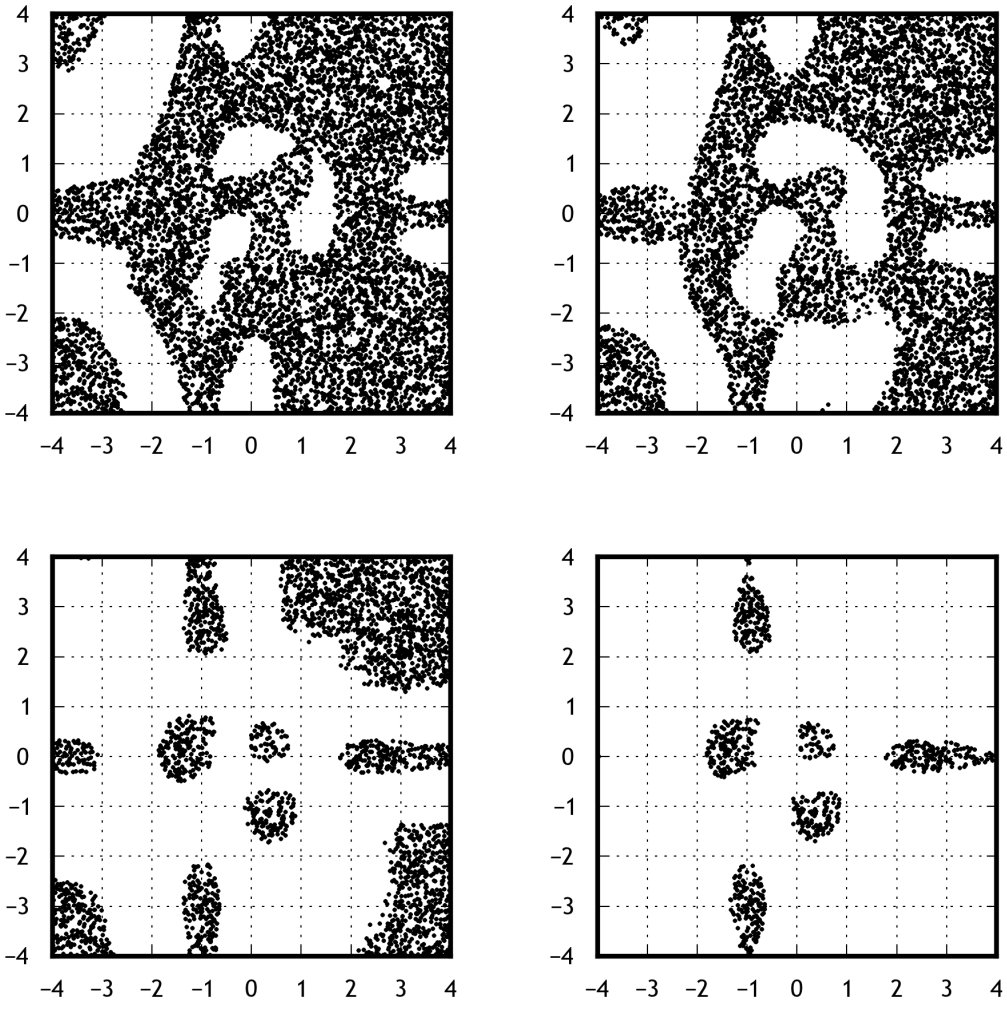
ISSN 2223-3792

Машинное обучение и анализ данных

2014 год **Том 1, номер 10**



Машинное обучение и анализ данных

Journal of Machine Learning and Data Analysis

Журнал «Машинное обучение и анализ данных» публикует новые теоретические и обзорные статьи с результатами научных исследований в области теоретических основ информатики и её приложений. Цель журнала — развитие теории машинного обучения, интеллектуального анализа данных и методов проведения вычислительных экспериментов. Принимаются статьи на английском и русском языках.

Журнал включен в российский индекс научного цитирования РИНЦ. Информация о цитировании статей находится на сайте Российского индекса научного цитирования. ISSN 2223-3792, номер свидетельства о регистрации ЭЛ № ФС 77-55486.

- Архив журнала http://www.ccas.ru/jmlda/
- Новостной сайт http://jmlda.org/
- Электронная система подачи статей http://jmlda.org/papers/

Тематика журнала:
— классификация, кластеризация, регрессионный анализ,
— алгебраический подход к проблеме синтеза корректных алгоритмов,
— многомерный статистический анализ,
— выбор моделей и сложность,
— предсказательное моделирование,
— статистическая теория обучения,
— методы прогнозирования временных рядов,
— методы обработки и распознавания сигналов,
— методы оптимизации в задачах машинного обучения и анализа данных,
— методы визуализации данных,
— обработка и распознавание речи и изображений,
— анализ и понимание текста,
— информационный поиск,
— прикладные задачи анализа данных.

Вычислительный центр Российской академии наук
Московский физико-технический институт
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»

Москва, 2014

Содержание

Оценка состояния геомагнитного поля на основе совмещения вейвлет-преобразования с радиальными нейронными сетями*

***О. В. Мандрикова*[1,2], *Е. А. Жижикина*[1,2]**
oksanam1@mail.ru; ekaterinazh1@mail.ru
[1]Институт космофизических исследований и распространения радиоволн ДВО РАН, с. Паратунка, Камчатский край, Российская Федерация; [2]Камчатский государственный технический университет, г. Петропавловск-Камчатский, Российская Федерация

Предложен метод оценки степени возмущенности геомагнитного поля, основанный на совместном применении кратномасштабного вейвлет-преобразования с радиальными нейронными сетями. Определены разномасштабные составляющие регистрируемых данных геомагнитного поля, характеризующие степень его возмущенности, и изучена их структура. Предложен способ формирования радиального слоя нейронной сети, позволяющий существенно уменьшить количество используемых примеров и повысить качество решения задачи классификации геомагнитных данных. Апробация метода выполнялась на данных станции «Паратунка», Камчатский край (регистрацию данных выполняет ИКИР ДВО РАН).

Ключевые слова: *геомагнитные данные; нейронные сети; вейвлет-преобразование; классификация данных; магнитное поле Земли*

Estimation of degree of the geomagnetic field disturbance based on the combined use of wavelet transform with radial neural networks*

***O. V. Mandrikova*[1,2], *E. A. Zhizhikina*[1,2]**
[1]Institute of Cosmophysical Research and Radio Wave Propagation Far Eastern Branch of the Russian Academy of Sciences, Paratunka, Kamchatka Region, Russia; [2]Kamchatka State Technical University, Petropavlovsk-Kamchatsky, Russia

The present paper is focused on the development of theoretical tools and software for the analysis of the geomagnetic field parameters and for the estimation of the geomagnetic field condition using modern methods of pattern recognition and digital signal processing. Existing methods for the geomagnetic data analysis do not allow to identify some regularities in the data and lead to the loss of important information.

A method based on the combined use of the wavelet transform and radial neural networks has been proposed. This method allows to study subtle structural features of the geomagnetic data and to extract informative components which characterize the disturbance degree of the geomagnetic field.

In the present paper, geomagnetic data structure was studied in detail, the signs of the geomagnetic activity increasing were defined and classes for the radial layer of the neural network were offered. Furthermore, a way of forming a radial layer was proposed. This way allows to significantly reduce the number of examples and to improve the quality of the

*Работа выполнена при финансовой поддержке РНФ, проект № 14-11-00194 и ФСР МФП НТС (программа «УМНИК»), дог. 0006065.

geomagnetic data classification. On the basis of combination of decisions of the developed neural networks, a decision rule to estimate the geomagnetic field condition in the automatic mode has been suggested.

The method has been successfully tested on the geomagnetic data that were kindly provided to the authors by the Institute of Cosmophysical Research and Radio Wave Propagation (Paratunka, Kamchatka Region, Russia). Using the proposed method in combination with other methods and approaches allows to enhance the quality of geomagnetic data automatic processing during space weather forecast.

Keywords: *geomagnetic data; neural networks; wavelet transform; data classification; Earth's magnetic field*

Введение

Работа направлена на создание теоретических и программных средств анализа параметров геомагнитного поля и оценки его состояния по данным наземных обсерваторий с применением современных методов распознавания образов и цифровой обработки сигналов. Регистрируемые геомагнитные данные имеют сложную структуру, подвержены влиянию внешних факторов различной физической природы, что значительно усложняет процесс их изучения. Они содержат разномасштабные локальные особенности, имеющие различную форму и несущие основную информацию о состоянии поля [1, 2]. Существующие средства обработки и анализа геомагнитных данных имеют следующие недостатки:

1. Недостаточная степень автоматизации и существенные погрешности в работе систем [1, 3, 4].
2. Существующие методы анализа геомагнитных данных не позволяют выявлять некоторые закономерности в данных и приводят к потере важной информации [4, 5].

Для исследований предлагается метод, основанный на совместном применении вейвлет-преобразования [6, 7] с радиальными нейронными сетями [8]. Применение вейвлет-преобразования позволяет изучать тонкие особенности структуры геомагнитных данных и выделять информативные составляющие [1, 2]. В основе радиальных нейронных сетей лежит непараметрический байесовский классификатор, позволяющий выделять в вариациях поля классификационные признаки, характеризующие степень его возмущенности [8].

В работе на основе кратномасштабного вейвлет-преобразования детально изучена структура геомагнитных данных (на примере горизонтальной компоненты геомагнитного поля), выделены признаки, характеризующие степень возмущенности поля и предложены классы для радиального слоя нейронной сети. На основе построения образов классов предложен способ формирования радиального слоя, позволяющий существенно уменьшить количество используемых примеров и повысить качество решения задачи классификации геомагнитных данных. Для различных компонент вариаций геомагнитного поля построены нейронные сети, выполняющие оценку их степени возмущенности. На основе комбинации решений нейронных сетей предложено решающее правило по оценке состояния геомагнитного поля.

Апробация метода, выполненная на данных станции «Паратунка» (Камчатский край), подтвердила его эффективность. Использование метода в комплексе с другими методами и подходами позволяет повысить качество результатов автоматической обработки геомагнитных данных при проведении прогноза комической погоды.

Описание метода

Кратномасштабное вейвлет-преобразование данных и выделение компонент. В качестве базового пространства регистрируемых дискретных данных $f_0(t)$ рассмотрим замкнутое пространство с разрешением $j = 0$:

$$V_0 = clos_{L^2(R)}(2^0\varphi(2^0t - k)) : k \in Z),$$

порожденное скэйлинг-функцией $\varphi \in L^2(R)$ [7, 6]. На основе кратномасштабного вейвлет-преобразования до уровня m можно получить представление данных в виде [7, 6]:

$$f_0(t) = \sum_{j=-1}^{-m} g\left[2^j t\right] + f\left[2^{-m}t\right], \text{ где } g\left[2^j t\right] \in W_j,\ f\left[2^{-m}t\right] \in V_{-m}, \tag{1}$$

W_j - пространство с разрешением j, порожденное вейвлет-базисом $\Psi_{j,n}(t) = 2^{j/2}\Psi(2^j t - n)$; компоненты $g\left[2^j t\right] = \sum\limits_n d_{j,n}\Psi_{j,n}(t)$, где $d_{j,n} = \langle f, \Psi_{j,n}\rangle$, являются детализирующими, характеризуют локальные свойства данных; компонента $f\left[2^{-m}t\right] = \sum\limits_k c_{-m,k}\varphi_{-m,k}(t)$, $c_{-m,k} = \langle f, \varphi_{-m,k}\rangle$ является аппроксимирующей.

В работе [2] показано, что коэффициенты детализирующих компонент вейвлет-преобразования $d_{j,n}$ (см. соотношение (1)) характеризуют степень возмущенности геомагнитного поля и в периоды возрастания возмущений существенно увеличиваются их абсолютные значения (рис. 1). Поэтому за меру геомагнитной возмущенности коэффициента логично принять его абсолютное значение.

Рассмотрим три возможных состояния вариации поля:

(1) «спокойное» состояние (1-й класс);
(2) «слабовозмущенное» состояние (2-й класс);
(3) «возмущенное» состояние (3-й класс).

В соответствии с введенными состояниями рассмотрим функцию

$$Z_{j,n}^M(t) = \left|d_{j,n}^M(t)\right|,$$

где $M = 1, 2, 3$, $z_{j,n}^1$ — абсолютные значения коэффициентов компонент разрешения j «спокойных» вариаций поля, $z_{j,n}^2$ — абсолютные значения коэффициентов компонент разрешения j «слабовозмущенных» вариаций поля, $z_{j,n}^3$ — абсолютные значения коэффициентов компонент разрешения j «возмущенных» вариаций поля.

Анализ распределений функций $Z_{j,n}^M$, $M = 1, 2, 3$, представленных для разрешения $j = -6$ на рис. 2, показывает, что их диапазоны значений имеют значительные наложения, что существенно затрудняет задачу разделения образов. Следуя результатам работы [2], в качестве *меры возмущенности компоненты* разрешения j определим величину

$$S_j = \max_n |d_{j,n}|.$$

Значения величин S_{-4} и S_{-6} для вариаций поля с различными состояниями представлены на рис. 3, 4. Анализ рис. 3 и 4 показывает, что диапазоны значений этих величин также имеют наложение, которое обусловлено сложным разномасштабным характером процесса, а также отсутствием четких границ между анализируемыми классами. Учитывая данные особенности исследуемого процесса, введем в рассмотрение следующие подклассы.

Для 1-го класса:

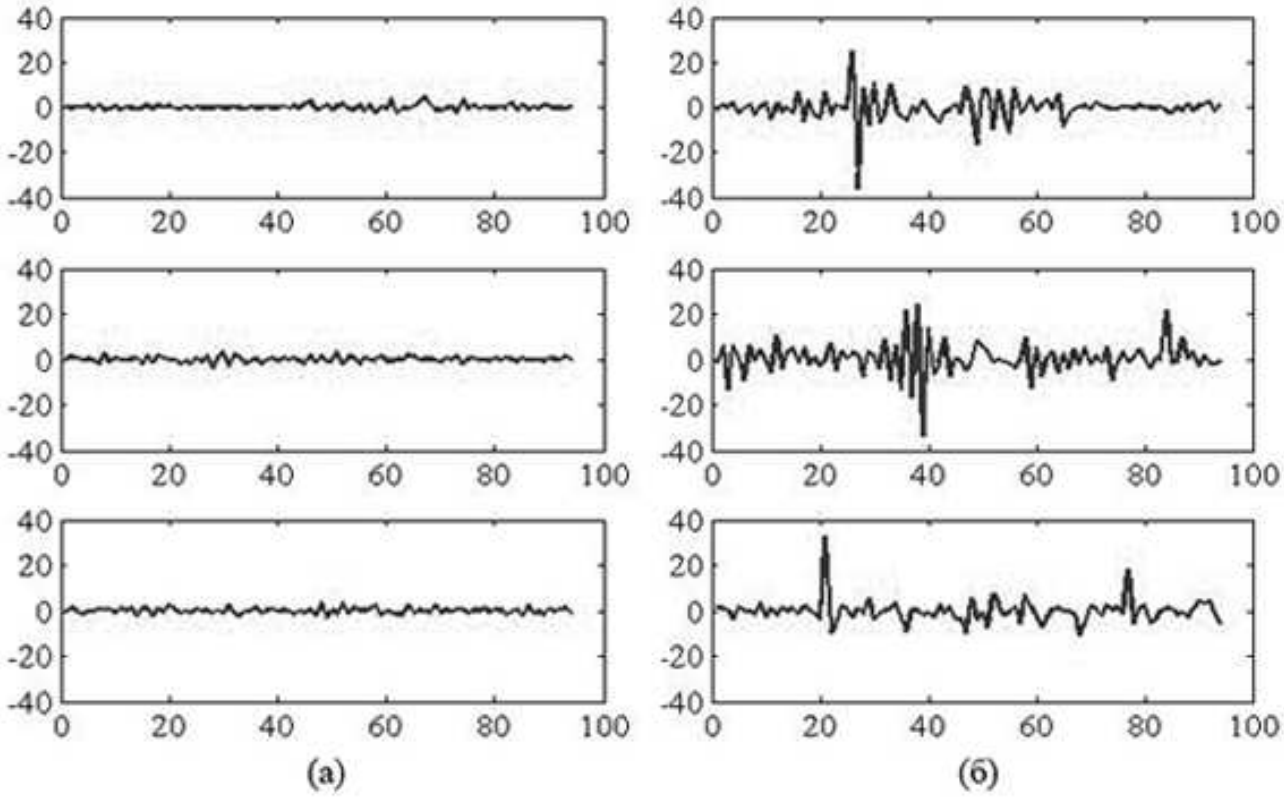


Рис. 1. Детализирующие компоненты вариаций геомагнитного поля, полученные с помощью вейвлета Добеши 3-го порядка: (а) в период спокойного геомагнитного поля, (б) в период возмущенного геомагнитного поля

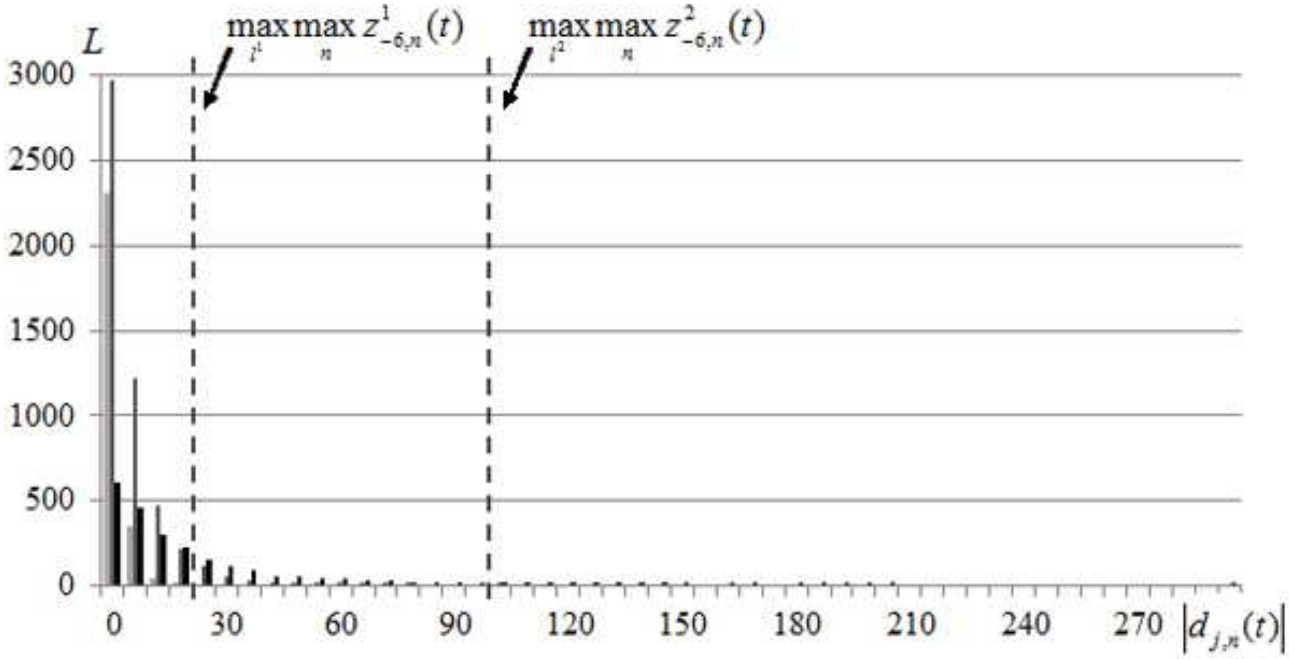


Рис. 2. Распределения функций $z^M_{-6,n}(t)$: светло-серый цвет — $M = 1$; темно-серый цвет — $M = 2$, черный цвет — $M = 3$ (l^M — индекс анализируемой вариации поля состояния M)

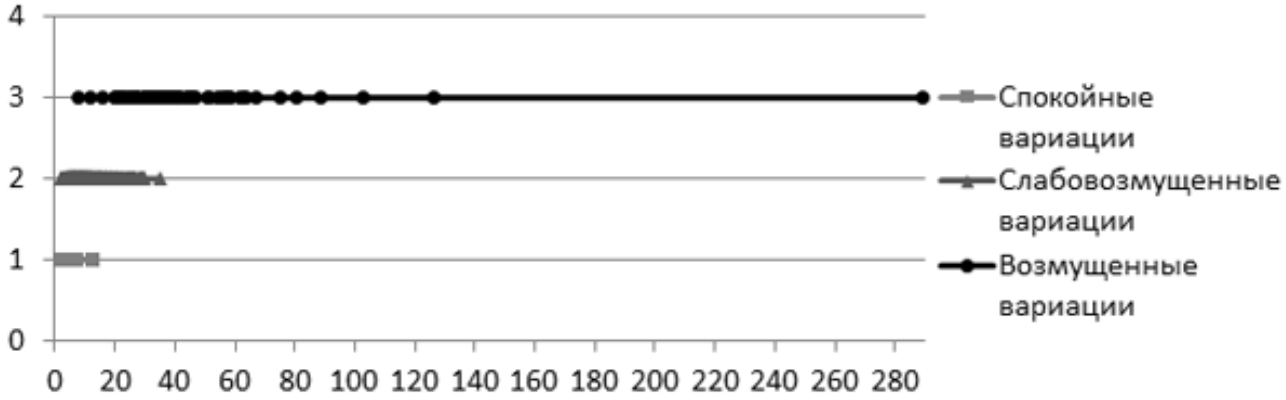


Рис. 3. Максимумы амплитуд коэффициентов детализирующих компонент разрешения $j = -4$

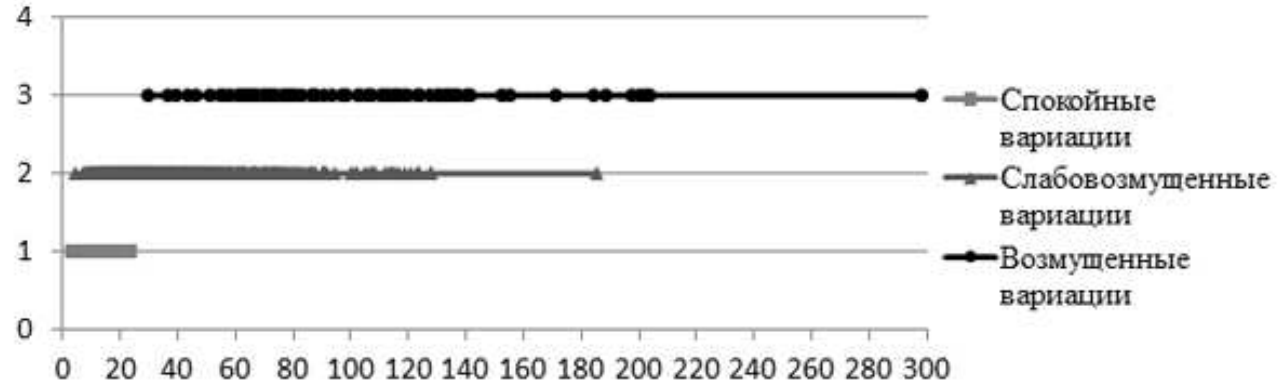


Рис. 4. Максимумы амплитуд коэффициентов детализирующих компонент разрешения $j = -6$

(1) «α-спокойные»: $\max\limits_n |d_{j,n}(t)| \leqslant T_j^{\alpha_1}$;
(2) «β-спокойные»: $\max\limits_n |d_{j,n}(t)| > T_j^{\alpha_1}$.

Для 2-го класса:

(1) «α-слабовозмущенные»: $\max\limits_n |d_{j,n}(t)| \leqslant T_j^{\alpha_2}$;
(2) «β-слабовозмущенные»: $\max\limits_n |d_{j,n}(t)| > T_j^{\alpha_2}$.

Для 3-го класса:

(1) «α-возмущенные»: $\max\limits_n |d_{j,n}(t)| \leqslant T_j^{\alpha_3}$;
(2) «β-возмущенные»: $\max\limits_n |d_{j,n}(t)| > T_j^{\alpha_3}$.

Для выбора порогов $T_j^{\alpha_i}$, $i = 1, 2, 3$ рассмотрим критерий наименьшей частоты ошибок (достигаемый путем оценки апостериорного риска [9]). В этом случае пороги определяются путем минимизации апостериорного риска [9].

Построение радиального слоя нейронной сети. Радиальные нейронные сети [8] имеют три слоя: входной слой; скрытый слой примеров (радиальный слой), содержащий признаки классов; выходной линейный слой, определяющий вероятность принадлежности входного образа к классу.

В радиальном слое выполняется следующее преобразование входных сигналов:

1. Оценка состояния нейронов на основе функции взвешивания $r = ||p - w||b$, где p — вектор входа; w — вектор весов; b — смещение.

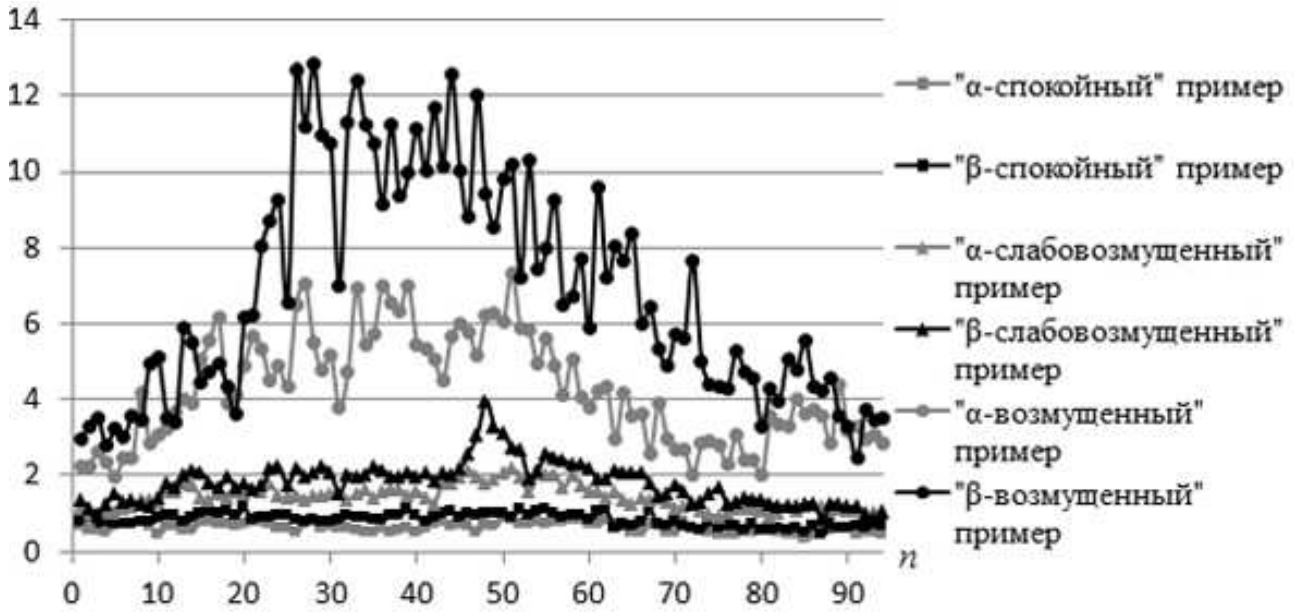


Рис. 5. Примеры-образы подклассов радиальной нейронной сети для $j = -4$

2. Используя функцию активации e^{-r^2}, оценка меры близости входного сигнала и примера.

Когда расстояние r между вектором примера w и входным сигналом p уменьшается, выход радиальной базисной функции приближается к значению «1», в противном случае — к значению «0».

Поскольку абсолютные значения коэффициентов детализирующих компонент $d_{j,n}$ характеризуют степень возмущенности геомагнитного поля, они могут быть определены в качестве признаков классов радиальной нейронной сети, выполняющей разделение образов на «спокойные», «слабовозмущенные» и «возмущенные».

С целью оптимизации структуры нейронной сети при формировании радиального слоя для каждого введенного подкласса k_i построим его *пример-образ*:

$$P_j^{k_i}(t) = \frac{\sum\limits_{u=1}^{U} D_{j,u}^{k_i}(t)}{U} \tag{2}$$

где $D_{j,u}^{k_i} = \left\{ \left| d_{j,n}^{k_i,u} \right| \right\}_{n \in Z}$; u — номер компоненты подкласса k_i; U — количество компонент подкласса k_i.

