сформировать список его стартовых символов.

Практическая значимость работы заключается в том, что рассмотренные алгоритмы являются основой для реализации компонентов описания и проверки синтаксиса языкового инструментария, позволяющего создавать текстовые динамически настраиваемые предметно-ориентированные языки.

В дальнейшем предполагается разработать алгоритмы трансформации моделей, созданных с помощью DSL, в код на целевом языке программирования.

## Список литературы

1. **IC: Предприятие 8.** Встроенный язык. URL: [http://v8.lc.ru/overview/Term\\_000000020.htm](http://v8.lc.ru/overview/Term_000000020.htm) (дата обращения: 21.11.2017).
2. **Pegden C. D.** The Evolution of Simulation Languages // *Advances in Modeling and Simulation. Simulation Foundations, Methods and Applications.* Cham: Springer, 2017. P. 81–96.
3. **Buch J. P., Laursen J. S., Soensen L. C., Ellekilde L.-P., Kraft D., Schultz U. P., Petersen H. G.** Applying Simulation and a Domain-Specific Language for an Adaptive Action Library // *Simulation, Modeling, and Programming for Autonomous Robots. SIMPAR 2014. Lecture Notes in Computer Science.* Cham: Springer, 2014. Vol. 8810. P. 86–97.
4. **Дерябин Н. Б., Жданов Д. Д., Соколов В. Г.** Внедрение языка сценариев в программные комплексы оптического моделирования // *Программирование.* 2017. № 1. С. 40–53.
5. **Сухов А. О., Медведева Е. Ю.** Подход к разработке языкового инструментария для создания текстовых пред-

метно-ориентированных языков // Информационные технологии. 2018. Т. 24, № 1. С. 10–16.

6. **Haase A.** Introduction to OpenArchitectureWare 4.1.2 // Model-Driven Development Tool Implementers Forum 2007. URL: <https://pdfs.semanticscholar.org/a6fa/d330ed634810f3f002658443ab361eb3c423.pdf> (дата обращения: 21.11.2017).

7. **Voelter M.** Language and IDE Modularization and Composition with MPS // Generative and Transformational Techniques in Software Engineering IV. Lecture Notes in Computer Science. Berlin: Springer, 2013. P. 383–430.

8. **Charles Ph., Fuhrer R. M., Sutton S. M.** IMP: a Meta-Tooling Platform for Creating Language-Specific IDEs in Eclipse // Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering. NY. 2007. P. 485–488.

9. **Holldobler K., Roth A., Rumpe B., Wortmann A.** Advances in Modeling Language Engineering // Model and Data Engineering. MEDI 2017. Lecture Notes in Computer Science. Cham: Springer, 2017. Vol. 10563. P. 3–17.

10. **Kats L. C. L., Visser E.** The Spoofox Language Workbench: Rules for Declarative Specification of Languages and IDEs // Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications. NY. 2010. P. 444–463.

11. **Nijholt A.** Context-Free Grammars. Covers, Normal Forms, and Parsing. Berlin: Springer, 1980. 253 p.

12. **Mogensen T.** Introduction to Compiler Design. Berlin: Springer, 2017. 258 p.

13. **Koranne S.** Compiler Construction // Handbook of Open Source Tools. Boston: Springer, 2010. P. 241–284.

**L. N. Lyadova**, Associate Professor, e-mail: LLyadova@hse.ru,

**A. O. Sukhov**, Associate Professor, e-mail: ASuhov@hse.ru,

**E. Yu. Medvedeva**, Master Student, e-mail: medvedevaeyu@mail.ru,

National Research University Higher School of Economics, Perm, 614070, Russian Federation

## Syntactic Parsing Algorithms for Textual Domain-Specific Languages

*Over time an increasing number of software packages have provided users with domain-specific languages designed to customize the system in accordance with changing operating conditions and user needs. However, it is necessary to create language workbenches for such languages. The authors suggested an approach to developing toolkits for creating textual dynamically customizable domain-specific languages that allow eliminating the limitations of most existing language workbenches. The purpose of the study is to develop algorithms for parsing language grammar and checking syntactic correctness of the programs. These algorithms are necessary for implementing a language workbench.*

*As a means of representing the transitional results of the domain-specific language grammar analysis a parsing pseudo tree is used, which makes it possible to improve the efficiency of language constructions searching at the stage of checking the program's syntactic correctness.*

*The parsing algorithm is based on the left-recursive descent method with backtracking. Despite the exponential complexity, the algorithm makes it possible to provide the possibility of dynamically changing the language description.*

*Functions of language syntax highlighting and code completing also use the parsing pseudo tree in their work. These functions simplify the process of model building and reading and reduce the number of errors in it.*

**Keywords:** grammar analysis, syntax checking, domain-specific languages, textual languages, language workbench, left-recursive descent parser, formal grammars, parse tree, metalanguage, model-oriented approach

DOI: 10.17587/it.24.648-656

### References

1. **IS:Predpriyatje 8.** Vstroennyj jazyk, available at: [http://v8.1c.ru/overview/Term\\_00000020.htm](http://v8.1c.ru/overview/Term_00000020.htm) (Access at: 21.11.2017) (in Russian).

2. **Pegden C. D.** The Evolution of Simulation Languages, *Advances in Modeling and Simulation. Simulation Foundations, Methods and Applications*, Cham, Springer, 2017, pp. 81–96.

3. **Buch J. P., Laursen J. S., Soensen L. C., Ellekilde L.-P., Kraft D., Schultz U. P., Petersen H. G.** Applying Simulation and a Domain-Specific Language for an Adaptive Action Library, *Simulation, Modeling, and Programming for Autonomous Robots. SIMPAR 2014. Lecture Notes in Computer Science*, Cham, Springer, 2014, vol. 8810, pp. 86–97.

4. **Derjabin N. B., Zhdanov D. D., Sokolov V. G.** Vnedrenie jazyka scenarijev v programmnye komplekxy opticheskogo modelirovanija (Embedding the Script Language into Optical Simulation Software), *Programirovanie*, 2017, no. 1, pp. 40–53 (in Russian).

5. **Suhov A. O., Medvedeva E. Ju.** Podhod k razrabotke jazykovogo instrumentarija dlja sozdaniya tekstovyh predmetno-orientirovannyh jazykov (Approach to Language Workbench Development for Textual Domain-Specific Languages Creation), *Informacionnye Tehnologii*, 2018, vol. 24, no. 1, pp. 10–16 (in Russian).

6. **Haase A.** Introduction to OpenArchitectureWare 4.1.2, Model-Driven Development Tool Implementers Forum 2007,

available at: <https://pdfs.semanticscholar.org/a6fa/d330ed634810f3f002658443ab361eb3c423.pdf> (Access at: 21.11.2017).

7. **Voelter M.** Language and IDE Modularization and Composition with MPS, *Generative and Transformational Techniques in Software Engineering IV. Lecture Notes in Computer Science*, Berlin, Springer, 2013, vol. 7680, pp. 383–430.

8. **Charles Ph., Fuhrer R. M., Sutton S. M.** IMP: a Meta-Tooling Platform for Creating Language-Specific IDEs in Eclipse, *Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering*, NY, 2007, pp. 485–488.

9. **Holldobler K., Roth A., Rumpe B., Wortmann A.** Advances in Modeling Language Engineering, *Model and Data Engineering. MEDI 2017. Lecture Notes in Computer Science*, Cham, Springer, 2017, vol. 10563, pp. 3–17.

10. **Kats L. C. L., Visser E.** The Spoofox Language Workbench: Rules for Declarative Specification of Languages and IDEs, *Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications*, NY, 2010, pp. 444–463.

11. **Nijholt A.** Context-Free Grammars. Covers, Normal Forms, and Parsing, Berlin, Springer, 1980, 253 p.

12. **Mogensen T.** Introduction to Compiler Design, Berlin, Springer, 2017, 258 p.

13. **Koranne S.** Compiler Construction, *Handbook of Open Source Tools*, Boston, Springer, 2010, pp. 241–284.



**В. И. Васильев**, д-р техн. наук, проф., e-mail: vasilyev@ugatu.ac.ru,  
**А. М. Вульфин**, канд. техн. наук, доц., e-mail: vulfin.alexey@gmail.com,  
**М. Б. Гузайров**, д-р техн. наук, проф., e-mail: guzairov@ugatu.su,  
**А. Д. Кириллова**, магистр, e-mail: kirillova.andm@gmail.com,  
Уфимский государственный авиационный технический университет

### Интервальное оценивание информационных рисков с помощью нечетких серых когнитивных карт

*Рассматривается возможность получения интервальных количественных оценок рисков информационной безопасности с помощью нечетких серых когнитивных карт (Fuzzy Grey Cognitive Maps). Обсуждаются вопросы построения нечетких серых когнитивных карт на основе обработки знаний и опыта экспертов. Рассмотрены особенности применения данного класса когнитивных моделей на примере задачи оценки информационных рисков.*

**Ключевые слова:** информационные риски, когнитивное моделирование, интервальные оценки, нечеткие серые когнитивные карты

#### Введение

Проблема оценки информационных рисков является одной из центральных проблем, вызывающих интерес у специалистов в области информационной безопасности (ИБ). Сегодня известно большое число методов и подходов (CRAMM, OCTAVE, COBRA, MSAT, Risk-Watch, АванГард и др.), цель которых — дать качественную или количественную оценку рисков ИБ и в конечном итоге сформировать определенные рекомендации по выбору состава защитных мер, направленных на обеспечение заданного уровня защищенности информационной системы [1–3]. Вопросам оценки информационных рисков посвящены разделы ряда стандартов по ИБ (ГОСТ Р ИСО/МЭК 15408, 27001-27005, 13335, 18045, СТО БР ИББС и др.), руководящие документы ФСТЭК (Федеральной службы по техническому и экспертному контролю) России. В то же время, в силу наличия значительных факторов неопределенности проблема остается во многом открытой и требует для своего решения применения все новых подходов, базирующихся, в частности, на применении технологий интеллектуального анализа данных и когнитивного моделирования.

В последние годы внимание многих исследователей привлекают возможности, которые предоставляет для решения проблемы оценки рисков ИБ такое направление в изучении сложных, плохо формализуемых систем, как моделирование на основе построения нечетких когнитивных карт (НКК) (Fuzzy Cognitive Maps) [4–8]. В соответствии со сложившейся классификацией различают: простые (классические) НКК, обобщенные НКК, реляционные НКК, нечеткие продукционные НКК, НКК в базисе "истина—ложь—неопределенность" и многие другие. Известны успешные примеры применения аппарата НКК для решения задач оценки рисков нарушения ИБ [9–13].

Важное место среди семейства НКК занимают нечеткие "серые" когнитивные карты (НСКК) (Fuzzy Grey Cognitive Maps, FGCM), впервые предложенные в 2010 г. Хосе Салмероном [14]. Основное отличие НСКК от других разновидностей НКК — использование интервальных оценок (диапазонов) значений переменных состояния концептов и весов связей между этими концептами вместо использования значений (термов) лингвистических переменных, описываемых с помощью нечетких чисел или функций принадлежности нечет-

ких множеств, как это традиционно делается в НКК. Операции нечеткой логики заменяются при этом интервальной арифметикой над "серыми" (интервальными) числами (grey numbers).

Нечеткие серые когнитивные карты (которые с равным успехом можно назвать также "интервальными" НКК) считаются удачным расширением НКК, поскольку они лучше соответствуют представлениям экспертов, обладают большей интерпретируемостью и представляют больше степеней свободы лицу, принимающему решение (ЛПР) на основании результатов моделирования. Отсюда понятен тот интерес, который проявляется к применению НСКК в различных технических приложениях [8, 14]. Очевидно, что применение НСКК для решения задач интервального оценивания рисков ИБ имеет свои перспективы (ранее аналогичные предложения высказывались в работах [15, 16]).

Ниже приведены понятия "серой" системы, "серого" числа и "серой" переменной, рассмотрены особенности построения НСКК. На конкретном примере обсуждается методика расчета интервальных оценок информационных рисков с использованием НСКК.

### 1. Теоретические основы построения нечетких серых когнитивных карт

Фундаментом построения НСКК является теория серых систем (Grey Systems Theory), предложенная в 1989 г. Дж. Денгом [17]. Предметом изучения данной теории являются объекты и системы с высокой неопределенностью, представленные малыми выборками неполных и неточных данных. В зависимости от имеющейся известной информации изучаемые системы при этом делятся на три вида:

- "белые" системы (внутренняя структура и свойства системы полностью известны);
- "серые" системы (известна частичная информация о системе);
- "черные" системы (внутренняя структура и свойства системы полностью неизвестны).

В соответствии с терминологией теории серых систем НСКК — это когнитивная модель системы в виде ориентированного графа, заданного с помощью следующего набора множеств:

$$\text{НСКК} = \langle C, F, W \rangle, \quad (1)$$

где  $C = \{C_i\}$  — множество концептов (вершин графа) ( $i = 1, 2, \dots, n$ );  $F = \{F_{ij}\}$  — множество связей между концептами (дуг графа);  $W = \{W_{ij}\}$  — множество отношений между концептами, определяющих веса указанных связей (дуг графа),  $(i, j) \in \Omega$ . Здесь  $\Omega = \{(i_1, j_1), (i_2, j_2), \dots, (i_L, j_L)\}$  — множество пар индексов смежных (связанных между собой) вершин,  $L \leq n(n - 1)$ .

В отличие от традиционного понимания НКК, веса связей НСКК задаются с помощью "серых" (интервальных) чисел  $\otimes W_{ij}$ , определяемых как

$$\begin{aligned} \otimes W_{ij} &\in [\underline{W}_{ij}, \overline{W}_{ij}], \\ \underline{W}_{ij} &< \overline{W}_{ij}, \{\underline{W}_{ij}, \overline{W}_{ij}\} \in [-1, 1], \end{aligned} \quad (2)$$

где  $\underline{W}_{ij}$  — нижняя граница серого числа  $\otimes W_{ij}$ ;  $\overline{W}_{ij}$  — верхняя граница серого числа. Таким образом, вес связи между  $i$ -м и  $j$ -м концептами ( $C_i \rightarrow C_j$ ) может принимать любое значение в пределах заданного диапазона изменения  $[\underline{W}_{ij}, \overline{W}_{ij}] \in [-1, 1]$ . В частном случае, когда  $\underline{W}_{ij} = \overline{W}_{ij}$ , получаем  $\otimes W_{ij} \in [\underline{W}_{ij}, \underline{W}_{ij}]$  — "белое" (четкое, обычное) число.

Предполагается, что изменение состояния концептов во времени описывается уравнениями

$$\begin{aligned} \otimes X_i(k+1) &= f \left( \otimes X_i(k) + \sum_{\substack{j=1 \\ (j \neq i)}}^n \otimes W_{ji} \otimes X_j(k) \right), \quad (3) \\ i &= 1, 2, \dots, n, \end{aligned}$$

где  $\otimes X_i(k)$  — "серая" (интервальная) переменная состояния  $i$ -го концепта  $C_i$ , которая в каждый момент времени  $k = 0, 1, 2, \dots$  принимает некоторое значение внутри определенного интервала (диапазона изменения), заданного границами  $\underline{X}_i(k)$  и  $\overline{X}_i(k)$ ;  $f(\cdot)$  — нелинейная функция активации  $i$ -го концепта, отображающая значения аргумента в интервал  $[-1, 1]$ . В качестве функции активации  $f(\cdot)$ , как правило, принимаются:

- а) линейная функция с ограничением:

$$f(x) = \begin{cases} x, & \text{если } |x| \leq 1, \\ \text{sign } x, & \text{если } |x| > 1; \end{cases} \quad (4)$$

- б) двухполярная сигмоидная функция (гиперболический тангенс):

$$f(x) = (1 - e^{-x}) / (1 + e^{-x}) = \text{th} \left( \frac{x}{2} \right); \quad (5)$$

в) однополярная сигмоида:

$$f(x) = 1/(1 + e^{-x}). \quad (6)$$

Для решения системы уравнений (3) требуется задать начальные значения переменных состояния  $\otimes X_i(0)$ , которые также должны рассматриваться как серые числа  $\otimes X_i(0) \in [\underline{X}_i(0), \bar{X}_i(0)]$ . Наибольший интерес обычно представляет получение равновесного (установившегося) решения, которое представляет собой "серый" вектор  $\lim_{k \rightarrow \infty} [\otimes X_i(k)] = \otimes X^* \in [\underline{X}^*, \bar{X}^*]$  или предельный цикл (странный аттрактор).

Для определения устойчивости установившегося решения  $\otimes X^*$  можно воспользоваться теоремой [18], согласно которой единственное равновесное (установившееся) решение уравнений вида (3) ("неподвижная точка") существует в том и только в том случае, если выполняется условие

$$\left( \sum_{i,j=1}^n W_{ij}^2 \right)^{1/2} < H, \quad (7)$$

где значение положительной константы  $H$  зависит от выбора функции активации концептов:  $H = 1$  для функции (4);  $H = 2$  для функции (5);  $H = 4$  для функции (6). Очевидно, что проверка выполнения условия для уравнений (3) должна проводиться для верхних границ серых чисел  $\bar{W}_{ij}$  ( $i, j = 1, 2, \dots, n$ ).

Более подробную информацию о построении НСКК и особенностях их применения можно найти в работах [14] или [8] (Chapter 14: Using Fuzzy Grey Cognitive Maps for Industrial Processes Control / Salmeron J. L., Papageorgiou E. I., pp. 237–252).

## 2. Пример оценки рисков с помощью НСКК

Допустим, что требуется оценить риски, связанные с нарушением конфиденциальности и целостности информации вследствие воздействия ряда угроз на информационные ресурсы (активы). Пусть НСКК для рассматриваемой ситуации принимает вид, представленный на рис. 1.

Здесь 1 — концепт  $C_1$ , представляющий собой угрозу, связанную с попыткой несанкционированного доступа (НСД) к информации; 2 — концепт  $C_2$ , представляющий угрозу, свя-

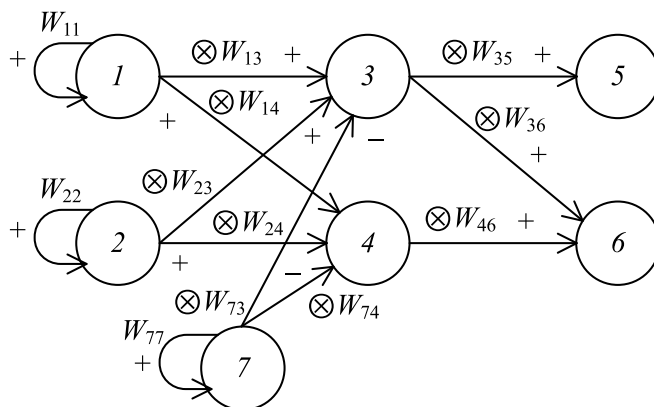


Рис. 1. Нечеткая серая когнитивная карта для оценки рисков ИБ

занную с вредоносным программным воздействием (вирусными атаками); 3 — концепт  $C_3$ , характеризующий целевой объект угрозы — базу данных (БД), размещенную на сервере; 4 — концепт  $C_4$ , характеризующий электронный документооборот (ЭДО) организации; 5 — концепт  $C_5$ , характеризующий потенциальный ущерб, вызванный нарушением конфиденциальности информации; 6 — концепт  $C_6$ , характеризующий потенциальный ущерб вследствие нарушения целостности информации.

Переменные состояния:  $\otimes X_1$  — вероятность возникновения угрозы типа НСД за определенный период времени;  $\otimes X_2$  — вероятность возникновения угрозы типа "Вредоносное программное воздействие/вирусы" за тот же период времени;  $\otimes X_3$  — доля утраченных или искаженных записей в БД к их общему количеству;  $\otimes X_4$  — доля времени, затрачиваемого на простой или восстановление нормальной работы ЭДО, по отношению к общему времени;  $\otimes X_5$  — ущерб от нарушения конфиденциальности информации;  $\otimes X_6$  — ущерб от нарушения целостности;  $\otimes X_7$  — стоимость контрмер по защите информации. Связи между концептами  $W_{13}, W_{14}, W_{23}, W_{24}, W_{35}, W_{36}, W_{46}$  считаются положительными, т. е. для пары концептов  $C_i \rightarrow C_j$  увеличение переменной  $X_i$  приводит к увеличению переменной  $X_j$ , а связи  $W_{73}, W_{74}$  — отрицательными, т. е. увеличение переменной  $X_i$  приводит к уменьшению переменной  $X_j$ . Все переменные  $\otimes X_1 \div X_7$  считаются нормированными; их значения принадлежат интервалу  $[0, 1]$ .

Будем полагать, что при выборе серых значений весов  $\otimes W_{ij}$  эксперт начинает с выбора "центров" соответствующих интервалов  $W_{ij}^0$ , ориентируясь на некоторую нечеткую шкалу, наподобие той, которая представлена в табл. 1.

Таблица 1

Оценка силы связи между концептами

Лингвистическое значение силы связи	Числовой диапазон
Не влияет	0
Очень_слабая	(0; 0,15]
Слабая	(0,15; 0,35]
Средняя	(0,35; 0,6]
Сильная	(0,6; 0,85]
Очень_сильная	(0,85; 1]

Следующим шагом, определяющим действия эксперта, будет выбор границ интервала  $[W_{ij}, \bar{W}_{ij}]$ , определяющего серое значение силы связи  $\otimes W_{ij}$ . Это могут быть равноотстоящие от центрального значения  $W_{ij}^0$  числа, например:  $\otimes W_{ij} \in [W_{ij}^0 - \delta_{ij}, W_{ij}^0 + \delta_{ij}]$ , где  $\pm \delta_{ij}$  — разброс оценки относительно центра  $W_{ij}^0$ , но возможны и другие варианты.

Допустим, что эксперт оценил значения весов связей НСКК (рис. 1) определенным образом (табл. 2).

В табл. 2 в отдельном столбце приведены значения уровня "серости" (greyness) соответствующих "серых" чисел, определяемого как отношение размаха серого числа к общей длине диапазона его изменения  $[-1, 1]$ :

$$\Phi(\otimes W_{ij}) = |\bar{W}_{ij} - W_{ij}|/2. \quad (8)$$

Заметим, что концепты  $C_1, C_2, C_7$  на рис. 1 имеют собственные циклы положительной обратной связи с весами  $W_{11} = W_{22} = W_{77} = 1$ . Это указывает на то, что данные концепты выступают в качестве независимых источников входных сигналов НСКК, отражающих воздействия на смежные концепты со стороны внешней

Таблица 2

Значения весов связей НСКК

Вес связи	Значение веса связи	Серость (разброс оценки)
$W_{13}$	[0,65; 0,85]	0,1
$W_{14}$	[0,6; 0,75]	0,075
$W_{23}$	[0,6; 0,8]	0,1
$W_{24}$	[0,5; 0,7]	0,075
$W_{35}$	[0,6; 0,8]	0,1
$W_{36}$	[0,7; 0,85]	0,075
$W_{46}$	[0,5; 0,7]	0,1
$W_{73}$	[-0,6; -0,4]	0,1
$W_{74}$	[-0,6; -0,3]	0,15

среды (в работе [19] такие концепты названы драйверами).

Примем в качестве функции активации  $f(\cdot)$  концептов  $C_3, C_4, C_5, C_6$  двухполярную сигмиду (5). Проверка выполнения условия (7) для данных, приведенных в табл. 2, показывает, что

$$\left( \sum_{i,j=3}^6 \bar{W}_{ij}^2 \right)^{1/2} = \sqrt{2,98} < 2,$$

т. е. установившиеся состояния НСКК для рассмотренных ниже сценариев будут устойчивы. Значения весов связей, выходящих из драйверов, т. е. концептов  $C_1, C_2$  и  $C_7$ , согласно работе [18], в данном случае не учитываются.

Переходя непосредственно к расчетной части моделирования с помощью НСКК, рассмотрим следующие сценарии моделирования.

*A. Угроза "Несанкционированный доступ" при отсутствии дополнительных мер защиты (контрмер), что соответствует начальным условиям*

$$\begin{aligned} \otimes X(0) &= \\ &= ([0,8; 1], [0; 0], [0; 0], [0; 0], [0; 0], [0; 0], [0; 0]). \end{aligned} \quad (9)$$

*B. Угроза "Вредоносное программное воздействие" при отсутствии дополнительных контрмер, что соответствует начальным условиям*

$$\begin{aligned} \otimes X(0) &= \\ &= ([0; 0], [0,8; 1], [0; 0], [0; 0], [0; 0], [0; 0], [0; 0]). \end{aligned} \quad (10)$$

*C. Угроза "Несанкционированный доступ" при использовании дополнительных контрмер, что соответствует начальным условиям*

$$\begin{aligned} \otimes X(0) &= \\ &= ([0,8; 1], [0; 0], [0; 0], [0; 0], [0; 0], [0; 0], [0,8; 1]). \end{aligned} \quad (11)$$

*D. Угроза "Вредоносное программное воздействие" при использовании дополнительных контрмер, что соответствует начальным условиям*

$$\begin{aligned} \otimes X(0) &= \\ &= ([0; 0], [0,8; 1], [0; 0], [0; 0], [0; 0], [0; 0], [0,8; 1]). \end{aligned} \quad (12)$$

В качестве указанных контрмер могут выступать, например, межсетевые экраны, системы обнаружения атак, антивирусные программы и т. п.

Нетрудно видеть, что для сценария A расчетная схема моделирования существенно упрощается и принимает вид, представленный на рис. 2.

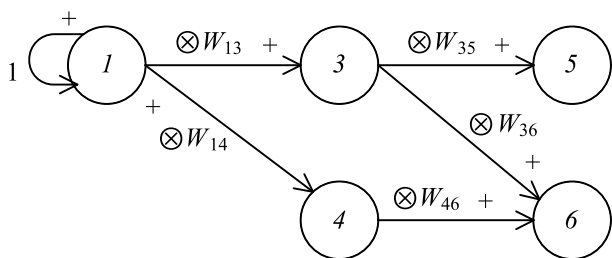


Рис. 2. Схема оценки рисков с помощью НСКК

Таблица 3

Верхние границы оценок состояния

$\bar{X}_i$	$k$							
	1	2	3	4	5	6	7	8
$\bar{X}_3$	0,40	0,55	0,60	0,62	0,63	0,63	0,63	0,63
$\bar{X}_4$	0,36	0,50	0,55	0,57	0,58	0,58	0,58	0,58
$\bar{X}_5$	0	0,16	0,21	0,33	0,39	0,42	0,43	0,43
$\bar{X}_6$	0	0,29	0,50	0,60	0,64	0,66	0,66	0,66

Таблица 4

Нижние границы оценок состояния

$\underline{X}_i$	$k$							
	1	2	3	4	5	6	7	8
$\underline{X}_3$	0,25	0,37	0,42	0,44	0,45	0,45	0,45	0,45
$\underline{X}_4$	0,24	0,34	0,39	0,41	0,42	0,42	0,42	0,42
$\underline{X}_5$	0	0,07	0,14	0,19	0,22	0,24	0,25	0,25
$\underline{X}_6$	0	0,15	0,28	0,37	0,41	0,44	0,45	0,45

Учитывая монотонный характер зависимостей  $X_5 = f_1(X_1, X_3)$  и  $X_6 = f_2(X_1, X_3, X_4)$ , можно отдельно провести оценку сначала верхних границ переменной  $\otimes X_5$  и переменной  $\otimes X_6$ :

$$\bar{X}_5 = f_1(\bar{X}_1, \bar{X}_3); \bar{X}_6 = f_2(\bar{X}_1, \bar{X}_3, \bar{X}_4),$$

а затем — аналогично оценку нижних границ  $\otimes X_5$  и  $\otimes X_6$ :

$$\underline{X}_5 = f_1(\underline{X}_1, \underline{X}_3); \underline{X}_6 = f_2(\underline{X}_1, \underline{X}_3, \underline{X}_4).$$

Для исходных данных, приведенных в табл. 2, соответствующие схемы НСКК для оценки верхней и нижней границ  $\otimes X_5$  и  $\otimes X_6$  принимают вид, представленный на рис. 3, а, б.

Далее, используя уравнения (3), можно рассчитать значения  $\bar{X}_i(k)$  и  $\underline{X}_i(k)$ , определяющие изменение верхних и нижних границ переменных состояния концептов  $X_i(k)$  во времени ( $k = 1, 2, \dots$ ). Выполнив соответствующие расчеты для схем

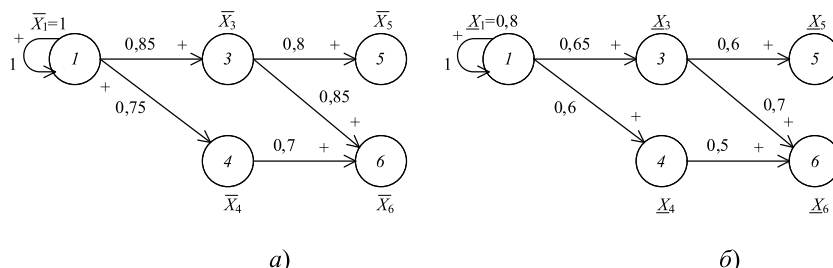


Рис. 3. Схемы НСКК для оценки верхней и нижней границ рисков ИБ

НСКК, представленных на рис. 3, а, б, для начальных условий (9) получим данные, приведенные в табл. 3, 4.

Как видно из табл. 3, 4, переменные состояния  $\bar{X}_i(k)$  и  $\underline{X}_i(k)$  за 7..8 тактов достигают своих установившихся значений, что является следствием выполнения условий устойчивости (7). В частности, для схемы на рис. 3, а имеем:

$$\left( \sum_{i,j=3}^6 \bar{W}_{ij}^2 \right)^{1/2} = \sqrt{1,85} = 1,36 < 2, \text{ т. е. условие (7)}$$

выполняется. Таким образом, серый вектор состояния НСКК  $\otimes X(k)$  сходится к установившемуся значению

$$\otimes X^*|_A = ([0,8;1], [0;0], [0,45;0,63], [0,42;0,58], [0,25;0,43], [0,45;0,66]),$$

а искомые оценки рисков ИБ вследствие нарушения конфиденциальности и целостности информации будут определяться серыми числами:

$$\otimes X_5^*|_A \in [0,25;0,43]; \otimes X_6^*|_A \in [0,45;0,66]. \quad (13)$$

Значения "серости" для указанных установившихся значений переменных состояния следующие:

$$\Phi_1^*|_A = 0,2; \Phi_2^*|_A = 0; \Phi_3^*|_A = 0,18; \Phi_4^*|_A = 0,16; \Phi_5^*|_A = 0,18; \Phi_6^*|_A = 0,21.$$

Следуя аналогичной процедуре, можно провести оценку диапазонов изменения рисков ИБ для сценария В (схема НСКК на рис. 4, а), сценария С (схема НСКК на рис. 4, б) и сценария D (схема НСКК на рис. 4, в).

После выполнения соответствующих расчетов с помощью уравнений (3) для начальных условий (10)—(12) получаем

для сценария В (рис. 4, а):

$$\otimes X_5^*|_B \in [0,23;0,42]; \otimes X_6^*|_B \in [0,41;0,65]; \quad (14)$$

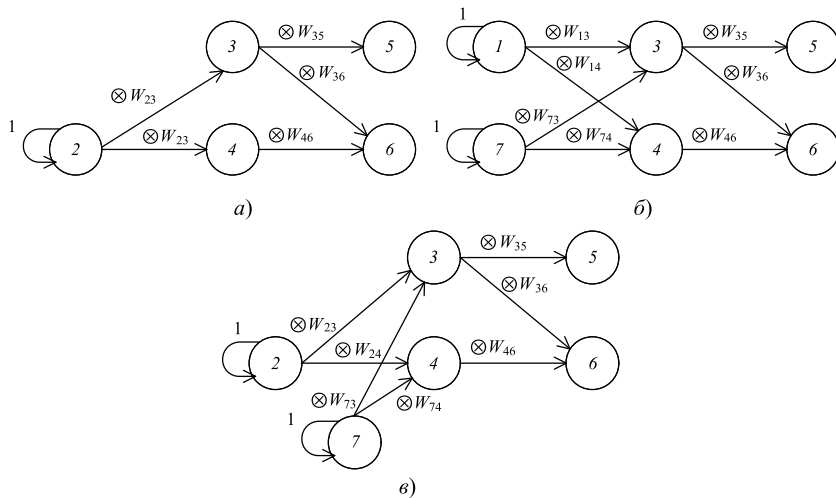


Рис. 4. Схемы НСКК для оценки рисков ИБ

для сценария *C* (рис. 4, б):

$$\otimes X_5^*|_C \in [0,1; 0,17]; \quad \otimes X_6^*|_C \in [0,22; 0,28]; \quad (15)$$

для сценария *D* (рис. 4, в):

$$\otimes X_5^*|_D \in [0,09; 0,15]; \quad \otimes X_6^*|_D \in [0,17; 0,22]. \quad (16)$$

В целях большей наглядности представим полученные результаты в виде диаграмм (рис. 5, а, б), где по оси абсцисс отложены значения чисел  $\otimes X_5$  и  $\otimes X_6$ , а по оси ординат — указания на соответствующий сценарий (вариант) моделирования.

Как видно из рис. 5, обе рассмотренные угрозы ("НСД" и "Вредоносное программное воздействие") при отсутствии дополнительных контрмер по защите информации (сценарии *A* и *B*) приводят к значительным рискам, причем ущерб от нарушения целостности информации ( $\otimes X_6$ ) превышает ущерб от нарушения ее конфиденциальности ( $\otimes X_5$ ). Применение дополнительных контрмер позволяет в 2–2,5 раза снизить соответствующие риски. Диапазон интервальных оценок ("серость" чисел  $\otimes X_5$  и  $\otimes X_6$ ) при переходе от стратегии *A* и *B* к стратегиям *C* и *D* при этом уменьшается примерно в той же пропорции, т. е. в 2,5–3 раза, что и абсолютные значения верхней и нижней границ этих чисел.

В целом, на основе полученных результатов можно сделать следующие общие выводы:

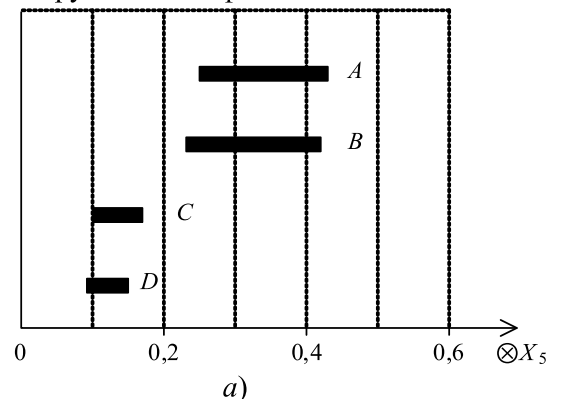
1) применение НСКК позволяет перейти от "точечных" оценок мнений экспертов (что обычно подвергается сомнению) к более мягким интервальным оценкам исходных дан-

ных и, как следствие, к получению интервальных оценок конечных результатов, что является, с одной стороны, более достоверным, а с другой стороны, предоставляет ЛПР большой материал для принятия окончательного решения с учетом его опыта и предпочтений;

2) рассмотренные выше интервальные оценки в представлении исходных данных (табл. 2) могут, вообще говоря, отражать не "осторожность" конкретного эксперта в оценке силы взаимосвязей между концептами, а разброс мнений группы экспертов, имеющих свое собственное представление об изучаемой проблеме;

3) являясь расширением классических НКК Б. Коско, НСКК сохраняют наглядность, интерпретируемость и способность к обучению на реальных данных, т. е. общепризнанные преимущества технологий когнитивного моделирования сложных, плохо формализуемых систем;

Нарушение конфиденциальности



Нарушение целостности

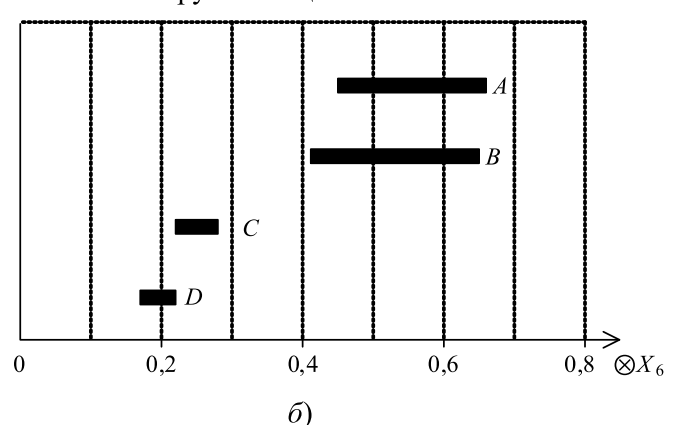


Рис. 5. Диаграммы значений рисков ИБ для различных сценариев

4) следующим шагом в исследовании возможностей применения НСКК для решения задачи оценки информационных рисков могло бы быть применение интуиционистских НКК [20], позволяющих отразить в интервальных оценках ("серых" числах) степень нерешительности (неуверенности, сомнения) экспертов в правильности этих оценок, т. е. внести элементы психологического анализа в процедуру получения этих оценок и соответственно моделирования плохо формализуемых процессов.

### Заключение

Предложен подход к оценке информационных рисков в компьютерных системах, основанный на применении технологии когнитивного моделирования с использованием нечетких серых (интервальных) когнитивных карт. В отличие от классических способов построения нечетких когнитивных карт, в данном случае для оценки силы взаимосвязей между концептами используются интервальные оценки ("серые" числа), характеризующие некоторую меру естественной неопределенности (размытости) в суждениях эксперта или группы экспертов относительно взаимовлияния указанных концептов. В качестве численного примера построения и анализа НСКК рассмотрена задача оценки рисков (потенциального ущерба) от нарушения конфиденциальности и целостности информации, вызванных воздействием на информационные активы угроз типа "Несанкционированный доступ" и "Вредоносное программное воздействие/вирусы". Проанализированы основные этапы реализации соответствующей процедуры когнитивного моделирования. Отмечаются несомненные преимущества применения НСКК для решения задачи оценки информационных рисков, связанные с получением более достоверных оценок исходных данных и предоставлением лицу, принимающему решения, больше степеней свободы для принятия окончательного и более обоснованного решения по существу изучаемого вопроса.

### Список литературы

1. Петренко С. А., Симонов С. В. Управление информационными рисками. Экономически оправданная безопасность. М.: ДМК Пресс, 2005. 384 с.

2. Астахов А. М. Искусство управления информационными рисками. М.: ДМК Пресс, 2010. 312 с.

3. Аникин И. В. Методы оценки и управления рисками информационной безопасности в корпоративных информационных сетях. Казань: Ред.-издат. центр "Школа", 2015. 224 с.

4. Kosko B. Fuzzy Cognitive Maps // Intern. Journal of Man-Machine Studies. 1986. Vol. 1. P. 65–75.

5. Силов В. Б. Принятие стратегических решений в нечеткой обстановке. М.: ИНПРО-РЕС, 1995. 228 с.