Выполнение данной процедуры позволит существенно уменьшить количество используемых примеров, в отличие от традиционного подхода, используемого в радиальных нейронных сетях [8]. В этом случае для каждого подкласса в радиальном слое сети будет создан только один нейрон (с весами примера-образа подкласса). Полученные примеры подклассов $P_j^{k_i}$ для разрешений $j = -4$ и $j = -6$ показаны на рис. 5 и 6.

Учитывая разномасштабность исследуемого процесса, для каждой детализирующей компоненты будем создавать нейронную сеть, выполняющую оценку ее состояния. При выполнении процедуры (1), следуя работе [10], ограничимся уровнем разложения $m = 6$. Для оценки состояния геомагнитного поля будем использовать следующее *решающее правило*:

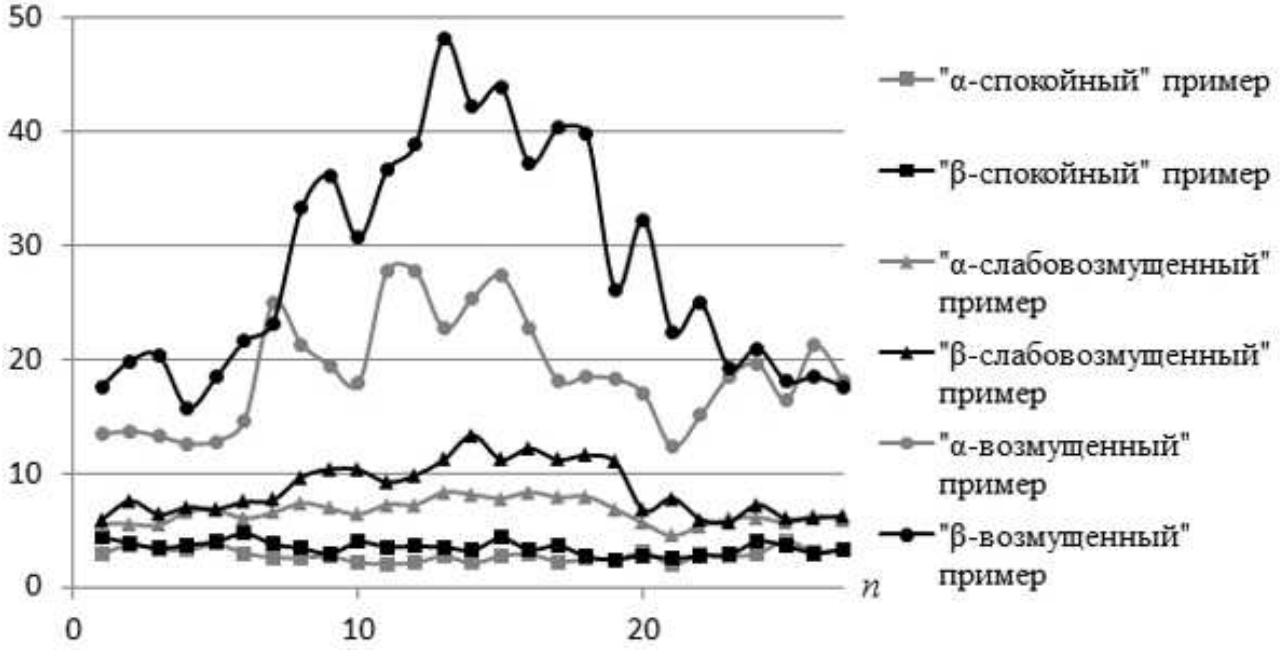


Рис. 6. Примеры-образы подклассов радиальной нейронной сети для $j = -6$

(1) если все компоненты имеют «спокойное» состояние, либо только одна компонента является «слабовозмущенной», то будем считать, что вариация является «спокойной» (соответствует спокойному состоянию геомагнитного поля);
(2) если хотя бы одна из компонент вейвлет-преобразования имеет «возмущенное» состояние, то будем считать, что вариация является «возмущенной» (соответствует возмущенному состоянию геомагнитного поля);
(3) в остальных случаях будем считать, что вариация имеет «слабовозмущенное» состояние (соответствует слабовозмущенному состоянию геомагнитного поля).

Результаты экспериментов

В процессе исследований было проанализировано 100 «спокойных», 190 «слабовозмущенных» и 86 «возмущенных» вариаций геомагнитного поля станции «Паратунка» (Камчатский край, с. Паратунка) за 2002, 2005 и 2008 гг.

При создании примеров-образов «спокойными» считались вариации, у которых суточный суммарный индекс геомагнитной активности K (K-индекс) не превышал значения 10. Слабовозмущенными считались вариации, у которых суточный суммарный K-индекс имел значения в диапазоне от 11 до 18. Возмущенными считались вариации, суточный суммарный индекс K-индекс которых превышал значение 18.

Для оценки эффективности предлагаемого метода обработка данных выполнялась также методом на основе традиционной архитектуры радиальной нейронной сети, на вход которой подавались исходные вариации геомагнитного поля (без применения вейвлет-преобразования).

Примеры-образы радиального слоя такой сети, в соответствии с процедурой (2), создавались следующим образом:

$$P^{k_i}(t) = \frac{\sum\limits_{u=1}^{U} f_{0,u}^{k_i}(t)}{U},$$

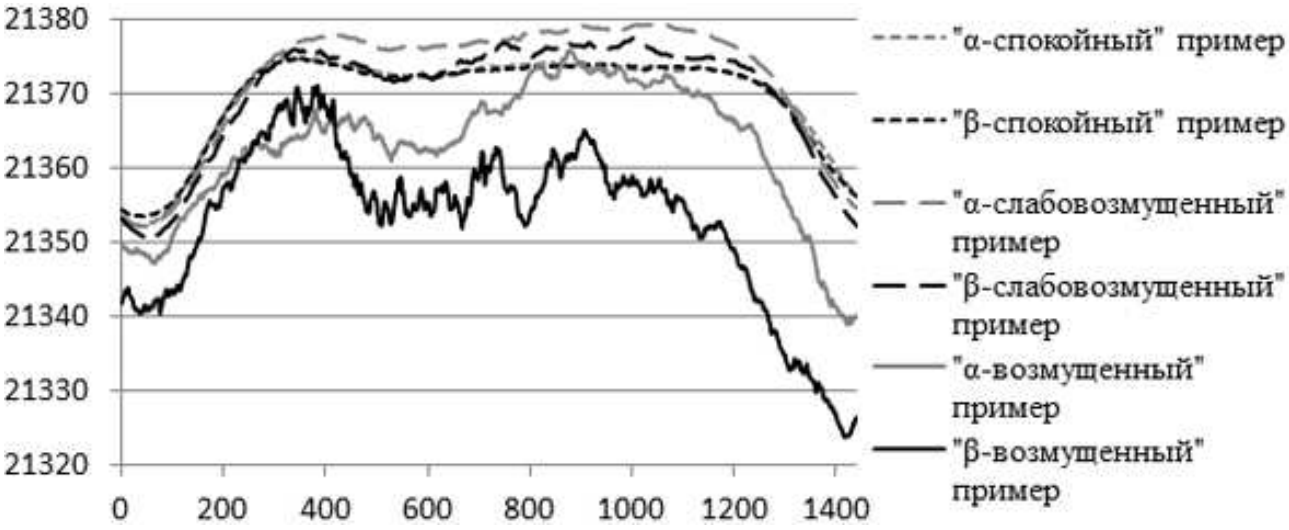


Рис. 7. Примеры-образы подклассов радиального слоя нейронной сети (построена без применения вейвлет-преобразования)

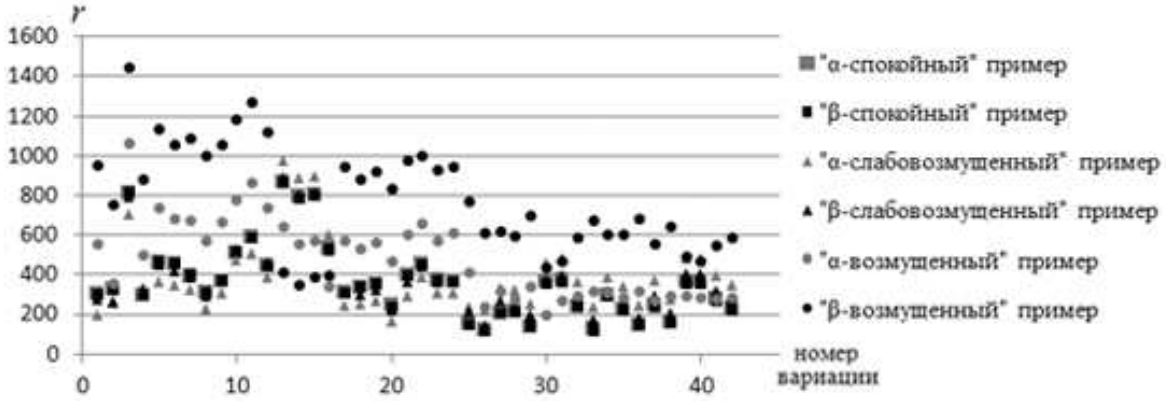


Рис. 8. Состояния нейронов радиального слоя традиционной нейронной сети при подаче на ее вход «α-спокойных» вариаций (без применения вейвлет-преобразования)

где $f_{0,u}^{k_i}(t)$ — исходная вариация подкласса k_i; u — номер вариации подкласса k_i; U – – количество вариаций подкласса k_i. Полученные таким образом примеры подклассов показаны на рис. 7.

На рис. 8 и 9, в качестве примера, показаны состояния нейронов радиального слоя созданных нейронных сетей при подаче на их вход «α-спокойных» вариаций. Анализ рис. 8 и 9 показывает, что применение вейвлет-преобразования позволяет существенно повысить достоверность классификации данных.

В табл. 1 представлены результаты погрешности оценки состояния геомагнитного поля предлагаемым методом и на основе радиальной сети, построенной традиционным способом. Результаты табл. 1 подтверждают эффективность предлагаемого метода и показывают, что на его основе погрешность решения задачи значительно меньше, чем в случае применения традиционной архитектуры радиальной сети.

Заключение

В работе описан автоматический метод оценки состояния геомагнитного поля, основанный на совмещении вейвлет-преобразования с радиальными нейронными сетями, и вы-

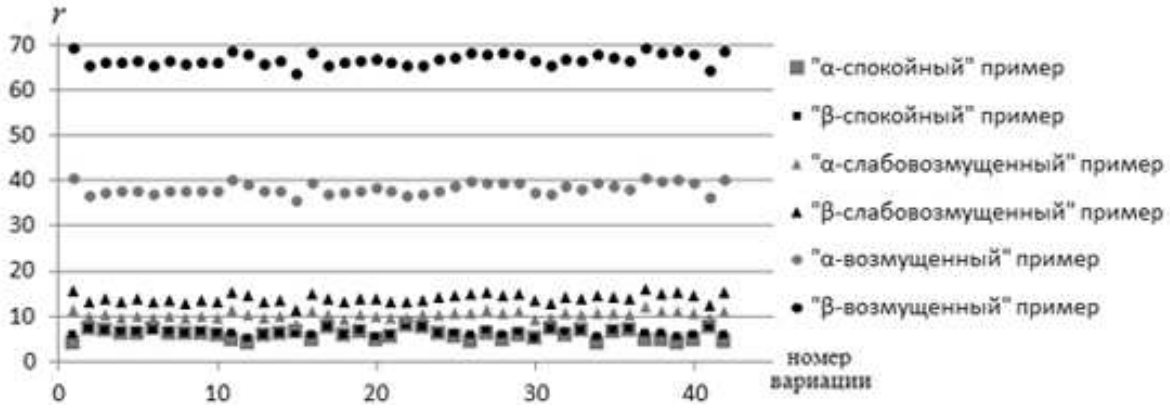


Рис. 9. Состояния нейронов радиального слоя нейронной сети компоненты разрешения $j = -4$ при подаче на ее вход «α-спокойных» компонент вариаций

Таблица 1. Погрешность оценки состояния геомагнитного поля (%)

Подкласс	Традиционная архитектура сети	Предлагаемый метод						
		$j = -1$	$j = -2$	$j = -3$	$j = -4$	$j = -5$	$j = -6$	Решающее правило
α-«спокойный»	64,29	7,14	0	0	0	0	0	0
β-«спокойный»	81,03	13,79	15,52	6,90	1,72	3,45	3,45	13,79
α-«слабовозмущеный»	26	34	32	17	15	16	17	1
β-«слабовозмущеный»	28,89	25,56	14,44	1,11	4,44	2,22	14,44	11,11
α-«возмущеный»	43,18	20	18	14	11	16	9	0
β-«возмущеный»	23,81	19,05	9,52	4,76	7,14	0	7,14	0

полнена оценка его эффективности. Анализ полученных результатов показал, что предлагаемый метод позволяет значительно повысить эффективность автоматической обработки геомагнитных данных в задачах выделения возмущений и оценки состояния магнитного поля Земли. В экспериментах использовались вариации геомагнитного поля, полученные на станции «Паратунка», Камчатский край (регистрацию данных выполняет ИКИР ДВО РАН).

Литература

[1] *Mandrikova O. V., Smirnov S. E., Solov'ev I.S.* Method for determining the geomagnetic activity index based on wavelet packets // *Geomagnetism and Aeronomy*, 2012. Vol. 52, no. 1. P. 111–120.

[2] *Mandrikova O. V., Solovyev I. S., Geppener V. V., Klionskiy D. M., Al-Kasasbeh R. T.* Analysis of the Earth's magnetic field variations on the basis of a wavelet-based approach // *Digital Signal Processing*, 2013. No. 23. P. 329–339.

[3] *Афраймович Э. Л., Перевалова Н. П.* GPS-мониторинг верхней атмосферы Земли. Иркутск: ГУ НУ РВХ ВСНЦ СО РАМН, 2006. 480 с.

[4] *Nowozynski K.* Calculate geomagnetic activity K indices using the Adaptative Smoothing method, 2007. URL: `http://www.intermagnet.org/Software_e.html`.

[5] *Будько Н. И., Зайцев А. Н., Карпачев А. Т., Козлов А. Н., Филиппов Б. П.* Космическая среда вокруг нас. Троицк: ТРОВАНТ, 2006. 232 с.

[6] *Daubechies I.* Ten Lectures on wavelets. Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001. 464 p.

[7] *Малла С.* Вэйвлеты в обработке сигналов. М.: Мир, 2005. 671 с.

[8] *Хайкин С.* Нейронные сети: полный курс. М.: Издательский дом «Вильямс», 2006. 1104 с.

[9] *Левин Б. Р.* Теоретические основы статистической радиотехники. М.: Радио и связь, 1989. 656 с.

[10] *Мандрикова О. В., Жижикина Е. А.* Оценка степени возмущенности геомагнитного поля на основе совмещения вейвлет-преобразования с радиальными нейронными сетями // *17 Междунар. конф. по мягким вычислениям и измерениям.* С.-Пб.: СПбГЭТУ «ЛЭТИ», 2014. С. 223–226.

References

[1] *Mandrikova O. V., Smirnov S. E., Solov'ev I.S.* 2012. Method for determining the geomagnetic activity index based on wavelet packets. *Geomagnetism and Aeronomy* 52(1):111–120.

[2] *Mandrikova O. V., Solovyev I. S., Geppener V. V., Klionskiy D. M., Al-Kasasbeh R. T.* 2013. Analysis of the Earth's magnetic field variations on the basis of a wavelet-based approach. *Digital Signal Processing* 23:329–339.

[3] *Afraimovich E. L., Perevalova N. P.* 2006. *GPS-monitoring of the Earth upper atmosphere.* Irkutsk: SC RRS SB RAMS. 480 p. (In Russian.)

[4] *Nowozynski K.* 2007. Calculate geomagnetic activity K indices using the Adaptative Smoothing method. Available at: `http://www.intermagnet.org/Software_e.html`.

[5] *Bud'ko N. I, Zaitsev A. N., Karpachev A., T., Kozlov A. N., Filippov B. P.* 2006. *Space Around Us.* Troitsk: TROVANT. 232 p. (In Russian.)

[6] *Daubechies I.* 2001. *Ten lectures on wavelets.* Izhevsk: NITs «Regulyarnaya i Khaoticheskaya Dinamika». 464 p.

[7] *Mallat S.* 1999. *A wavelet tour of signal processing.* San Diego, CA: Academic Press. 671 p.

[8] *Haykin S.* 1999. *Neural networks: A comprehensive foundation.* New Jersey: Prentice Hall. 842 p.

[9] *Levin B. R.* 1989. *Theoretical foundations of statistical radio engineering.* Moscow: Radio and Communications. 656 p. (In Russian.)

[10] *Mandrikova O. V., Zhizhikina E. A.* 2014. Estimation of degree of the geomagnetic field disturbance on the basis of the combined use of wavelet transform with radial neural networks. *17th Conference (International) on Soft Computing and Measurements.* St. Petersburg. 223–226.

Вычислительная технология оценки степени выпуклости многоэкстремальной функции*

А. Ю. Горнов, Т. С. Зароднюк
gornov@icc.ru, tz@icc.ru
Институт динамики систем и теории управления СО РАН, ул. Лермонтова, 134, г. Иркутск, Россия

Предложена методика определения степени выпуклости функции, основанная на ее стохастической аппроксимации на всей исследуемой области. Основной идеей подхода является поточечное исследование выпуклости функции по случайно выбранным направлениям и систематизация полученной информации с целью получения интегральной оценки выпуклости. Эффективность предложенной технологии продемонстрирована на ряде модельных примеров небольших размерностей, для которых построены и визуализированы области выпуклости функций.

Ключевые слова: *выпуклость функции; методы оптимизации; глобальный экстремум*

Computing technology for estimation of convexity degree of the multiextremal function*

A. Yu. Gornov, T. S. Zarodnyuk
Institute for System Dynamics and Control Theory of SB RAS, 134 Lermontov Str., Irkutsk, Russia

Background: Optimization problems arise in the application of mathematical modeling method. Advance in applying of mathematical modeling depends on how successfully the researcher can construct a valid model and, primarily, on the convexity or nonconvexity of the involved functions. It can be argued that the class of convex functions is mathematically well studied. However, the situation greatly changes in the case of nonconvex problems.

Methods and Results: This paper proposes a technique of determining the degree of the function convexity based on its stochastic approximation for the considerable area. The main idea of the approach is the pointwise study of the function convexity on the stochastic selected areas and systematization of this information to obtain an integrated estimate of convexity. The effectiveness of the proposed technology is demonstrated on a number of model examples of small dimensions, for which the areas of convexity are constructed and visualized.

Concluding Remarks: The selection of functional used in the mathematical modeling can be produced to choose more convenient for optimization analysis with the application of the proposed computing technology. This technique allows one to demonstrate "the areas of convexity-nonconvexity" for problems of small dimensions. The algorithm can be easily parallelized. The efficiency of the considered approach is investigated on a number of test and model problems. The obtained numerical results allow to expect for the creation of a new computational software useful in solving practical problems in various scientific and technical fields.

Keywords: *function convexity; optimization methods; global extremum*

*Работа выполнена при финансовой поддержке РФФИ, проект № 15-37-20265.

Введение

Задачи оптимизации естественно возникают при применении метода математического моделирования. При этом во многих ситуациях исследователь имеет определенную свободу в выборе модели, при помощи которой целесообразно изучать рассматриваемые явления, процессы или данные. Успех в применении математического моделирования самым серьезным образом зависит от того, насколько удачно удалось сформировать адекватную модель и, в первую очередь, от выпуклости или невыпуклости привлеченных функций.

Свойство выпуклости является, очевидно, одним из самых удобных свойств при практическом анализе и оптимизации функциональных зависимостей. Оптимизации выпуклых функций посвящена необозримая научная литература, созданы эффективные теоретические подходы, разработан и основательно исследован большой набор численных алгоритмов, реализованы программные средства, имеется большой опыт практических применений (см., например, [1–8]). Неявным образом большая часть результатов для выпуклых функций перенесена на близкий класс унимодальных функций, во многих случаях более практически значимый. Можно утверждать, что класс выпуклых и унимодальных функций неплохо освоен математически. Однако ситуация самым существенным образом меняется при рассмотрении невыпуклых задач. Как правило, априори не удается получить эффективных оценок числа локальных экстремумов, осложняется теоретический анализ задач с недифференцируемыми функциями, на несколько порядков растут вычислительные затраты на достижение глобальных решений и т. д. Несмотря на многолетние усилия многих высококвалифицированных специалистов (см., например, [6, 9–14]), на практике, для задач глобальной оптимизации существенных размерностей, решаемых недетерминированными алгоритмами, никогда нет гарантий, что полученные результаты вычислений отражают наилучший из возможных вариантов решения.

В работе предлагается простая методика определения степени выпуклости функции, основанная на ее стохастической аппроксимации на всей исследуемой области:

$$f(x) \to \min, a \leqslant x \leqslant b,$$

где $f(x) : R^n \to R$, $f(x)$ — непрерывно дифференцируема и в общем случае невыпукла, $a, b \in R^n$.

Основной идеей подхода является поточечное исследование выпуклости функции по случайно выбранным направлениям и систематизация полученной информации с целью получения интегральной оценки выпуклости. Эффективность предложенной технологии демонстрируется на ряде модельных примеров небольших размерностей, для которых построены и визуализированы области выпуклости функций.

Алгоритм поиска областей выпуклости функции

1. Задаются алгоритмические параметры: ns – число стохастических проб; np —- число направлений, по которым осуществляется оценка степени выпуклости; h — шаг для оценки степени выпуклости; C – параметр, отвечающий за учитываемый порог выпуклости функции; N^1_{vip} — число случайных направлений, по которым функция оказалась выпуклой в выбранной точке; N^2_{vip} — число точек, в которых степень выпуклости функции равна 1 (стартовое значение $N^2_{\text{vip}} = 0$).
   Для всех $k = \overline{1, \text{ns}}$
2. Выбирается значение независимой переменной $x^k_i = \widehat{x}^k_i$, где $\widehat{x}^k_i$ — случайное значение из заданного отрезка $[\overline{x}^k_i, \underline{x}^k_i]$, $N^1_{\text{vip}} = 0$, $i = \overline{1, n}$.

3. Вычисляется значение минимизируемой функции в данной точке $\widehat{f}^k = f(\widehat{x}^k)$.
   Для всех $j = \overline{1, \text{np}}$:
   (а) выбираются случайные параметры p_i^j и соответствующие точки $\overline{y}_i^j = \widehat{x}_i^k + p_i^j h$ и $\underline{y}_i^j = \widehat{x}_i^k - p_i^j h$, $i = \overline{1, n}$;
   (б) вычисляются значения функции в полученных точках $\overline{f} = f(\overline{y}^j)$ и $\underline{f} = f(\underline{y}^j)$;
   (в) выполняется оценка степени выпуклости функции в исследуемой точке $S_{\widehat{x}^k}^j = \overline{f} + \underline{f} - 2\widehat{f}^k$: если $S_{\widehat{x}^k}^j < 0$, то f по выбранному направлению является вогнутой, иначе — выпуклой ($N_{\text{vip}}^1 = N_{\text{vip}}^1 + 1$).
4. Вычисляется общая степень выпуклости функции $S_{\widehat{x}^k} = N_{\text{vip}}^1/\text{np}$ в точке $\widehat{x}_i^k$.

   (а) Если $S_{\widehat{x}^k} > C$, то запоминаем исследуемую точку.
   (б) Если $S_{\widehat{x}^k} = 1$, то функция f является выпуклой в точке $\widehat{x}_i^k$ по всем рассмотренным направлениям ($N_{\text{vip}}^2 = N_{\text{vip}}^2 + 1$).
5. Итоговая степень выпуклости функции во всей допустимой области $S_f = N_{\text{vip}}^2/\text{ns}$.
6. Выполняется графическая визуализация картины выпуклости функции.
   Алгоритм завершен.

Одним из результатов работы алгоритма является значение степени выпуклости функции S_f, позволяющее оценить ее сложность. Степень выпуклости естественным образом зависит от размеров допустимой области.

Для задач небольшой размерности информативна получаемая картина выпуклости функции, на которую напрямую влияет параметр C. При нулевом его значении вся область определения функции будет заполнена (никакие из рассматриваемых точек не будут отброшены). С увеличением этого параметра все больше точек перестают достигать требуемой степени выпуклости. При выборе крайнего значения ($C = 100\%$) отображаются только точки, в которых ни по одному случайному направлению не нарушалось свойство выпуклости исследуемой функции.

Вычислительные эксперименты

Проведено тестирование предложенного алгоритма, в результате которого выбраны значения основных алгоритмических параметров, устранены неточности его программной реализации. В статье представлены результаты исследования 10-ти тестовых задач небольшой размерности. Произведена оценка степени выпуклости S_f и построены картины выпуклости для рассматриваемых многоэкстремальных функций, соответствующие разным значениям параметра C, фиксирующего учитываемый порог выпуклости (рис. 1–10).

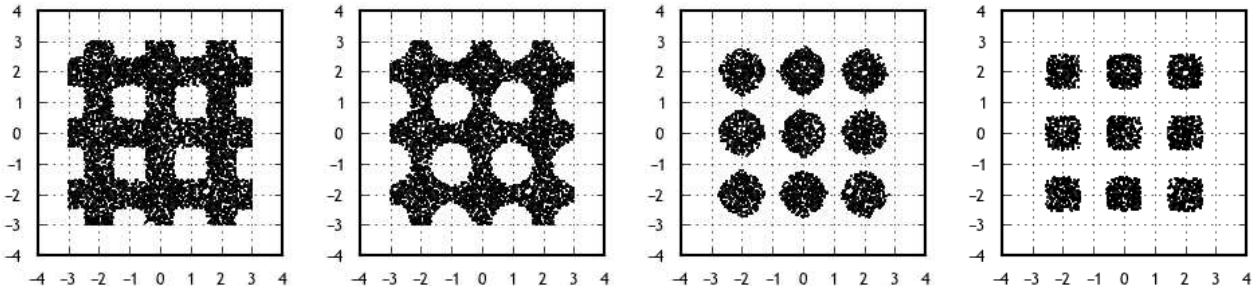
Тестовый пример 1

$f_1(x) = \sin(\pi x_1 - 0{,}5\pi) + \sin(\pi x_2 - 0{,}5\pi) + 0{,}1(x_1^2 + x_2^2) \to \min$, $x_1, x_2 \in [-3, 3]$.

Степень выпуклости данной функции S_{f_1}, полученная в результате работы алгоритма, равна 0,259.

В табл. 1 отображен фрагмент результатов проведенных вычислительных экспериментов для тестовой функции $f_1(x)$. Представлены случайные точки из допустимой области $\widehat{x}^k$, $k = \overline{1, 20}$, и значения степеней выпуклости для каждой из них $S_{\widehat{x}^k}^j$. При увеличении значения параметра C все большее число точек перестает достигать выбранного порога выпуклости функции, что естественным образом отображается на рис. 1.

Тестовый пример 2 (функция Растригина)



(а) (б) (в) (г)

Рис. 1. Области выпуклости тестовой функции 1 при разных значениях параметра: (а) $C = 20$; (б) 40; (в) 60; (г) 80

Таблица 1. Результат работы алгоритма для первых случайных проб в тестовом примере 1

k	$\widehat{x}_1^k$	$\widehat{x}_2^k$	$C = 0$	$C = 0{,}20$	$C = 0{,}40$	$C = 0{,}60$	$C = 0{,}80$
1	0,62414	2.3488	0,00	—	—	—	—
2	−1,7029	2,3488	1,00	1,00	1,00	1,00	1,00
3	1,8136	−2,5077	0,97	0,97	0,97	0,97	0,97
4	−1,2763	−1,3144	0,00	—	—	—	—
5	1,9267	−2,1066	1,00	1,00	1,00	1,00	1,00
6	1,72630	0,20775	1,00	1,00	1,00	1,00	1,00
7	1,8399	2,0492	1,00	1,00	1,00	1,00	1,00
8	2,7546	−1,0509	0,00	—	—	—	—
9	−1,9680	−0,51916	0,89	0,89	0,89	0,89	0,89
10	1,5229	0,75407	0,21	0,21	—	—	—
11	−0,77342	1.,152	0,00	—	—	—	—
12	−2,0478	−1,4416	0,79	0,79	0,79	0,79	–
13	−2,7265	−0,40888	0,38	0,38	—	—	—
14	0,75612	−2,8267	0,00	—	—	—	—
15	2,3304	−2,8998	0,36	0,36	—	—	—
16	1,2342	0,88662	0,00	—	—	—	—
17	−2,5920	1,0757	0,00	—	—	—	—
18	−0,16534	−1,8077	1,00	1,00	1,00	1,00	1,00
19	0,34112	0,041409	1,00	1,00	1,00	1,00	1,00
20	2,0490	−1,0385	0,49	0,49	0,49	—	—

$f_2(x) = x_1^2 + x_2^2 - \cos 18x_1 - \cos 18x_2 \to \min$, $x_j \in [-1, 1]$, $j = 1, 2$.