6. Stylios C. D., Georgopoulos V. C., Groumpos P. P. Introducing the theory of fuzzy cognitive maps in distributed systems // Proc. of the Twelfth IEEE Intern. Symposium on Intelligent Control, 16–18 July 1997, Istanbul, Turkey, 1997. P. 55–60.

7. Papageorgiou E. I. Review of Fuzzy Cognitive Maps Research During the Last Decade // IEEE Trans. on Fuzzy Systems. 2013. Vol. 21, N. 1. P. 66–79.

8. Papageorgiou E. I. (Ed.) Fuzzy Cognitive Maps for Applied Sciences and Engineering: From Foundations to Extensions and Learning Algorithms // Intelligent Systems Reference Library 54, Springer Science & Business Media. 2014. Vol. 54. 411 p.

9. Гузаиров М. Б., Васильев В. И., Кудрявцева Р. Т. Системный анализ информационных рисков с применением нечетких когнитивных карт // Инфокоммуникационные технологии. 2007. Т. 5, № 4. С. 42–48.

10. Степанова Е. С., Машкина И. В., Васильев В. И. Разработка модели угроз на основе построения нечеткой когнитивной карты для численной оценки риска нарушения информационной безопасности // Известия ЮФУ. Технические науки. Тематич. выпуск "Информационная безопасность". 2010. № 11 (112). С. 31–40.

11. Ажмухамедов И. М. Динамическая нечеткая когнитивная модель оценки уровня информационной безопасности информационных активов вуза // Вестник АГТУ. Сер.: Управление, вычислительная техника и информатика, 2012, № 2. С. 137–141.

12. Yebiah-Bouteng E. O. Using fuzzy cognitive maps (FCMs) to evaluate the vulnerabilities with ICT assets disposal policies // Intern. Journal on Electrical & Computer Science (IJECS-IJENS). 2012. Vol. 12, N. 05. P. 20–31.

13. Васильев В. И., Вульфин А. М., Кудрявцева Р. Т. Анализ и управление рисками информационной безопасности с использованием технологии когнитивного моделирования // Доклады ТУСУР. 2017. Т. 20, № 4. С. 61–66.

14. Salmeron J. L. Modelling grey uncertainty with Fuzzy Grey Cognitive Maps // Expert Systems with Applications. Dec. 2010. Vol. 37, Iss. 12. P. 7581–7588.

15. Shishkin V. M., Savkov S. V. The Method of Interval Estimation of Risk-Analysis System // Proc. of the Second Intern. Conference on Security of Information and Networks (SIN'09), Oct. 6–10. 2009. Famagusta, North Cyprus, 2009. P. 3–7.

16. Савков С. В., Шишкин В. М. Разработка системы интервального оценивания информационных рисков // Изв. вузов. Приборостроение. 2011. Т. 54, № 9. С. 38–43.

17. Deng J. L. Introduction to grey systems theory // Journal on Grey Systems, 1989. N. 1. P. 1–24.

18. Boutalis Y., Kottas T., Christodoulou M. On the existence and uniqueness of solutions for the concept values in fuzzy cognitive maps // Decision and Control, 2008. CDC 2008. 47<sup>th</sup> IEEE Conference, Cancun: IEEE. 2008. P. 98–104.

19. Knight Ch. J. K., Lloyd D. J. B., Penn A. S. Linear and Sigmoidal Fuzzy Cognitive Maps: An Analysis of Fixed Points. URL: [www.inescid.pt/indicators/Ficheros/175.pdf](http://www.inescid.pt/indicators/Ficheros/175.pdf), свободный (дата обращения: 08.04.2018).

20. Papageorgiou E. I., Iakovidis D. Intuitionistic fuzzy cognitive maps // IEEE Trans. on Fuzzy Systems, DOI: 10.1109/TFUZZ.2012.2214224.

V. I. Vasilyev, Professor, e-mail: vasilyev@ugatu.ac.ru,  
A. M. Vulfin, Professor, e-mail: vulfin.alexey@gmail.com,  
M. B. Guzairov, Professor, e-mail: guzairov@ugatu.su,  
A. D. Kirillova, Master of Sc., e-mail: kirillova.andm@gmail.com,  
Ufa State Aviation Technical University

## Interval Estimation of Information Risks with use of Fuzzy Grey Cognitive Maps

*The possibility of obtaining the interval quantitative estimates of information security risks with use of Fuzzy Grey Cognitive Maps is considered. The issues of constructing the fuzzy grey cognitive maps on the basis of knowledge processing and experts experience are discussed. The peculiarities of applying the given class of cognitive models on the example of evaluating the information risks are considered.*

**Keywords:** information risks, cognitive modelling, interval estimates, fuzzy grey cognitive maps

DOI: 10.17587/it.24.657-664

### References

1. Petrenko S. A., Simonov S. V. *Upravlenie informatsionnymi riskami. Ekonomicheski opravdannaya bezopasnost* (Information risk management. Economically proved security), Moscow, DMK Press, 2005. 384 p. (in Russian).
2. Astahov A. M. *Iskusstvo upravleniya informatsionnymi riskami* (The art of information risk management), Moscow, DMK Press, 2010. 312 p. (in Russian).
3. Anikin I. V. *Metody otsenki i upravleniya riskami informatsionnoj bezopasnosti v korporativnykh informatsionnykh setyah* (Methods for assessing and managing information security risks in corporate information networks), Kazan, Red.-izdat. centr "Shkola", 2015, 224 p. (in Russian).
4. Kosko B. Fuzzy Cognitive Maps, *Intern. Journal of Man-Machine Studies*, 1986, vol. 1, pp. 65–75.
5. Silov V. B. *Prinyatie strategicheskikh reshenij v nechetkoy obstanovke* (Making strategic decisions in a fuzzy environment), Moscow, INPRO-RES, 1995, 228 p. (in Russian).
6. Stylios C. D., Georgopoulos V. C., Groumpos P. P. Introducing the theory of fuzzy cognitive maps in distributed systems, *Proc. of the Twelfth IEEE Intern. Symposium on Intelligent Control*, 16–18 July 1997, Istanbul, Turkey, 1997, pp. 55–60 (in Russian).
7. Papageorgiou E. I. Review of Fuzzy Cognitive Maps Research During the Last Decade, *IEEE Trans. on Fuzzy Systems*, 2013, vol. 21, no. 1, pp. 66–79.
8. Papageorgiou E. I. (Ed.) Fuzzy Cognitive Maps for Applied Sciences and Engineering: From Foundations to Extensions and Learning Algorithms, *Intelligent Systems Reference Library 54*, Springer Science & Business Media, 2014, vol. 54. 411 p.
9. Guzairov M. B., Vasilyev V. I., Kudryavtseva R. T. Sistemnyy analiz informatsionnykh riskov s primeneniem nechetkikh kognitivnykh kart (The system analysis of information risks with application of fuzzy cognitive maps) *Infokommunikatsionnye Tekhnologii*, 2007, vol. 5, no. 4, pp. 42–48 (in Russian).
10. Stepanova E. S., Mashkina I. V., Vasilev V. I. Razrabotka modeli ugroz na osnove postroeniya nechetkoy kognitivnoy karty dlya chislennoy otsenki riska narusheniya informatsionnoj bezopasnosti (Development of threats model on the basis of fuzzy cognitive maps contraction for information risk numerical estimation), *Izvestiya YuFU. Tekhnicheskije nauki/Tematich. vypusk "Informatsionnaya bezopasnost"*, 2010, no. 11 (112), pp. 31–40 (in Russian).
11. Azhmuhamedov I. M. Dinamicheskaya nechetkaya kognitivnaya model otsenki urovnya informatsionnoj bezopasnosti informatsionnykh aktivov vuza (Dynamic fuzzy cognitive model for assessing the level of information security of university's information assets), *Vestnik AGTU. Ser.: Upravlenie, Vychislitel'naya Tehnika i Informatika*, 2012, no. 2, pp. 137–141 (in Russian).
12. Yebiah-Bouteng E. O. Using fuzzy cognitive maps (FCMs) to evaluate the vulnerabilities with ICT assets disposal policies, *Intern. Journal on Electrical & Computer Science (IJECS-IJENS)*, 2012, vol. 12, no. 05, pp. 20–31.
13. Vasilyev V. I., Vulfin A. M., Kudryavtseva R. T. Analiz i upravlenie riskami informatsionnoj bezopasnostju s ispol'zovaniem tehnologii kognitivnogo modelirovaniya (Analysis and management of information security risks with using cognitive modeling technology), *Doklady TUSUR*, 2017, vol. 20, no. 4, pp. 61–66 (in Russian).
14. Salmeron J. L. Modelling grey uncertainty with Fuzzy Grey Cognitive Maps, *Expert Systems with Applications*, Dec. 2010, vol. 37, iss. 12, pp. 7581–7588.
15. Shishkin V. M., Savkov S. V. The Method of Interval Estimation of Risk-Analysis System, *Proc. of the Second Intern. Conference on Security of Information and Networks (SIN'09)*, Oct. 6–10, 2009, Famagusta, North Cyprus, 2009, pp. 3–7.
16. Savkov S. V., Shishkin V. M. Razrabotka sistemy interval'nogo otsenivaniya informatsionnykh riskov (Development of a system of interval estimation of information risks), *Izv. vuzov. Priborostroenie*, 2011, vol. 54, no. 9, pp. 38–43 (in Russian).
17. Deng J. L. Introduction to grey systems theory, *Journal on Grey Systems*, 1989, no. 1, pp. 1–24.
18. Boutalis Y., Kottas T., Christodoulou M. On the existence and uniqueness of solutions for the concept values in fuzzy cognitive maps, *Decision and Control, 2008. CDC 2008. 47<sup>th</sup> IEEE Conference*, Cancun, IEEE, 2008, pp. 98–104.
19. Knight Ch. J. K., Lloyd D. J. B., Penn A. S. Linear and Sigmoidal Fuzzy Cognitive Maps: An Analysis of Fixed Points, available at: [www.inescid.pt/indicators/Ficheros/175.pdf](http://www.inescid.pt/indicators/Ficheros/175.pdf) (accessed 8 April 2018).
20. Papageorgiou E. I., Iakovidis D. Intuitionistic fuzzy cognitive maps, *IEEE Trans. on Fuzzy Systems*, DOI: 10.1109/TFUZZ.2012.2214224.



Н. А. Игнатьев, д-р физ.-мат. наук, проф., e-mail: n\_ignatev@rambler.ru,  
Национальный университет Узбекистана им. М. Улугбека

## Выбор собственного пространства объекта с использованием нелинейных преобразований признаков

*Рассмотрено описание объектов в собственном пространстве из латентных признаков. Для формирования пространства использованы правила иерархической агломеративной группировки исходных признаков. Предложена методика отбора информативного набора признаков объекта и вычисление меры сходства его с другими объектами по этому набору. Методика позволяет относить объект к числу эталонов либо идентифицировать его как шумовой.*

**Ключевые слова:** иерархическая агломеративная группировка, оценки объекта, логические закономерности, латентные признаки объекта, нелинейные преобразования.

### Введение

Разработка методов поиска собственного признакового пространства основана на идее [1], что в окрестности каждого объекта существует своя логическая закономерность. Результаты поиска востребованы при построении информационных моделей в предметных областях со сложной структурой отношений между объектами. Особый интерес представляют методы отбора информативных латентных признаков, инвариантных к масштабам измерений данных, комбинаторная сложность реализации которых позволяет получать результаты за приемлемое время.

Примерами предметных областей со сложной структурой отношений между объектами является медицина, психология и т. д. В медицине симптомы и синдромы людей с одинаковой патологией могут сильно отличаться друг от друга. Информативный набор признаков объекта может быть использован при выборе метода лечения, определении тяжести заболевания, группы инвалидности и т. д. Получить сравнимый по значимости эффект от методов отбора наборов признаков (показателей), информативных на всей обучающей выборке, практически невозможно.

В работе предлагается собственное пространство для описания объекта латентными

признаками формировать с помощью алгоритма иерархической агломеративной группировки исходных признаков. Как отдельная задача рассматривается выбор информативных латентных признаков объекта. Краткие сведения о возможности использования латентных признаков для выбора собственного пространства объекта содержатся в работе [2]. Необходимость (потребность) применения нелинейных отображений исходных признаков в латентные по правилам иерархической агломеративной группировки объясняется следующим образом:

- начиная с некоторой (изначально неизвестной) размерности исходного признакового пространства отношения близости между объектами становятся размытыми;
- структура отношений между объектами зависит от масштабов измерений количественных признаков и используемых мер близости.

Использование правил иерархической агломеративной группировки при формировании набора латентных признаков для описания объекта позволяет:

- гарантировать единственность решения независимо от масштабов измерений и выбранных мер близости;
- упорядочивать латентные признаки по отношению информативности;
- проводить отбор информативных латентных признаков для описания объекта инва-

риантных к масштабам измерений исходных признаков.

Методика отбора индивидуальных информативных наборов разнотипных признаков с помощью локальных метрик объектов описана в работе [3]. Для отбора был использован критерий на основе максимальной разницы частот встречаемости представителей (объектов) двух классов  $K_1$  и  $K_2$  в последовательности, упорядоченной по значениям расстояний от указанного объекта по его локальной метрике. Экстремальное значение критерия, полученное по медицинским данным с описанием состояния больных и практически здоровых индивидуумов, было предложено интерпретировать как индекс здоровья. Для нахождения экстремума использовали эвристические пошаговые алгоритмы отбора. Было показано, что различные схемы отбора признаков (последовательное удаление малоинформативных либо последовательное включение наиболее информативных) не давали схожих результатов.

Свойство инвариантности к масштабам шкал измерений является атрибутом нелинейного отображения групп исходных (сырых) разнотипных (номинальных и количественных) признаков на числовую ось. Результатом отображения являются латентные признаки. Описание допустимого объекта в рамках пространства из информативных латентных признаков необходимо для нахождения индивидуальной меры сходства (различия) с другими объектами. Эта мера отражает отношения между объектами и служит средством для принятия решения. Открываются перспективы обнаружения и интерпретации скрытых закономерностей в базах данных с единых методологических позиций.

Выбор критериев из некоторого конечного множества для принятия решения относительно объекта является трудно формализуемой проблемой. Предлагается рассматривать эту проблему как задачу распознавания из двух классов. Объекты классов могут быть представлены в виде оппозиции: больной — практически здоровый, богатый — бедный и т. д. Индивидуальная мера сходства (различия) в данном случае является аналогом свертки критериев.

Вычисление информативных наборов признаков можно применять для отслеживания изменения (мониторинга) состояния объекта. Например, изменение степени тяжести заболевания пациента в медицине, экологического

состояния окружающей среды, уровня кредитоспособности клиента банка и т. д.

Предлагается методика выбора информативных признаков объекта, основанная на использовании трех критериев. В рамках этой методики можно оценивать степень выраженности логических закономерностей относительно объекта по его информативному набору признаков. Значения оценок позволяют обосновывать использование объектов в качестве эталонов либо рассматривать их как шумовые.

### 1. Формирование собственного пространства объекта из латентных признаков

Рассматривается задача распознавания в стандартной постановке. Объекты обучения заданы через множество  $E_0 = \{S_1, \dots, S_m\}$ , разделенное на два непересекающихся подмножества (класса)  $K_1$  и  $K_2$ ,  $E_0 = K_1 \cup K_2$ . Описание объектов выполняется с помощью набора из  $n$  разнотипных признаков  $X(n) = (x_1, \dots, x_n)$ ,  $\xi$  из которых измеряются в интервальных шкалах,  $(n - \xi)$  — в номинальной.

Выбор собственного пространства для описания объекта  $S_d \in E_0$ ,  $d = 1, \dots, m$ , набором латентных признаков  $Y(\tau) = (y_1, \dots, y_\tau)$ ,  $\tau < n$  проводится с помощью правил алгоритма иерархической агломеративной группировки. Алгоритм группировки разбивает набор признаков  $X(n)$  на непересекающиеся группы  $X(k_1), \dots, X(k_r)$ ,  $k_1 + \dots + k_r \leq n$ . Нелинейное отображение представителей каждой группы на числовую ось образует новый латентный признак в описании объекта.

Считается, что известны критерии для отбора набора информативных латентных признаков  $Y(k) \subset Y(\tau)$  объекта. Требуется для указанного объекта  $S \in E_0$  определить:

- собственное пространство из латентных признаков  $Y(\tau)$ ,  $\tau < n$ ;
- информативный набор признаков  $Y(k) \subset Y(\tau)$ ,  $k \geq 0$ ;
- оценку объекта  $S$  на информативном наборе  $Y(k)$ .

Обозначим через  $I, J$  множество индексов соответственно количественных и номинальных признаков в исходном наборе  $X(n)$ . Для выбора латентных признаков в собственном пространстве объекта  $S_d \in E_0$ ,  $S_d = (a_{d1}, \dots, a_{dn})$  проведем предобработку данных следующим образом. Значения признаков объекта  $S = (b_1, \dots, b_n)$ ,  $S \in E_0$  преобразуем как

$$b_i = \begin{cases} |a_{di} - b_i|, i \in I, \\ 1, a_{di} = b_i, i \in J, \\ -1, a_{di} \neq b_i, i \in J. \end{cases} \quad (1)$$

Преобразованные по (1) признаки считаются измеренными в количественной шкале измерений, множество номеров которых идентифицируются как  $I = \{1, \dots, n\}$ . Для вычисления значений латентных признаков используют правила иерархической агломеративной группировки. Латентные признаки, полученные на  $p$ -м шаге группировки, обозначим как  $x_j^p, j \in I, p \geq 0$ . При  $p = 0, |I| = n$ . Упорядоченное множество значений признака  $x_j^p$  объектов из  $E_0$  разделим на два интервала  $[c_1^{jp}, c_2^{jp}], (c_2^{jp}, c_3^{jp}]$ , каждый из которых рассматривается как градация номинального признака.

Пусть  $u_i^1, u_i^2$  — число значений признака  $x_j^p, j \in I, p \geq 0$  класса  $K_i, i = 1, 2$  соответственно в интервалах  $[c_1^{jp}, c_2^{jp}], (c_2^{jp}, c_3^{jp}], |K_i| > 1, v$  — порядковый номер элемента упорядоченной по возрастанию последовательности  $r_{j_1}, \dots, r_{j_v}, \dots, r_{j_m}$  значений  $x_j^p$  у объектов из  $E_0$ , определяющий границы интервалов как  $c_1^{jp} = r_{j_1}, c_2^{jp} = r_{j_v}, c_3^{jp} = r_{j_m}$ . Критерий

$$\left( \frac{\sum_{i=1}^2 u_i^1 (u_i^1 - 1) + u_i^2 (u_i^2 - 1)}{\sum_{i=1}^2 |K_i| (|K_i| - 1)} \right) \times \left( \frac{\sum_{d=1}^2 \sum_{i=1}^2 u_i^d (|K_{3-i}| - u_{3-i}^d)}{2|K_1||K_2|} \right) \rightarrow \max_{c_1^{jp} < c_2^{jp} < c_3^{jp}} \quad (2)$$

позволяет вычислять оптимальное значение границы  $c_2^{jp}$  для интервалов  $[c_1^{jp}, c_2^{jp}]$  и  $(c_2^{jp}, c_3^{jp}]$ .

Экстремум критерия (2) используется в качестве веса  $w_j^p$  ( $0 \leq w_j^p \leq 1$ ) признака  $x_j^p$ . При  $w_j^p = 1$  значения признака  $x_j^p$  у объектов из классов  $K_1$  и  $K_2$  не пересекаются между собой. Особенность использования критерия (2) на данных, полученных по (1) при  $j \in J$ , заключается в том, что число различных значений признака равно числу классов. В этом случае граница между двумя классами определяется как  $c_2^{jp} = (c_1^{jp} + c_3^{jp})/2$ .