Тестовый пример 3 (функция Розенброка)

$f_3(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \to \min$, $x_1 \in [-5, 10]$, $x_2 \in [0, 15]$.

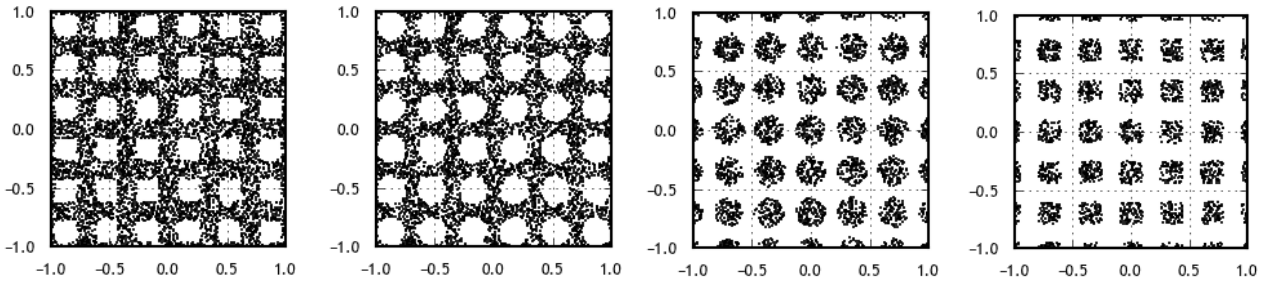
Тестовый пример 4 (функция Camel)

$$f_4(x) = \left(4 - 2{,}1x_1^2 + \frac{1}{3}x_1^4\right) x_1^2 + x_1x_2 + 4x_2^2(x_2^2 - 1) \to \min,\ x_1 \in [-2, 2],\ x_2 \in [-1, 1].$$

Тестовый пример 5

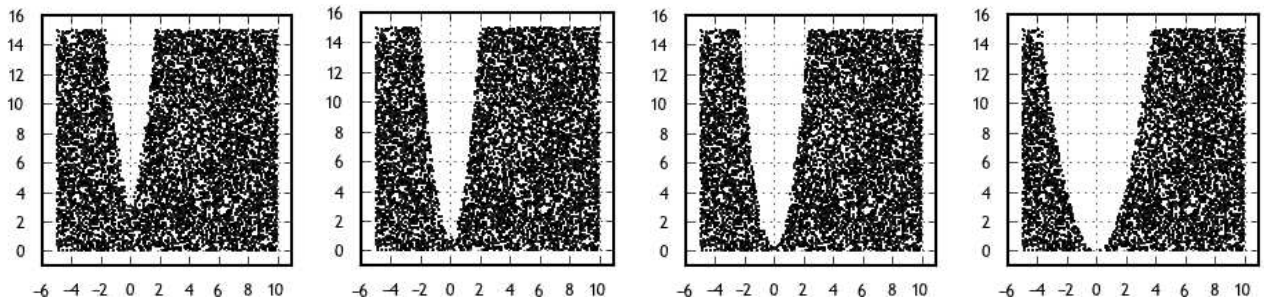
$f_5(x) = \cos x_1 \cos x_2 e^{-(x_1-\pi)^2-(x_2-\pi)^2} \to \min$, $x_j \in [-10, 10]$, $j = 1, 2$.

Тестовый пример 6



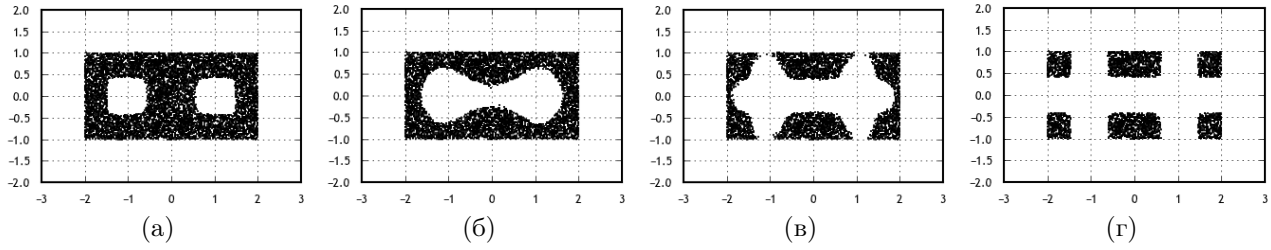
(а) (б) (в) (г)

Рис. 2. Области выпуклости тестовой функции 2 ($S_{f_2} = 0{,}229$) при разных значениях параметра: (а) $C = 20$; (б) 40; (в) 60; (г) 80



(а) (б) (в) (г)

Рис. 3. Области выпуклости тестовой функции 3 ($S_{f_3} = 0{,}649$) при разных значениях параметра: (а) $C = 20$; (б) 40; (в) 60; (г) 99



(а) (б) (в) (г)

Рис. 4. Области выпуклости тестовой функции 4 ($S_{f_4} = 0{,}334$) при разных значениях параметра: (а) $C = 20$; (б) 60; (в) 80; (г) 99

$$f_6(x) = 0{,}5 + \left(\frac{\sin(\sqrt{x_1^2 + x_2^2 + 1})^2 - 0{,}5}{(1 + 0{,}001(x_1^2 + x_2^2))^2} \right) \to \min,\ x_j \in [-4, 4],\ j = 1, 2.$$

Тестовый пример 7

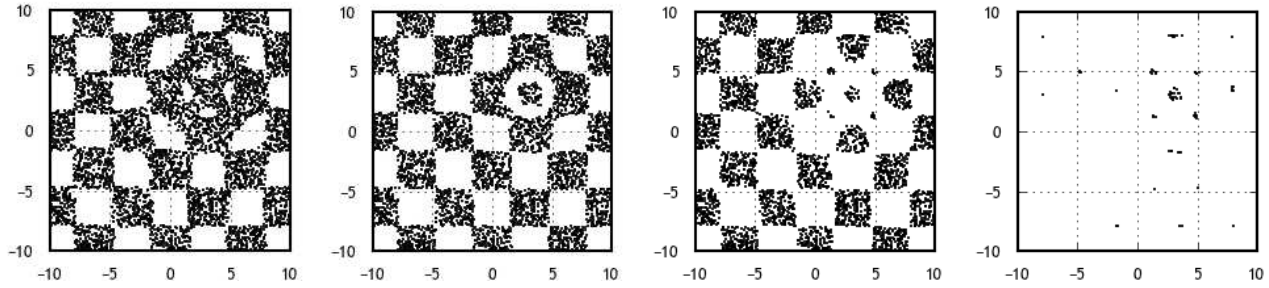
$f_7(x) = x_1^2 \dfrac{x_2}{x_1^4 + x_2^2} \to \min,\ x_j \in [-4, 4],\ j = 1, 2.$

Тестовый пример 8

$f_8(x) = \sin x_1 x_2 \to \min,\ x_j \in [-6, 6],\ j = 1, 2.$

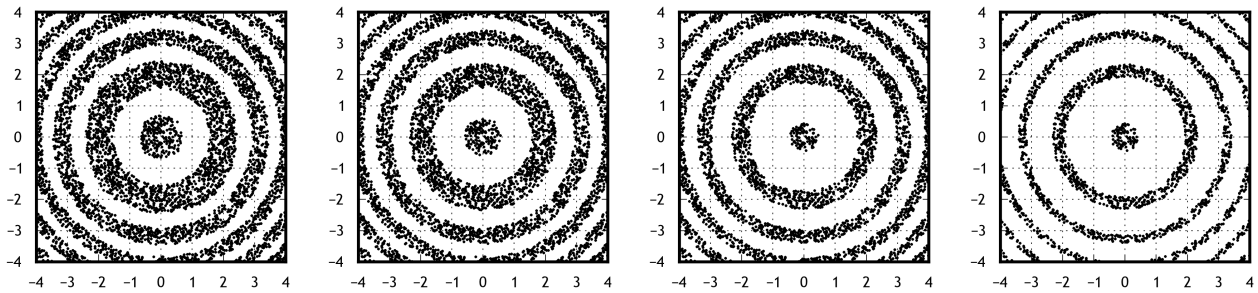
Тестовый пример 9

$f_9(x) = 3(1-x_1)^2 e^{-x_1^2-(x_2+1)^2} - 10(0{,}2x_1 - x_1^3 - x_2^3) e^{-x_1^2-x_2^2} - e^{-(x_1+1)^2 x_2^2/3} \to \min,\ x_j \in [-4, 4],$ $j = 1, 2.$



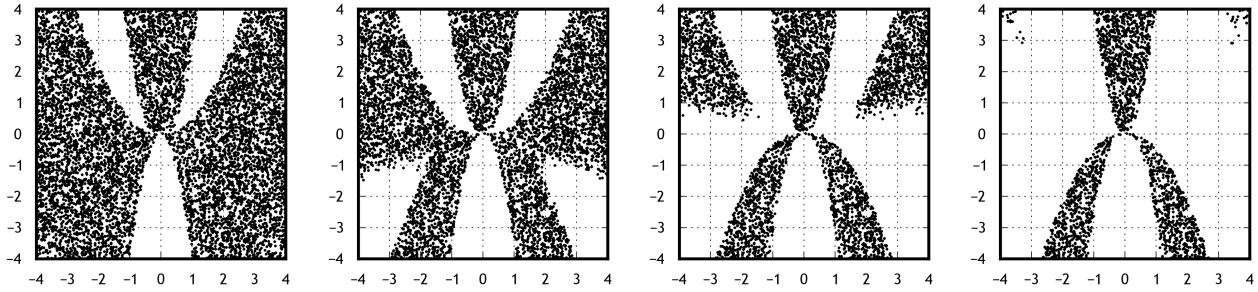
(а) (б) (в) (г)

Рис. 5. Области выпуклости тестовой функции 5 ($S_{f_5} = 0{,}008$) при разных значениях параметра: (а) $C = 20$; (б) 40; (в) 80; (г) 99



(а) (б) (в) (г)

Рис. 6. Области выпуклости тестовой функции 6 ($S_{f_6} = 0{,}250$) при разных значениях параметра: (а) $C = 20$; (б) 40; (в) 80; (г) 99



(а) (б) (в) (г)

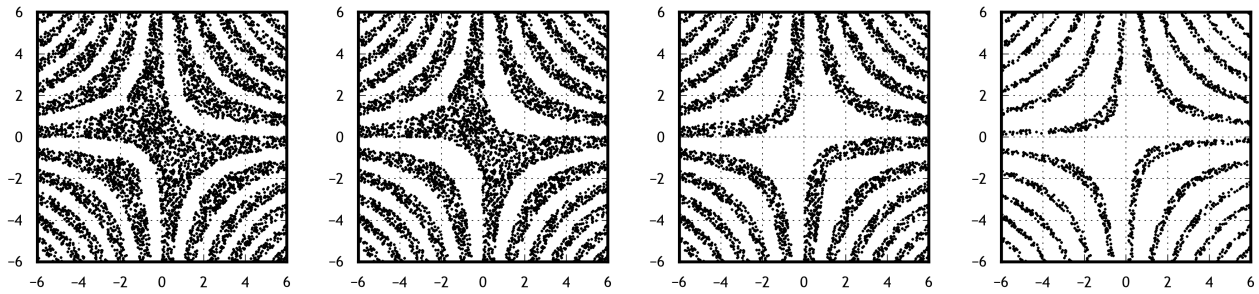
Рис. 7. Области выпуклости тестовой функции 7 ($S_{f_7} = 0{,}075$) при разных значениях параметра: (а) $C = 20$; (б) 40; (в) 60; (г) 80

Тестовый пример 10

$f_{10}(x) = x_1^3 x_2^2 + \sin x_1 - \ln|x_2| + 10 \to \min$, $x_j \in [-8, 8]$, $j = 1, 2$.

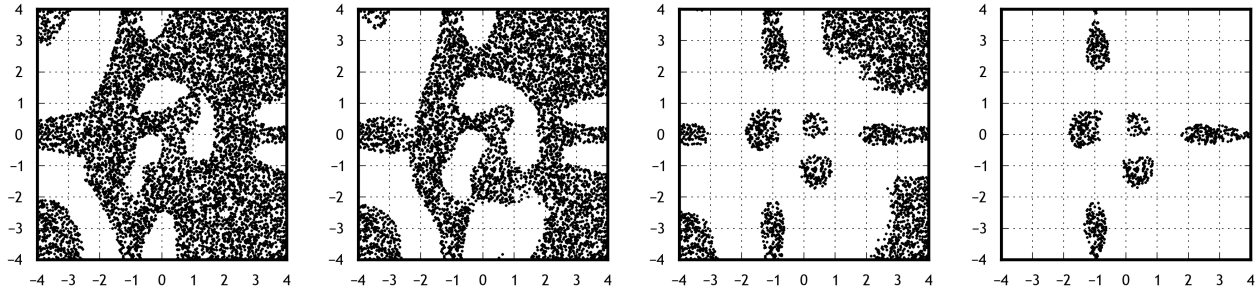
Заключение

С применением предлагаемой вычислительной технологии возможно производить селекцию используемых при моделировании функциональных зависимостей с целью выбора более удобных для последующего оптимизационного анализа. Для задач небольших размерностей технология позволяет выполнить визуализацию «областей выпуклости-



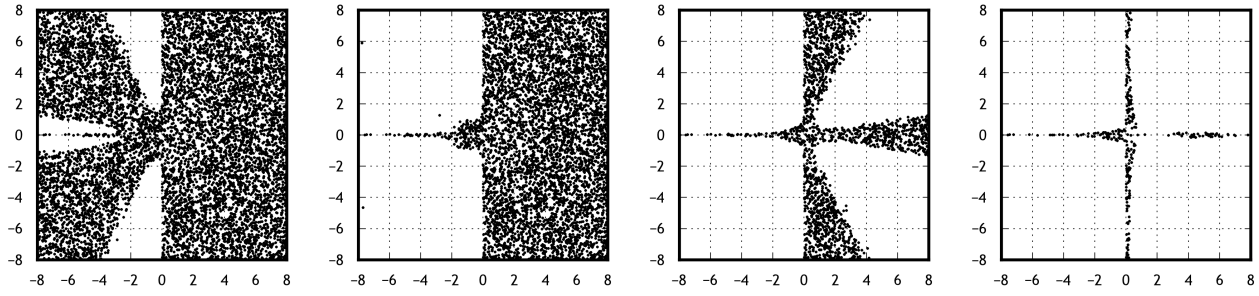
(а) (б) (в) (г)

Рис. 8. Области выпуклости тестовой функции 8 ($S_{f_8} = 0{,}219$) при разных значениях параметра: (а) $C = 20$; (б) 40; (в) 80; (г) 99



(а) (б) (в) (г)

Рис. 9. Области выпуклости тестовой функции 9 ($S_{f_9} = 0{,}080$) при разных значениях параметра: (а) $C = 20$; (б) 40; (в) 80; (г) 99



(а) (б) (в) (г)

Рис. 10. Области выпуклости тестовой функции 10 ($S_{f_{10}} = 0{,}026$) при разных значениях параметра: (а) $C = 20$; (б) 40; (в) 80; (г) 99

невыпуклости». Алгоритм может быть легко распараллелен. Работоспособность предложенного подхода исследована на ряде тестовых и модельных задач. Полученные результаты численных экспериментов позволяют надеяться на создание нового вычислительного инструмента, полезного при решении практических задач из различных научно-технических областей.

Литература

[1] *Скоков В. А.* Пакет анализа оптимизационных экономических моделей ППП «ПАОЭМ ЕС ЭВМ». М.: ЦЭМИ, 1981. 117 с.

[2] *Поляк Б. Т.* Введение в оптимизацию. М.: Наука, 1983. 384 с.

[3] *Гилл Ф., Мюррей У., Райт М.* Практическая оптимизация. М.: Мир, 1985. 509 с.

[4] *Деннис Дж., Шнабель Р.* Численные методы безусловной оптимизации и решения нелинейных уравнений. М.: Мир, 1988. 440 с.

[5] *Murray W., Gill P. E., Saunders M. A.* SNOPT: An SQP algorithm for large-scale constrained optimization // *SIAM J. Optim.*, 2002. No. 12. P. 979–1006.

[6] Encyclopedia of optimization / Eds. C. A. Floudas, P. M. Pardalos. 2nd ed. Springer, 2009. 4646 p.

[7] *Нестеров Ю. Е.* Введение в выпуклую оптимизацию. М.: МЦНМО, 2010. 280 с.

[8] *Sachsenberg B., Schittkowski K.* NLPIP: A Fortran implementation of an SQP-IPM algorithm for solving large-scale nonlinear optimization problems. *User's guide, Version 2.0.* Department of Computer Science, University of Bayreuth, 2013. Report. 29 p.

[9] *Евтушенко Ю. Г.* Методы решения экстремальных задач и их применение в системах оптимизации. М.: Наука, 1982. 432 р.

[10] *Shary S. P.* A surprising approach in interval global optimization // *Reliable Computing*, 2001. Vol. 7, no. 6. P. 497–505.

[11] *Pardalos P., Romeijn E.* Handbook of global optimization. Dordrecht: Kluwer Acad. Publ., 2002. Vol. 2. 569 p.

[12] *Zhigljavsky A. A., Zilinskas A. G.* Stochastic global optimization. Berlin: Springer, 2008. 362 p.

[13] *Сергеев Я. Д., Квасов Д. Е.* Диагональные методы глобальной оптимизации. М.: Физматлит, 2008. 352 р.

[14] *Rios L. M., Sahinidis N. V.* Derivative-free optimization: A review of algorithms and comparison of software implementations // *J. Global Optim.*, 2013. No. 56. P. 1247–1293.

References

[1] *Skokov V. A.* 1981. *Software for optimization analysis of economic models SAP "POAEM ES EVM".* M.: CEMI. 117 p.

[2] *Polyak B. T.* 1983. *Introduction to optimization.* M.: Nauka. 384 p.

[3] *Gill F., Murray D., Wright M.* 1985. *Practical optimization.* M.: Mir. 509 p.

[4] *Dennis J., Schnabel R.* 1988. *Numerical methods for unconstrained optimization and nonlinear equations.* M.: Mir. 440 p.

[5] *Murray W., Gill P. E., Saunders M. A.* 2002. SNOPT: An SQP algorithm for large-scale constrained optimization *SIAM J. Optim.* 12:979–1006.

[6] *Floudas C. A., Pardalos P. M., eds.* 2009. *Encyclopedia of optimization*, 2nd ed. Springer. 4646 p.

[7] *Nesterov Yu. E.* 2010. *Introduction to convex optimization.* M.: MCCME. 280 p.

[8] *Sachsenberg B., Schittkowski K.* 2013. NLPIP: A Fortran implementation of an SQP-IPM algorithm for solving large-scale nonlinear optimization problems. *User's guide, Version 2.0.* Department of Computer Science, University of Bayreuth. Report. 29 p.

[9] *Evtushenko Yu. G.* 1982. *Methods for solving extremal problems and their applications in optimization systems.* M.: Nauka. 432 p.

[10] *Shary S. P.* 2001. A surprising approach in interval global optimization *Reliable Computing* 7(6):497–505.

[11] *Pardalos P., Romeijn E.* 2002. *Handbook of global optimization.* Dordrecht: Kluwer Acad. Publ. Vol. 2. 569 p.

[12] *Zhigljavsky A. A., Zilinskas A. G.* 2008. *Stochastic global optimization.* Berlin: Springer. 362 p.

[13] *Sergeev J. D., Kvasov D. E.* 2008. *Diagonal global optimization methods.* M.: Fizmatlit, 352 p.

[14] *Rios L. M., Sahinidis N. V.* 2013. Derivative-free optimization: A review of algorithms and comparison of software implementations. *J. Global Optim.* 56:1247–1293.

Связность базового скелета и параметрический дескриптор формы*

К. В. Жукова, И. А. Рейер
kz@pisem.net,reyer@forecsys.ru
Москва, Вычислительный центр РАН

Рассматривается изменение с ростом точности аппроксимации устойчивого скелетного представления формы — базового скелета. Этот процесс моделируется стиранием ребер определенными парами кривых. При этом базовый скелет может разделиться на несколько связных компонент. Монотонность и непрерывность изменения позволяют рассматривать параметрическое семейство базовых скелетов и строить масштабируемую гранично-скелетную модель формы, описывающую свойства границы при разных степенях детализации. Для анализа свойств формы, проявляющихся при различных значениях точности, используется параметрический дескриптор, представляющий собой множество вершин выпуклых углов границы аппроксимирующей объект многоугольной фигуры с определенной оценкой значимости. В работе представлено обобщение алгоритма вычисления оценок значимости выпуклых особенностей для случаев нарушения связности базового скелета и исследуется возможность использования параметрического дескриптора для различных типов нарушения связности.

Ключевые слова: *анализ формы; скелетное представление; базовый скелет; параметрический дескриптор формы*

Skeleton base connectivity and parametric shape descriptor*

K. V. Zhukova and I. A. Reyer
Dorodnicyn Computing Centre of RAS, Moscow, Russia

A skeleton base connectivity is described. A skeleton base is a stable shape representation constructed with the use of a polygonal figure approximating the shape. The change of a skeleton base with the growth of the approximation accuracy value is modeled by erasing of edges of the skeleton by pairs of curves. The composition and location of erasing curves is defined by the boundary elements generating an edge, a certain subset of convex boundary vertices, and the accuracy value. A skeleton markup is a set of points of skeleton corresponding to essential changes of the skeleton base's structure. A skeleton markup defines a marked skeleton, in which every edge is erased by a unique pair moving in one direction. A skeleton markup may have points where the skeleton base's connectivity changes. Monotonic and continuous change of a skeleton base allows one to examine the family of skeleton bases and to construct a variously detailed boundary-skeleton shape model. An analysis of this family allows one to calculate significance estimations for curvature features generated by convex vertices of the polygon's boundary. The set of convex vertices with their significance estimations is used as a shape descriptor. In the paper, a generalization of the procedure of curvature features significance estimation for the cases of changes of skeleton base's connectivity is proposed.

Keywords: *shape analysis; skeletal shape representation; skeleton base; parametric shape descriptor*

*Работа выполнена при финансовой поддержке РФФИ, проекты №№ 11-07-00462 и 14-07-00736.

Введение

В задачах машинного зрения часто требуется проводить классификацию объектов по их признакам. Одной из основных характеристик формы объекта, которую можно использовать в качестве признакового описания, является граница объекта. Установлено, что человеческий глаз анализирует форму объекта, опираясь на выпуклости и вогнутости границы. Таким образом, граница отражает особенности формы и делает возможным поиск, основанный на сходстве. Для нахождения особенностей границы используются различные методы, например, методы представления контуров в виде последовательности особенностей-примитивов (выпуклостей и вогнутостей) [1, 2]. При аппроксимациях объекта с различной точностью особенности формы будут проявляться в соответствии со своей значимостью — чем более ярко выражена особенность, тем дольше она сохраняется при уменьшении точности аппроксимации. Поэтому, помимо нахождения особенностей, нужно получить оценку значимости каждой особенности границы. Таким образом, возникает задача построения такого дескриптора формы, который содержит информацию об особенностях формы объекта на заданном уровне аппроксимации. Один из подходов для решения такой задачи представляют собой методы «обнаружения углов» (corner detection) [3, 4]. Идея этих методов состоит в отборе точек локальных экстремумов границы по пороговому значению. Оценкой значимости особенности здесь является абсолютная величина кривизны фрагмента границы, соответствующего этой особенности, поэтому для каждого уровня аппроксимации нужно проводить вычисления этих величин.

Другим популярным инструментом является масштабируемая модель кривизны границы (curvature scale space) [5, 6], которая основана на аппроксимации границы кусочно-гладкой кривой, сглаживании этой кривой и выявлении экстремумов или нулей кривизны границы при разных степенях сглаживания.

Описанные подходы для анализа особенностей границы используют понятие кривизны границы, поэтому для реализации этих методов нужно либо применять дискретные модели кривизны, либо аппроксимировать границу кривыми высших порядков. Эта необходимость является недостатком существующих методов.

Для решения поставленной задачи мы используем параметрическое семейство гранично-скелетных моделей формы, строящихся на основе аппроксимирующей объект многоугольной фигуры [7] и состоящих из базового скелета фигуры [8, 9] и границы объединения множества базовых кругов. С ростом величины точности аппроксимации базовый скелет монотонно и непрерывно изменяется (в смысле расстояния Хаусдорфа) [10]. Анализируя изменение моделей при росте величины точности аппроксимации, для каждой вершины выпуклого угла аппроксимирующего многоугольника можно получить оценку значимости — минимальную величину точности, при которой соответствующая вершине особенность границы исключается из граничного описания. Полученный набор вершин выпуклых углов многоугольной фигуры с сопоставленными им величинами точности аппроксимации мы используем в качестве дескриптора формы.

При исследовании характера изменения базового скелета возможны ситуации, когда базовый скелет перестает быть связным, начиная с некоторого значения точности. В данной работе рассмотрены возможные варианты нарушения связности и сформулированы правила вычисления дескриптора для них.

Базовый скелет и дескриптор формы

Введем основные определения и обозначения. Пусть P — односвязная многоугольная фигура на плоскости R^2 с евклидовым расстоянием $d(\cdot,\cdot)$, ε — некоторое неотрицательное число, $H(\cdot,\cdot)$ — расстояние Хаусдорфа между множествами.

Определение 1. *Пустым кругом фигуры P с центром в точке p и радиусом $r \geqslant 0$ называется замкнутое множество точек $\tilde{C}_r(p) = \{q : q \in R^2, d(p,q) \leqslant r\}$ такое, что $\tilde{C}_r(p) \subset P$.*

Определение 2. *Максимальным пустым кругом называется пустой круг, который не содержится ни в каком другом пустом круге.*

Определение 3. *Скелетом фигуры P называется множество центров всех ее максимальных пустых кругов.*

Определение 4. *Круг C называется ε-допустимым кругом для P, если существует круг C' из множества максимальных пустых кругов P такой, что $H(C', C' \bigcup C) \leqslant \varepsilon$.*

Определение 5. *Круг C называется максимальным ε-допустимым кругом для P, если:*
1) C является ε-допустимым кругом для P;
2) C не содержится целиком ни в каком другом ε-допустимом для P круге.

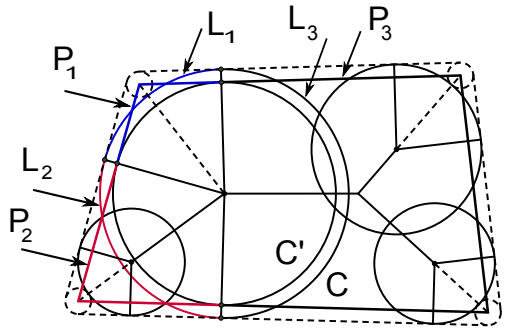


Рис. 1. Базовый скелет

В работе [10] показано, что множество центров максимальных ε-допустимых кругов для P совпадает со множеством центров максимальных пустых кругов для P, и максимальному пустому кругу с центром в точке p и радиусом r соответствует максимальный ε-допустимый круг с центром в p и радиусом $r + \varepsilon$.

Пусть C – максимальный ε-допустимый круг для P. Точки, в которых соответствующий максимальный пустой круг C' касается границы фигуры, разбивают границу на фрагменты $P_1, P_2, \ldots, P_n$, $n \geqslant 2$, а радиусы круга C, проходящие через эти точки, разбивают окружность круга C на дуги $L_1, L_2, \ldots, L_n$ (рис.1).

Определение 6. *Максимальный ε-допустимый круг C называется базовым кругом для многоугольной фигуры P, если $\exists i, j : i \neq j, 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n$ такие, что $H(P_i, L_i) \geqslant \varepsilon$ и $H(P_j, L_j) \geqslant \varepsilon$.*

Определение 7. *Базовым скелетом $S_{base}(P, \varepsilon)$ многоугольной фигуры P называется множество центров всех базовых кругов фигуры.*

Таким образом, базовый скелет P является подмножеством скелета P.

Обозначим U_i, $i = 1, \ldots, n$ — подмножества вершин границы, принадлежащих фрагментам, на которые разбивается граница точками касания круга C'. Пусть d_i максимальное расстояние от центра круга до точек из множества U_i. Упорядочим расстояния d_i, $i = 1, \ldots, n$, по возрастанию:

$$d_1 \leqslant d_2 \leqslant \ldots \leqslant d_{n-1} \leqslant d_n,$$

и выберем такое подмножество U_j, что соответствующее расстояние d_j является вторым по величине. Выбранное подмножество вершин границы обозначим U', а наиболее удаленную от центра круга точку U' будем обозначать f.

В работе [10] показано, что базовый скелет при росте ε измененяется монотонно и непрерывно в смысле расстояния Хаусдорфа. Данный процесс моделируется «стиранием» ребер скелета парами кривых — парабол и гипербол (рис. 2). Для ребра, порожденного парой сегментов границы, стирающими кривыми являются параболы с фокусом в точке f множества U' и директрисами, параллельными порождающим ребро сегментам границы. Ребро, порожденное парой вершин границы, стирают две гиперболы с фокусами в точке f и одной из порождающих ребро вершин. В случае ребра, порожденного вершиной и сегментом границы, стирающие кривые — парабола с фокусом в точке f и директрисой, параллельной сегменту границы, и гипербола с фокусами в точке f и вершине границы.

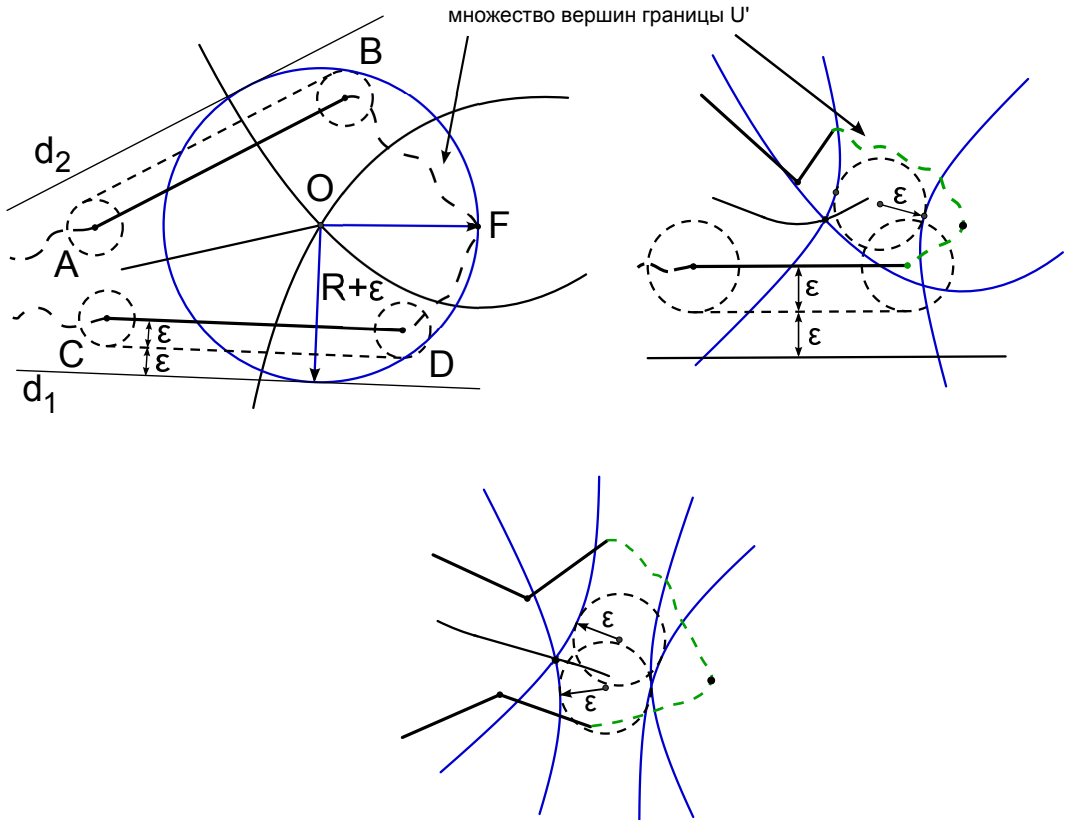


Рис. 2. Стирание кривыми ребра: (а) порожденного парой сегментов границы; (б) порожденного вершиной и сегментом границы; (в) порожденного парой вершин границы

Очевидно, что пара стирающих кривых не всегда является постоянной для ребра скелета. Во-первых, на ребре могут находиться точки, в которых происходит смена дальней

точки f в пределах множества U' (точки пересечения ребра с диаграммой Вороного дальней точки [11] для U'). Во-вторых, возможны ситуации, когда для разных фрагментов ребра множество U' различно: смена множества происходит в так называемых центральных точках [10], равноудаленных от точек f нескольких фрагментов границы. Кроме того, ветви пары кривых могут стирать в противоположных направлениях два соседних фрагмента ребра, имеющих общую точку, в которой одна из кривых касается ребра, а вторая кривая вырождается в луч. Все такие внутренние точки ребер, в которых происходит изменение пары стирающих кривых или направления стирания (точки пересечения с диаграммой Вороного дальней точки, центральные точки, точки касания), вместе с исходными вершинами скелетного графа образуют разметку [10], в которой для каждого ребра скелета однозначно определена пара стирающих кривых и направление стирания (рис. 3).

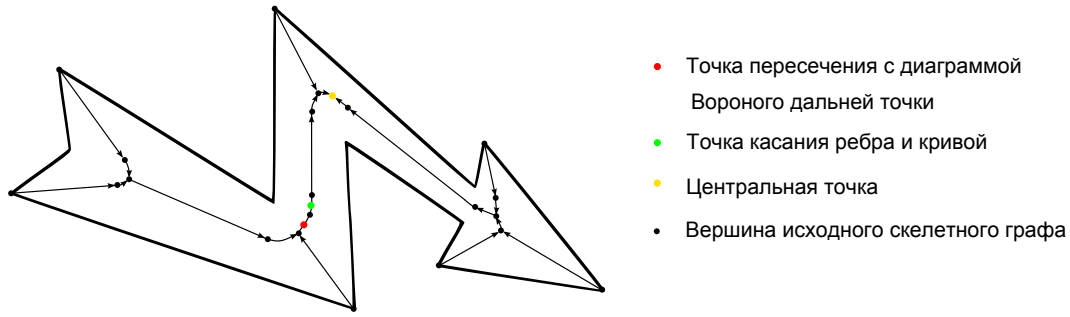


Рис. 3. Разметка скелетного графа

Процесс стирания начинается с терминальных вершин скелета и монотонно и непрерывно продолжается «вглубь» фигуры с ростом значения точности ε. Это значит, что для фигуры опеределено параметрическое семейство базовых скелетов. При этом дуга базовой окружности с центром в терминальной вершине базового скелета аппроксимирует с известной точностью соответствующий участок границы. Поэтому границу объединения множества всех базовых кругов можно рассматривать в качестве модели контура фигуры, отражающей те свойства границы, которые являются существенными для данной точности аппроксимации (рис. 4). Таким образом, анализируя изменение гранично-скелетных моделей формы, можно оценить значимость особенностей (чем существеннее особенность, тем дольше она сохраняется в модели формы) и построить параметрический дескриптор, который представляет собой набор выпуклых вершин границы с определенной оценкой значимости [12].

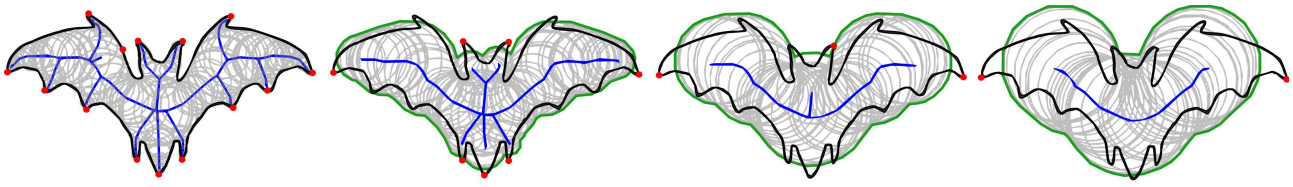


Рис. 4. Параметрическое семейство моделей формы

Рассмотрим «идеальный» случай, когда в базовом скелете нет точек нарушения связности и стирание заканчивается в центральной точке (рис. 5). Заметим, что выпуклые вершины границы являются терминальными вершинами скелета фигуры, поэтому далее

для удобства будем говорить о вычислении оценок значимости для терминальных вершин скелета. Идею вычисления дескриптора можно описать, проведя аналогию с соревнованиями по бегу: из каждой терминальной вершины одновременно стартуют бегуны. В вершинах, в которые прибегают несколько участников (т. е. в вершинах скелета степени больше двух), тот, кто прибежал раньше остальных, останавливается и выбывает из гонки, а значение точности в этой вершине присваивается соответствующей исходной терминальной вершине в качестве оценки значимости. Тот же, кто прибежал в вершину последним, продолжает двигаться дальше. Заканчивается соревнование, когда все оставшиеся бегуны встречаются в центральной точке. Количество бегунов, добравшихся до центральной точки, равно степени этой вершины в размеченном скелетном графе.

Опишем алгоритм вычисления оценок значимости более формально. Двигаемся от терминальных вершин по направлению стирания, запоминая значение точности в вершине. Если значение в вершине для текущего входящего (по направлению стирания) ребра не является максимальным из всех входящих ребер, то процесс останавливается и это значение присваивается исходной терминальной вершине в качестве оценки значимости. Таким образом, входящие ребра определяют, для каких терминальных вершин процесс вычисления оценки закончен, а для какой продолжится дальше. Исходящие ребра определяют дальнейшее направление. Продолжаем двигаться по ребрам, пока не окажемся в центральной точке, у которой нет исходящих ребер. Значение точности в ней будет максимально и дуги базовой окружности в центром в этой точке будут аппроксимировать соответствующие фрагменты границы фигуры.

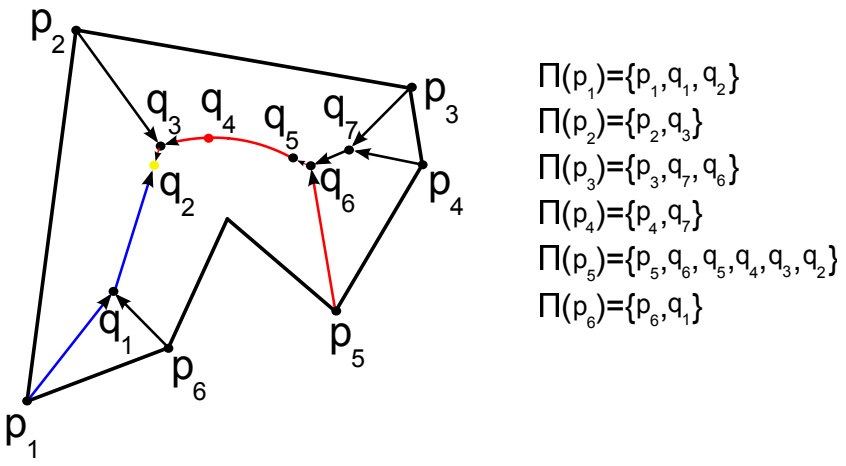


Рис. 5. Построение путей для терминальных вершин

Для каждой терминальной вершины скелета p будем строить путь из вершин размеченного скелета, упорядоченных по направлению стирания: $\Pi(p) = \left\{q_1^{\varepsilon^1}, \ldots, q_k^{\varepsilon^k} \mid \varepsilon^i = \max\{\varepsilon, \varepsilon \in E(q_i)\}\right\}$, где $E(q) = \{\varepsilon_1, \ldots, \varepsilon_{n-1}\}$ — набор значений ε вершины q, при которых стираются входящие в q ребра. Обозначим через $\Upsilon(p) = \{\varepsilon^1, \ldots, \varepsilon^k\}$ — соответствующую последовательность значений точности. Очевидно, что эта последовательность возрастающая. За оценку значимости вершины p принимается максимальный (т. е. последний) элемент из $\Upsilon(p)$:

$$\Psi(p) = \max\left\{\varepsilon^i \mid \varepsilon^i \in \Upsilon(p)\right\} . \tag{1}$$

Таким образом, каждой терминальной вершине p соотвествует упорядоченная последовательность вершин размеченного скелета $\Pi(p)$. На рис. 5 приведен пример построения путей для вершин фигуры.

Более сложные ситуации возникают, если в разметке скелетного графа присутствуют точки нарушения связности: при некотором значении точности из такой точки начинается стирание ребра в противоположных направлениях и базовый скелет разделяется на две части. Опишем, как и в каких случаях это происходит.

Точки нарушения связности

Нарушение связности в точках пересечения ребер скелета с ребрами диаграммы Вороного дальней точки. Рассмотрим фрагмент фигуры на рис. 6а.

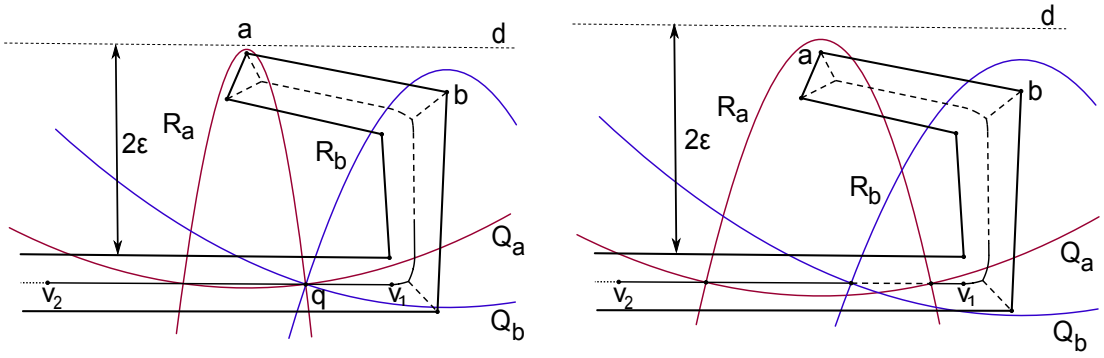


Рис. 6. Нарушение связности в точке пересечения ребра скелета с ребром диаграммы Вороного дальней точки (а,б)

На ребре v_1v_2 находится точка q, в которой ребро пересекается с ребром диаграммы Вороного дальней точки для соответствующего множества вершин границы и происходит смена стирающей кривой. Эта точка равноудалена от вершин a и b границы. Стирающими кривыми для ребра v_1v_2 являются две пары парабол: (R_a, Q_a) для части ребра qv_1 и (R_b, Q_b) для части ребра qv_2. При некотором значении точности все четыре параболы будут пересекаться в точке q, а при дальнейшем увеличении значения ε две пары парабол будут стирать ребро в разных направлениях, т. е. в точке q произойдет нарушение связности (рис. 6, б).

Нарушение связности в точке касания ребра и стирающей кривой. Рассмотрим случай, когда оба элемента, порождающие ребро скелета, являются сегментами (рис. 7). При $\varepsilon < \varepsilon_*$ стирающие параболы не пересекают ребро. При достижении точностью значения ε_* одна из парабол касается ребра, а вторая вырождена и представляет собой луч. При последующем увеличении ε в точке касания происходит нарушение связности и ребро стирается параболами в противоположных направлениях. Значение ε_* в точке касания равно половине расстояния от точки f до сегмента границы, определяющего вырожденную параболу.

Нарушение связности и центральные точки. В случае, если на ребре есть центральная точка, то возможны три варианта стирания ребра: две пары кривых стирают ребро от концевых точек к центральной точке (такая точка в [10] названа центральной точкой I типа); две пары кривых стирают ребро от центральной точки к концевым точкам с нарушением связности (центральная точка II типа); ребро стирается в одном направлении

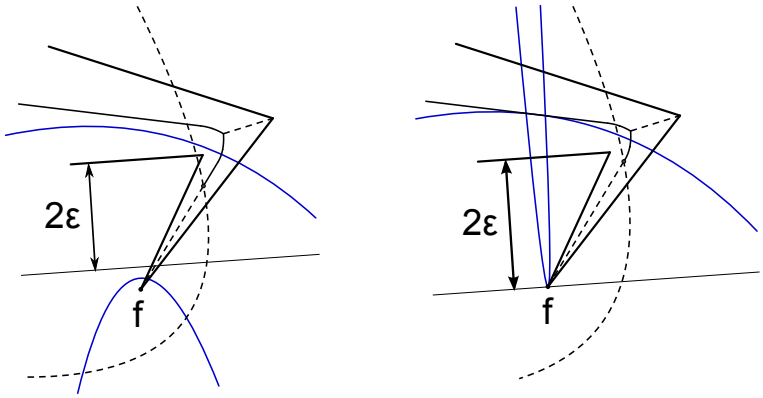


Рис. 7. Нарушение связности в точке касания ребра и стирающей кривой

и при переходе через центральную точку меняется пара стирающих кривых (центральная точка III типа). В случае, когда точка нарушения связности является точкой касания, она также может быть и центральной точкой (II типа). Рассмотрим фигуру на рис. 8. Точка z лежит на ребре, порожденном парой сегментов границы, и является точкой касания. При некотором значении точности произойдет касание стирающей параболы и ребра в точке z и из нее начнется стирание ребра в противоположных направлениях. Кроме этого, точка z является центральной точкой: она равноудалена от «дальних» точек f_1 и f_2 соответствующих множеств U_1' и U_2' вершин границы. Базовый скелет распадается на два фрагмента, стирание каждого из которых заканчивается в точках t_1 и t_2 соответственно.

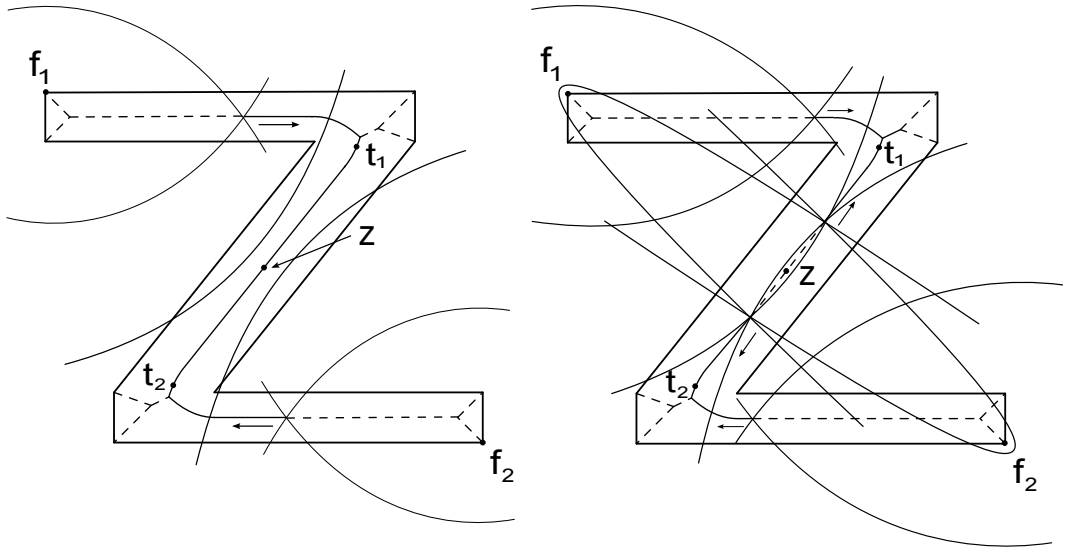


Рис. 8. Центральная точка II типа — точка нарушения связности

Теперь рассмотрим пример с центральной точкой III типа, в которой стирание не заканчивается и не начинается, но происходит смена множества U' (рис. 9). После того, как базовый скелет разделяется в точке нарушения связности z, стирание продолжается через точку v пересечения с диаграммой Вороного дальней точки и через центральную точку III типа c. При этом при переходе через c меняется пара стирающих кривых (так как меняется множество U' для ребра). В данном примере фрагмент скелета, содержащий точку t_1 исчезнет раньше, чем фрагмент с t_2, и останется только часть скелета, которая

соответствует особенности f_2, а для особенности f_1 уже не будет соответствующих ветвей скелета. Это означает, что в получившейся модели формы дуга базовой окружности с центром в терминальной вершине аппроксимирует только часть границы фигуры, а для оставшегося фрагмента границы нет аппроксимирующей его дуги базовой окружности.

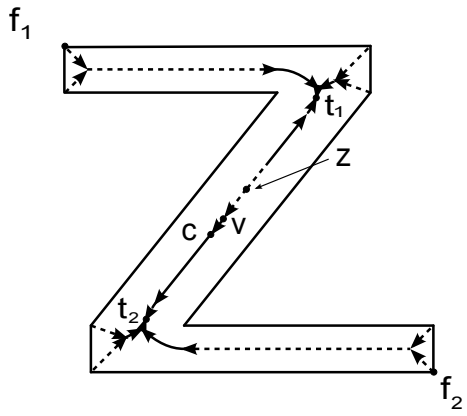


Рис. 9. Центральная точка III типа

На рис. 10 приведены еще два примера, когда в результате остается часть скелета, соответствующая только одной особенности. Для фигуры, изображенной на рис. 10, а часть скелета, соответствующая особенности p_1, исчезнет при достижении кривыми центральной точки I типа c, в то время как часть, соответствующая особенности p_2, исчезнет позже. На рис. 10, б приведен пример, в котором отсутствуют точки нарушения связности. Стирание части скелета, соответствующей особенности p_1, закончится в центральной точке III типа c, дальше стирание продолжат кривые, относящиеся к особенности p_2.

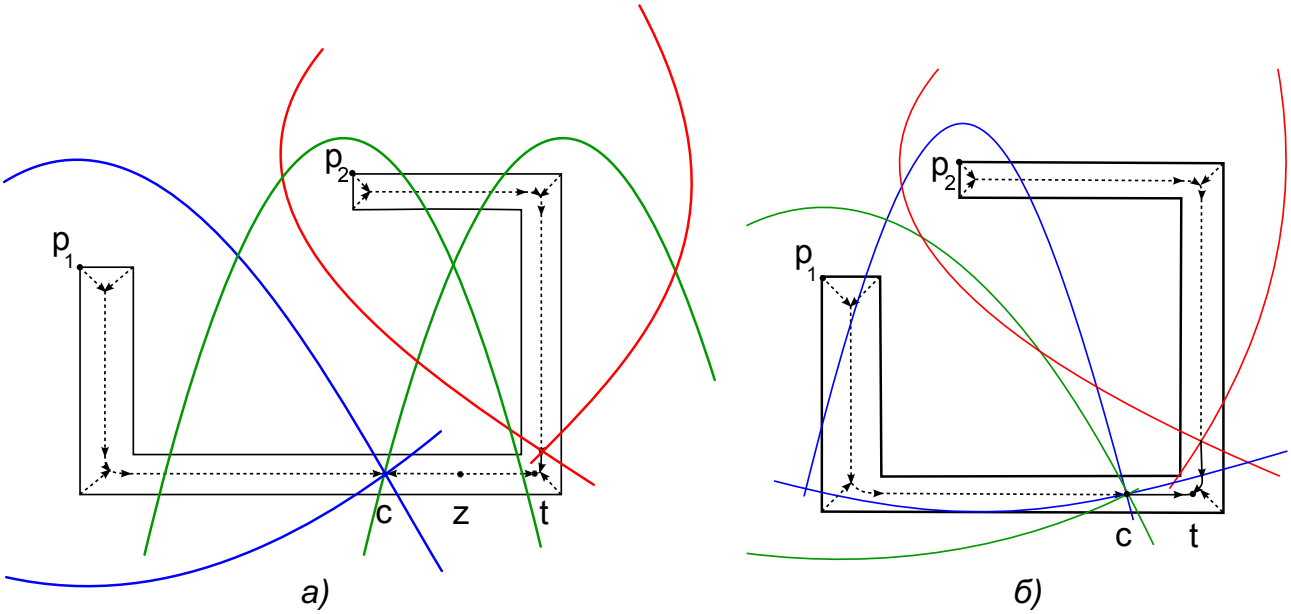


Рис. 10. В модели формы остается только одна особенность: (а) при наличии точки нарушения связности, (б) при отсутствии нарушения связности

В размеченном скелете может быть несколько центральных точек. На рис. 11 изображена фигура, при стирании скелета которой произойдет нарушение связности в двух точках z_1 и z_2, в результате чего базовый скелет разделится на три фрагмента. Каждый из этих фрагментов содержит центральную точку I типа (c_1, c_2, c_3), в которой закончится стирание фрагмента. При этом значения точности в центральных точках не совпадают: $\varepsilon_{c_1} < \varepsilon_{c_2} < \varepsilon_{c_3}$. Таким образом, в конце останется только один фрагмент базового скелета, содержащий центральную точку c_3 с максимальным значением точности.

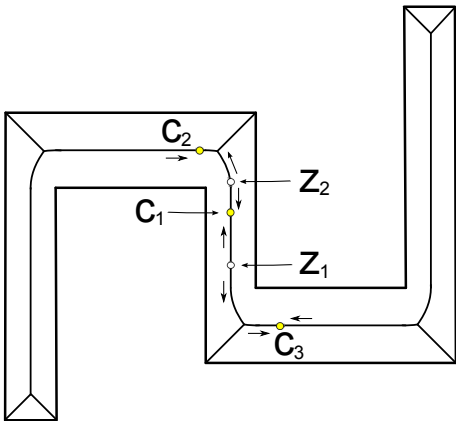


Рис. 11. Несколько точек нарушения связности и центральных точек

Нарушение связности в вершине скелета. Итак, нарушение связности возможно в точках пересечения с диаграммой Вороного дальней точки, в точках касания для ребер, порожденных парой сегментов границы, и в центральных точках II типа. Помимо этого, точкой нарушения связности может оказаться и вершина скелета, которая не относится ни к какому из перечисленных выше типов точек разметки. Рассмотрим пример на рис. 12: в вершине q происходит нарушение связности и ребра, выходящие из вершины q, стираются парами кривых, порожденными одной точкой f. Отметим, что в этом случае в стирании участвуют три параболы, так как исходящие из q ребра имеют общий порождающий сегмент границы.