Для вычислений без учета масштаба изменений проведем предобработку признаков объекта  $S_r = (a_{r1}, \dots, a_{rn}), S_r \in E_0$  с помощью (2) как  $a_{ri}^0 = (a_{ri} - c_2^{i0}) / (c_3^{i0} - c_1^{i0}), i = 1, \dots, n$ . Значение комбинации  $b_{rij}^p$  по паре признаков  $(x_i^p, x_j^p)$ ,

$0 \leq p < n, i, j \in I, i \neq j$  объекта  $S_r = \{a_{ru}^p\}_{u \in I}, S_r \in E_0$  вычисляется таким образом:

$$b_{rij}^p = \eta_{ij} (t_i w_i^p a_{ri}^p + t_j w_j^p a_{rj}^p) + (1 - \eta_{ij}) t_{ij} w_{ij}^p (a_{ri}^p a_{rj}^p - c_2^{ijp}) / (c_3^{ijp} - c_1^{ijp}),$$

$$i, j \in I, t_i, t_j \in \{-1, 1\}, \eta_{ij} \in [0, 1],$$

где  $w_i^p, w_j^p, w_{ij}^p$  — веса признаков, определяемые по (2) соответственно по множеству значений  $x_i^p, x_j^p$ , и их произведения  $x_i^p x_j^p$  на  $E_0$ , значения  $t_{ij}, t_i, t_j \in \{-1, 1\}, \eta_{ij} \in [0, 1]$  выбираются по экстремуму функционала

$$\varphi(p, i, j) = \frac{\min_{S_r \in K_1} b_{rij}^p - \max_{S_r \in K_2} b_{rij}^p}{\max_{S_r \in E_0} b_{rij}^p - \min_{S_r \in E_0} b_{rij}^p} = \max_{t_{ij}, t_i, t_j \in \{-1, 1\}, \eta_{ij} \in [0, 1]} b_{rij}^p. \quad (3)$$

Экстремум функционала (3) интерпретируется как отступ между объектами классов  $K_1$  и  $K_2$  по множеству значений по паре признаков  $(x_i^p, x_j^p), 0 \leq p < n, i, j \in I, i \neq j$ .

Обозначим через  $\{z_{ij}^p\}_{i, j \in I}, p \geq 0$  — квадратную матрицу размера  $(n - p) \times (n - p)$ , значение элемента  $z_{ij}^p$  которой при  $p = 0$  определяется как

$$z_{ij}^p = \begin{cases} w_i^p, i = j, \\ \text{значению (2) по } \{b_{rij}^p\}_{r=1}^m, i \neq j, \end{cases} \quad (4)$$

через  $\Gamma_\eta, \eta > 0$  — подмножество номеров признаков из  $X(n)$ . Пошаговая реализация алгоритма иерархической агломеративной группировки будет такой.

**Шаг 1.**  $p = 0, \lambda c = 0, \eta = 1$ . Выполнять  $\Gamma_\eta = \{\eta\}, \text{Margin}_\eta = -2, \eta = \eta + 1$  пока  $\eta \leq n$ .

**Шаг 2.** Вычислить значения элементов матрицы  $\{z_{ij}^p\}_{i, j \in I}$  по (4).

**Шаг 3.** Выделить  $\Phi = \{z_{uv}^p | z_{uv}^p \geq \max(w_u^p, w_v^p)\}$  и  $u \neq v, u, v \in I$ . Если  $\Phi = \emptyset$ , то переход к шагу 9.

**Шаг 4.** Вычислить  $\lambda n = \max_{z_{uv}^p \in \Phi} z_{uv}^p$ . Выделить  $\Delta = \{(s, t), s, t \in I | z_{st}^p = \lambda n \text{ и } s < t\}$ . Определить пару  $\{i, j\}, i < j$  как

$$\{i, j\} = \begin{cases} \Delta, |\Delta| = 1, \\ \{s, t\}, (s, t) \in \Delta \text{ и} \\ \varphi(p, s, t) > \max_{(u, v) \in \Delta \setminus (s, t)} \varphi(p, u, v). \end{cases}$$

**Шаг 5.** Если  $\lambda n > \lambda c$  или  $\lambda n = \lambda c$  и  $\text{Margin}_i < \varphi(p, i, j)$ , то  $\Gamma_i = \Gamma_i \cup \Gamma_j, \Gamma_j = \emptyset, \text{Margin}_i = \varphi(p, i, j)$ , переход к шагу 7.

**Шаг 6.** Вывод номеров признаков из  $\Gamma_i, \Gamma_i = \emptyset, I = I \setminus \{i\}$ , переход к шагу 3.

**Шаг 7.**  $p = p + 1$ ,  $I = I \setminus \max(i, j)$ ,  $k = \min(i, j)$ ,  $\lambda c = \lambda n$ . Заменить значения признаков в описании объекта  $S_r = \{a_{ru}^{p-1}\}_{u \in I}$ ,  $r = 1, \dots, m$  на

$$a_{ru}^p = \begin{cases} a_{ru}^{p-1}, u \in I \setminus \{k\}, \\ b_{rj}^p, u = k. \end{cases}$$

**Шаг 8.** Для каждой пары  $(u, v)$ ,  $u, v \in I$  определить значение

$$z_{uv}^p = \begin{cases} z_{uv}^{p-1}, u \in I \setminus \{k\}, v \in I, \\ \text{значению (2) на } \{a_{rv}^p\}_{r=1}^m, u = k, v \in I. \end{cases}$$

Если  $n - p > 1$ , то переход к шагу 3.

**Шаг 9.** Конец.

## 2. Отбор информативных наборов признаков объекта

Формирование собственного пространства из латентных признаков по алгоритму из разд. 1 в общем случае не решает проблему обнаружения устойчивых логических закономерностей в окрестности объекта. Для решения этой проблемы необходимо использовать отбор информативного набора признаков.

Пусть для объекта  $S_d \in K_p$ ,  $p = 1, 2$  по алгоритму из разд. 1 определен его собственный набор латентных признаков  $Y(\mu)$ ,  $1 \leq \mu < n$ . Покажем, что значения, вычисленные по критерию (2), являются необходимым но недостаточным условием включения признака  $y_t \in Y(\mu)$ ,  $t = 1, \dots, \mu$  в информативный набор. Для проверки достаточности предлагается использовать два критерия.

Упорядочим множество объектов  $E_0$  по возрастанию значений признака  $y_t$  при описании их по набору  $Y(\mu)$  как

$$S_{d_1}, S_{d_2}, \dots, S_{d_m}. \quad (5)$$

Для оценки объекта  $S_d \in K_p$  по (5) применим критерий

$$F(S_d, y_t) = \max_{1 \leq i \leq m} \left( \frac{z_p(i)}{|K_p \cap E_0|} - \frac{z_{3-p}(i)}{|K_{3-p} \cap E_0|} \right), \quad (6)$$

где  $z_p(i)$ ,  $z_{3-p}(i)$  — число объектов в  $\{S_{d_1}, \dots, S_{d_i}\} \subset E_0$  из (5) из класса  $K_p$  и  $K_{3-p}$  соответственно.

Пусть по критерию (2) на (5) определены границы интервалов  $[c_1, c_2]$ ,  $(c_2, c_3]$ . Значение

критерия устойчивости объекта  $S_d \in K_p$  по латентному признаку  $y_t \in Y(\mu)$  вычисляется как

$$\Omega(y_t) = \theta_1(1 - \theta_2), \quad (7)$$

где

$$\theta_1 = \frac{|\{S_i \in K_p \mid y_t \in [c_1, c_2]\}|}{|K_p|},$$

$$\theta_2 = \frac{|\{S_i \in K_{3-p} \mid y_t \in [c_1, c_2]\}|}{|K_{3-p}|}.$$

Критерий (7) является мультипликативным, критерий (6) — аддитивным. Множество допустимых значений (6) и (7) принадлежат интервалу  $[0, 1]$ . Интерес представляет анализ значений этих критериев и использование их для отбора информативного набора признаков объекта. Методологически критерии (2), (6) и (7) связаны друг с другом. Так как с помощью критерия (2) формируется набор латентных признаков  $Y(\mu)$ ,  $\mu < n$ , то его использование предшествует применению критериев (6) и (7).

Рассмотрим условия отбора информативных признаков на  $Y(\mu)$  с использованием критериев (2), (6) и (7). Упорядочим возможные варианты на включение (не включение) признака  $y_t \in Y(\mu)$  в информативный набор по их приоритетности следующим образом:

- а) значения по (6) и (7) больше значения по (2);
- б) максимальное значение из (6) и (7) больше значения по (2);
- в) значение по (2) больше максимального значения по (6) и (7).

Истинность условия по варианту в) по всем латентным признакам указывает на то, что объект не отличается от объектов противоположного класса и может считаться шумовым.

Для формирования информативного набора  $Y(k) \subset Y(\mu)$  рекомендуется использовать условия вариантов а) и б). В качестве оценки объекта  $S_d \in K_p$  предлагается выбирать максимальное значение критерия (2) на признаках из набора  $Y(k)$ . Одной из форм использования оценок является селекция обучающих выборок.

Для мониторинга состояний объекта рекомендуется формировать обучающие выборки в виде оппозиции, например, класс практически здоровых людей и больных гипертонией в медицине. На практике выбор собственного пространства объекта при числе непересекающихся классов больше двух можно свести к задаче с двумя классами. С помощью информатив-

ных признаков можно визуализировать отношения объектов в одно-, двух- и трехмерном пространстве. Такое представление позволяет проводить анализ данных аналогично предложенному в методе локальной геометрии [1].

*О нелинейном отображении многомерных данных в пространство низкой размерности.* Нелинейное отображение многомерных данных [4] использовалось для представления геометрической структуры матрицы исходных данных в пространстве малой размерности (как правило, двух- или трехмерное) с сохранением, по возможности, расстояний между ними. С помощью методов отображения пытались минимизировать некоторую функцию потерь (меру искажения), характеризующую значение рассогласования расстояний между первоначальными и полученными векторами в пространстве малой размерности. Меры искажения, как правило, основаны на сравнении попарных расстояний между точками в исходном пространстве и пространстве отображения.

В данной работе близость между объектами представляет интерес с точки зрения ее влияния на степень истинности гипотезы о компактности классов. Методы обнаружения логических закономерностей очень чувствительны к размерности пространства, в котором вычисляется значение близости. Для исследования этой проблемы в работе [5] предложено использовать меру компактности классов и выборки в целом. Значение компактности связано с результатом разбиения объектов классов на непересекающиеся группы. При разбиении использовали свойство связанности объектов по оболочкам классов. Оболочки определяли как подмножество граничных объектов классов по заданной метрике.

### 3. Вычислительный эксперимент

Для эксперимента была взята выборка данных German из работы [6]. Выборка представлена 1000 объектами, разделенными на два класса  $K_1$  и  $K_2$ . В качестве объектов рассмотрены заемщики банка из Германии. Показатели заемщиков описываются 20 признаками, 7 из которых измеряются в количественных шкалах, 13 — в номинальных. В табл. 1 представлена последовательность выбора собственного пространства из латентных признаков для ряда объектов по результатам иерархической агломеративной группировки.

Как видно из табл. 1, объект 778 имеет оценку 0,2794 по критерию (2) на латентном признаке, полученном из комбинации  $((x_3, x_{15}), x_9, x_7)$ . Выбор этого признака в качестве информативного следует из того, что значение 0,3330 по критерию (7) больше 0,2794. Для объекта 702 в состав информативного набора входят латентные признаки  $(((((x_1, x_5), x_3), x_2), x_6), x_{13}), x_{15}), x_{20})$  и  $((x_{12}, x_{14}), x_{11}), x_4)$ . Оценка объекта 0,3479 является нелинейной комбинацией объединения исходных признаков в латентный в такой последовательности  $(((((проверка статуса, сумма кредита), кредитная история), продолжительность), состояние сбережений), возраст), корпус), иностранец)$ .

В табл. 2 приведены значения оценок для ряда объектов и последовательность формирования латентных признаков, на которых они получены.

Анализ табл. 2 показывает, что информативные латентные признаки объектов отлича-

Таблица 1

Выбор собственного признакового пространства объектов

№ объекта (класс)	Формирования групп признаков	Значение критерия		
		(2)	(6)	(7)
778(1)	$(((((x_1, x_5), x_{13}), x_{20}), x_{10}), x_8)$	0,3522	0,0	0,0848
	$(x_2, x_{14})$	0,2875	0,0243	0,1403
	$((x_4, x_6), x_{12}), x_{16})$	0,2936	0,0043	0,1318
	$((x_3, x_{15}), x_9, x_7)$	0,2794	0,1786	0,3330
	$(x_{17}, x_{19})$	0,2535	0,0438	0,2167
	$(x_{11}, x_{18})$	0,2502	0,0324	0,2440
702(2)	$(((((x_1, x_5), x_3), x_2), x_6), x_{13}), x_{15}), x_{20})$	0,3479	0,3467	0,4248
	$((x_{12}, x_{14}), x_{11}), x_4)$	0,2807	0,1881	0,3050
	$(x_7, x_8)$	0,2650	0,0057	0,1737
	$((x_9, x_{16}), x_{10}), x_{18}), x_{19})$	0,2635	0,0171	0,1780
	$(x_{17})$	0,2499	0,0686	0,2410

Таблица 2

Оценки объектов по латентным признакам

№ объекта (класс)	Формирование латентного признака	Оценка объекта
100(1)	$(((((про продолжительность, другие способы платежей), состояние сбережений), кредитная история), существующие кредиты)$	0,2934
490(1)	$((кредитная история, собственный телефон), личный статус), работа)$	0,2607
16(2)	$((проверка статуса, сумма кредита), иностранец), возраст)$	0,3352
918(2)	$((проверка статуса, сумма кредита), продолжительность), иностранец)$	0,3450

ются как по составу, так и по порядку включения исходных признаков.

### Заключение

Разработан алгоритм иерархической агломеративной группировки для формирования собственного пространства объекта из латентных признаков. Описаны условия выбора информативных признаков из собственного пространства. Результаты вычислений востребованы для наполнения баз знаний и построения информационных моделей в слабо формализованных предметных областях. Например, при медицинской компьютерной диагностике можно будет определять индивидуальный набор информативных показателей больного и рекомендовать ему соответствующий курс лечения.

### Список литературы

1. Дюк В. А. Методология поиска логических закономерностей в предметной области с нечеткой системологией: На примере клинико-экспериментальных исследований: Дисс. ... докт. тех. наук: Санкт-Петербург. 2005. 309 с.
2. Игнат'ев Н. А. Индексирование объектов по индивидуальным наборам информативных признаков // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2016. № 4 (37). С. 27–35.
3. Ignat'ev N. A., Mirzaev A. I. The Intelligent Health Index Calculation System // Pattern Recognition and Image Analysis. 2016. Vol. 26. N. 1. P. 73–77.
4. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика. Классификация и снижение размерности. М.: Финансы и статистика, 1989. 607 с.
5. Игнат'ев Н. А. О мере компактности объектов классов // Труды конференции "Проблемы современной топологии и ее приложения". Ташкент. 11–12 мая 2017 г. С. 283–285.
6. Machine Learning Repository. URL: <http://archive.ics.uci.edu/ml/datasets>.

N. A. Ignatiev, D. Sc., Professor, e-mail: [n\\_ignatev@rambler.ru](mailto:n_ignatev@rambler.ru),  
National University of Uzbekistan named after M. Ulugbek, Tashkent

## Selecting an Object's Own Space Using Nonlinear Transformations of Features

*The description of objects in their own space from latent features is considered. For the formation of the space, the rules of a hierarchical agglomeration grouping of initial features are used. The rules provide the property of invariance to the scales of measurement for non-linear mapping of groups of raw polytypical (nominal and quantitative) features onto the numerical axis. The need to apply non-linear displays of initial features to latent ones according to the rules of the hierarchical agglomeration group is explained as follows:*

- from with some (initially unknown) dimension of the source features space, the relations of proximity between the objects of classes become blurred;
- the structure of relations between objects depends on the scale of measurements of quantitative features and on the adjacency measures used.

*As a separate problem, the choice of informative latent features of the object is considered. The description of an admissible object within a space of informative latent features is necessary for finding an individual measure of similarity (difference) with other objects. This measure reflects the relationship between the objects and serves as a means for decision-making. It is practically impossible to obtain a comparable effect from the methods of selecting sets of features that are informative on the whole training sample. The technique of selection of an informative set of features of an object by three criteria is described. Measure of similarity in this set allows relate the object to the number of standards or to identify it as noisy. The prospects of discovering and interpreting hidden regularities in databases from unified methodological positions are revealed. For example, in the field of medical computer diagnostics, one can choose an individual course of treatment for a patient.*

**Keywords:** hierarchical agglomerative grouping, object estimates, logical regularities, latent features of the object, nonlinear transformations, class compactness

DOI: 10.17587/it.24.665-670

### References

1. Dyuk V. A. Metodologiya poiska logicheskikh zakonomenostey v predmetnoy oblasti s nechetkoy sistemologiyey: Na primere kliniko-eksperimental'nykh issledovaniy: (Methodology of the search for logical regularities in the subject domain with fuzzy systemology: On the example of clinical and experimental studies), Diss. ... dokt. tekh. nauk: Sankt-Peterburg, 2005, 309 p. (in Russian).
2. Ignat'yev N. A. Indeksirovaniye ob'yektov po individual'nym naboram informativnykh priznakov (Indexing of objects on individual sets of informative features), *Vestnik Tomskogo gosudarstvennogo universiteta. Upravleniye, vychislitel'naya tekhnika i informatika*, 2016, no. 4 (37), pp. 27–35 (in Russian).

3. Ignat'ev N. A., Mirzaev A. I. The Intelligent Health Index Calculation System. *Pattern Recognition and Image Analysis*, 2016, vol. 26. no. 1, pp. 73–77.
4. Ayvazyan S. A., Bukhshtaber V. M., Yenyukov I. S., Meshalkin L. D. *Prikladnaya statistika. Klassifikatsiya i snizheniye razmernosti* (Applied statistics. Classification and reduction of dimensionality), Moscow, Finansy i statistika, 1989, 607 p. (in Russian).
5. Ignat'yev N. A. O mere kompaktnosti ob'yektov klassov (About measure of compactness of objects of classes), *Trudy konferentsii "Problemy sovremennoy topologii i yeyo prilozheniya"*, Tashkent, 2017, pp. 283–285 (in Russian).
6. Machine Learning Repository, available at: <http://archive.ics.uci.edu/ml/datasets>

**В. П. Май**, канд. техн. наук, вед. науч. сотр., e-mail: may@iacp.dvo.ru,  
Институт автоматизации и процессов управления ДВО РАН, г. Владивосток

### Система объемной визуализации синоптических объектов

*Представлена усовершенствованная реализация интерактивной системы объемной визуализации и анализа синоптических объектов, предназначенной для автоматизации исследований динамики атмосферы и океана. Описана новая архитектура и функциональные возможности системы. Приведена программная структура разработанной системы визуализации. Реализованы методы визуализации физических полей синоптических объектов. Для увеличения вычислительной производительности обрабатываемых и визуализируемых данных применены шейдер- и CUDA-технологии многопроцессорной обработки на базе графических процессоров. Опытная эксплуатация и тестирование системы на модельных и реальных данных показали ее практическую эффективность.*

**Ключевые слова:** визуальный анализ, визуализация объемов, синоптические данные, скалярные и векторные поля, интерактивный интерфейс, шейдер- и CUDA-технологии, тропический циклон, аномалии

#### Введение

Объемная визуализация (*volume rendering*) является важным инструментом в задачах исследования динамики синоптических объектов атмосферы и океана. Разработка соответствующих программно-аналитических средств ведется довольно давно, однако потребность в совершенствовании методов/алгоритмов визуализации сохраняет свою актуальность в связи с необходимостью быстрой обработки больших объемов данных (приведение исходных данных к желаемому виду) и повышением требований к скорости и эффективности визуализации 3D-данных, облегчающей пространственное восприятие синоптических объектов. Необходимость в разработке специализированных средств 3D-графики определяется спецификой исследуемых синоптических объектов, имеющих многоплановый характер рассматриваемых физических полей скалярного и векторного типа с возможностью комбинированной визуализации и потребностью в наращиваемом наборе разных методов/алгоритмов визуализации и интерактивности.