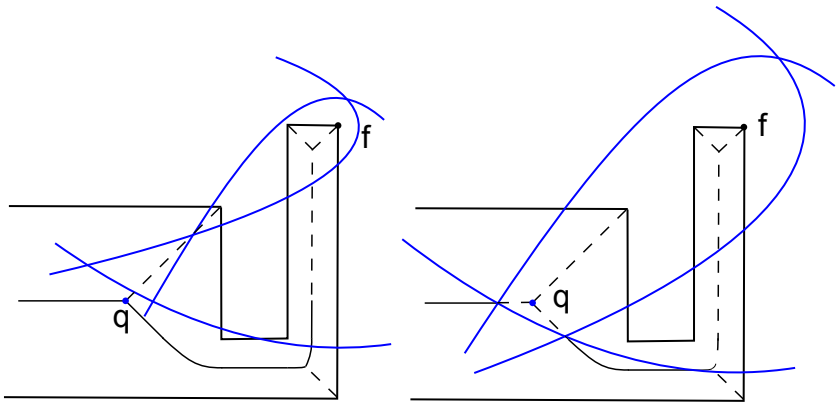


Рис. 12. Нарушение связности в вершине скелета

Нарушение связности и параметрический дескриптор формы

Рассмотрим пример на рис. 13а. Для фрагмента ребра, содержащего точку нарушения связности z, множество U' состоит из точек p_1, p_2, p_3, p_4. При некотором значении точности из точки нарушения связности z начинается стирание двумя парами кривых. В результате выполнения описанного выше алгоритма построения дескриптора стирание одного фрагмента скелета начнется из точки p_2 и закончится в точке t, а оценка значимости вершины p_2 $\Psi(p_2) = \varepsilon_t$. Для другой части фигуры стирание начнется из точки p_6 и закончится в центральной точке c, оценка значимости вершины p_6 $\Psi(p_6) = \varepsilon_c$. Рассмотрим модель фи-

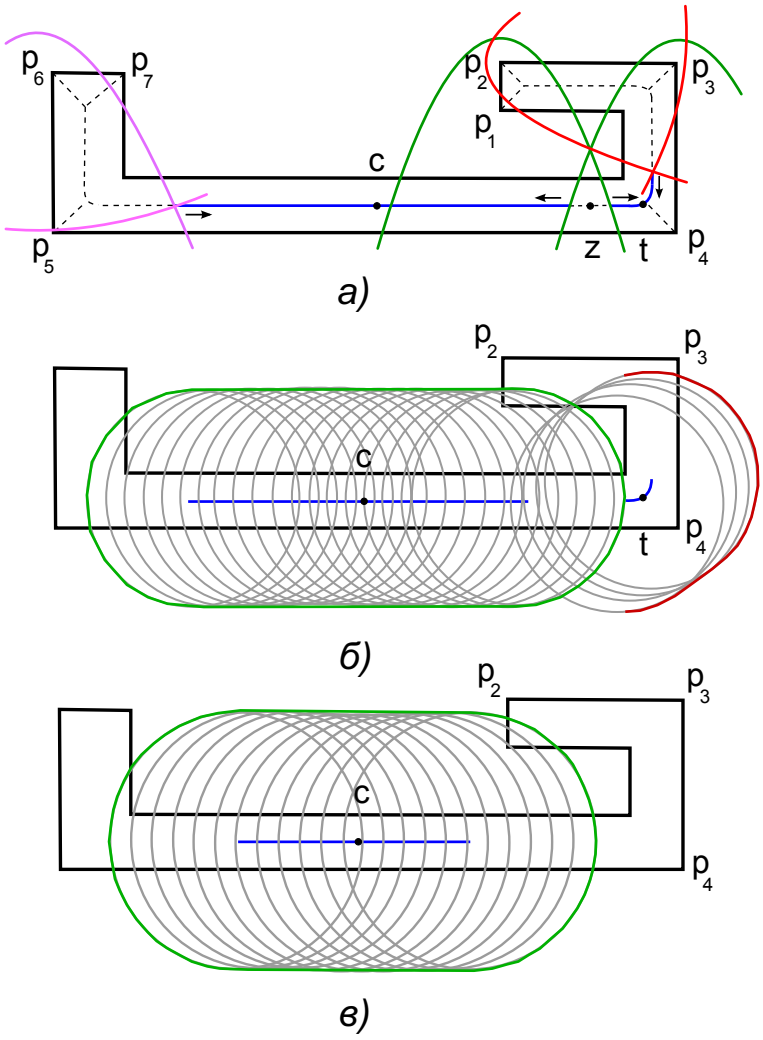


Рис. 13. Стирание в случае нарушения связности и модели формы

гуры для некоторой точности ε, такой, что $\varepsilon_z < \varepsilon < \varepsilon_t$, т. е. при которой в базовом скелете уже произошло нарушение связности, но еще присутствует ребро с точкой t (рис. 13, б). Эта модель состоит из двух компонент, одна из которых аппроксимирует фрагмент границы $p_1p_2p_3p_4$, при этом оценка значимости этой особенности равна $\Psi(p_2)$. Однако при точности $\varepsilon > \varepsilon_t$ в модели останется одна компонента (рис. 13, в) и можно видеть, что в ней есть дуга базовой окружности, аппроксимирующая особенность $p_1p_2p_3p_4$ и, значит, оценка значимости этой особенности больше, чем ε_t. Поэтому при построении дескриптора нуж-

но учитывать и ребра, содержащие точки нарушения связности. Адаптируем алгоритм построения дескриптора для случаев с нарушением связности.

Вернемся к примеру с соревнованиями по бегу. Будем считать, что в момент нарушения связности из точки z стартуют два бегуна в противоположных направлениях. Один из них продолжит бежать к центральной точке c, а второй отправится в точку t и сообщит спортсмену, прибежавшему в t из вершины p_2, о том, что за него продолжает соревнование первый бегун из точки z. Таким образом, для вершины p_2 нужно следить за результатами двух бегунов и в конце соревнования выбрать того, кто прибежит позднее.

Для каждой вершины p будем строить путь $\Pi(p)$ в размеченном скелетном графе. При прохождении ребра помечаем его как «пройденное». Построение пути прекращается, если выполнено одно из условий:

(1) значение ε для текущего входящего в вершину q ребра не максимально из значений всех входящих в q ребер;

(2) q является центральной точкой с максимальным значением ε из значений всех центральных точек;

(3) количество исходящих из q ребер не равно 1.

Алгоритм состоит из четырех шагов:

Шаг 1. Для каждой терминальной вершины p строим путь $\Pi(p)$. При этом построение пути для p завершено, если $\Pi(p)$ заканчивается по условиям (1) или (2). Обозначим множество «недостроенных» путей $\overline{\Pi}$.

Шаг 2. Если есть точки нарушения связности, то останутся непройденные ребра (по крайней мере, выходящие из точки нарушения связности). Из каждой точки нарушения связности q строим выходящие из нее два пути $\Pi^1(q)$ и $\Pi^2(q)$, используя только непройденные ребра и прекращая построение, если выполнено одно из условий (1), (2), (3). Обозначим множество таких пар путей $Z = \{(\Pi^1(q), \Pi^2(q))\}$.

Шаг 3. Для каждого недостроенного пути $\Pi(p)$ из $\overline{\Pi}$ ищем среди пар из Z такую пару $(\Pi^1(q_k), \Pi^2(q_k))$, один из путей которой $\Pi^i(q_k), i = 1, 2$ заканчивается в последней вершине $\Pi(p)$, и добавляем к пути $\Pi(p)$ вершины $\Pi^i(q_k)$ в обратном направлении, а также вершины второго пути пары $\Pi^j(q_k)$, $i, j = 1, 2, i \neq j$. Найденная пара удаляется из множества Z. Если последняя вершина в полученном пути $\Pi(p)$ удовлетворяет условию (1) или (2), то построение пути завершено и он удаляется из множества $\overline{\Pi}$.

Повторяем шаг 3, пока множество Z не пусто.

Шаг 4. Назначение оценок вершинам. За оценку значимости вершины p принимаем максимальное значение точности из значений вершин построенного для p пути $\Pi(p)$.

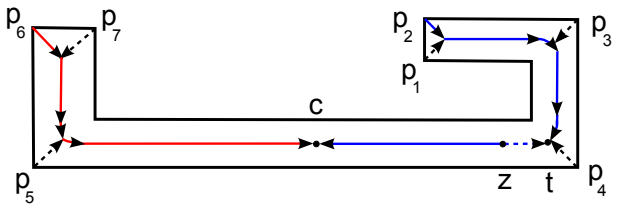


Рис. 14. Построение путей в случае нарушения связности

Вернемся к фигуре на рис. 13. Для вершин p_1, p_3, p_4, p_5 и p_7 пути состоят из двух вершин скелета – терминальной и другой концевой вершиной соответствующего терминального ребра (обозначены прерывистой линией на рис. 14). Для вершины p_6 путь $\Pi(p_6)$

состоит из вершин скелета от p_6 до центральной точки c. Оценкой значимости p_6 является значение точности в c: $\Psi(p_6) = \varepsilon_c$. Для вершины p_2 путь $\Pi(p_2)$ состоит из вершин скелета от p_2 до c, причем вершины фрагмента скелета от точки z до t добавляются в $\Pi(p_2)$ в направлении, обратном стиранию: от t до z. Оценкой значимости вершины p_2 является значение точности в точке c: $\Psi(p_2) = \varepsilon_c$.

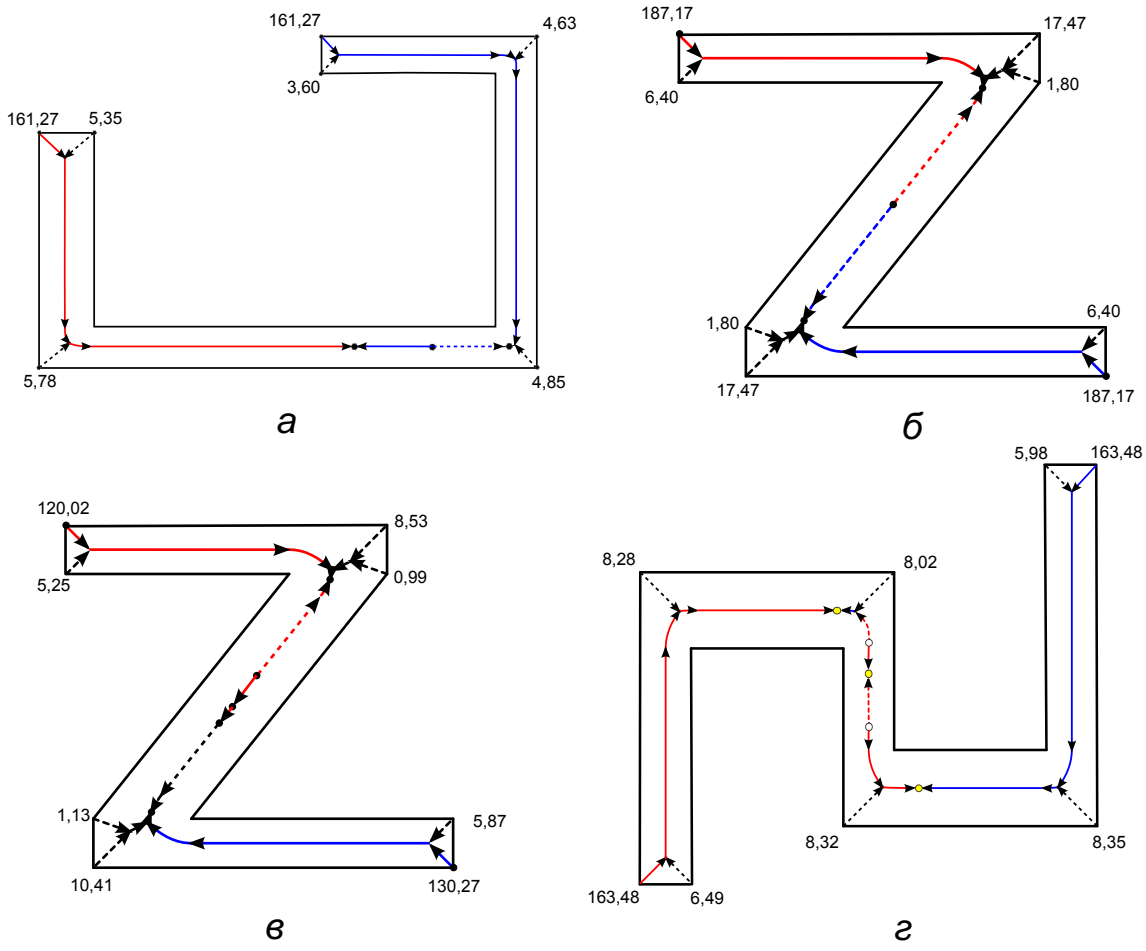


Рис. 15. Оценки значимости особенностей форм. Размеры фигур: (а) 462×309; (б) 495×495; (в) 376×371; (г) 442×445

На рис. 15 представлен результат работы алгоритма построения дескриптора с вычисленными значениями оценок значимости особенностей границ.

Выводы

В работе описаны возможные ситуации нарушения связности базового скелета и представлено обобщение алгоритма вычисления параметрического дескриптора формы для таких случаев. Показано, что в ряде случаев при больших значениях точности можно получить гранично-скелетные модели формы, в которых определнному фрагменту границы не сопоставлено ни одного базового круга с аппроксимирующей этот фрагмент дугой окружности. Таким образом, начиная с некоторого значения точности, полученные модели формы могут, вообще говоря, не быть корректными. Анализ размеченного скелетного графа дает возможность заранее определить, возникнет ли такая ситуация. Вопрос выбо-

ра допустимого диапазона значений точности и соответствующего набора моделей форм в конкретной задаче остается на усмотрение исследователя.

Литература

[1] *Rosin P. L.* Multiscale representation and matching of curves using codons // *CVGIP: Graphical Models and Image Processing*, 1993. Vol. 55, no. 4. P. 286–310.

[2] *Galton A., Meathrel R.* Qualitative Outline Theory // *IJCAI'99 Proceedings of the 16th international joint conference on Artificial intelligence,* 1999. Vol. 2. P. 1061–1066.

[3] *Koplowitz J., Plante S.* Corner detection for chain codes curves // *Pattern Recognition*, 1995. Vol. 28, no. 6. P. 843–852.

[4] *Ray B. K., Pandyan R.* ACORD — an adaptive corner detector for planar curves // *Pattern Recognition*, 2003. Vol. 36. P. 703–708.

[5] *Dudek G., Tsotsos J. K.* Shape representation and recognition from mutliscale curvature // *Computer Vision Image Understanding*, 1997. Vol. 68, no. 2. P. 170–189.

[6] *Abbasi S., Mokhtarian F., Kittler J.* Curvature scale space image in shape similarity retrieval // *MultiMedia Syst.*, 1999. Vol. 7. P. 467–476.

[7] *Местецкий Л. М.* Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. М.: Физматлит, 2009. 288 с.

[8] *Местецкий Л. М., Рейер И. А.* Непрерывное скелетное представление изображения с контролируемой точностью // *Тр. 13-й Междунар. конф. ГРАФИКОН-2003.* Москва, 2003. С. 246–249.

[9] *Жукова К. В., Рейер И. А.* Параметрическое семейство гранично-скелетных моделей формы // *Математические методы распознавания образов: 14-я Всеросс. конф.: Сб. докл.*, 2009. С. 346–350.

[10] *Жукова К. В., Рейер И. А.* Параметрическое семейство базовых скелетов многоугольной фигуры // *Машинное обучение и анализ данных*, 2012. Т. 1, № 4. С. 391–410.

[11] *Препарата Ф., Шеймос М.* Вычислительная геометрия: введение. М.: Мир, 1989. 478 с.

[12] *Жукова К. В., Рейер И. А.* Параметрический дескриптор формы на основе гранично-скелетной модели // *Математические методы распознавания образов: 15-я Всеросс. конф.: Сб. докл.*, 2011. С. 408–411.

References

[1] *Rosin P. L.* 1993. Multiscale representation and matching of curves using codons. *CVGIP: Graphical Models and Image Processing* 55(4):286–310.

[2] *Galton A., Meathrel R.* 1999. Qualitative outline theory. *16th Joint Conference (international) on Artificial Intelligence Proceedings* 2:1061–1066.

[3] *Koplowitz J., Plante S.* 1995. Corner detection for chain codes curves. *Pattern Recognition* 28(6):843–852.

[4] *Ray B. K., Pandyan R.* 2003. ACORD — an adaptive corner detector for planar curves. *Pattern Recognition* 36:703–708.

[5] *Dudek G., Tsotsos J. K.* 1997. Shape representation and recognition from mutliscale curvature. *Computer Vision and Image Understanding* 68(2):170–189.

[6] *Abbasi S., Mokhtarian F., Kittler J.* 1999. Curvature scale space image in shape similarity retrieval. *MultiMedia Syst.* 7:467–476.

[7] *Mestetskii L. M.* 2009. *Continuous morphology of binary images: Figures, skeletons and circulars.* Moscow: Fizmatlit Publ. 288 p. (In Russian.)

[8] *Mestetskii L. M., Reyer I. A.* 2003. Continuous skeletal representation of image with controllable accuracy. *Conference (International) Graphicon Proceedings.* Moscow. 246–249. (In Russian.)

[9] *Zhukova K. V., Reyer I. A.* 2009. Parametric family of boundary-skeletal shape models. *14th Russian Conference on Mathematical Methods for Pattern Recognition (MMPR-14) Proceedings.* Suzdal. 346–350. (In Russian.)

[10] *Zhukova K. V., Reyer I. A.* 2012. Parametric family of skeleton bases of a polygonal figure. *Machine Learning Data Analysis* 1(4):391–410. (In Russian.)

[11] *Preparata F., Shamos M.* 1985. *Computational Geometry.* N. Y.: Springer-Verlag.

[12] *Zhukova K. V., Reyer I. A.* 2011. Parametric shape descriptor based on a boundary-skeletal model. *15th Russian Conference on Mathematical Methods for Pattern Recognition (MMPR-15) Proceedings.* Petrozavodsk. 408–411. (In Russian.)

Помехоустойчивый морфологический алгоритм обнаружения вилочного погрузчика на видео*

В. О. Черноусов, А. В. Савченко
v.chernousov@mail.ru
НИУ Высшая школа экономики, Москва, Россия

Исследуется задача обнаружения движущегося вилочного погрузчика на видео при наличии помех, в которой точность традиционного сопоставления локальных дескрипторов (SURF, SIFT, FAST, ORB) не достаточна. Предложен новый алгоритм, на первом этапе которого на кадре выделяются движущиеся объекты, после чего на передней части объекта находится потенциальная область вил и груза. На втором этапе выделяются контуры, затем с помощью морфологических преобразований вычисляются элементарные геометрические признаки объекта. Показано, что такой подход позволяет на 7% и 50% понизить вероятности ложной тревоги и пропуска события, соответственно, при детектировании пустого погрузчика по сравнению с методом FAST, является устойчивым к аддитивному шуму, а обработка одного кадра происходит в среднем на 30 мс быстрее.

Ключевые слова: *обнаружение объектов на видео; зашумленная среда; motion history image; морфология*

A noise-resistant morphological algorithm of video-based moving forklift truck detection*

V. O. Chernousov, A. V. Savchenko
National Research University Higher School of Economics, Moscow, Russian Federation

Background: The problem of video-based detection of the moving forklift truck is explored. It is shown that the detection quality of the state-of-the-art local descriptors (SURF, SIFT, FAST, ORB) is not satisfactory if the resolution is low and the lighting is changed dramatically.
Methods: In this paper, it is proposed to use a simple mathematical morphological algorithm to detect the presence of a cargo on the forklift truck. At first, the movement direction is estimated by the updating motion history image method and the front part of the moving object is obtained. Next, contours are detected and binary morphological operations in front of the moving object are used to estimate simple geometric features of empty forklift.
Results: The authors' experimental study shows that the best results are achieved if the bounding rectangles of empty forklift contours are used as an object validation rule. Namely, FAR and FRR of empty cargo detection is 7% and 50% lower than FAR and FRR of the FAST descriptor. The proposed method is much more resistant to the effect of additive noise. The average frame processing time for the morphological algorithm is 5 ms (compare with 35 ms of FAST method).
Concluding Remarks: The proposed morphological method is task specific and can be used only for forklift truck detection. Additional detection principles need to be added to adopt algorithm for other moving object detection in noisy environment.

Keywords: *video-based object detection; noisy environment; motion history image; binary morphology*

*Работа выполнена за счет средств гранта Российского научного фонда (проект №14-41-00039).

Введение

В настоящее время все актуальнее становится исследование способов применения алгоритмов машинного зрения в прикладных задачах автоматического контроля состояния производства. Например, в данной работе рассматривается прикладная задача обнаружения движущегося вилочного погрузчика и выделение его основных атрибутов (направление движения, наличие/отсутствие груза на вилах) для автоматизации производственных линий в существующей автоматизированной системе управления.

Классический подход для решения задачи детектирования текстурированного объекта основывается на использовании популярных алгоритмов компьютерного зрения SIFT [1], SURF [2], ORB [3], FAST [4] и пр. [5]. Эти алгоритмы ищут ключевые точки на эталонном изображении объекта, вычисляют их дескрипторы и сравнивают их с дескрипторами кадра, полученными аналогичным образом. К сожалению, такой подход существенно зависит от качества изображения и часто показывают высокие вероятности ошибок I (False Reject Rate, FRR) и/или II рода (False Accept Rate, FAR), особенно при наличии на изображении помех (вариативное освещение, частичное выпадение объекта из кадра и т. п.).

Целью проведенного исследования является снижение влияния фактора зашумленности изображения на точность обнаружения движущегося объекта на видео [6] при помощи использования морфологических операций [7]. В этом случае силуэт движущегося объекта может быть получен методом MHI (Motion History Image) [8], после чего на основе полученной информации об объекте можно вычислить его ключевые характеристики, применяя дополнительные морфологические преобразования изображения [9, 10]

Работа организована следующим образом: во втором разделе формулируется задача обнаружения пустого движущегося погрузчика. В третьем разделе проводится экспериментальное сопоставление традиционных методов обнаружения и разработанного морфологического алгоритма и анализируется устойчивость точности к аддитивному шуму. В заключительном разделе представлены основные выводы, сделанные по результатам проведенного исследования.

Материалы и методология

Пусть имеется база данных (БД) модельных изображений детектируемых объектов и заданы их атрибуты (размер, предполагаемая скорость движения и пр.). Задача распознавания состоит в том, чтобы на входящем потоке видеоданных определить положение движущегося объекта, отчертив его ограничивающим прямоугольником и путем определения его ключевых признаков, отнести его к одному из искомых классов [6]. Нами рассматривается прикладной вариант такой задачи, в котором необходимо обнаружить движущийся вилочный погрузчик и определить его тип: пустой (рис. 1, а) или с грузом (рис. 1, б) [11]. Для указанной задачи известны следующие условия:

1) погрузчик движется в кадре горизонтально или вертикально с небольшими возможными отклонениями;

2) погрузчик является самым большим (занимает более половины кадра), но не единственным движущимся объектом на видео;

3) существуют два типа погрузчиков: вилочный и с одним длинным шестом для груза.

Поставленная задача традиционно решается методом поиска ключевых точек (алгоритм 1) на каждом изображении видеопоследовательности и сравнением их локальных дескрипторов с искомым изображением (SIFT [1], SURF [2]). Данный метод подразумевает устойчивость искомого объекта к аффинным преобразованиям, однако, исходя из особенностей задачи, где погрузчик движется только под углами 0° и 180°, сравнение

(а) Пустой вилочный погрузчик (б) Погрузчик с грузом

Рис. 1. Примеры изображений детектируемого объекта

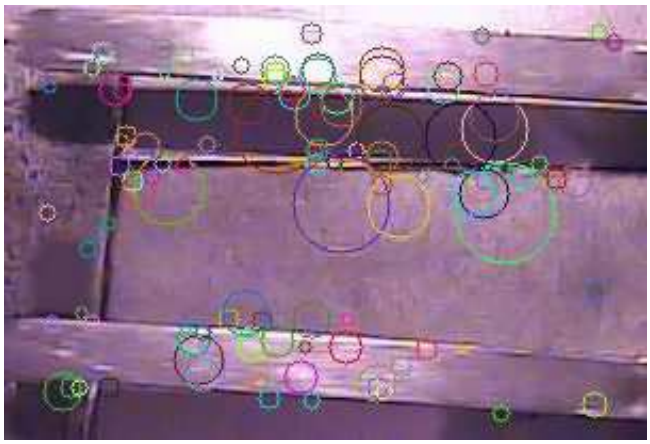


Рис. 2. Искомое изображение пустых вил с выделенными ключевыми точками

дескрипторов производится только с учетом горизонтальной или вертикальной ориентации объекта (в функции Compare). У данного алгоритма имеются следующие параметры: M_0 — количество совпадений дескрипторов эталонного и найденного на видео объектов, при котором эти объекты признаются идентичными; M_1 — порог количества кадров видео (в % от общего числа), на которых должен быть обнаружен объект, при достижении которого он считается найденным на видео; N_f — количество кадров с обнаруженным движущимся объектом; N_m — количество кадров, на которых был обнаружен искомый объект в процессе работы алгоритма. Функция Area отвечает за расчет площади переданной ей области. К сожалению, точность этого подхода (алгоритм 1) существенно зависит от качества входного видеосигнала. Поэтому уже небольшие помехи могут значительно ухудшить точность распознавания.

Для повышения качества обнаружения на зашумленных данных в настоящей работе предложено учитывать специфику задачи для разработки менее универсального, но более точного алгоритма. Используется информация о минимальном размере погрузчика относительно размера кадра (больше половины) — площадь также рассчитывается функцией Area, об отсутствии других объектов подобного размера и данные о том, что погрузчик движется на видео горизонтально или вертикально лишь с небольшими отклонениями. Предложенный алгоритм (алгоритм 2) состоит из двух частей. В первой части производится определение наличия движущегося объекта и направления его движения

Data: Последовательность кадров $\{X(t)\}$, $t = \overline{1,T}$, изображение пустых вил X_{trn}, Порог совпадений дескрипторов M_0, Порог количества кадров с обнаруженным объектом на видео M_1
Result: $True$ если пустые вилы найдены, $False$ иначе

```
kpts_trn := DetectAndCalc(X_trn); // поиск ключевых точек и расчет их дескрипторов для X_trn
N_m := 0, N_f := 0;
for t = 1, ..., T do
    // проверка области движения на соответствие минимальному размеру в 0.5 кадра
    if Area(frame_moving_region) > 0.5 * Area(frame) then
        N_f := N_f + 1; kpts_t := DetectAndCalc(t); // поиск ключевых точек и расчет их дескрипторов для t
        match := Compare(kpts_t, kpts_trn); // рассчитаем количество совпадающих дескрипторов
        if match > M_0 then
            N_match := N_match + 1;
        end
    end
end
if (N_match/N_f) > M_1 then
    return True;
else
    return False;
end
```

Algorithm 1: Принцип работы алгоритма локальных дескрипторов

на видео: последовательно идущий кадр $X(t)$ сравнивается с предыдущим X_{bg} методом MHI (применяется адаптивное пороговое преобразование методом Гаусса, где пороговое значение $T(x, y)$ есть взвешенная сумма blocksize×blocksize окрестности пикселя (x, y) — C)(рис. 4, а) [8]. Затем для обнаруженной движущейся области строится ограничивающий прямоугольник. Заметим, что для традиционного алгоритма, основанного на поиске ключевых точек и сравнении их дескрипторов, задача обнаружения движения не является необходимой, поскольку оно может быть извлечено из матрицы аффинного преобразования, получаемой вследствие работы алгоритма RANSAC [9].

На втором этапе выполняются операции обнаружения контуров в передней части обнаруженного движущегося объекта (который был отмечен ограничивающим прямоугольником) с помощью оператора Кэнни [12] (рис. 4, б). После этого строятся ограничивающие прямоугольники для больших контуров (размер определяется исходя из размера кадра и найденной на нем движущейся области, чтобы отсеять контуры помех), среди которых затем выбирается пара прямоугольников при помощи определенных критериев сравнения. Они классифицируются как ограничивающие прямоугольники пустых вил погрузчика (рис. 4, в).

Алгоритм требует установки следующих параметров: тип фильтрации, используемый при предварительной обработке изображения и размерность ядра этого фильтра; минимальный размер движущегося сегмента кадра на видео (по условиям технического зада-

```
   Data: Последовательность кадров {X(t)}, t = 1,T
   Result: True если пустые вилы найдены, False иначе
 1 X_bg := X(1); // Считаем самый первый кадр фоновым
 2 rectangles := 0, object_found := False, contours := 0;
 3 for t = 2,...,T do
 4 |   X_cur := X(t) − X_bg;
 5 |   Filter(X_cur);
 6 |   //применяем адаптивное пороговое преобразование к X_cur;
 7 |   X_g := ToGrayscale(X_cur);
 8 |   Filter(X_g);
 9 |   UpdateMotionHistory(X_g); //обновляем историю движений изображением X_g;
10 |   bound_rect = BoundingBox(X_g); // ищем контур движущегося силуэта и строим
   |   его ограничивающий прямоугольник
11 |   // проверка области движения на соответствие минимальному размеру
12 |   if Area(bound_rect) > 0.5 * Area(frame) then
13 |   |   DetectMoveDirection(X_g, X_bg); contours := Canny(X_g); //ищем всевозможные
   |   |   контуры в движущейся области
14 |   |   MorphClosure(X_g, rect, contours); //выполняем морфологическое замыкание
   |   |   контуров с прямоугольным элементом
15 |   |   for each cont in contours do
16 |   |   |   // размер контура должен быть больше определенного порога
17 |   |   |   if Size(cont) > threshold then
18 |   |   |   |   rect := BoundingBox(cont);
19 |   |   |   |   if motion.isHorizontal AND rect.width >
   |   |   |   |   rect.height OR motion.isVertical AND rect.width < rect.height then
20 |   |   |   |   |   rectangles+ = BoundingBox(cont);
21 |   |   |   |   end
22 |   |   |   end
23 |   |   end
24 |   |   //производим выборку ограничивающих прямоугольников по заданным
   |   |   критериям
25 |   |   if ∀i ∈ {1,...,5}C_i = true then
26 |   |   |   object_found := True;
27 |   |   end
28 |   end
29 |   X_bg := X(t);
30 end
31 if object_found then
32 |   return True;
33 else
34 |   return False;
35 end
```

Algorithm 2: Принцип работы морфологического алгоритма

ния — не менее половины кадра). Также требуется выбрать критерии отбора ограничивающих прямоугольников, которые проверяются на завершающем этапе работы алгоритма.

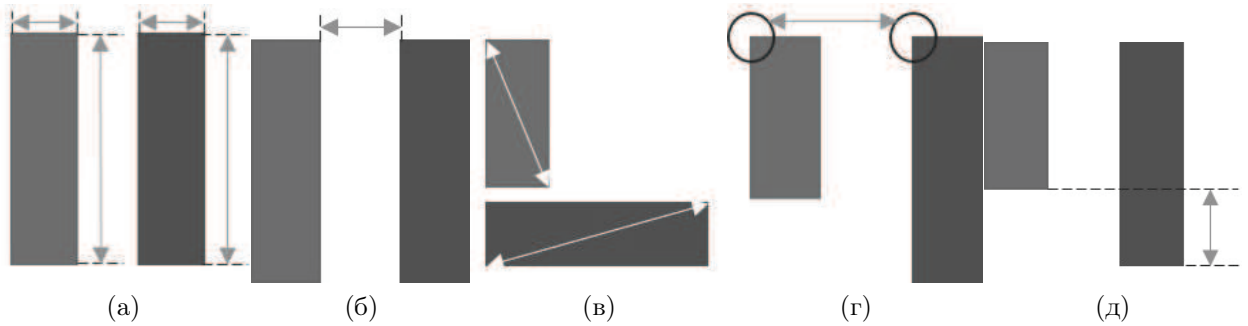
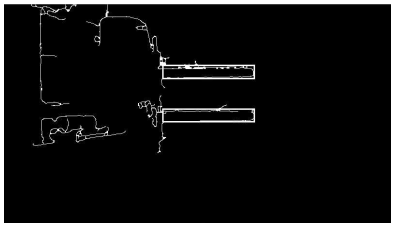


Рис. 3. Критерии отбора ограничивающих прямоугольников: (а)Width $> W_1$ AND Height $> H_1$; (б)Distance(Rect1, Rect2) $< D$; (в)|Area(Rect1) − Area(Rect2)| $< A$; (г)$R_2 <$ |Rect1.x − Rect2.x| $< R_1$; (д) |Width1 − Width2| $< W$

(а) Результат вычитания двух последовательных кадров

(б) Погрузчик с грузом



(в) Кадр с погрузчиком после морфологической обработки с обнаруженными вилами (помечены белыми прямоугольниками)

Рис. 4. Обработка кадра в алгоритме

Были реализованы следующие критерии отбора: (C_1) — минимальная длина и ширина прямоугольника должна быть больше $W_1 = const$ и $H_1 = const$, соответственно; (C_2) — максимальное расстояние между прямоугольниками (вилы не могут быть расставлены слишком широко) не больше фиксированной величины $D = const$; (C_3) — максимальная разность площадей прямоугольников (поправка на небольшое отклонение вил от горизонтального/вертикального расположения) должна быть меньше $A = const$; (C_4) — максимальное и минимальное отклонение расположения по осям (разница в расположении верхних левых углов по оси X при горизонтальном движении и оси Y при вертикальном движении) находится в диапазоне между числами $R_1 = const$ и $R_2 = const$; (C_5) — максимальная разность длин меньше $W = const$ (поправка на небольшой поворот либо частичное выпадение вил из кадра).

Результаты экспериментальных исследований

Популярные текстурные алгоритмы (SURF, SIFT, ORB и FAST), а также предложенный морфологический алгоритм были реализованы в прототипе программного продукта обнаружения вилочного погрузчика на видео для проведения сравнительного эксперимента. Для реализации использовалась библиотека OpenCV 2.4 [13]. На вход программе подается набор видеоданных, который затем обрабатывается любыми из указанных алгоритмов по выбору пользователя, после чего результаты работы программы выводятся в файл или непосредственно на экран. Агрегация результатов в процессе работы алгоритмов производится отдельно для каждого видео. Итоговая оценка качества работы алгоритмов определяются из показателей FRR и FAR обнаружения пустого погрузчика. Детектирование проводится независимо для каждого кадра. При оценке верного обнаружения (true detection) объекта для текстурных алгоритмов берется порог количества кадров M_1 по отношению к общему количеству кадров видео (алгоритм 1). Для морфологического алгоритма такой порог не используется — если объект был обнаружен хотя бы на одном кадре, он считается присутствующим на видео.

Пользовательский интерфейс разработанной программы позволяет выбирать различные типы фильтрации изображения и размер ядра накладываемого фильтра, задействовать или отключать любой из пяти заложенных алгоритмов при обработке видео, также есть возможность выбора любого количества видеоданных, как с указанием конкретных файлов, так и загрузкой папок.

Для проведения эксперимента было выбрано 24 видеофайла, с разрешением 1280×720, на каждом их которых имеется движущийся в определенном направлении вилочный погрузчик (см. рис. 1) (загружены из БД компании Intelligent Security Systems) [14]. На 10 видео погрузчик движется без груза, на 14 оставшихся — с грузом. Камера, с которой записывалось видео, расположена строго перпендикулярно к плоскости пола, где движется погрузчик. Съемка велась со стационарных камер видеонаблюдения в больших складских помещениях во время процессов загрузки/отгрузки товара в разное время суток. Этим обусловлено наличие таких помех, как резкое изменение освещения и затенения, частичное выпадение объекта из кадра, наличие множества иных движущихся в разные стороны объектов. Продолжительность видео — от 15 до 25 с, средняя частота кадров в секунду — 25. Погрузчик (с грузом или без него) является самым крупным движущимся объектом на видео.

Эксперимент был разбит на три части. В первой было оценено качество работы алгоритма, основанного на сравнении локальных дескрипторов ключевых точек. Было использовано четыре метода вычисления и сравнения дескрипторов (SURF, SIFT, ORB, FAST). FAR определяется как $N_{\text{fork\_miss}}/N_{\text{fork}}$, а FRR как $N_{\text{false\_det}}/N_{\text{no\_fork}}$, где $N_{\text{fork\_miss}}$ — количество видео, содержащих вилы с грузом, ошибочно распознанных как пустые, N_{fork} — количество видео с грузами, $N_{\text{false\_det}}$ — число видео, на которых пустые вилы ошибочно не были обнаружены, $N_{\text{no\_fork}}$ — количество видео с пустыми вилами. В эксперименте использовались различные значения параметров для каждого текстурного метода. Разрешение исходных видео данных: 1280×720 и 800×465 (искусственно уменьшено с исходного), разрешение изображения искомого объекта — 303×205 (вырезано из исходных данных).

В результате проведения первой части эксперимента были определены оптимальные значения параметров первого алгоритма, при которых алгоритмы показывают наименьшее количество FAR и FRR обнаружения пустых вил. Порог совпадений при сравнении дескрипторов M_0: SURF — 25, SIFT — 19, ORB — 40, FAST — 30. Рассмотренные значения порогов для SURF {16, 20, 25}, для SIFT {16, 20, 25}, ORB {16, 20, 25}, FAST {16, 20,

Таблица 1. Результаты работы алгоритма локальных дескрипторов

Значения	SURF	SIFT	ORB	FAST
Порог совпадений дескрипторов M_0	25	19	40	30
Порог кадров с обнаруженным объектом M_1, %	1	2	20	5
FRR, %	0	65	87	75
FAR, %	64	53	0	13
Время обработки кадра, мс	75,1	188,8	35,5	44,5

25}. Оптимальное значение порога M_1: SURF — 1%, SIFT — 2%, ORB — 2%, FAST — 5%. Рассмотренные значения порога: SURF {0.5%, 1%, 2%}, SIFT {1%, 2%, 4%}, ORB {10%, 20%, 25%}, FAST {1%, 5%, 7%}. При оптимальных значениях параметров были получены результаты, представленные в табл. 1.

Как видно, значения FRR и FAR велики даже при оптимальном подборе значении параметров. Это в первую очередь связано с тем, что пустые вилы на выбранных видео практически не выделяются из фона (из-за схожей цветовой насыщенности). Очевидно, низкое качество обнаружения и вычислительная эффективность традиционного подхода не являются удовлетворительными для его практического применения.

Во второй части эксперимента протестирована реализация морфологического алгоритма на качество обнаружения. Была выявлена зависимость точности алгоритма от условий отбора ограничивающих прямоугольников для контуров вил. Были рассмотрены следующие значения параметров: тип фильтрации предварительной обработки — медианная, гауссова, оконный фильтр и нормализованный оконный фильтр [9, 15]; размеры ядра фильтра — от 1 до 9; минимальный размер движущейся области кадра — 0,3 и 0,5. Экспериментально были определены значения параметров, при которых алгоритм показывает наибольшую точность обнаружения при неизменных критериях отбора прямоугольников: медианная фильтрация в окрестности 5×5, при размере движущейся области в 0,5 кадра. Наилучшее качество детектирования было получено для следующих значений параметров критериев отбора прямоугольников: $H_1 = 150$ (длина), $W_1 = 35$ (ширина), $D = 150$, $A = 1500$, $R_1 = 150, R_2 = 80$, $W = 20$. В результате проведения эксперимента было выявлено, что все названные критерии, за исключением максимальной разности площадей прямоугольников, при добавлении их как условий в процесс отбора существенно понижают FRR поиска пустых вил, лишь незначительно повышая значение FAR. Наименьшее FRR — 16% и 5% — FAR достигается при совместном использовании пяти критериев. Кроме того, разработанный алгоритм очень быстро обрабатывает каждый кадр — порядка 5 мс, что позволяет получать результаты в реальном времени. Зависимость морфологического алгоритма от критериев отбора ограничивающих прямоугольников контуров вил представлена в табл. 2.

В третьей части эксперимента проводится тестирование реализованных алгоритмов на устойчивость к шумам различного типа. Всего рассматривалось три типа помех, являющихся типичными для условий поставленной задачи: аддитивный белый гауссов шум, случайный импульсный шум и частичное осветление/затемнение кадра. Так, импульсные помехи могут возникать от тиристорных регуляторов и ламп дневного света или из-за токов питания устройств, участвующих в обработке сигнала (синхронные с сетью помехи). Белый гауссов шум может возникать при плохих условиях приема сигнала с камеры. Пе-

Таблица 2. Зависимость точности морфологического алгоритма от критериев проверки

Критерии проверки	Минимальная длина и ширина	Разность площадей	Максимальное удаление друг от друга	Максимальная ширина	Максимальная разность длин
FRR, %	0	0	7,5	11,8	16,3
FAR, %	78,9	78,9	34,1	15,5	5
Время обработки кадра, мс	4,8	4,9	4,9	5,1	5,2

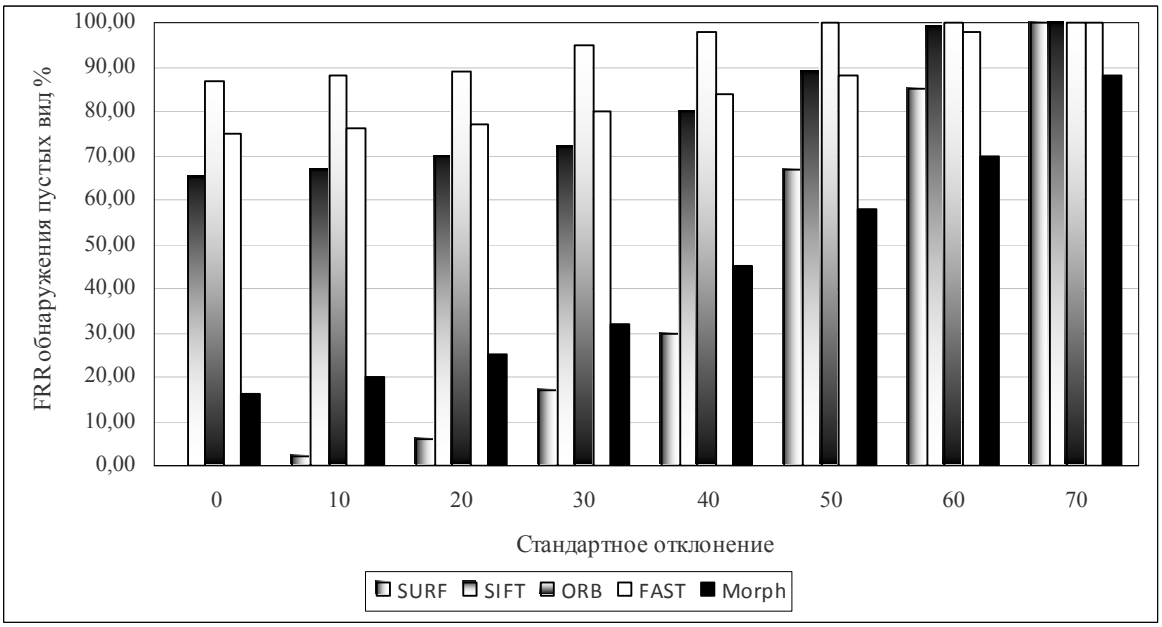


Рис. 5. Зависимость значения FRR от дисперсии аддитивного белого шума

репады освещения типичны при движении по неравномерно освещенным помещениям или когда источники света расположены не на потолке, а на стенах и освещают помещение под разными углами. Для получения гауссова шума генерируется нормально-распределенная случайная величина с нулевым средним и переменной дисперсией (параметр, характеризующий уровень шума), прибавляющаяся к каждому пикселю каждого кадра, затем значение итогового пикселя устанавливается в пределах [0, ..., 255]. Зависимость качества рассматриваемых алгоритмов (значения FRR и FAR) от дисперсии шума показана на рис. 5 и 6.

При использовании случайного импульсного шума (рис. 11), фиксируется один числовой параметр — количество модифицируемых пикселей изображения по отношению к общему числу пикселей одного кадра. Далее случайным образом выбирается координата модифицируемого пикселя и новое значение его яркости. Соответственно, чем больше количество изменяемых точек, тем сильнее помехи. Зависимость качества детектирования алгоритмов от этого параметра показана на рис. 7 (для значения FRR) и на рис. 8 (FAR).

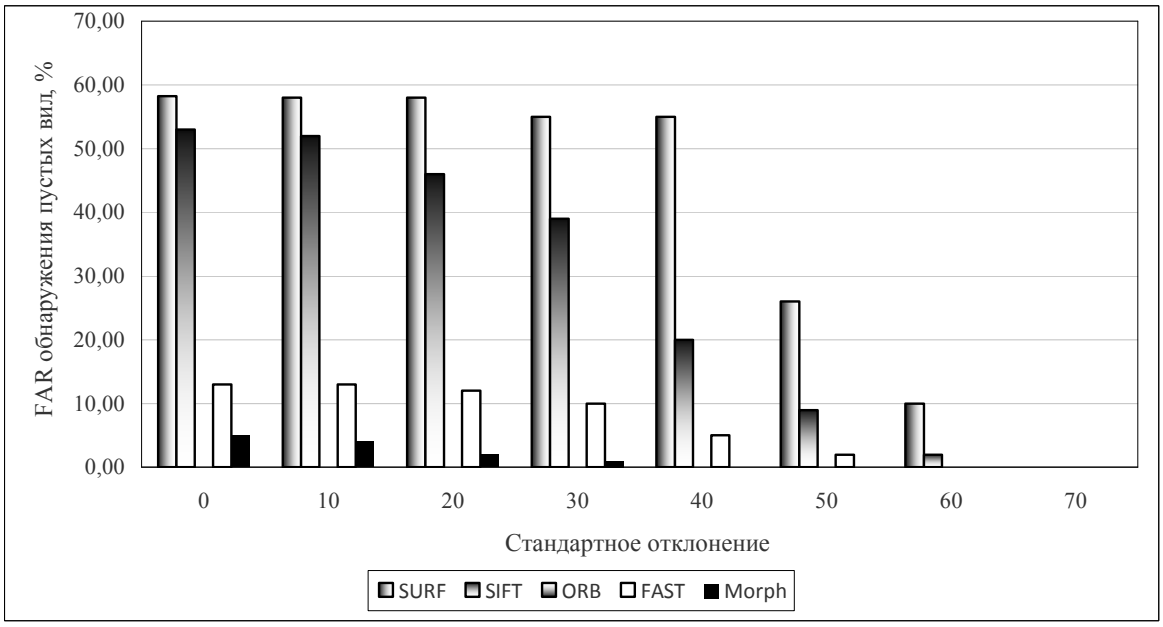


Рис. 6. Зависимость значения FAR от дисперсии аддитивного белого шума

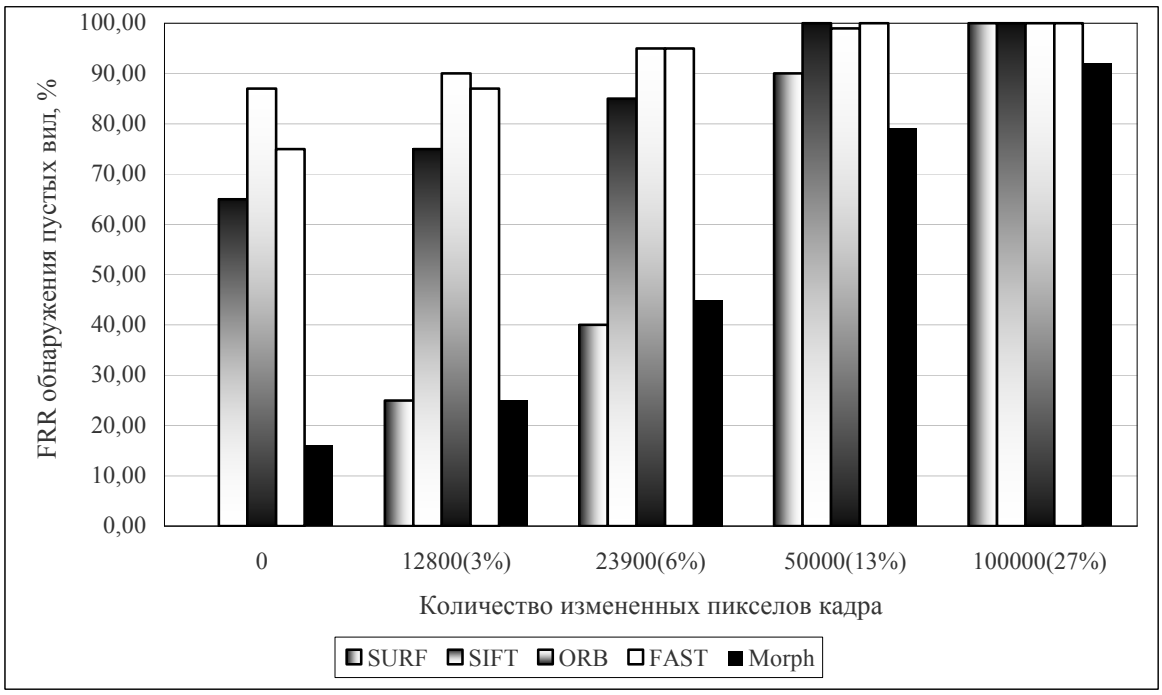


Рис. 7. Зависимость значения FRR количества пикселей с импульсным шумом

Для добавления перепадов освящения на видео использовалось динамическое увеличение яркости пикселей одной половины и ее понижение на другой половине кадра на фиксированную величину (интенсивность), которая и служит параметром этого шума. Динамика изменения значений FAR и FRR показана на рис. 9 (FRR) и на рис. 10 (FAR).

Результаты этого эксперимента подтверждают предположение о большей устойчивости к помехам предложенного алгоритма. Заметим, что при очень высоком уровне шума точность разработанного алгоритма (алгоритм 2) становится близкой к точности традицион-

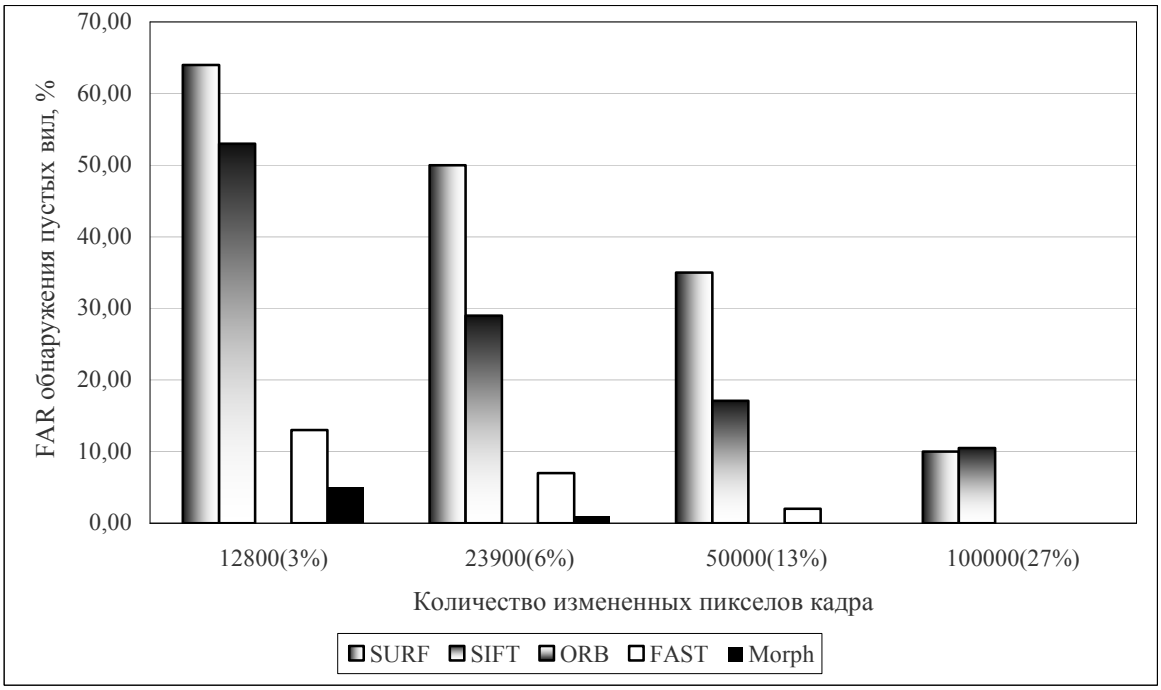


Рис. 8. Зависимость значения FAR от количества пикселей с импульсным шумом

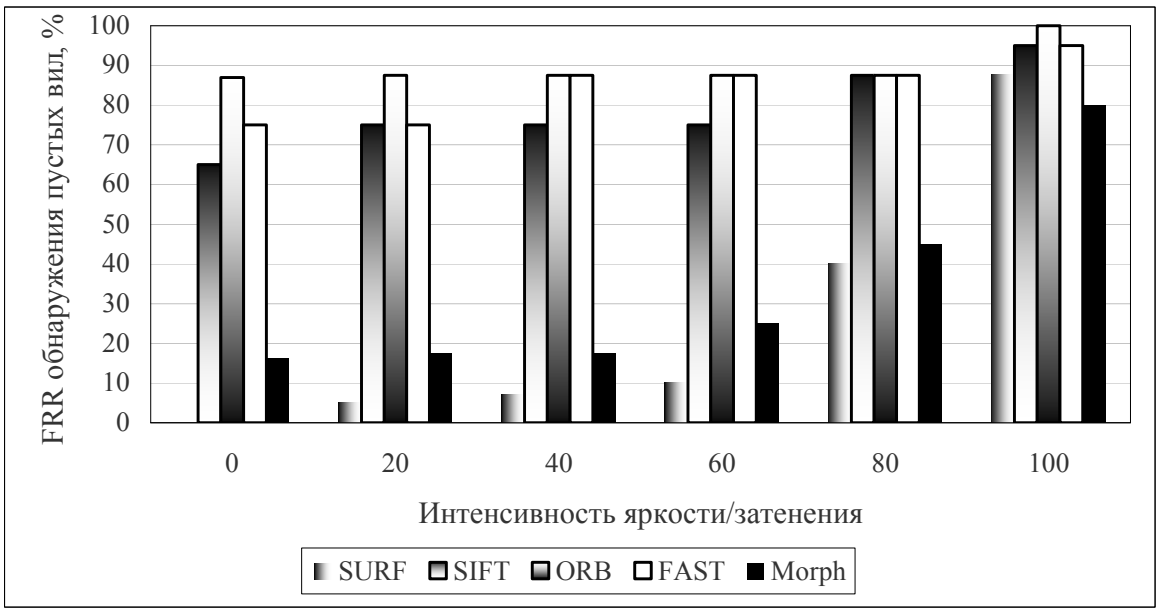


Рис. 9. Зависимость значения FRR от интенсивности яркости/затенения

ного подхода (алгоритм 1), что означает неудовлетворительное качество решения задачи при значительных помехах.

Заключение

В результате проведения исследования был разработан морфологический алгоритм обнаружения движущегося объекта на видео, который обеспечивает на 40% и 27% (разница между усредненными значениями FRR и FAR (FAR + FRR/2) текстурных алгоритмов и морфологического алгоритма при обнаружении пустых вил) более точный резуль-

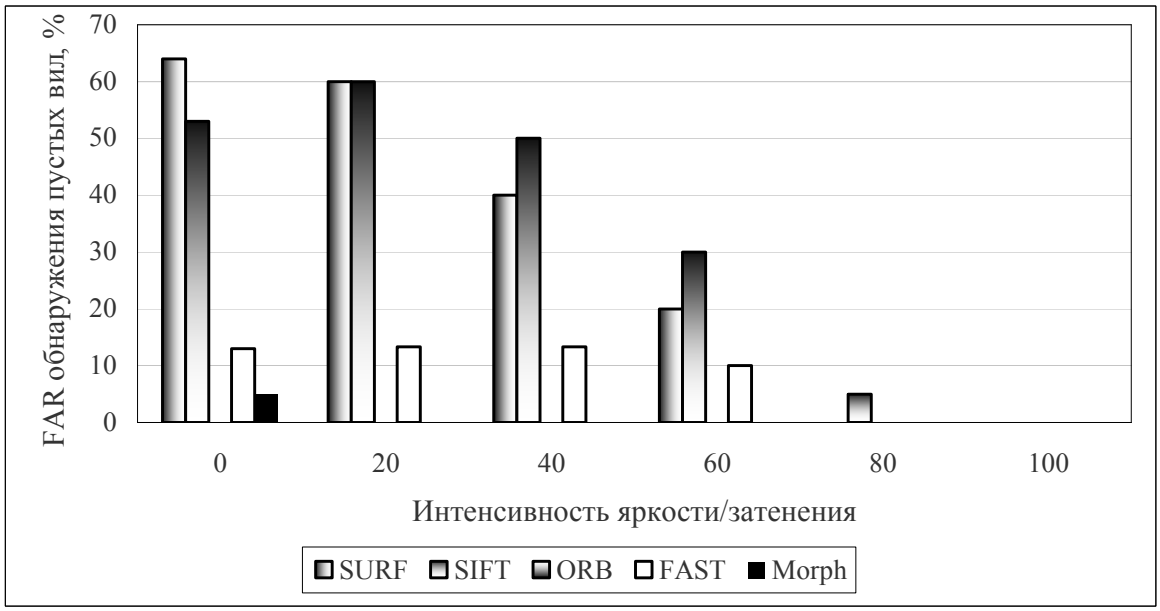


Рис. 10. Зависимость значения FAR от интенсивности яркости/затенения

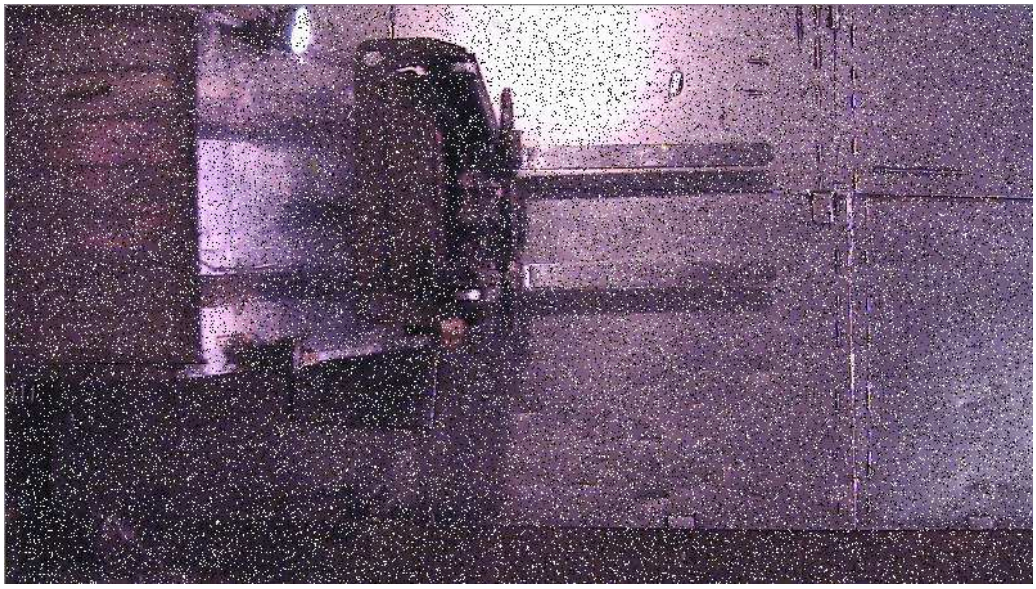


Рис. 11. Пример изображения погрузчика со случайным импульсным шумом

тат и в 7 раз (5 мс против 35 мс у ORB) более быструю обработку кадра [16], нежели традиционный метод, основанный на сравнении локальных дескрипторов ключевых точек. Экспериментально подтвержден факт сильного влияния помех вида игры света и тени, а также разного рода случайных помех на точность работы текстурных алгоритмов [1, 2, 3, 4] (рис. 5 и 7). Также было отмечено, что методы локальных дескрипторов неэффективны в плане скорости работы, когда изображение имеет много потенциальных ключевых точек и их приходится вычислять для каждого кадра. Оба недостатка были устранены в предложенном алгоритме, где игра света и тени обрабатывается оператором Кэнни [12], случайный шум удаляется фильтрацией и морфологическими преобразованиями. Производительность алгоритма повышена за счет использования только простых преобразований изображения.