Существует ряд зарубежных и отечественных программных разработок в этой области.

Из известных систем, отвечающих этим требованиям, можно выделить систему CAMVis от NASA [1, 2] и систему визуализации ураганов в 3D-окружении с использованием системы стереовидения VERTEX при поддержке проекта NOAA [3]. Эти системы отличаются большой степенью информативности и высоким качеством визуализации с высокопроизводительной аппаратной поддержкой. Однако они являются уникальными и не тиражируемыми, поскольку характеризуются достаточно сложной программной архитектурой с привязкой к специализированному вычислительному оборудованию и специфике форматов данных и решаемых задач в конкретном центре мониторинга.

Общедоступные системы имеют ряд недостатков, в особенности недостаточный уровень интерактивности интерфейса и информативности визуализации, а также ограниченный набор поддерживаемых форматов данных и методов визуализации. Поэтому в настоящее время проблема разработки систем визуализации синоптических объектов сохраняет свою актуальность.

Необходимость создания специализированной системы объемной визуализации синоптических объектов была продиктована потреб-

ностями Центра спутникового мониторинга Дальневосточного отделения Российской академии наук (ДВО РАН). С участием потенциальных пользователей-метеорологов сформулированы следующие требования к системе визуализации синоптических объектов.

1. Интерактивный удобный интерфейс системы визуализации с высокой степенью наглядности, позволяющий: исследовать многоплановый характер физических полей скалярного и векторного типа с необходимостью комбинированной визуализации; выполнять работу с различными специализированными форматами входных данных; объединять данные от разных спутников; осуществлять управление данными, параметрами и методами визуализации с хронологией времени для динамических данных.

2. Нарастаемый набор методов/алгоритмов визуализации и обработки данных с учетом многоплановости физических полей атмосферы и океана и исследуемых синоптических объектов.

3. Эффективная реализация алгоритмов, направленная на повышение скорости графической обработки и визуализации синоптических данных, в том числе с применением аппаратных графических возможностей и многоядерной архитектуры современных центральных процессоров.

В соответствии с этими требованиями была разработана система визуализации синоптических объектов [4, 5]. Эксплуатация системы подтвердила эффективность ее применения для определенных задач. В то же время необходимость повысить производительность системы и удобства работы с ней пользователи потребовали изменения ее архитектуры и интерфейса. В настоящей работе представлена усовершенствованная версия системы объемной визуализации синоптических данных. В новой версии реализованы: расширенные функциональные возможности для работы с данными; более высокая производительность обработки данных за счет применения GPU-параллелизма, что позволяет избавиться от привязки к стационарному вычислительному оборудованию; усовершенствованная модульная структура, обеспечивающая нетрудоемкое расширение алгоритмической базы без ограничений; современный эргономичный интерфейс, обеспечивающий эффективность и удобство в работе исследователя.

## Структура и функциональные возможности системы визуализации

Структура системы визуализации, разработанная с учетом обеспечения возможности наращивания и модификации алгоритмической базы, представлена на рис. 1. В настоящей версии программы реализована поддержка двух типов подключаемых модулей: загрузки данных и визуализации. Для каждого вида модулей разработана спецификация взаимодействия и потоков данных. Для модулей загрузки определяется место хранения данных и методы доступа к хранилищу из различных частей программного комплекса. Каждый модуль загрузки работает только с определенной группой данных и от определенного источника данных, что хотя и требует разработки множества модулей, но зато позволяет точно учитывать специфику обрабатываемых данных. Результатом работы модуля загрузки является не только загрузка, но и интерпретация данных и приведение их к принятому формату. Для модулей визуализации определена процедура интерактивных настроек параметров визуализации в режиме диалога. При этом реализована не только возможность одновременной визуализации нескольких видов синоптических данных (температура, влажность, давление, ветер), но и одновременное применение нескольких методов визуализации для одного набора данных (объемный рендеринг + изоповерхности + плоские срезы).

Разработанную систему можно использовать для визуализации различных пространственных статических и динамических полей

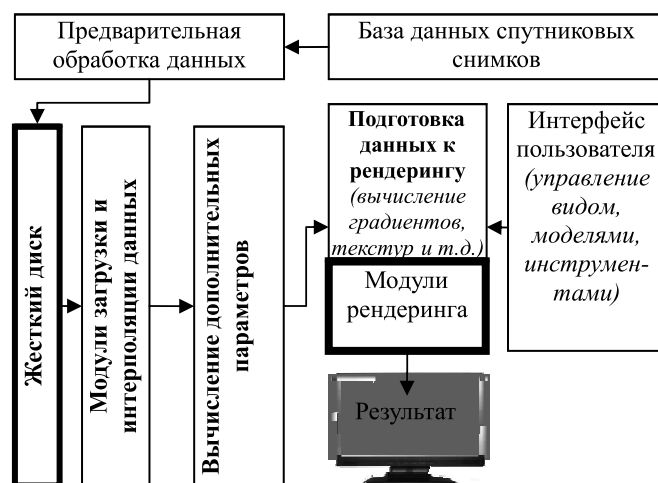


Рис. 1. Структура системы



скалярного и векторного типа. Вместе с тем она, прежде всего, ориентирована на применение в исследовании синоптических объектов в задачах динамики атмосферы и океана. Основной прикладной функциональной направленностью при этом являются:

- визуальное обнаружение важнейших пространственно-временных характеристик атмосферы, в частности влажности, положения тропопаузы;

- визуализация изоповерхностей температуры атмосферы;

- интерактивный анализ атмосферно-океанических аномалий, определение их сущности, происхождения, положения и времени жизни.

Для обеспечения указанных проблемно-ориентированных возможностей в интерфейсе системы предусмотрены следующие графические функции:

- интерактивное управление модулями загрузки и визуализации;

- темпоральные и пространственные срезы;

- гибкое управление интерфейсом;

- интерактивная обработка данных;

- выбор методов визуализации и инструментов для проведения визуального анализа данных.

К инструментарию относятся такие режимы, как:

- управление камерой ("Arcball", свободный полет, плоский режим, автоматическое слежение за динамическими объектами);

- управление данными (отсечение скаляров, фильтрация);

- управление визуализацией;

- управление динамическими полями.

Функциональной особенностью системы является возможность одновременной работы с множеством алгоритмов фильтрации и визуализации, работа с темпоральными срезами данных, с интерактивным выводом текущих визуальных и информационных данных. Все это требует создания большого количества визуальных форм, диалогов и информационных окон. Каждое окно является определенным инструментом для управления данными или визуализацией. Отсутствие систематизации диалогов в начальной версии породило проблему неудобства их использования: время от времени исследователю приходилось отвлекаться на поиск необходимого диалога, который мог быть не активен и закрыт другим диалогом. Таким образом, возникла необходимость в разработке более дружелюбного интерфейса.

Новый интерфейс программной системы, построенный на основе современной технологии Docking Windows, позволяет конфигурировать взаимное расположение окон с автоматической привязкой, масштабированием, системой вкладок (по аналогии с современными интернет-обозревателями). Гибкая система настроек взаимного расположения окон запоминается системой и используется при следующем запуске программы автоматически, что позволяет исследователю персонализировать интерфейс (рис. 2, см. вторую сторону обложки).

### Архитектура системы

В архитектуре программной системы учитывается специфика работы с синоптическими данными. Под синоптическими данными подразумеваются скалярные и векторные поля распределения погодных характеристик. Данные могут быть одномерными (показания зондов, стационарных радиолокационных станций), двумерными (погодные карты, спутниковые снимки облачности, льдов и т. д.) и пространственными (поля ветра, давления, температуры и т. д.). Кроме того, система визуализации должна работать со стандартными, используемыми синоптиками, форматами данных. При этом данные могут быть как статическими, так и изменяемыми во времени, т. е. динамическими. Динамические данные за определенный период могут быть представлены набором файлов данных или одним файлом — базой данных. В настоящей версии реализованы модули загрузки данных, способные загружать динамические данные обоих типов. Для загрузки многофайловых данных указывается их корневой каталог, после чего система запоминает имена доступных файлов, а также даты, которым они соответствуют (в зависимости от типа данных дата может быть получена из имени файла или метаданных внутри него). В момент, когда происходит смена текущей даты, выполняется загрузка соответствующего файла. Если соответствующего файла не существует, то загружаются ближайшие файлы по дате с двух сторон, после чего между ними выполняется интерполяция, результат которой отправляется для визуализации. Аналогичным образом происходит работа с базами данных: первым этапом считаются доступные даты, которые в дальнейшем

используются для выборки данных согласно структуре базы данных.

В настоящей версии реализован механизм автоматического включения актуальных данных при движении по шкале времени, расширяемой автоматически и охватывающей все включенные в проект динамические данные. Статические данные включены в визуализацию независимо от текущего значения шкалы времени.

Для эффективной и единообразной работы с разнотипными данными в предлагаемой архитектуре программной системы были реализованы следующие дополнительные возможности:

1) управление данными с помощью штатных средств интерфейса; подключение программ для предобработки данных (построение аномалий, перевод из текстовых профилей в бинарный унифицированный формат);

2) введение нового вида модулей — модулей предобработки, преобразующих исходные данные: вычисление градиентов, фильтрация, интерполяция, интерактивное разрежение сеток, сечение данных и т. д.;

3) разработка и реализация древовидной архитектуры управления данными, которая позволяет многоэтапную ступенчатую предобработку загруженных данных и обеспечивает визуализацию связей подключенных модулей;

4) сохранение текущего состояния проекта.

### Визуальный анализ

Программная система включает ряд функциональных возможностей, необходимых для проведения эффективного визуального анализа синоптических данных и направленных на улучшение понимания погодных явлений, а именно:

- трехмерная визуализация синоптических данных, с использованием различных алгоритмов графического представления; пользователь может поворачивать и перемещать камеру в удобное для него место, включая внутреннюю структуру визуализируемых объектов; область визуализации содержит инструмент для определения положения и направления камеры, а также инструмент для перемещения между датами;
- управление параметрами метода визуализации с помощью панели настройки, содержащей параметры, соответствующие выбранному методу;
- трехмерный информационный курсор, который показывает координаты (долготу,

широту и высоту) синоптического объекта под курсором мыши;

- комбинированная визуализация нескольких наборов данных, которая может быть применена для одновременной визуализации нескольких наборов данных как одного, так и множества объектов;
- управление хронологией динамических данных через временную шкалу (с возможностью интерполяции в местах отсутствия данных), а также через панель в окне визуализации, позволяющую перемещаться между доступными в наборе датами;
- визуализация скалярных и векторных полей; статическая анимация для методов многочастичной визуализации векторных полей;
- анимация динамических наборов данных, привязанных к временным диапазонам, управление анимацией выполняется с помощью временной шкалы со следующими функциями: старт, пауза, стоп, зацикливание воспроизведения, запись с настройками для создания фильмов;
- возможность создавать видеофильмы на основе анимации динамических наборов данных, а также делать снимки области визуализации.

### Алгоритмы визуализации объемов

В настоящей версии системы нашли применение следующие алгоритмы анимационной визуализации скалярных и векторных полей:

- алгоритм объемной текстурной визуализации;
- алгоритм объемной многочастичной визуализации;
- алгоритм визуализации изоповерхностей;
- алгоритм трассировки объемов на шейдерах.

Все перечисленные методы реализованы на OpenGL.

*Алгоритм объемной текстурной визуализации* базируется на использовании аппаратно поддерживаемых 3D-текстур. Для подготовки этапа рендеринга в рассматриваемом объеме строится семейство секущих плоскостей, параллельных плоскости экрана. Поле скалярных данных по определенной схеме интерпретируется как 3D-текстура, и графический акселератор берет на себя дальнейшую интерполяцию данных между ячейками сетки. Закраска плоскостей происходит средствами 3D-акселератора с использованием подготовленной и загруженной в видеопамять 3D-текстуры. При визуализации

данных можно оперировать такими параметрами, как цвет, прозрачность, освещенность. Вычислительные затраты метода сводятся к вычислению трехмерных текстурных координат, которые присваиваются вершинам примитивов, являющихся приближением секущих плоскостей. За счет использования прозрачности, реализованной в 3D-ускорителях, все, не относящиеся к исследуемому объему пиксели, остаются невидимыми, в результате на экране формируется реалистичное изображение.

*Алгоритм объемной многочастичной визуализации* заключается в отображении на экран частиц (в нашем случае это сферы малого диаметра) в соответствии со значением скаляра в рассматриваемых точках объема. Для реализации наглядности отображения скалярного поля варьируются следующие параметры: цвет частиц, освещенность, вероятность появления. Реализуется анимация с постоянной сменой положений частиц (мельтешение). В результате этого пользователь по характеру распределения частиц на последовательности кадров может судить о структуре скалярного поля. Для достижения наилучшего визуального эффекта пользователь может выбирать число частиц и их размер.

*Алгоритм визуализации изоповерхностей* основан на конвертировании воксельного представления изоповерхностей в полигональное представление с использованием метода "марширующих кубиков" [6], что позволяет обеспечить аппаратную поддержку визуализации изоповерхностей скалярных полей.

*Алгоритм трассировки объемов на шейдерах* [7] основывается на применении 3D-текстур, текстуры ColourLookUpTable (CLUT) и шейдер-технологии, основанной на использовании возможностей современных графических ускорителей. На первом этапе выполняется загрузка данных в видеопамять. Данные преобразуются к формату текстур с учетом того, что загружаемая в видеопамять текстура должна содержать значения в интервале  $[0, 1]$  по каждой компоненте. Для этого выполняется нормировка вычисляемых градиентов на скалярном поле и смещение значений на 0,5. Поле скаляров также приводится к интервалу  $[0, 1]$ . Таким образом, в каждой точке воксельной решетки имеются три значения вектора градиента и одно значение скалярного поля. После этого данные загружаются в четырехкомпонентную 3D-текстуру, где первые три компонента определяют нормаль в точке, а четвертая — норми-

рованное значение скалярного поля. На втором этапе в видеопамять загружается текстура CLUT. Создание текстур CLUT происходит с помощью редактора CLUT (рис. 3, см. вторую сторону обложки). Цветовая компонента берется из верхней строки текстуры, а компонента прозрачности из нижней.

Далее выполняется загрузка вершинного и пиксельного шейдеров в графический ускоритель. Поскольку известны физические размеры скалярного поля, есть привязка к системе координат меркаторской проекции. Операциями масштабирования и сдвига в буфер кадра вводится куб со стороной ребра, равной единице, при этом конвейер OpenGL для закраски куба использует ранее загруженные шейдеры. OpenGL настраивается таким образом, чтобы были видны только те грани куба, которые находятся ближе к наблюдателю. В результате шейдерная программа получает точку входа, преобразованную в систему координат текстуры. Теперь следует в вершинном шейдере вычислить вектор наблюдения, т. е. вектор от точки наблюдателя до точки входа. Источник освещения располагается в позиции наблюдателя, при этом вектор наблюдения и вектор освещения совпадают, что существенно сокращает время расчетов. Во фрагментном шейдере происходит накопление цвета по принципу "от ближнего к дальнему" до тех пор, пока луч не выйдет за пределы куба, или накопленная прозрачность не достигнет значения "пиксель непрозрачный". На каждом шаге накопления цвета из текстуры данных считывается значение преобразованного градиента и скаляра в точке.

При одновременной работе нескольких алгоритмов визуализации [7] с использованием Z-буфера могут возникать ситуации некорректного отображения результатов рендеринга. Во избежание такой ситуации при совместной работе метода трассировки лучей с другими алгоритмами, необходимо обеспечивать правильную последовательность выполнения: алгоритм трассировки должен обрабатывать после работы всех других алгоритмов. Это необходимо для того, чтобы до трассировки лучей был сформирован Z-буфер. Непосредственно в алгоритме трассировки при накоплении цвета на каждой итерации движения по лучу выполняется следующая проверка: находится ли очередная точка к наблюдателю ближе, чем фрагмент, занесенный в буфер кадра другими алгоритмами.

При реализации анимации во времени расчет параметров для визуализации (вычисление текстур, градиентов, минимаксных значений скалярных полей и построение полигональных моделей) выполняется незамедлительно. Анимация в пространстве поддерживается стандартным образом за счет упомянутых выше алгоритмов. Следует отметить, что модульная структура системы позволяет легко расширять базу алгоритмов визуализации.

### Многопроцессорная обработка

Работа с синоптическими данными — это работа с данными больших масштабов и объемов, что требует значительных вычислительных ресурсов, поэтому наиболее актуальным остается вопрос производительности. Данные подобного типа имеют, как правило, сеточную регулярную структуру. Нерегулярные данные могут быть регуляризованы на предварительном этапе обработки данных согласно актуальной модели. Задача обработки больших массивов данных эффективно решается с помощью технологий массового параллелизма с применением современной технологии параллельных вычислений на GPU CUDA [8]. Например, предварительная обработка исходных данных "оператором Собеля" с изменяемым размером ядра [9]. Размер ядра фильтра может быть изменен в любое время с помощью специальной панели в интерфейсе. Следует отметить, что ввод модулей с поддержкой CUDA не потребовал изменений архитектуры программной системы, что является следствием заложенного в архитектуру принципа универсальности при подключении новых и замене имеющихся модулей.

Поскольку физические поля заданы в узлах регулярных 3D-решеток, операции над ними

носят сеточный характер, что удобно для организации параллельных вычислений. Анализ показал, что этапы обработки синоптических данных в зависимости от их трудоемкости и подверженности распараллеливанию могут быть распределены между центральным процессором (CPU) и графическим ускорителем (GPU) следующим образом (рис. 4). Этап загрузки, восполнения данных и вычисления аномалий выполняется на CPU в многопоточном режиме, интерполяция и вычисление дополнительных параметров — на CUDA-ядре GPU. Подготовка данных к визуализации распределяется между CPU и GPU. Визуализация эффективно осуществляется средствами шейдер-технологии.

К дополнительным параметрам, которые вычисляются на регулярной сетке, относятся давление насыщенного водяного пара, относительная влажность, абсолютная влажность, массовая доля водяного пара, связь парциального давления и точки росы, массовая доля водяного пара.

Поскольку в каждой ячейке сетки рассчитывается несколько параметров (температура, влажность, давление и др.), все вычисления в ячейке можно разделить на независимые нити. Скорость вычислений на CUDA в данном случае будет сильно зависеть от эффективности использования операций чтения/записи в глобальную память GPU, так как это одна из самых затратных операций. Важной особенностью GPU является возможность объединения нескольких обращений к глобальной памяти в одну операцию над блоком (транзакцию). Для этого нужно обеспечить выполнение четырех условий:

- 1) нити должны обращаться к 32-битовым словам, давая при этом в результате один 64-байтовый блок;
- 2) общий блок обязательно должен быть выровнен в памяти;
- 3) все 16 слов должны лежать в пределах данного блока;
- 4) нити должны обращаться к словам последовательно, т. е.  $k$ -я нить должна обращаться к  $k$ -му слову.