Разработанный метод, однако, имеет существенный недостаток, так как во многом полагается на априорную информацию о специфике поставленной задачи (размеры движущегося объекта, наличие других объектов на видео и т. д.). Следовательно, его нельзя будет применить для обнаружения других типов объектов без внесения дополнительной логики работы в механизм распознавания. С точки зрения развития данной темы следующим шагом, на наш взгляд, может стать дальнейшее повышение точности алгоритма (с использованием комбинирования рассмотренных методов), а также его адаптация для других практически важных задач распознавания движущихся объектов в зашумленной среде.

Литература

[1] *Lowe D.* Distinctive image features from scale-invariant keypoints // *Int. J. Comput. Vision*, 2014. Vol. 60, no. 2. P. 91–110.

[2] *Bay H., Ess A., Tuytelaars T., Van Gool L.* SURF: Speeded up robust features // *Comput. Vision Image Understanding*, 2008. Vol. 110, no. 3. P. 346–359.

[3] *Rublee E., Rabaud V., Konolige K., Bradski G.* ORB: An efficient alternative to SIFT or SURF // *IEEE Conference (International) on Computer Vision (ICCV)*, 2011.

[4] *Rosten E., Drummond T.* Machine learning for high-speed corner detection // *European Conference on Computer Vision*, 2006. Vol. 1. P. 430–443.

[5] *Savchenko A. V.* Probabilistic neural network with homogeneity testing in recognition of discrete patterns set // *Neural Networks*, 2013. Vol. 46. P. 227–241.

[6] *Savchenko A. V.* Adaptive video image recognition system using a committee machine, optical memory and neural networks // *Optical Memory Neural Networks (Information Optics)*, 2012. Vol. 21, no. 4. P. 219–226.

[7] *Chien S.-Y., Ma S.-Y., Chen L.-G.* Efficient moving object segmentation algorithm using background registration technique // *IEEE Trans. Circuits Syst. Video Technol.*, 2002. Vol. 12, no. 7. P. 577–586.

[8] *Ahad Md., Atiqur Rahman, Tan J. K., Kim H., Ishikawa S.* Motion history image: its variants and applications // *Machine Vision Appl.*, 2012. Vol. 23, no. 2. P. 255–281.

[9] *Shapiro L. G, Stockman G. C.* Computer vision. Prentice Hall, 2001.

[10] Mathematical morphology: From theory to applications. / Eds. L. Najman, H. Talbot. Wiley-ISTE, 2010.

[11] ISS (Intelligence Secure Systems). URL: `http://www.iss.ru/`

[12] *Canny J. A.* Computational approach to edge detection // *IEEE Computer Society*, 1986. P. 679–698.

[13] OpenCV library. URL: `http://opencv.willowgarage.com/wiki/`

[14] ISS video dataset. URL: `ftp://isstemp:isstemp@ftpsupport.iss.ru/Loaders/Video/`

[15] *Sonka M., Hlavac V., Boyle R.* Image processing, analysis, and machine vision. 4th ed. Cengage Learning, 2014.

[16] *Savchenko A. V.* Directed enumeration method in image recognition // *Pattern Recognition*, 2012. Vol. 45, no. 8. P. 2952–2961.

Установившиеся режимы в модели Хёнинга и ее модификациях*

***В. Л. Макаров*[1], *Л. А. Бекларян*[1], *Ф. А. Белоусов*[1,2]**
beklar@cemi.rssi.ru
[1]ЦЭМИ РАН, Москва, Нахимовский пр., 47; [2]НИУ ВШЭ, Москва, ул. Мясницкая, 20

Рассматривается модель Хенинга поведения популяции и ее модификации. Приводятся модификации, в которых преодолеваются некоторые недостатки модели Хенинга, связанные с эффектом гибели популяции в результате внутрипопуляционных противоречий. Эта тема является важной для изучения, так как подобные явления наблюдаются как в дикой природе, так и в истории человеческой цивилизации. Определяется также модель, в которой, в отличие от модели Хенинга и его модификаций, взаимодействие между агентами определяются эндогенно, т. е. взаимодействие, основанное на реакции типа «инстинкт», заменяется на взаимодействие с использованием элементов «этики».

Ключевые слова: *агенто-ориенированное моделирование; модель Хенинга; вымирание популяции; фактор мести; ассиметрия во взаимоотношениях*

Steady regimens in Henning model and its modifications*

***V. L. Makarov*[1], *L. A. Beklaryan*[1], *and F. A. Belousov*[1 2]**
[1]CEMI RAS, Moscow, Nachimovsky prospect 47; [2]NRU HSE, Moscow, Myasnutskaya str., 20

Background: This paper is based on work of Peter A. Henning published in Lecture Notes in Economics and Mathematical Systems in 2008. In his model, Henning explores an effect of death of populations owing to intrinsic causes. This subject is interesting to study, since such phenomena may be observed both in unexplored wilderness and in human civilization. Degree of survival of population depends mainly on level of aggression between agents in this population. There are many papers on interspecific aggression. In particular, it was considered in one of the modifications of well-known Sugar model. One can highlight another paper of S.Younger published in Journal of Artificial Societies and Social Simulation in 2005, in which interspecific aggression and, in particular, revenge are also considered. Other works studying positive interspecific influence may also be mentioned. For example, there are papers where populations with altruistic agents are considered. These questions are studied by S. Bowles and E. Blume.

Methods: This paper consists of two parts. The first one is dedicated to consideration of Henning model and its modifcations. In modifications of the model, the authors try to overcome some disadvantages which initial Henning model has. In particular, such factors as revenage and asymmetry are considered. In the second part, the same questions are considered by using another model, construction of which differs significantly from Henning model and to a greater extent, it is similar to already mentioned Sugar model.

Results: The main distinction of the second model from the first one is endogeneity of behavior rules between agents. In other words, if in Henning model and its modifications the rules of interaction between agents are determined randomly, in the second model these interaction rules are determined based on conditions of agents and conditions of environment.

*Работа выполнена при финансовой поддержке РФФИ, проект № 12-01-00768.

Concluding Remarks: In modifications of Henning model, the factors of revenage and asymmetry are studied. It was shown that if these factors are not included in the model, then death of population is not observed. In the second part, where behavior between agents is determined endogeneously, many other interesting regularities are found.

Keywords: *agent-based modeling; Henning model; population death; revenge factor; asymmetry in relationship*

Введение

Работа посвящена изучению динамики развития популяций в зависимости от разных типов взаимодействий между особями внутри популяций, а также отдельно изучаются условия, при которых наблюдается эффект вымирания таких популяций. В первой части статьи такое исследование делается на основе модели Хенинга [1], а также ее модификаций, для которых будут характерны различные правила поведения между особями, а именно: будут изучаться влияния таких факторов, как внутрипопуляционная агрессия и, в частности, месть. Во второй части эти же вопросы будут изучаться на основе другой модели, которая по своей конструкции сильно отличается от модели Хенинга и в большей степени перекликается с хорошо известной «Сахарной моделью» [2]. Тем не менее в «Сахарной модели» тема вымирания популяций не поднимается. По своей структуре эти модели имеют много общего, однако они заметно отличаются. Отличием является то, что рассмотренная здесь модель является более простой по сравнению с «Сахарной моделью», что позволяет в явном виде описать механизм, благодаря которому происходит вымирание популяции.

Рассматриваемая в статье вторая модель отличается от первой эндогенностью правил взаимодействия между особями в популяции, т. е. если в модели Хенинга и ее модификациях взаимодействия между агентами определяется случайным образом, то во второй модели эти правила определяются на основе состояния агентов и состояния внешней среды.

Основной целью работы является исследование условий, при которых наблюдается вымирание популяции вследствие внутрипопуляционных противоречий между особями. Интуитивно понятно, что в ситуации, когда особи в большей степени негативно влияют друг на друга вероятность вымирания популяции увеличивается. Интересно построить такую модель, в которой внутри популяции положительное и негативное влияние особей друг на друга сбалансированы, но при этом эффект вымирания популяции сохраняется. В статье авторы попытались построить такую модель.

Вопрос внутривидовой агрессии представляет немалый научный интерес. Этой теме посвящен целый ряд научных работ. В частности внутривидовая агрессия была затронута в одной из модификаций упомянутой выше «Сахарной модели» [2]. Кроме этого можно выделить статью S. Younger [3], в которой также изучаются вопросы насилия и отдельно исследуется фактор мести. Однако вопрос вымирания целых популяций в подобных работах либо вообще не затронут, либо ему не уделяется какого либо значимого внимания.

Изучение вопроса вымирания популяций можно также найти в так называемых эпидемиологических моделях, в которых рассматривается распространение эпидемии по некоторой территории, населенной агентами. Однако эпидемию в такой ситуации можно расценивать как некоторый внешний негативный фактор. Здесь же важно смоделировать эффект вымирания популяции за счет исключительно внутренних причин.

Другой важной особенностью модели, которая представлена во второй части работы, является то, как агент определяет свое отношение к находящемуся рядом агенту. В рас-

сматриваемом случае оно определяется благодаря тому, какой инстинкт доминирует в сознании агента. В предложенной здесь модели будут два инстинкта — это инстинкт потребления (чтобы не умереть с голоду) и инстинкт размножения. Если в особи доминирует инстинкт потребления (т. е. особь голодна), то она рассматривает другие особи в качестве конкурентов и по возможности атакует их. С другой стороны, если особь не голодна, то начинает доминировать инстинкт к размножению, в результате чего другие особи рассматриваются как партнеры для рождения новой особи. В уже указанных работах «Сахарной модели» и работах S. Younger также рассматривается вражда между особями, но возникающая по социальным причинам, т. е. из-за того, что они принадлежат либо к разным социальным группам, либо к разным племенам. Другими словами, принципиальным различием является то, что отношение агентов друг к другу в представленной здесь модели может меняться несколько раз в течение их жизни, тогда как в других моделях оно не меняется либо может меняться, скажем, один раз за жизнь агента.

Можно также отметить работы, которые изучают наличие положительных внутрипопуляционных взаимодействий между особями. Например, изучаются популяции, в которых есть агенты, способные на альтруистические поступки. Среди работ, в которых рассматриваются указанные вопросы, можно выделить статьи С. Боулза [4, 5], а также работу E. Blume [6]. При этом стоит отметить, что основным методом исследования в работе E. Blume [6] является широко распространенный теоретико-игровой подход. В таких моделях поведение агентов определяется исходя из условий равновесия (в частности, равновесия по Нэшу). Однако среди существенных недостатков использования такого метода можно отметить так называемое усреднение, которое предполагает идентичность всех агентов по всем параметрам, что, очевидно, далеко от реальности. Данного недостатка лишен агенто-ориентированный подход, который используется авторами представленной работы.

Общий вывод из большинства работ достаточно интуитивен — чем больше негативного во взаимоотношениях между особями, тем менее жизнеспособна популяция в целом. И наоборот — чем больше благотворного влияния особи оказывают друг на друга, тем более живучей оказывается популяция. В этой статье акцент будет сделан не только на влияние положительных и негативных факторов на живучесть популяции, но и влияние остальных факторов, относящихся как к характеристикам самой популяции, так и характеристикам окружающей среды.

Модель Хенинга

Описание Модели Хенинга. Модель Хенинга характеризуется вектором состояний Ω, который представляет из себя n-мерный вектор, элементы которого могут принимать значения либо 0, либо 1:

$$\mathbf{\Omega} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix},$$

где $x_i \in \{0, 1\}$, $i = \overline{1, n}$.

Если $x_i = 0$, то i-й агент мертв, если $x_i = 1$, то агент считается живым. Из периода в период состояние системы меняется — какие-то агенты умирают, какие-то рождаются. Состояние системы в момент времени k обозначим через Ω^k.

Опишем алгоритм перехода из состояния Ω^{k-1} в состояние Ω^k. В нулевой период предполагается, что все агенты живы, т. е. Ω^0 есть n-мерный вектор, все элементы которого равны 1. Вводится в рассмотрение $(n\times n)$-матрица перехода M^k, которая в нулевой период совпадает с единичной матрицей.

Опишем процедуру, по которой осуществляется переход от матрицы M^{k-1} к M^k. В каждой строке матрицы M^{k-1} выбирается один элемент и заменяется на некоторую случайную величину, распределенную равномерно на интервале от -1 до 1. Так, матрица M^1 примет вид:

$$M^1 = \begin{pmatrix} 1 & \cdots & \alpha_{1,j_1} & \cdots & 0 \\ \vdots & 1 & \cdots & \alpha_{2,j_2} & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_{n,j_n} & \cdots & 1 \end{pmatrix},$$

где для всех $i \in \{1, ..., n\}$ случайно выбирается целое значение $j_i \sim \overline{U}[1, n]$, а также некоторое значение $\alpha_{i,j_i} \sim U[-1, 1], i = \overline{1, n}$. Здесь $U[-1, 1]$ — равномерное распределение в диапазоне $[-1, 1]$, а $\overline{U}[1, n]$ — целочисленное равномерное распределение, в соответствии с которым случайным и равновероятным образом выбирается целое число из диапазона от 1 до n. Описанную процедуру обозначим через F, т. е. для любого $k \in \mathbb{N} \cup \{0\}$ будет справедливо $F(M^{k-1}) = M^k$.

Далее вычисляется промежуточный параметр $\omega^k = M^k\Omega^k$. Тогда вектор состояний Ω^{k+1} будет определяться по правилу

$$\Omega_j^{k+1} = \begin{cases} 1 & \text{если } \omega_j^k > 0\,; \\ 0 & \text{если } \omega_j^k \leqslant 0\,. \end{cases}$$

Таким образом элемент M_{ij}^k матрицы M^k, $k = 1, 2, ...$ характеризует воздействие j-го агента на агента i в период k. Если $M_{ij}^k > 0$, то воздействие на агента i положительно, если $M_{ij}^k < 0$, то воздействие отрицательно. Итоговое воздействие на агента i в k-й период определяется по формуле $\sum\limits_{j=1}^{n} M_{ij}^k\Omega_j^k$. Формально итоговое воздействие на агента i это произведение i-й строки матрицы M^k на вектор состояний Ω^k. Если это произведение больше 0, то особь либо выживает, либо рождается. В противном случае особь считается мертвой.

Описанный выше алгоритм преобразования матрицы M^{k-1} в матрицу M^k означает, что в каждый период времени изменение воздействия на любого агента $i = \overline{1, n}$ происходит со стороны другого, случайно выбранного агента j_i, на случайную величину α_{i,j_i}.

Определим также такие понятия, как установившийся режим и равновесная численность популяции.

Установившимся режимом или, что то же самое, равновесным состоянием развития популяции будем называть такую динамику изменения количества особей в популяции, при которой это количество варьируется вокруг некоторого постоянного значения. Равновесной численностью популяции, соответственно, будет обозначаться такое постоянное значение, вокруг которого изменяется численность популяции.

Результаты работы модели Хенинга. Был проведен ряд численных реализаций по модели Хенинга. Главный результат этих реализаций состоит в том, что, вопреки утверждениям из [1], в рамках этой модели, при достаточно большом значении максимального количества особей популяции (т. е. при достаточно большом значении параметра n), полного вымирания популяции не наблюдается. На рис. 1 и 2 приведена одна из реализаций модели Хенинга в случае, когда $n = 100$, $T = 5000$.

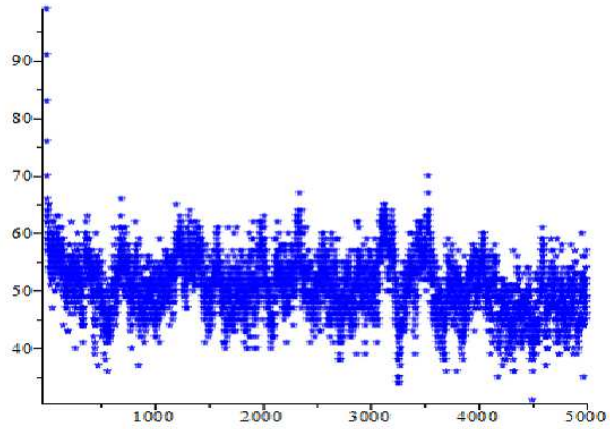
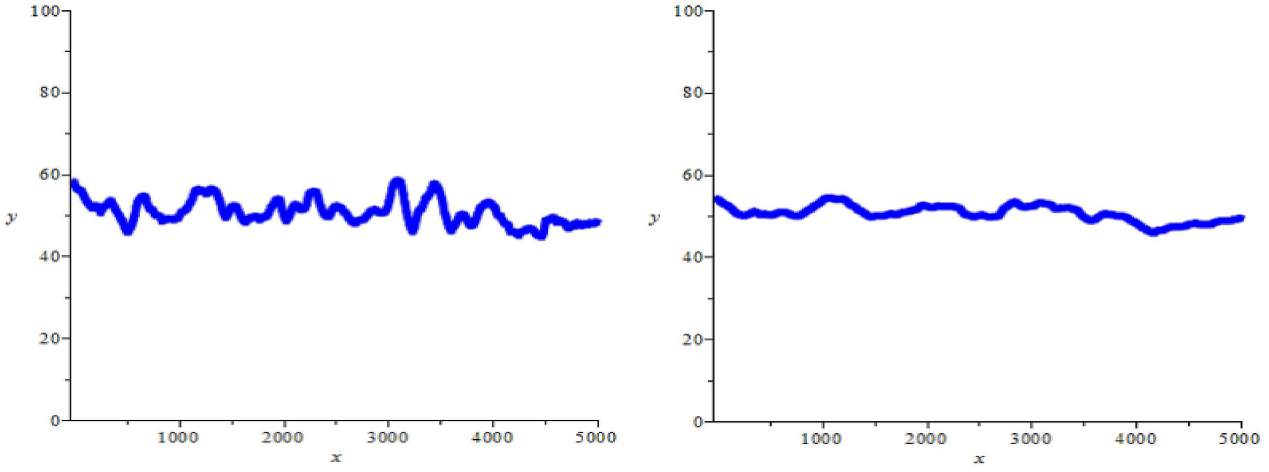


Рис. 1. Результат работы модели Хенинга в случае $n = 100$, $T = 5000$



(а) Усреднение за 100 периодов (б) Усреднение за 400 периодов

Рис. 2. Результат работы модели Хенинга в случае $n = 100$, $T = 5000$

Модификация модели Хенинга. Введение антисимметрии. Для получения эффекта вымирания всей популяции введем в модель Хенинга некоторые изменения, а именно: в процедуре F, которая была определена при описании исходной модели Хенинга, случайные величины α_{i,j_i} имеют распределение $U[-1, 1]$. Заменим это распределение на другое $U[-1, a]$, где $a \leqslant 1$. Оказалось, что величина параметра a является решающим для динамики популяции. Можно экспериментально определить максимальное значение a, при котором наблюдается эффект вымирания популяции. Численные эксперименты показали, что такое значение равно $a = 0,74$. На рис. 3 и 4 приведена одна из реализаций этой модели в случае $n = 100$, $T = 5000$.

Таким образом, видно, что одним из способов реализации эффекта вымирания популяции является введение антисимметрии во взаимоотношениях между особями путем увеличения количества негативных воздействий по сравнению с количеством положительных воздействий.

Модификация модели Хенинга. Введение фактора мести. В этом разделе рассмотрим другую модификацию модели Хенинга. Дадим агентам возможность реаги-

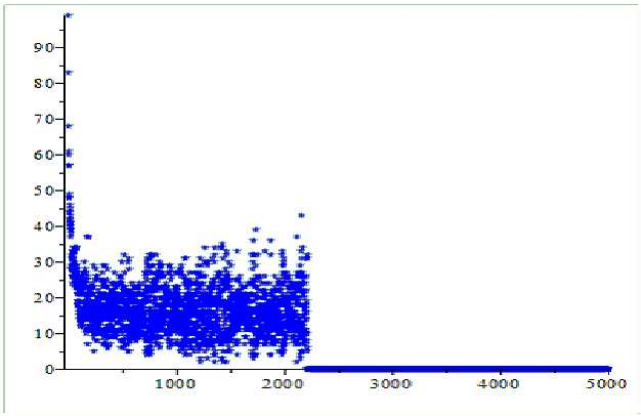
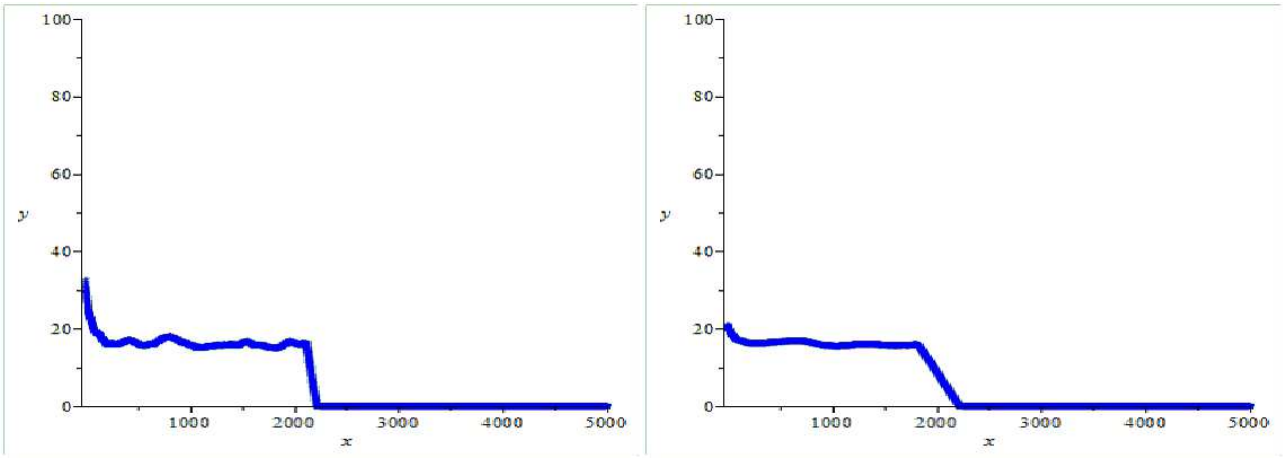


Рис. 3. Результат работы модели Хенинга с антисимметрией в случае $n = 100$, $T = 5000$



(а) Усреднение за 100 периодов (б) Усреднение за 400 периодов

Рис. 4. Результат работы модели Хенинга с антисимметрией в случае $n = 100$, $T = 5000$

ровать на оказываемое на них воздействие. Процедура F, описанная выше, останется без изменений, а именно: в каждый момент времени k в каждом столбце i матрицы M^{k-1} произвольно выбранный элемент M^{k-1}_{i,j_i} заменяется на случайную величину $\alpha_{i,j_i} \sim U[-1,1]$. Далее, введем дополнение в эту процедуру. А именно: после ее выполнения выбираются только отрицательные значения α_{i,j_i}, $i = \overline{1,n}$. Если $\alpha_{i,j_i} < 0$, то на место симметричного элемента $M_{j_i,i}$ будет ставиться случайно реализованная величина, распределенная по закону $U[-1,0]$. Таким образом, если некоторый агент оказывает негативное воздействие на другого агента, то этот агент также окажет негативное воздействие на первого агента. То есть образуется популяция из мстительных агентов. На рис. 5 и 6 представлены результаты одной из реализаций такой модели.

Эксперименты показывают, что в среднем популяция с мстительными агентами живет меньше периодов по сравнению с популяцией с антисимметрией во взаимоотношениях. Помимо этого можно отметить, что в случае наличия фактора мести, по сравнению со случаем наличия антисимметрии, динамика развития популяции быстро выходит на установившийся режим.

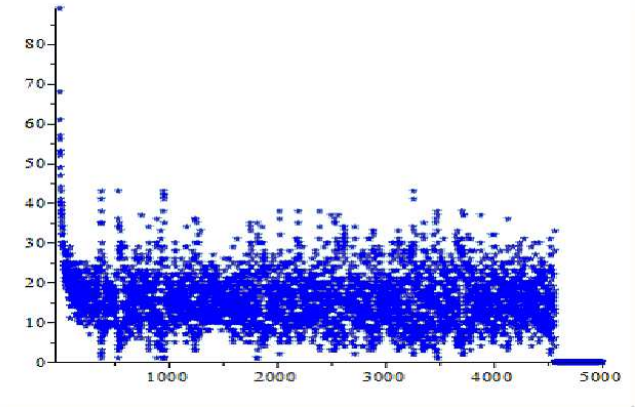
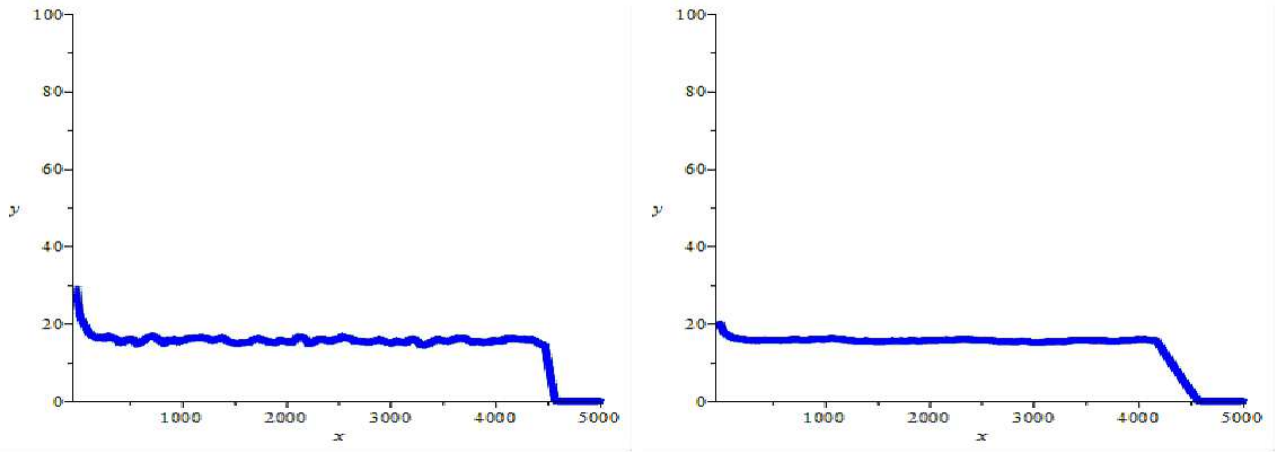


Рис. 5. Результат работы модели Хенинга с мстительными агентами в случае $n = 100$, $T = 5000$



(а) Усреднение за 100 периодов (б) Усреднение за 400 периодов

Рис. 6. Результат работы модели Хенинга с мстительными агентами в случае $n = 100$, $T = 5000$

Помимо этого можно отметить, что в случае наличия фактора мести популяция быстрее выходит на установившийся режим развития по сравнению со случаем наличия антисимметрии.