Словом в нашем случае является 32-битовое число с плавающей запятой. Условие 2 выполняется еще при загрузке данных в GPU. Сетка блоков для CUDA-ядра формируется таким обра-

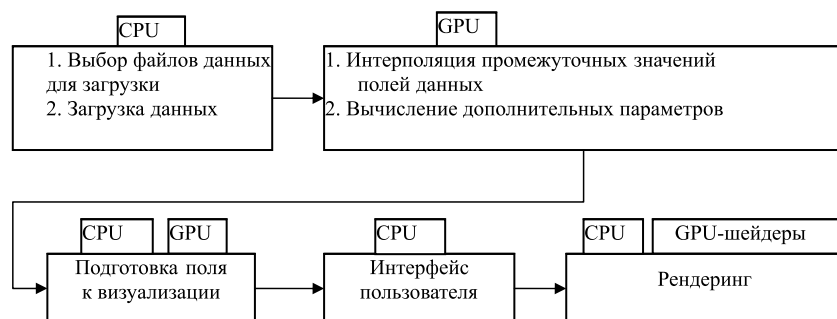


Рис. 4. Распределение вычислений между CPU и GPU

зом, чтобы обработке подверглись все данные 3D-сетки. А сетка нитей в блоке имеет размерность (16,  $X$ ). Организация блока, кратного 16 потокам, обеспечивает выполнение условий 1 и 3. Значение  $X$  выбирается таким образом, чтобы максимально эффективно использовать регистры мультипроцессора при вычислении конкретных параметров, так как каждый параметр имеет свою формулу и, таким образом, свою сложность. Последнее условие также обеспечивается тем, что последовательно расположенные нити могут обрабатывать последовательные ячейки сетки. Эксперименты показали, что указанное распределение вычислительной нагрузки между CPU и GPU носит оптимальный характер.

### Эффективность работы системы

Для оценки эффективности системы были проведены эксперименты по визуализации как для реальных, так и для модельных данных. При этом удовлетворено одно из требований к системе относительно возможности ее эксплуатации не только на многопроцессорной вычислительной системе, но и на персональных ЭВМ с обычным аппаратным обеспечением. Конфигурация компьютера, на котором проводили тесты: AMD Phenom II x4 965 3800+, 4 GBytes memory, video — GeForce Gtx 460 1024 MBytes.

Работа системы визуального анализа синоптических объектов для реальных данных может быть проиллюстрирована на примере моделирования динамики развития тропического циклона. Тропический циклон, в рамках задачи визуализации, можно определить как возмущение в атмосфере, в синоптике называемое аномалией — отклонением от нормы. Нормой считается среднестатистическая температура в районе. Процесс моделирования и визуализации динамики развития циклона имеет свои специфические особенности получения и обработки синоптических данных:

— использование темпоральной и пространственной интерполяции в целях преодоления сложности выбора из множества спутниковых данных тех, которые относятся к одному конкретному циклону за период от его зарождения до распада;

— необходимость сложной предварительной обработки синоптических данных, включающей такие этапы, как получение и фильтра-

ция данных от спутника, получение объемных данных, пространственное восполнение, регуляризация, расчет аномалий, вычисление дополнительных параметров (точка росы, относительная влажность).

Для визуализации циклонов используют данные спутниковых измерений пространственных физических полей в атмосфере (температуру, давление, влажность) в районах прохождения циклона. Для того чтобы отследить динамику развития циклона необходимо, чтобы движущийся циклон находился в зоне видимости спутников от момента его зарождения до момента его распада с учетом движения циклона. После первичной обработки спутниковые данные представляются в виде профилей (регулярная решетка) по уровням высоты над землей.

При работе с данными возникают две трудности. Первая трудность традиционная — отсутствие значений в некоторых точках и наличие ошибочных измерений. Вторая трудность связана с тем, что орбитальные спутники не способны обеспечить требуемую непрерывную "видимость" циклона, поскольку спутники движутся по фиксированным орбитам, а траектория движения циклона непредсказуема. Поэтому необходимо использовать совокупные измерения двух и более спутников, например спутников NOAA 15, 16, 18, 19. Однако два соседних спутниковых измерения, как правило, разделены не только во времени, но и могут относиться к различным регионам. Временные интервалы между измерениями могут изменяться от 30 мин до 12 ч. Если предположить, что циклон за такие интервалы времени меняется незначительно, то можно применить приемы пространственно-темпорального восполнения данных. Эксперимент показал, что в данном случае можно ограничиться простыми методами — восполнение по ближайшему "соседу" и линейная интерполяция. При этом применялись два способа совместного использования данных: а) объединение измерений; б) пересечение измерений. Каждый из них имеет свои очевидные преимущества и недостатки.

Для наглядного представления циклона в динамике предлагается использовать предварительно вычисленные карты температурных аномалий тропического циклона. Под аномалией понимается отклонение от среднего значения [10]. Анализ аномалий в пространстве и времени позволяет метеорологам судить о физике процесса. Если известно, что профили в

данном районе содержат циклон и положение центра циклона также известно, то на каждом уровне высоты можно построить среднее значение параметра (нормальное состояние). Для вычисления среднего берут значения из области периферии циклона, так как известно, что на периферии циклона состояние параметров близко к нормальному. Формально на каждом уровне высоты для вычисления среднего используют данные внутри кольца с внутренним радиусом 250 км от центра, и внешним радиусом 500 км, за пределами которого могут происходить процессы, уже не относящиеся к исследуемому циклону. Среднее значение температуры рассчитывается внутри кольца, центр которого совпадает с центром циклона. Аномалию на уровне вычисляют как разность измеренного и среднего значений. Затем для каждого уровня строят одномерную функцию зависимости значения аномалии от расстояния до центра циклона. Таким образом, получают распределение аномалий по высоте и удаленности от центра циклона.

При визуализации аномалий четко выделяется область, которая является ядром циклона. В ядре циклона сконцентрированы максимальные аномалии. Для визуализации ядра циклона используется пространственная регулярная решетка, центр которой помещается в центр циклона. Каждая ячейка решетки заполняется значением из карты аномалий в зависимости от удаленности от центра циклона. Размер решетки определяет компромисс между качеством визуализации и трудоемкостью вычислений. В приведенном ниже примере оптимальным выбором была решетка  $64 \times 64 \times 64$ .

Работа системы проиллюстрирована на примере визуализации реальных данных тропического циклона Melor со временем жизни с 29.09.2009 по 10.10.2009. Профили циклона получены по данным AMSU спутников NOAA 15, 16, 18, 19. Траектория циклона взята из базы данных KITAMOTO Asanobu @ National Institute of Informatics. Для визуализации температурных аномалий ядра циклона применена палитра цветов CLUT.

На рис. 5 (см. вторую сторону обложки) приведен пример визуализации реальных данных поля распределения температуры в циклоне с помощью алгоритма объемного шейдерного рендеринга. Можно наблюдать центр циклона и зону максимальных аномалий (отклонение от нормального значения), так назы-

ваемое "ядро". Среднее значение температуры рассчитывается внутри кольца, центр которого совпадает с центром циклона.

Для оценки эффективности системы также были использованы модельные данные динамики распределения влажности, температуры и давления прогнозной модели WRF, представленные скалярными и векторными полями, соответствующими временному периоду с 25.04.2012 по 29.04.2012, в одном из районов Приморского края.

На рис. 6, 7 (см. вторую сторону обложки) приведены примеры визуализации данных модели WRF.

Оценивая полученные на этапе тестирования результаты с точки зрения конечного пользователя, можно отметить, что система позволяет визуально обнаруживать важнейшие пространственно-временные характеристики атмосферы. На данном этапе система способна визуализировать изотемпературные поверхности атмосферы. Результаты визуализации могут быть использованы для интерактивного анализа атмосферных/океанических аномалий, определения их сущности, происхождения, положения и времени жизни.

## Заключение

В работе представлена действующая версия системы визуализации пространственных данных статических и динамических скалярных и векторных полей. Тестирование системы на реальных и модельных данных показало ее практическую эффективность. Реализован современный эргономичный интерфейс, обеспечивающий эффективность и удобство в работе исследователя.

Реализованные возможности многопроцессорной обработки прикладных данных по технологии CUDA и рендеринга на шейдерах обеспечили приемлемую скорость обработки больших объемов спутниковых данных и режим анимационной визуализации.

Система ориентирована на применение в задачах исследования динамики атмосферы и океана в качестве инструмента автоматизации научных исследований. Структура системы предусматривает развитие алгоритмической базы статической и анимационной визуализации, расширение типов используемых данных, повышение производительности вычислений.

В настоящее время система используется в центре спутникового мониторинга ДВО РАН для визуального контроля результатов первичной обработки оперативных спутниковых измерений и для исследования тропических циклонов на ретроспективных данных.

*Работа поддержана программой Президиума РАН № 1.33П "Фундаментальные проблемы математического моделирования"*

#### Список литературы

1. Green B., Henze C., Shen B.-W. Development of a scalable concurrent visualization approach for high temporal- and spatial-resolution models // AGU 2010 Western Pacific Geophysics Meeting, Taipei, Taiwan, June 22–25, 2010.
2. Shen B.-W., Bryan G., Tao W.-K., Henze C., Cheung S., Li J.-L. F., Mehrotra P. Coupling NASA Advanced Multi-Scale Modeling and Concurrent Visualization Systems for Improving Predictions of Tropical High-Impact Weather (CAMVis) // Earth Science Technology Forum (ESTF) 2010. Arlington, Virginia, June 22–24. 2010.
3. Wang Shao Rong, Rui You, Sheng Li, Yi Song Chen, Guo Ping Wang. Realtime Visualization of Hurricane in Distributed

Environment // Applied Mechanics and Materials (Vol. 40–41), Volume: Advances in Science and Engineering. 2010. P. 948–954.

4. Melman S. V., Bobkov V. A., May V. P. The system for visualization of synoptic objects // Proceedings 21 International Conference on Computer Graphics and Vision. GraphiCon 2011, September 26–30, 2011. Lomonosov Moscow State University. P. 103–106.

5. Бобков В. А., Мельман С. В., Май В. П. Визуализация динамики циклонов // Информационные технологии. 2012. № 2. С. 49–54.

6. Lorensen W. E., Cline H. E. Marching Cubes: a high resolution 3D surface reconstruction algorithm // SIGGRAPH 87, Conference Proceedings. 1987. P. 163–169.

7. Rossler F., Botchen R. P., Ertl T. Dynamic Shader Generation for GPU-Based Multi-Volume Ray Casting // Computer Graphics and Applications, IEEE. Sept.-Oct. 2008. Vol. 28, Is. 5. P. 66–77.

8. Lourenco L. H. A., Weingaertner D., Todt E. Efficient implementation of Canny Edge Detection Filter for ITK using CUDA // Proc. 13th Symp. Comput. Syst. Petropolis. Brazil. 17–19 Oct 2012. IEEE, 2012. P. 33–40.

9. Gupta S., Mazumdar S. G. Sobel Edge Detection Algorithm // International Journal of Computer Science and Management Research. 2013. Vol. 2, Is. 2. P. 1578–1583.

10. Kidder S. Q., Goldberg M. D., Zehr R. M., DeMaria M., Purdom J. F. W., Velden C. S., Grody N. C., Kusselson S. J. Satellite analysis of tropical cyclones using the Advanced Microwave Sounding Unit (AMSU) // Bulletin of the American Meteorological Society. 2000. No. 81. P. 1241–1259.

V. P. May, PhD, Leading Researcher, e-mail: may@iacp.dvo.ru,  
Institute of Automation and Control Processes of the Far Eastern Branch  
of the Russian Academy of Sciences, Vladivostok

## The Volume Visualization System of Synoptic Objects

*The volume visualization (volume rendering) is an important instrument in the problems of investigating dynamic of the synoptic objects of atmosphere and ocean. There is a strong demand to perfect the methods/algorithms of rendering in connection with a necessity of quick processing of large data volumes and gain in requirements on the speed and efficiency of 3D data rendering lightening the spatial perception of the synoptic objects. The well-known universally public program systems in this field have a number of disadvantages, particularly: deficient level of interface interactivity and informational content of visualization, and also there is the limited set of the supported data formats and visualization methods.*

*The advanced implementation of interactive program system of visualization and analysis of synoptic data intended for automation of atmosphere and ocean dynamic investigation is presented in the given work. The new architecture and functionality of the system are described. The program structure of the developed rendering system is given. The methods of rendering of the physical fields of synoptic objects are realized. To increase the calculating capacity of the processed and rendered data we used shader and CUDA technologies of the multiprocessor processing on a basis of graphic processors.*

*In the new version we realized: the extended functional possibilities for data running; higher performance of data processing at the sacrifice of GPU-parallelism that allows us to do away with binding to the stationary computing equipment; advanced modular structure providing not labor-consuming extension of an algorithmic base without restrictions; modern ergonomic interface providing efficiency and convenience in the researcher operating. The system was tested with the help of model and real data, and it showed the practical efficiency.*

**Keywords:** *visual analysis tool, volume visualization, synoptic objects, scalar fields, interactive interface, shaders, CUDA, tropical cyclone, anomaly*

DOI: 10.17587/it.24.671-680

## References

1. **Green B., Henze C., Shen B.-W.** Development of a scalable concurrent visualization approach for high temporal- and spatial-resolution models, *AGU 2010 Western Pacific Geophysics Meeting, Taipei, Taiwan, June 22–25, 2010*.
2. **Shen B.-W., Bryan G., Tao W.-K., Henze C., Cheung S., Li J.-L. F., Mehrotra P.** Coupling NASA Advanced Multi-Scale Modeling and Concurrent Visualization Systems for Improving Predictions of Tropical High-Impact Weather (CAMVis), *Earth Science Technology Forum (ESTF) 2010. Arlington, Virginia, June 22–24, 2010*.
3. **Wang Shao Rong, Rui You, Sheng Li, Yi Song Chen, Guo Ping Wang.** Realtime Visualization of Hurricane in Distributed Environment, *Applied Mechanics and Materials (vol. 40–41), Volume: Advances in Science and Engineering, 2010*, pp. 948–954.
4. **Melman S. V., Bobkov V. A., May V. P.** The system for visualization of synoptic objects, *Conference Proceedings. 21 International Conference on Computer Graphics and Vision. GraphiCon 2011, September 26–30, 2011. Lomonosov Moscow State University*, pp. 103–106.
5. **Bobkov V. A., Melman S. V., May V. P.** Vizualizatsiya dinamiki zyclonov (The visualization dynamic of cyclone), *Informatsionnye tekhnologii*, 2012, no. 2, pp. 49–54 (in Russian).
6. **Lorensen W. E., Cline H. E.** Marching Cubes: a high resolution 3D surface reconstruction algorithm, *SIGGRAPH 87, Conference Proceedings*, 1987, pp. 163–169.
7. **Rossler F., Botchen R. P., Ertl T.** Dynamic Shader Generation for GPU-Based Multi-Volume Ray Casting, *Computer Graphics and Applications, IEEE*, sept.-oct., 2008, vol. 28, iss. 5. pp. 66–77.
8. **Lourenco L. H. A., Weingaertner D., Todt E.** Efficient implementation of Canny Edge Detection Filter for ITK using CUDA, *Proc. 13th Symp. Comput. Syst. Petropolis, Brazil. 17–20 oct., 2012, IEEE, 2012*, pp. 33–40.
9. **Gupta S., Mazumdar S. G.** Sobel Edge Detection Algorithm, *International Journal of Computer Science and Management Research*, 2013, vol. 2, iss. 2, pp. 1578–1583.
10. **Kidder S. Q., Goldberg M. D., Zehr R. M., DeMaria M., Purdom J. F. W., Velden C. S., Grody N. C. and Kusselson S. J.** Satellite analysis of tropical cyclones using the Advanced Microwave Sounding Unit (AMSU), *Bulletin of the American Meteorological Society*, 2000, no. 81, pp. 1241–1259.

## ГЛАВНОЕ СОБЫТИЕ В ОБЛАСТИ ПРИБОРОСТРОЕНИЯ, ТОЧНЫХ ИЗМЕРЕНИЙ, МЕТРОЛОГИИ И ИСПЫТАНИЙ

МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ ИННОВАЦИОННЫЙ ФОРУМ

## ТОЧНЫЕ ИЗМЕРЕНИЯ – ОСНОВА КАЧЕСТВА И БЕЗОПАСНОСТИ

Москва, 15-17 мая 2019 года  
ВДНХ, Павильон №75

ОРГАНИЗАТОРЫ:

 **МИНПРОМТОРГ  
РОССИИ**

 **РОССТАНДАРТ**

ВЫСТАВОЧНЫЕ РАЗДЕЛЫ

 **METROEXPO**  
МЕТРОЛОГИЯ, ИЗМЕРЕНИЯ И ИСПЫТАНИЯ

 **CONTROL&DIAGNOSTIC**  
КОНТРОЛЬ И ДИАГНОСТИКА

 **LABTEST**  
ЛАБОРАТОРНЫЕ ИСПЫТАНИЯ

 **PROMAUTOMATIC**  
ПРОМЫШЛЕННАЯ АВТОМАТИЗАЦИЯ

 **RESMETERING**  
УЧЁТ ЭНЕРГОРЕСУРСОВ

 **WEIGHT SALON**  
ВЕСОВОЙ САЛОН



ЦИФРЫ И ФАКТЫ 2018 года:

Участники – 296 компаний из 24 стран мира

Посетители – 5046 специалистов из 63 регионов России

Площадь экспозиции – 6870 м<sup>2</sup>

ДИРЕКЦИЯ ФОРУМА

Тел./Факс: +7 (495) 937-40-23

E-mail: metrol@expoprom.ru

Спешите забронировать стенд [www.metrol.expoprom.ru](http://www.metrol.expoprom.ru)



# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИОМЕДИЦИНСКИХ СИСТЕМАХ

## INFORMATION TECHNOLOGIES IN BIOMEDICAL SYSTEMS

УДК 004.932.72'1, 57.087.1

DOI: 10.17587/it.24.681-688

**В. П. Новик**, инженер-исследователь, Iritech Inc., Москва, e-mail: novikvp@mail.ru,  
**И. А. Матвеев**, д-р техн. наук, и.о. гл. науч. сотр., e-mail: matveev@ccas.ru,  
**В. Ю. Леонов**, канд. физ.-мат. наук, науч. сотр.,  
Федеральный исследовательский центр "Информатика и управление"  
Российской академии наук, Москва

### Сравнение информативных признаков изображений радужки глаза методом оптимального пути

*В большинстве методов распознавания радужной оболочки глаза используются локальные текстурные признаки. Качество распознавания существенно зависит от точности их совмещения при сравнении. В работе представлен простой, при этом эффективный метод повышения точности совмещения и исследуется вопрос выбора оптимальных параметров вэйвлетов, формирующих шаблон радужной оболочки глаза. Показано, что улучшение точности совмещения позволяет использовать более информативные высокочастотные вэйвлеты. Численные эксперименты проведены на базах данных изображений ICE2005 и CASIA, находящихся в свободном доступе.*

**Ключевые слова:** оптимальный путь, биометрические признаки, распознавание по радужной оболочке глаза

#### Введение

В настоящее время распознавание по изображению радужной оболочки глаза (РОГ) является одним из основных методов биометрической идентификации. РОГ — кольцевая область, расположенная между зрачком и склерой. Текстура РОГ имеет большое число локальных деталей, составляющих уникальный "рисунок" признаков для каждого человека. Наиболее успешные методы распознавания используют эти признаки, получая их локально-спектральными преобразованиями, например сверткой с вэйвлетами Gabor или LoG. Такой подход был предложен уже в самых ранних работах [1, 2].

С тех пор было разработано большое число других способов описания и сравнения РОГ. В работах [3, 4] используются "ключевые точки" радужки, сравнивается их взаимное расположение. В работе [5] показаны возможности описания текстур РОГ признаками, основанными на порядковых статистиках. Используются глобальные признаки, такие как дескрипторы PCA, LDA или ICA [6, 7], признаки гистограмм [8], преобразование Фурье [9] и ряд других [10]. Все эти способы дают

ошибку распознавания  $EER \geq 1\%$ , что хуже, чем результаты методов, основанных на локальных описаниях. Также возможно использование ковариационных матриц градиентов [11], морфологического описания и сравнения [12] элементов РОГ. Использование локальных признаков вводит существенные требования к точности работы методов локализации границ радужки и совмещения частей изображений. Ошибки локализации приводят к нелинейным сдвигам между соответствующими участками сравниваемых шаблонов.

На рис. 1 схематически показан результат нормализации исходного изображения (а) с правильными границами (б), с ошибкой определения центра зрачка (в) и ошибкой определения центра радужки (г).

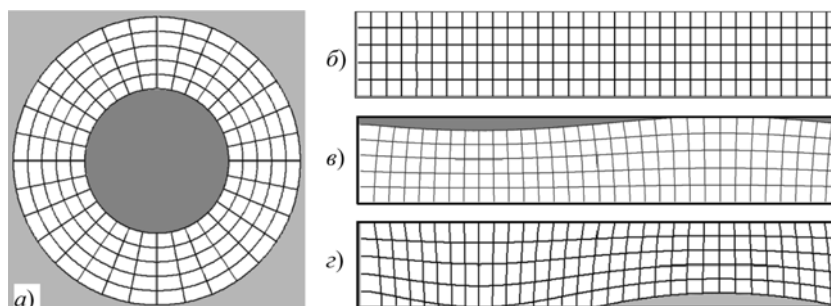


Рис. 1. Искажения нормализованного изображения

Деформация нормализованного изображения носит регулярный характер. Однако аналитически точно описать деформацию довольно сложно, требуется учет по крайней мере четырех параметров, которые могут быть определены с погрешностью (две координаты центра глаз и два радиуса окружностей, аппроксимирующих внутреннюю и внешнюю границы РОГ). Для более сложных моделей число параметров увеличивается. Например, для модели двух неконцентрических окружностей нужно учитывать уже шесть параметров. Задача аналитического определения деформаций радужки плохо обусловлена, ее численное решение практически неприменимо ввиду неустойчивости. Поэтому используют эвристические способы совмещения.

В случае умеренной ошибки (несколько процентов от радиуса радужки) определения границ процесс нормализации почти не искажает локальные признаки и сохраняет их сходство. По этой причине в большинстве работ на первом этапе проводят нормализацию с неточными значениями границ, затем для локальных участков одного шаблона выполняют поиск наилучших совпадений в другом шаблоне с некоторым окном сканирования. Размер окна сканирования (максимального взаимного смещения участков) задают исходя из ожидаемой ошибки определения границ. Понятно, что картина оптимальных смещений локальных участков в целом соответствует искажениям, вносимым неточностями определения границ, но с выбросами, которые вносятся ложными совпадениями. При значительном числе ложных совпадений возрастает число ошибок второго рода — случаев ложного определения двух несхожих шаблонов как близких. Исходя из этого, качество сравнения шаблонов может быть улучшено применением того или иного метода совместного анализа локальных смещений в целях исключения выбросов. В работе [13] рассмотрен случай произвольных смещений, которые затем связываются с помощью скрытой марковской модели. В работе [4] вычисление нелинейных смещений радужки осуществляется путем отслеживания соответствующих точек с использованием коррелятора специального вида. В [14] предлагаются глобальные уточнения (чтобы компенсировать угол обзора) и локальные уточнения (для компенсации ошибок в определении положений зрачка и радужки). В работах [15–17] оптимальные смещения участков радужки вычисляют с использованием деформируемого графа. Перечисленные подходы имеют большую вычислительную

сложность. В данной работе предлагается более простой и быстрый метод, основанный на вычислении оптимального пути на растре.

В следующем разделе приведена процедура формирования шаблона радужки. Затем дано применение метода оптимального пути для совмещения шаблонов. В последнем разделе описаны вычислительный эксперимент и его результаты.

## 1. Создание шаблона радужки из изображения

Рассмотрим тестовую базу данных, содержащую набор изображений глаз. Каждому изображению базы приписан идентификатор персоны, которой оно принадлежит, чтобы иметь возможность проверять правильность идентификации. В базе данных более одного человека и более одного изображения на каждого человека. Основная схема обработки представлена на рис. 2 (см. третью сторону обложки). Отдельные блоки представляют этапы выполнения алгоритма: исходное изображение глаза (*a*); локализация границ (*b*); нормализованное изображение и маска закрытости (*в*); извлечение признаков, формирующих шаблон (*г*); сравнение шаблонов (*д*); вычисление ошибок распознавания (*е*). Стрелки представляют взаимосвязь между блоками обработки.

На исходном изображении  $I(x, y)$  последовательностью методов, описанных в работе [18], вычисляются внутренняя и внешняя границы радужки и маска затенения (*occlusion*). Внутренней границей радужки является периметр зрачка, аппроксимируемый окружностью с центром и радиусом  $(x_p, y_p, z_p)$ . Внешняя граница радужки — окружность  $(x_l, y_l, z_l)$ , наилучшим образом аппроксимирующая границу радужка—склера. Маска затенения  $M(x, y)$  — бинарное изображение того же размера, что и исходное. Точки маски принимают ненулевое значение, если соответствующая им область радужки не искажена веками, ресницами, бликами или другими причинами. К исходному изображению и его маске применяется *нормализация* — преобразование, отображающее кольцо, заключенное между внешней и внутренней границами радужки, в прямоугольную область. В этой области горизонтальная ось  $0\phi$  приблизительно соответствует полярному углу, а вертикальная ось  $0\rho$  — радиальному перемещению от границ зрачка к внешней границе радужки. Верхнее и нижнее изображения на рис. 2, *в* являются нормализацией

исходного изображения  $I(x, y)$  и его маски затенения  $M(x, y)$  и обозначаются  $I(\varphi, \rho)$  и  $M(\varphi, \rho)$  соответственно.

Для нормализации изображения используется модель "rubber-sheet" [19].

Для каждой точки нормализованного изображения  $(\varphi, \rho)$ ,  $\rho \in [0; 1]$ ,  $\varphi \in [0; 2\pi]$  находятся соответствующие ей координаты на исходном изображении:

$$\begin{aligned} x(\varphi, \rho) &= (1 - \rho)x_1(\varphi) + \rho x_2(\varphi); \\ y(\varphi, \rho) &= (1 - \rho)y_1(\varphi) + \rho y_2(\varphi); \\ x_1(\varphi) &= x_p + r_p \cos(\varphi); y_1(\varphi) = y_p + r_p \sin(\varphi); \\ x_2(\varphi) &= x_l + r_l \cos(\varphi); y_2(\varphi) = y_l + r_l \sin(\varphi). \end{aligned} \quad (1.1)$$

Яркость точки нормализованного изображения получается билинейной интерполяцией:

$$\begin{aligned} I(\varphi, \rho) &= (1 - \{x\})(1 - \{y\})I(\lfloor x \rfloor, \lfloor y \rfloor) + \\ &+ \{x\}(1 - \{y\})I(\lfloor x \rfloor + 1, \lfloor y \rfloor) + \\ &+ (1 - \{x\})\{y\}I(\lfloor x \rfloor, \lfloor y \rfloor + 1) + \\ &+ \{x\}\{y\}I(\lfloor x \rfloor + 1, \lfloor y \rfloor + 1), \end{aligned} \quad (1.2)$$

где  $\lfloor a \rfloor$  и  $\{a\}$  обозначают дробную и целую часть числа  $a$ .

Признаки радужки  $P(\varphi, \rho)$  вычисляются с помощью свертки нормализованного изображения (1.2) с фильтром Лог-Габора, преобразование делается в спектральной области:

$$P(\varphi, \rho) = F^{-1}(F(I(\varphi, \rho))G(u)), \quad (1.3)$$

где  $F$  — преобразование Фурье;  $G(u) = \exp\left[-\frac{1}{2}\left(\frac{\ln \lambda + \ln u}{\ln \lambda - \ln \sigma}\right)^2\right]$  — фильтр Лог-Габора в спектральной области, параметры  $\sigma$  и  $\lambda$  определяют характеристики фильтра;  $u$  — аргумент, соответствующий углу  $\varphi$  в спектральной области.

Признаки, используемые для формирования шаблонов, вычисляются как бинарные значения вещественных и мнимых частей массива  $P(\varphi, \rho)$ :

$$\begin{aligned} T_{\Re}(\varphi, \rho) &= \begin{cases} 1, & \text{если } \Re(P(\varphi, \rho)) > 0, \\ 0, & \text{если } \Re(P(\varphi, \rho)) \leq 0, \end{cases} \\ T_{\Im}(\varphi, \rho) &= \begin{cases} 1, & \text{если } \Im(P(\varphi, \rho)) > 0, \\ 0, & \text{если } \Im(P(\varphi, \rho)) \leq 0. \end{cases} \end{aligned} \quad (1.4)$$

В общем виде шаблон может содержать любую совокупность локальных признаков, которые вычисляются в регулярной сетке. В данной работе для экспериментов используются

именно бинарные признаки. Таким образом, каждое изображение глаза  $I(x, y)$  преобразуется в шаблон  $T(\varphi, \rho)$  и соответствующую ему маску затенения  $M(\varphi, \rho)$ .

## 2. Сравнение шаблонов

Для любых двух бинарных шаблонов  $T_1$  и  $T_2$  можно вычислить нормализованное расстояние Хэмминга:

$$d_0(T_1, T_2) = \frac{1}{|\Omega|} \sum_{(\varphi, \rho) \in \Omega} T_1(\varphi, \rho) \oplus T_2(\varphi, \rho), \quad (2.1)$$

где  $\Omega = M_1 \cap M_2$  — пересечение незатененных областей двух сравниваемых шаблонов. Ввиду неопределенности угла поворота радужки используется более сложная формула расстояния. Вращение исходного изображения глаза эквивалентно циклическому сдвигу нормализованного изображения вдоль оси  $0\varphi$ . Поэтому один из шаблонов (вместе с маской) подвергается нескольким операциям сдвига и сравнения:

$$\begin{aligned} d(T_1, T_2) &= \min_{\psi \in [-S; S]} d(T_1, T_2, \psi); \\ d_0(T_1, T_2) &= \frac{1}{|\Omega|} \sum_{(\varphi, \rho) \in \Omega} T_1(\varphi, \rho) \oplus T_2(\varphi, \rho); \\ \Omega(\psi) &= M_1(\varphi + \psi) \cap M_2(\varphi), \end{aligned} \quad (2.2)$$

где  $\psi_n \in [S; S]$  — угол поворота изображения.

Разделим шаблон  $T_1$  на  $N$  непересекающихся секторов одинакового размера  $T_1^n$ ,  $n \in [1; N]$ , которые расположены вдоль угловой оси  $\varphi$ , как показано на рис. 3.

Каждый такой сектор может быть смещен на некоторый угол  $\psi_n \in [S; S]$  и сравнен с соответствующей частью шаблона  $T_2$  согласно (2.1). Вычисленные таким образом расстояния могут быть представлены в виде матрицы  $D = \{d_{\psi, n}\}$  с  $2S + 1$  строками и  $N$  столбцами. Можно отметить, что вычислительная сложность получения этой матрицы не превышает вычислительную сложность определения расстояния (2.2). В этих обозначениях расстояние (2.2) получается как минимальная сумма по строкам матрицы  $D$ :

$$d(T_1, T_2) = \min_{\psi} \sum_n d_{\psi, n}. \quad (2.3)$$

Угловые смещения всех секторов одинаковы, что можно назвать моделью недеформируемого "твердого тела". Вместе с тем угловые смещения могут быть сделаны независимыми,

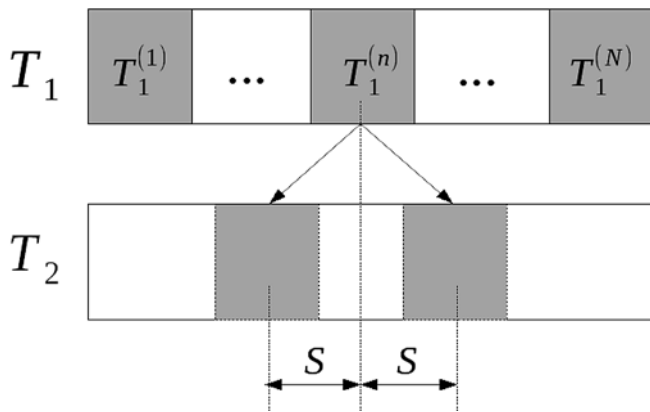


Рис. 3. Сравнение секторов с угловым сдвигом

если минимизировать расстояние по каждому сектору отдельно:

$$d(T_1, T_2) = \sum_n \min_{\psi} d_{\psi, n}, \quad (2.4)$$

что соответствует модели тела, абсолютно эластичного по отношению к вращениям, назовем такую модель "жидкой". Как показали численные эксперименты (см. разд. 3), эта модель дает меньшую ошибку распознавания, чем модель "твердого тела" (2.3). По мере увеличения числа секторов ошибка растет вследствие увеличения вероятности обнаружения ложных совмещений для небольших секторов.

Таким образом, возникает идея, что можно улучшить точность совмещения шаблонов, если ввести некоторые ограничения на взаимное движение секторов. То есть способ деформаций шаблона должен быть промежуточным между "твердой" (2.3) и "жидкой" (2.4) моделями.

Гладкость преобразования нормализации означает, что значения  $\psi_n$  для соседних секторов должны быть близки. Значения для от-

даленных секторов могут сильно различаться. Вследствие цикличности преобразования значения углов  $\psi_1$  и  $\psi_N$  также должны быть близки. Таким образом, возникает задача выбора последовательности элементов матрицы  $\{d_{\psi, n}\}$  со следующими требованиями: должен быть единственный элемент в каждом столбце; положение элементов в соседних столбцах изменяется не более, чем на единицу; сумма значений выбранных элементов минимальна. Эту задачу можно представить как поиск циклического пути минимальной стоимости в матрице  $\{d_{\psi, n}\}$ :

$$d(T_1, T_2) = \min_{(\psi_1, \dots, \psi_N)} \sum_{n=1}^N [d_{\psi, n} + C(\psi_n, \psi_{n+1})];$$

$$C(\psi_n, \psi_{n+1}) = \begin{cases} 0, & \text{если } \|\psi_n - \psi_{n+1}\| \leq 1, n \neq N, \\ 0, & \text{если } \|\psi_n - \psi_N\| \leq 1, n = N, \\ \infty, & \text{иначе.} \end{cases} \quad (2.5)$$

Задача (2.5) может быть решена одним из известных методов [20, 21]. На рис. 4 представлены примеры матриц  $\{d_{\psi, n}\}$  для трех типичных случаев. Матрицы имеют размеры  $N = 15$ ,  $S = 12$ . Большие значения элементов матрицы соответствуют более ярким элементам рисунка. Найденный оптимальный путь отмечен белыми точками.

Первая матрица (рис. 4, а) — сравнение двух шаблонов радужки одного человека (т. е. "своих" сравнений), когда все параметры границы обнаружены правильно (или имеют малые ошибки). В этом случае поворот не требуется, оптимальным путем является прямая линия, и для сравнения подходит даже простая модель "твердого тела". Матрица на рис. 4, б представляет сравнение "своих" шаблонов (принадлежащих одной персоне), но нормализация искажена ошибками локализации границ.

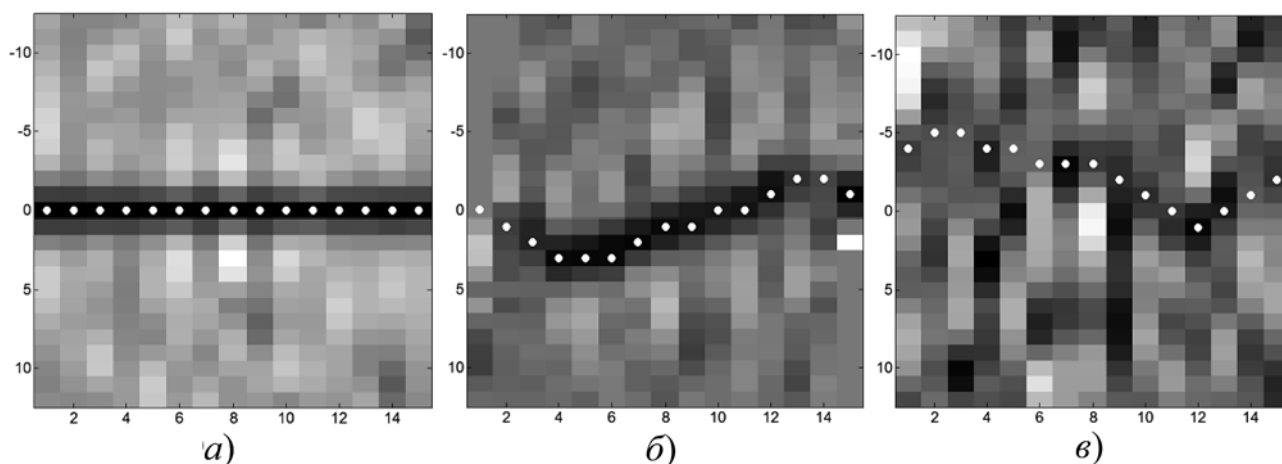


Рис. 4. Матрицы  $\{d_{\psi, n}\}$  для трех типичных случаев совмещения шаблонов

В этом случае можно видеть, что минимумы в столбцах расположены в некотором регулярном порядке, который образует темный "канал", по которому пролегает изогнутый, но гладкий оптимальный путь. Матрица на рис. 4,  $\epsilon$  — случай сравнения "чужих" шаблонов (принадлежащих разным людям). Независимо от точности локализации границ, минимумы матрицы сравнения в этом случае расположены хаотически, а оптимальный путь проходит через множество элементов с большими значениями матрицы расстояний. Итоговое значение суммарной стоимости пути также велико.

### 3. Оптимизация параметров алгоритмов построения и сравнения шаблонов

Очевидно, что число секторов  $N$ , на которые разбивается шаблон при сравнении, является важным параметром, влияющим на итоговое качество распознавания. Выбор оптимального значения этого параметра осуществлен экспериментально.

Для вычислительных экспериментов были использованы две общедоступные базы изображений радужной оболочки глаза: база ICE2005 — подмножество ND-Iris-0405 [22] и CASIA4-Lamp [23]. Размер изображений — 480 точек по вертикали и 640 точек по горизонтали. База ICE2005 содержит 2593 изображения 132 персон. Изображения получены биометрической системой LG2200. Для экспериментов использовалось подмножество всех левых радужек, содержащее 1527 изображений 119 персон. Число изображений радужной оболочки на одну персону в данной базе данных очень неравномерно и варьируется от одного до 31. Число возможных своих сравнений равно 15 357, число чужих сравнений — 1 149 744. CASIA4-Lamp содержит изображения глаз 411 персон, каждый глаз представлен 10...20 изображениями. Общее число изображений — 16 213. Это позволяет получить около 160 000 своих сравнений и приблизительно 130 млн чужих сравнений. База данных получена с использованием камеры IKEMB-100.

Обозначим  $\theta \in (0; 1)$  — порог, используемый для определения решения "допуск" (согласно расчетам, сравниваемые шаблоны принадлежат одной персоне) или "отказ" (эталон принадлежит разным персонам):

$$\text{Class} = \begin{cases} \text{допуск, если } d(T_1, T_2) \leq \theta, \\ \text{отказ, если } d(T_1, T_2) > \theta. \end{cases} \quad (3.1)$$

При проведении тестов персоны, которым принадлежат сравниваемые эталоны, известны. Поэтому сравнения эталонов разделяют на два класса: "свой" (*genuine*) и "чужие" (*impostor*). Решение классификатора (3.1) может совпадать с истинным или быть ошибочным. Качество исследуемого классификатора может быть оценено по числу ошибок классификации. Показатель *EER* (*equal error rate* — равенство ошибок первого и второго рода) определяется при равенстве относительного числа ошибочных отказов и ошибочных допусков, что зависит от порога классификации  $\theta$ :

$$EER = \frac{fn(\theta)}{fn(\theta) + fp(\theta)} = \frac{fp(\theta)}{fp(\theta) + tn(\theta)}, \quad (3.2)$$

где  $fn(\theta)$ ,  $fp(\theta)$  — число событий ошибочного отказа и допуска (ошибки первого и второго рода, соответственно);  $tn(\theta)$ ,  $tp(\theta)$  — число событий истинных отказов и допусков.

Выбор оптимального числа секторов. Поскольку *EER* зависит от степени разбиения шаблона на секторы, то можно определить функцию  $EER(N)$  и искать минимум этой функции:

$$EER^* = \min_N EER(N), \quad N = \arg \min_N EER(N). \quad (3.3)$$

На рис. 5 (см. третью сторону обложки) представлена зависимость *EER* от числа секторов  $N$ ,  $N \in [1, 30]$ , для тестируемых баз данных. График, обозначенный как NoLink, получен с использованием модели несвязанных секторов (2.4), график OptPath использует модель оптимального пути (2.5). Следует заметить, что начальные точки обоих графиков со значением  $N = 1$  совпадают и соответствуют модели "твердого тела" (2.3).

Анализ графиков позволяет сделать следующие выводы. "Твердая" модель почти всегда проигрывает моделям составного шаблона. "Жидкая" модель для малого числа секторов работает лучше, чем модель оптимального пути. Однако с увеличением числа секторов быстро достигает минимума, после чего ошибка распознавания увеличивается. Модель оптимального пути насыщается медленнее, но показывает меньшее значение ошибки в минимуме. Сравнивая результаты, получаемые по базам ICE2005 и CASIA, можно заметить, что при  $N = 1$  ICE2005 имеет большее значение *EER*, но с ростом  $N$  достигаются ошибки меньшие, чем в CASIA. Данный результат можно объяснить тем, что ICE2005 содержит большую долю изображений плохого качества, но при этом

Оптимальные параметры вэйвлетов в зависимости от числа секторов для метода оптимального пути

$N$	1	2	5	10	20	30	40
$\lambda^*$	10,0	9,0	7,6	7,1	6,4	6,3	6,3
$\sigma^*$	29,0	25,6	21,1	18,7	16,4	15,8	15,4
$EER^*$	1,76	0,94	0,65	0,55	0,49	0,51	0,53

Таблица 2

$EER$  для разных методов

Метод	ICE2005	CASIA
Zhang et al. [4]	—	0,59
Liu et al. [25]	0,63	—
He et al. [26]	0,53	—
"Твердая" модель ( $N = 1$ ), $\lambda_1$ , $\sigma_1$	2,72	1,76
"Жидкая" модель ( $N^* = 5$ ), $\lambda_1$ , $\sigma_1$	0,64	0,79
Оптимальный путь ( $N^* = 20$ ), $\lambda_1$ , $\sigma_1$	0,52	0,54
Оптимальный путь ( $N^* = 20$ ), $\lambda^*$ , $\sigma^*$	<b>0,50</b>	<b>0,49</b>

изображения этой базы весьма разнообразны и сильно отличаются друг от друга. На изображениях плохого качества результаты локализации границ РОГ часто неточны и использование "твердой" модели приводит к большой ошибке распознавания, но если неточность локализации компенсирована составными моделями, отличать разнообразные изображения проще, чем в базе CASIA. Другие модели позволяют компенсировать погрешность локализации и значительно уменьшить ошибку.

*Оптимизация параметров вэйвлетов.* До сих пор представленные рассуждения предполагали использование при распознавании некоторого определенного способа создания шаблонов (1.1)—(1.4), с фиксированными параметрами фильтра (1.3). Действительно, представленные выше результаты получены с использованием параметров  $\lambda = 10$ ,  $\sigma = 29$ , которые были определены как оптимальные для "твердой" модели в работе [24]. Там же было показано, что эти параметры зависят от точности локализации границ РОГ. Модели с разбиением на сектора частично компенсируют неточность локализации, соответственно, следует ожидать, что оптимальные параметры  $\lambda$ ,  $\sigma$  для них также примут иные значения. Экспериментально решена задача оптимизации:

$$EER^* = \min_{\lambda, \sigma} EER(\lambda, \sigma),$$

$$(\lambda^*, \sigma^*) = \arg \min_{\lambda, \sigma} EER(\lambda, \sigma) \quad (3.4)$$

для фиксированных значений  $N$ , результаты приведены в табл. 1 (для базы CASIA).

Из этой таблицы видно, что и апертура фильтра, и длина волны модуляции уменьшаются при росте числа секторов, т. е. при улучшении компенсации искажений, вызванных неточной локализацией. Однако для большого числа секторов эквивалентная ошибка классификации начинает расти, что объясняется увеличением числа ложных совпадений для слишком мелких секторов, уже не содержащих достаточно индивидуальных текстурных признаков. Оптимальное число секторов остается равным  $N^* = 20$ .

*Сравнение с ранее описанными методами.* Табл. 2 показывает ошибку  $EER$  (3.2), выраженную в процентах, сравниваемых методов для двух баз данных. Здесь  $\lambda_1 = 10$ ,  $\sigma_1 = 29$  — значения, оптимальные для "твердой" модели (т. е. с единственным сектором),  $\lambda^* = 6,4$ ,  $\sigma^* = 16,4$  — наилучшие значения для метода оптимального пути.

"Твердая" модель, которая представляет собой прямой аналог схемы Даугмана [2], имеет относительно низкую точность и проигрывает многим современным подходам. Совмещение шаблонов с использованием "жидкой" модели существенно улучшает результат, но все же уступает другим методам. Разработанный метод оптимального пути демонстрирует высокое качество распознавания, по критерию  $EER$  не уступающее известным решениям. При этом данный метод довольно прост как в алгоритмическом, так и в вычислительном плане.

### Заключение

В работе исследовано влияние разбиения радужной оболочки на сектора при совмещении шаблонов. Сравнивали три модели связи секторов: "твердую", "жидкую" и оптимального пути. Численные эксперименты показали, что третий подход может значительно повысить точность распознавания и превзойти по точности конкурирующие методы сравнения шаблонов. Минимум ошибки распознавания достигается при пяти секторах для "жидкой" модели совмещения и при 20 секторах для метода "оптимального пути". Дополнительная оптимизация параметров фильтров, выделяющих признаки, позволила улучшить результаты. Время работы алгоритма с  $N = 20$  составляет 150 мкс

для одного сравнения с процессором Intel Core i7-3770 и может быть кратно ускорено при использовании многопроцессорных систем.

#### Список литературы

1. Wildes R. Iris recognition: an emerging biometric technology // Proc. IEEE. 1997. Vol. 85, N. 9. P. 1348–1363.
2. Daugman J. High confidence visual recognition of persons by a test of statistical independence // IEEE Trans. PAMI. 1993. Vol. 15, N. 11. P. 1148–1161.
3. Павельева Е. А., Крылов А. С. Алгоритм сравнения изображений радужной оболочки глаза на основе ключевых точек // Информатика и ее применения. 2011. Т. 5, № 1. С. 68–72.
4. Zhang M., Sun Z., Tan T. Deformable DAISY Matcher for robust iris recognition // Proc. 18th IEEE Int. Conf. Image Processing. 2011. P. 3189–3192.
5. Sun Z., Tan T. Ordinal Measures for Iris Recognition // IEEE Trans. PAMI. 2009. Vol. 31, N. 12. P. 2211–2226.
6. Dorairaj V., Schmid N., Fahmy G. Performance evaluation of iris-based recognition system implementing PCA and ICA encoding techniques // Proc. SPIE. 2005. Vol. 5779. P. 51–58.
7. Erbilek M., Toygar O. Recognizing partially occluded irises using subpattern-based approaches // Proc. 24th Int. Symp. Computer and Information Sciences. 2009. P. 606–610.
8. Ives R., Guidry A., Etter D. An efficient iris recognition algorithm using phase-based image matching // Proc. IEEE Int. Conf. Image Processing. 2005. Vol. 2. P. 49–52.
9. Miyazawa K., Ito K., Aoki T., Kobayashi K., Nakajima H. Iris recognition using histogram analysis // Proc. 38th Asilomar Conference on Signals, Systems and Computers. 2004. Vol. 1. P. 562–566.
10. Bowyer K., Hollingsworth K., Flynn P. A Survey of Iris Biometrics Research: 2008–2010 // Handbook of Iris Recognition. London: Springer, 2013. P. 15–54.
11. Себряков Г. Г., Сошников В. Н., Кикин И. С., Ишутин А. А. Алгоритм распознавания объектов на оптико-электронных изображениях наземных сцен, основанный на оценке ковариационных матриц градиентных функций поля яркости // Вестник компьютерных и информационных технологий. 2013. N 7. С. 14–19.
12. Визильтер Ю. В., Горбачевич В. С., Желтов С. Ю., Рубис А. Ю., Воротников А. В. Морфлеты: новый класс древовидных морфологических описаний формы изображений // Компьютерная оптика. 2015. Т. 39, № 1. С. 101–108.
13. Kerekes R., Balakrishnan N., Thornton J., Savvides M., Kumar B. Graphical Model Approach to Iris Matching Under Deformation and Occlusion // Proc. IEEE Conf. Computer Vision and Pattern Recognition. 2007. P. 1–6.
14. Li X. Modeling Intra-class Variation for Nonideal Iris Recognition // Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005. P. 419–427.
15. Phang S., Boles W., Collins M. Tracking iris surface deformation using Elastic Graph Matching // Proc. 21st Int. Conf. Image and Vision Computing. 2006. P. 3–8.
16. Thainimit S., Alexandre L., de Almeida V. Iris surface deformation and normalization // Proc. 13th Int. Symp. Communications and Information Technologies. 2013. P. 501–506.
17. Songjang T., Thainimit S. Tracking and modeling human iris surface deformation // Proc. 12th Int. Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. 2015. P. 1–5.
18. Ганькин К. А., Гнеушев А. Н., Матвеев И. А. Сегментация изображения радужки глаза, основанная на приближенных методах с последующими уточнениями // Изв. РАН. ТиСУ. 2014. № 2. С. 80–94.
19. Daugman J. How iris recognition works // Proc. Int. Conf. Image Processing. 2002. Vol. 1. P. 33–36.
20. Селин П. С., Цурков В. И. Метод характеристических функций для классов сетей с фиксированными степенями узлов // Изв. РАН. ТиСУ. 2014. № 5. С. 28–37.
21. Есенков А. С., Кузовлев Д. И., Леонов В. Ю., Тизик А. П., Цурков В. И. Оптимизация сети и задачи с зацепляющимися переменными // Изв. РАН. ТиСУ. 2014. № 3. С. 71–87.
22. Phillips P., Scruggs W., O'Toole A., Flynn P., Bowyer K., Schott C., Sharpe M. FRVT 2006 and ICE 2006 Large-Scale Experimental Results // IEEE Trans. PAMI. 2010. Vol. 32, N. 5. P. 831–846.
23. Institute of Automation, Chinese Academy of Sciences. CASIA iris image database. 2010. URL: <http://biometrics.ideal-test.org/dbDetailForUser.do&id=4>.
24. Matveev I. A., Novik V. P. Dependency of optimal parameters of the iris template on image quality and border detection error // The Int. Archives Photogrammetry, Remote Sensing and Spatial Information Sciences. 2017. Vol. XLII-2/W4. P. 251–255.
25. Liu J. A Novel Image Deblurring Method to Improve Iris Recognition Accuracy // Proc. IEEE International Joint Conf. Biometrics. 2011. P. 1–8.
26. He Z., Tan T., Sun Z., Qiu X. Toward Accurate and Fast Iris Segmentation for Iris Biometrics // IEEE Trans. PAMI. 2009. Vol. 31, N. 9. P. 1670–1684.

V. P. Novik, Engineer-Researcher, Iritech Inc., Moscow, e-mail: novikvp@mail.ru,  
I. A. Matveev, D. Sc., Principal Researcher, e-mail: matveev@ccas.ru, V. Yu. Leonov, PhD, Researcher,  
Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow

## Matching of Informative Features of Iris Images with the Optimal Path Method

*The majority of iris recognition methods use local textural features. The accuracy of iris recognition depends significantly on the precision of local textural features alignment. The article proposes a simple and effective way to enhance the alignment accuracy. Also the problem of choosing the optimal parameters of wavelets forming the iris template is examined. It is shown that improvement of alignment accuracy allows using more informative high-frequency wavelets. Computational experiments were carried out with ICE2005 and CASIA bases from public domain.*

**Keywords:** iris recognition, biometric features, optimal path

DOI: 10.17587/it.24.681-688

## References

1. **Wildes R.** Iris recognition: an emerging biometric technology, *Proc. IEEE*, 1997, vol. 85, no. 9, pp. 1348–1363.
2. **Daugman J.** High confidence visual recognition of persons by a test of statistical independence, *IEEE Trans. PAMI*, 1993, vol. 15, no. 11, pp. 1148–1161.
3. **Pavel'eva E. A., Krylov A. S.** Algoritm sravneniya izobrazhenij raduzhnoj obolochki glaza na osnove klyuchevyh tochek (Algorithm for comparing images of the iris of the eye based on key points), *Informatika i eyo primeneniya*, 2011, vol. 5, no. 1, pp. 68–72 (in Russian).
4. **Zhang M., Sun Z., Tan T.** Deformable DAISY Matcher for robust iris recognition, *Proc. 18th IEEE Int. Conf. Image Processing*, 2011, pp. 3189–3192.
5. **Sun Z., Tan T.** Ordinal Measures for Iris Recognition, *IEEE Trans. PAMI*, 2009, vol. 31, no. 12, pp. 2211–2226.
6. **Dorairaj V., Schmid N., Fahmy G.** Performance evaluation of iris-based recognition system implementing PCA and ICA encoding techniques, *Proc. SPIE*, 2005, vol. 5779, pp. 51–58.
7. **Erbilek M., Toygar O.** Recognizing partially occluded irises using subpattern-based approaches, *Proc. 24th Int. Symp. Computer and Information Sciences*, 2009, pp. 606–610.
8. **Ives R., Guidry A., Etter D.** An efficient iris recognition algorithm using phase-based image matching, *Proc. IEEE Int. Conf. Image Processing*, 2005, vol. 2, pp. 49–52.
9. **Miyazawa K., Ito K., Aoki T., Kobayashi K., Nakajima H.** Iris recognition using histogram analysis, *Proc. 38th Asilomar Conference on Signals, Systems and Computers*, 2004, vol. 1, pp. 562–566.
10. **Bowyer K., Hollingsworth K., Flynn P.** A Survey of Iris Biometrics Research: 2008–2010, *Handbook of Iris Recognition*, London, Springer, 2013, pp. 15–54.
11. **Sebryakov G. G., Soshnikov V. N., Kikin I. S., Ishutin A. A.** Algoritm raspoznavaniya ob'ektov na optiko-jelektronnyh izobrazhenijah nazemnyh scen, osnovannyj na ocenke kovariacionnyh matric gradientnyh funkcij polja jarkosti (Algorithm for automatic recognition of ground objects in optoelectronic images, based on evaluation of feature covariance matrices built for image gradient functions), *Vestnik komp'yuternyh i informacionnyh tehnologij*, 2013, no. 7, pp. 14–19 (in Russian).
12. **Vizilter Yu. V., Gorbacevich V. S., Zheltov S. Yu., Rubis A. Yu., Vorotnikov A. V.** Morflety: novyj klass drevovidnyh morfologicheskikh opisanij formy izobrazhenij (Morphlets: a new class of tree-structured morphological descriptors of image shape), *Komp'yuternaja optika*, 2015, vol. 39, no. 1, pp. 101–108 (in Russian).
13. **Kerekes R., Balakrishnan N., Thornton J., Savvides M., Kumar B.** Graphical Model Approach to Iris Matching Under Deformation and Occlusion, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–6.
14. **Li X.** Modeling Intra-class Variation for Nonideal Iris Recognition, *Lecture Notes in Computer Science*. Berlin, Heidelberg, Springer, 2005, pp. 419–427.
15. **Phang S., Boles W., Collins M.** Tracking iris surface deformation using Elastic Graph Matching, *Proc. 21st Int. Conf. Image and Vision Computing*, 2006, pp. 3–8.
16. **Thainimit S., Alexandre L., de Almeida V.** Iris surface deformation and normalization, *Proc. 13th Int. Symp. Communications and Information Technologies*, 2013, pp. 501–506.
17. **Songjang T., Thainimit S.** Tracking and modeling human iris surface deformation, *Proc. 12th Int. Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2015, pp. 1–5.
18. **Gankin K. A., Gneushev A. N., Matveev I. A.** Segmentacija izobrazhenija raduzhki glaza, osnovannaja na priblizhennyh metodah s posledujushhimi utochnenijami (Iris image segmentation based on approximate methods with subsequent refinements), *Izv. RAN. TiSU*, 2014, vol. 53, no. 2, pp. 224–238 (in Russian).
19. **Daugman J.** How iris recognition works, *Proc. Int. Conf. Image Processing*, 2002, vol. 1, pp. 33–36.
20. **Selin P. S., Tsurkov V. I.** Metod harakteristicheskikh funkcij dlja klassov setej s fiksirovannymi stepenjami uzlov (Method of Characteristic Functions for Classes of Networks with Fixed Node Degrees), *Izv. RAN. TiSU*, 2014, vol. 53, no. 5, pp. 645–655 (in Russian).
21. **Esenkov A. S., Kuzovlev D. I., Leonov V. Yu., Tizik A. P., Tsurkov V. I.** Optimizacija seti i zadachi s zacepljajushhimisja peremennymi (Network Optimization and Problems with Coupling Variables), *Izv. RAN. TiSU*, 2014, vol. 53, no. 3, pp. 369–383 (in Russian).
22. **Phillips P., Scruggs W., O'Toole A., Flynn P., Bowyer K., Schott C., Sharpe M.** FRVT 2006 and ICE 2006 Large-Scale Experimental Results, *IEEE Trans. PAMI*, 2010, vol. 32, no. 5, pp. 831–846.
23. **Institute of Automation**, Chinese Academy of Sciences. CASIA iris image database. 2010, available at: <http://biometrics.idealtest.org/dbDetailForUser.do?id=4>
24. **Matveev I. A., Novik V. P.** Dependency of optimal parameters of the iris template on image quality and border detection error, *The Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017, vol. XLII-2/W4, pp. 251–255.
25. **Liu J.** A Novel Image Deblurring Method to Improve Iris Recognition Accuracy, *Proc. IEEE International Joint Conf. Biometrics*, 2011, pp. 1–8.
26. **He Z., Tan T., Sun Z., Qiu X.** Toward Accurate and Fast Iris Segmentation for Iris Biometrics, *IEEE Trans. PAMI*, 2009, vol. 31, no. 9, pp. 1670–1684.

---

---

### Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: [it@novtex.ru](mailto:it@novtex.ru)

Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 02.08.2018. Подписано в печать 24.09.2018. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT1018. 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 "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.

---