Выводы, связанные с модификациями модели Хенинга. Численные эксперименты показали, что в исходной модели Хенинга популяция достаточно быстро начинает развиваться вокруг некоторого установившегося режима и отсутствует эффект полного вымирания. Для получения эффекта полного вымирания популяции приходится вводить некоторую антисимметрию во взаимоотношениях так, чтобы негативного внутрипопуляционного взаимодействия было больше, чем положительного.

Важной особенностью данных моделей является тот факт, что взаимодействие агентов формируется экзогенно с помощью реакции типа «инстинкт». В модели, которая представлена ниже, эта особенность будет изменена, т. е. взаимодействие агентов будет формироваться эндогенно с помощью реакции с элементами «этики».

Модель с эндогенно заданными внутрипопуляционными взаимодействиями

Описание модели. В модели Хенинга и рассмотренных модификациях взаимодействия между особями формируются экзогенно. Другими словами, в модели не определено почему агенты начинают либо хорошо относиться к другим особям, либо враждебно. Это определяется случайным образом. Для преодоления этого недостатка рассмотрим модель, в которой эндогенно задается внутрипопуляционное взаимодействие между особями. Для этого вводится некоторый ограниченный ресурс, за который впоследствии может быть развернута конкурентная внутрипопуляционная борьба. В частности, правила, по которым особи могут вести себя агрессивно по отношению к своим сородичам, определяются эндогенно.

Разобьем описание модели на несколько этапов.

Основные элементы и характеристики модели. Модель описывается ($\dim\times\dim$)-матрицей, каждый элемент которой может принимать значения трех типов — больше нуля, равное нулю и отрицательное значение, равное -2. Опишем каждый тип в отдельности:

— если элемент равен 0, то поле считается пустым;
— если элемент — положительное число, то в данный период поле занято агентом, число характеризует уровень его здоровья;
— если значение поля равно -2, то это значит, что на нем появился (вырос) ресурс, потребляя который, агент увеличивает уровень здоровья на 2. Если же агент в какой-то период не потребляет этот ресурс, то уровень его здоровья уменьшается на 1.

Состояния агентов. В каждый период агенты ходят поочередно. Чем раньше особь родилась, тем раньше ей достанется право хода. Перемещаться агенты могут на одну клетку в любом направлении в пределах матрицы. У агентов есть два основных инстинкта, инстинкт потребления и инстинкт размножения. Какой инстинкт в данный момент преобладает, определяется состоянием, в котором находится тот или иной агент. Так, если агент не потреблял ресурс 5 или более периодов, то он считается голодным и инстинкт потребления для него является доминирующим, тогда как инстинкт к размножению отключается. В противном случае, если особь потребляла ресурс менее чем 5 периодов тому назад, инстинкт к размножению у нее доминирует над инстинктом к потреблению.

Взаимодействие между агентами. В рамках этой модели протестированы два типа взаимодействия между агентами. Условно их можно разделить на агрессивное поведение и неагрессивное поведение. На рис. 7 и 8 приведены блок-схемы алгоритмов, которым следуют неагрессивные и агрессивные особи соответственно. В случае модели с агрессивным поведением агентов внутрипопуляционное взаимодействие можно определить как симметричное, так как в зависимости от состояния агенты либо рожают новую особь, либо голодный агент атакует своего соседа (см. рис. 8), расценивая его как конкурента на ограниченный ресурс. При этом, атакуя, агент убивает свою жертву. Устанавливается также дополнительное ограничение — у двух агентов может родиться новая особь, только если в предыдущие 5 периодов обе особи не рожали других особей. Более того, в период репродуцирования уровень здоровья агентов уменьшается на 1,5 (а не на 1, как в обычном состоянии, если этот агент не принимал пищи). Новорожденная особь в период своего рождения не имеет права на ход.

Правило появления нового ресурса. В модели ресурс появляется с заданным темпом роста, т.е. каждый период на поле вырастает некоторое количество новых ресурсов, местопо-

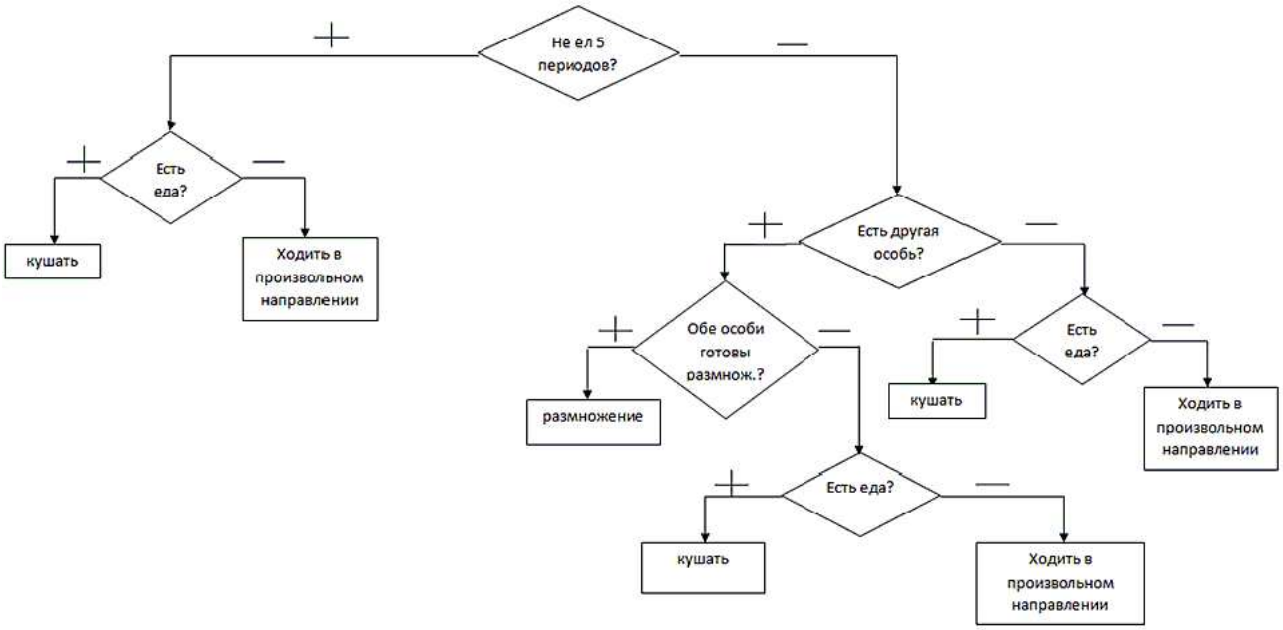


Рис. 7. Блок-схема алгоритма поведения агентов при отсутствии агрессивной составляющей

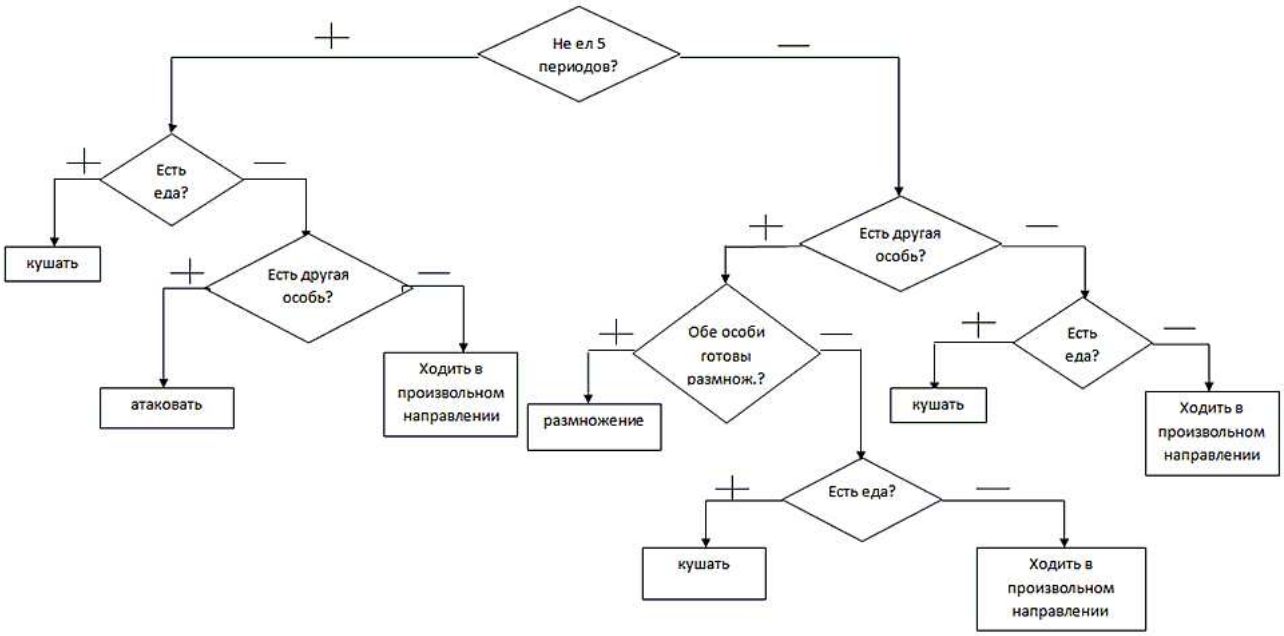


Рис. 8. Блок-схема алгоритма поведения агентов с агрессивной составляющей

ложение которых определяется случайно. В дальнейшем, этот параметр вне зависимости от размерности поля будет соответствовать появлению ресурса в 4 клетках матрицы на каждые 100 клеток (т. е. если матрица 20×20, то в этом случае ресурс появится в 16 клетках из 400).

Приведем таблицу основных характеристик модели:

1. Размерность поля (dim).
2. Начальный уровень здоровья.
3. Тип поведения агентов (агрессивный/неагрессивный).

4. Темп роста ресурса (каждый период на 100 клеток будет вырастать 4 единицы ресурса).
5. Пороговое значение голоден/сыт (5 или более периодов без ресурса = голоден).
6. Предельно возможный возраст агента (40 периодов).
7. Условия, при которых агенты могут размножаться.
8. Начальное количество агентов.

В данной работе акцент будет сделан на изучении влияния первых трех характеристик. Отметим также, что значения последней характеристики условно можно разделить на два класса. Начальное количество агентов слишком мало и популяция вымирает не успев достичь своего равновесного уровня развития и начальное количество агентов достаточно велико и популяция достигает своего установившегося режима развития.

Сравнение по типам взаимодействия агентов. Проведем сравнение динамики развития популяции для двух случаев взаимодействия агентов: агрессивное поведение агентов и неагрессивное поведение агентов. Остальные значимые параметры оставим фиксированными. А именно, размерность поля dim=10, начальный уровень здоровья положим равным 20. Результаты отражены на рис. 9.

Видно, что в первом случае популяция достаточно быстро выходит на некоторый установившийся режим развития и ее численность в среднем колеблется между 30 и 40. В случае же с агрессивным поведением агентов, видно, что численность популяции в среднем плавно снижается и в конце концов падает до нуля. Другие эксперименты также показали, что популяция с неагрессивным поведением агентов, при достаточном начальном количестве особей, не вымирает. В дальнейшем рассматриваются модели только с агрессивным поведением агентов и определяются пороговые значения параметров, при которых популяция начинает вымирать через какое-то количество периодов.

Влияние размерности поля. В этом разделе изучено влияние изменения размерности поля на живучесть популяции с агрессивным поведением агентов. На рис. 10 показан график динамики развития популяций, которые живут в поле размерности dim = 10, 14, 15, 16 и 20. При этом начальный уровень здоровья при рождении у особей везде равен 20.

Получаются достаточно интересные результаты. Так, видно, что популяция начинает выживать при размерности больше 15 (dim > 15). Чем выше размерность, тем больше продолжительность жизни популяции. Однако в данной реализации можно наблюдать, что при dim = 14 популяция получилась более живучей, чем при dim = 15. Более того, этот результат достаточно устойчив. То есть данная модель демонстрирует немонотонную зависимость живучести популяции (продолжительности жизни) от размерности. Далее, на рис. 11 приведем тот же график, однако теперь ось Y будет характеризовать плотность агентов, т. е. это величина равная количеству особей, деленное на количество всех полей в матрице ($\dim^2$).

Здесь следует обратить внимание на динамику популяций при dim = 16 и 20. Видно, что плотность расселения особей в этих случаях колеблется вокруг уровня 10%. Не исключено, что при прочих равных параметрах значение плотности расселения агентов является некоторым инвариантом относительно изменения размерности. Интересно также отметить, что популяции, которые вымирают, отклоняются от этой величины. В случае dim = 10 и 15 — это отклонение вниз, а в случае dim = 14 — это отклонении вверх.

Влияние начального уровня здоровья. Здесь проведено сравнение живучести популяции в зависимости от начального уровня здоровья при агрессивном поведении агентов. Сравнение сделано в случае, когда dim = 10, и в случае, когда dim = 20. В обоих

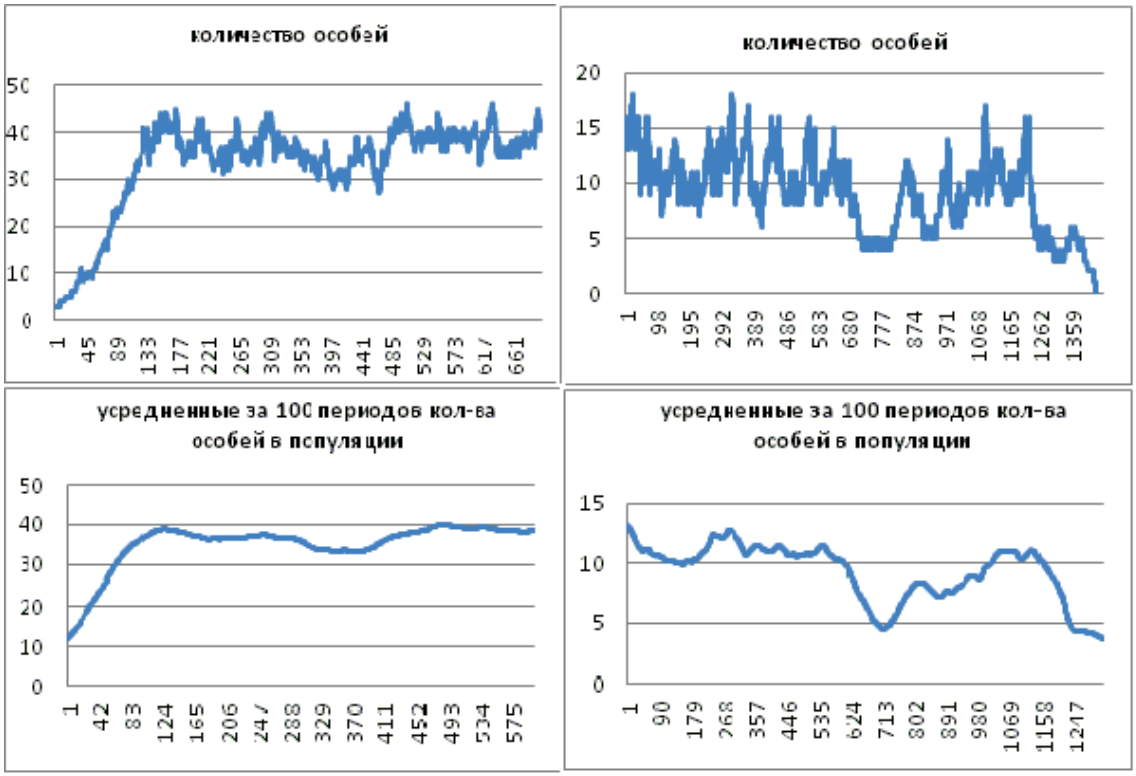


Рис. 9. Результат работы модели в случае неагрессивного поведения (левые графики) и агрессивного поведения агентов (правые графики)

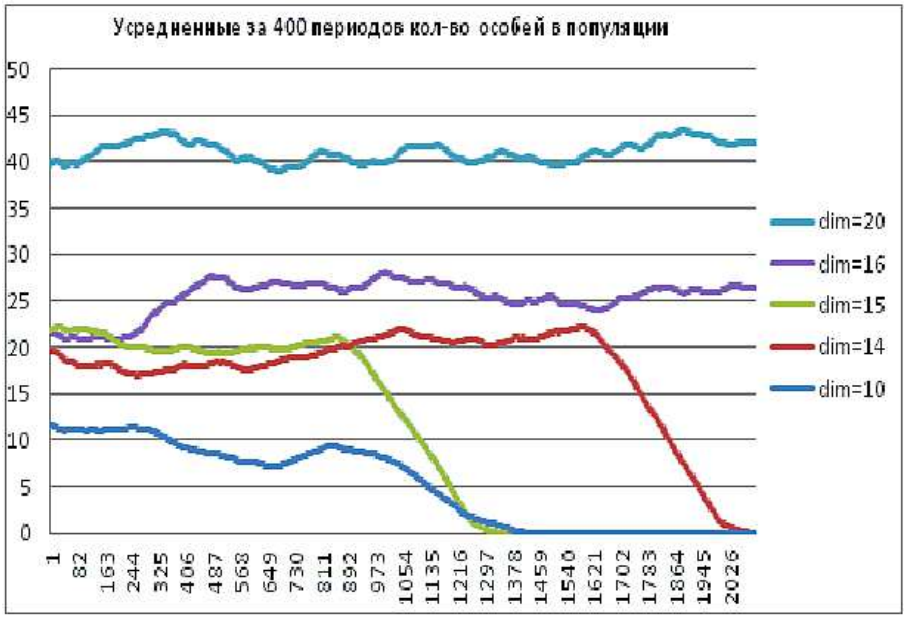


Рис. 10. Результат работы модели, динамика количества особей в популяции в зависимости от размерности поля dim

вариантах видны качественно разные зависимости динамики развития популяции от начального уровня здоровья агентов. Результаты отражены на рис. 12 и 13.

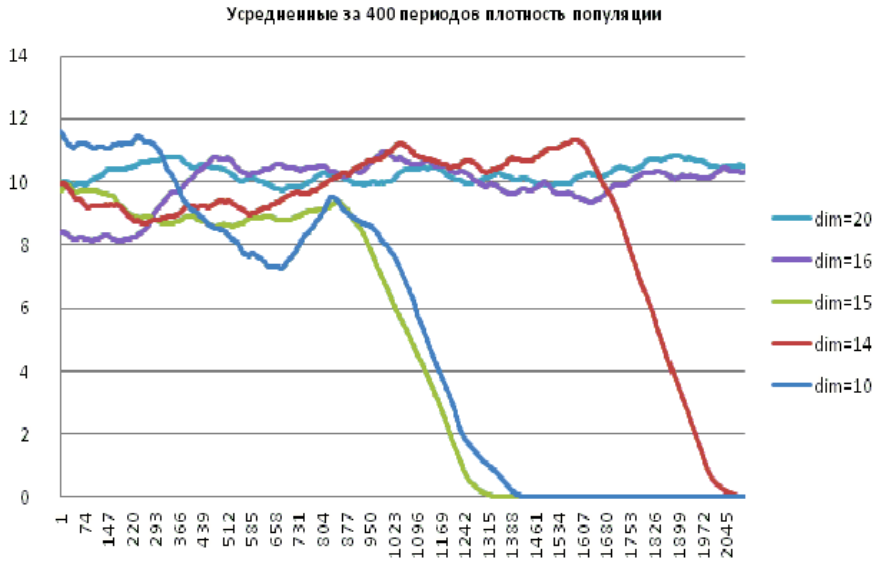


Рис. 11. Результат работы модели, динамика плотность популяции в зависимости от размерности поля dim

В случае dim = 10 видно, что все популяции рано или поздно вымирают, однако можно отметить, что зависимость продолжительности жизни популяции не монотонно зависит от начального уровня здоровья. А именно: выявляется некоторое оптимальное значение этого показателя, при котором продолжительность жизни популяции больше, чем при других значениях начального уровня здоровья. На графике видно, что из приведенных значений наибольшая продолжительность жизни популяции достигается при начальном уровне здоровья равном 20. При больших или меньших значениях этого параметра продолжительность жизни популяции меньше. Теперь можно проанализировать влияние тех же уровней начального здоровья в случае dim = 20.

В случае dim = 20 наблюдается другая динамика. Продолжительность жизни популяции монотонно зависит от начального уровня здоровья, т. е. чем выше этот показатель, тем либо выше период жизни популяции, либо выше среднее количество особей в установившемся режиме развития популяции. Пограничными значениями начального уровня здоровья являются 6 и 7. Интересно также отметить, что если при малых значениях этого показателя его изменения заметно влияют на динамику популяции (видно на примере сравнения начальном уровне здоровья равном 5 и 6), то при больших значениях эти изменения уже гораздо менее заметны. Так, при начальном уровне здоровья = 14 популяция в среднем развивается так же, как и при начальном уровне здоровья = 20.

Выводы. В модели с агрессивным поведением агентов присутствует как положительное влияние агентов друг на друга, так и отрицательное. Причем положительные и отрицательные влияния сбалансированы. Благодаря варьированию различных показателей выявлены граничные значения этих показателей, при которых популяция в конце концов вымирает.

Экспериментально было показано, что при неагрессивном поведении популяция не вымирает и численность особей колеблется вокруг некоторого установившегося режима, тогда как при агрессивном поведении живучесть популяции резко сокращается и зависит уже от других параметров. Так, одним из важнейших характеристик является размер-

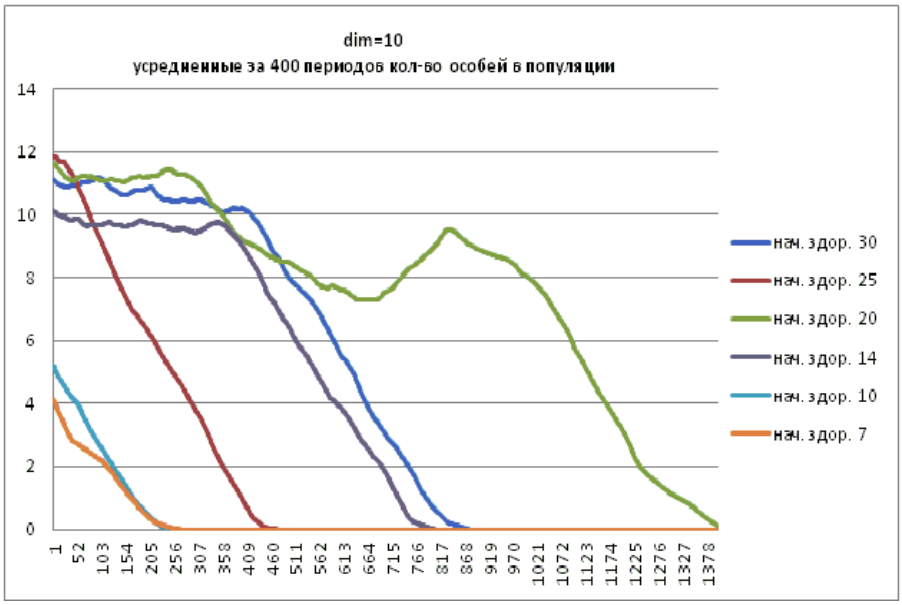


Рис. 12. Результат работы модели, динамика количества особей в популяции в зависимости от начального уровня здоровья при dim = 10

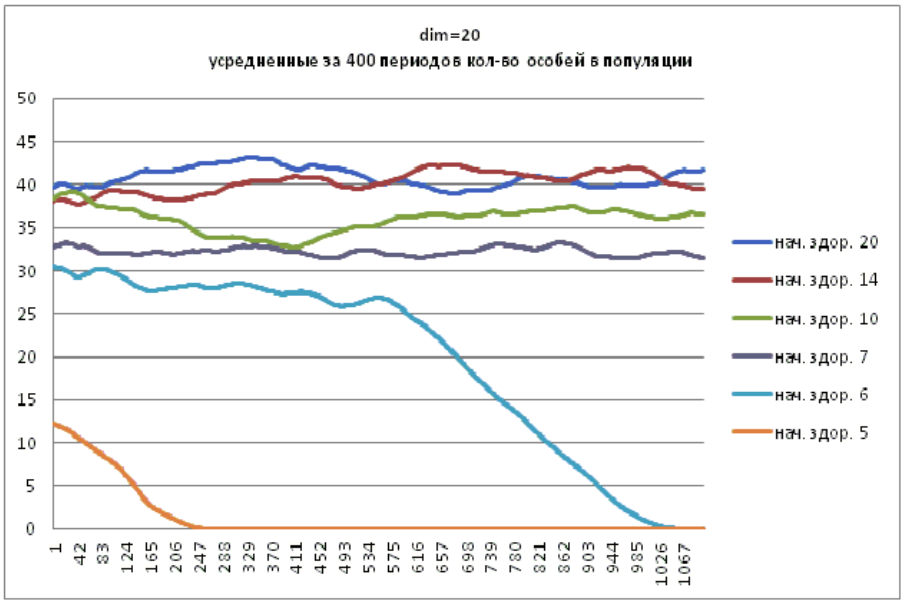


Рис. 13. Результат работы модели, динамика количества особей в популяции в зависимости от начального уровня здоровья при dim = 20

ность поля. Было получено, что при агрессивном поведении популяции, при прочих равных значениях остальных параметров, можно найти такое пороговое значение размерности, что при dim меньшем, чем это значение, популяция гарантированно будет вымирать через какое-то конечное время, при dim большем либо равном, чем это пороговое значение, популяция выживает, причем равновесное количество особей в популяции монотонно возрастает по *dim*. При этом в случае, когда популяция выживает, равновесное значение плотности населения (среднее количество особей, деленное на $\dim^2$), по всей видимости, является некоторым инвариантом относительно dim. С другой стороны, в случае, когда

популяция вымирает, время жизни популяции, вообще говоря, не монотонно зависит от размерности поля.

Изменения начального уровня здоровья также показывают интересные результаты. При достаточно малых значениях dim, когда популяция гарантированно вымирает, оказывается, что живучесть популяции зависит не монотонно от начального уровня здоровья, и существует некоторое оптимальное его значение, отклоняясь от которого как в большую, так и в меньшую сторону, живучесть популяции ухудшается. С другой стороны, при достаточно больших значениях dim живучесть популяции монотонно зависит от значения этого параметра (т. е. чем выше начальный уровень здоровья, тем выше равновесный уровень количества особей в популяции). При этом наблюдается эффект насыщения равновесного значения численности популяции при увеличении начального уровня здоровья, т. е. каждое следующее увеличение начального уровня здоровья дает все меньшую прибавку к равновесному уровню численности популяции. Если же начальный уровень здоровья сильно уменьшить, то можно найти пороговое значение этого параметра, при котором популяция будет вымирать.

Литература

[1] *Henning P. A.* Computational evolution. *Lecture notes in economics and mathematical systems ser.*, 2008. P. 175–193.

[2] *Epstein J., Axtell R.* Growing artificial societies: Social science from the bottom up. Washington, D.C.: Brookings Institution Press, 1996. 223 p.

[3] *Younger S.* Violance and revenage in egaliterian societies // *J. Artificial Societies Social Simulation*, 2005. Vol. 8, no. 4. P. 11. URL: `http://jasss.soc.surrey.ac.uk/8/4/11.html`.

[4] *Bowles S.* Individual interractions, group conflict, and the evolution of preferences // *Social Dynamics*, 2001. P. 155–190.

[5] *Bowles S., Choi J., Hopfensitz A.* The co-evolution of individual behaviors and social institutions // *J. Theor. Biol.*, 2003. Vol. 223, no. 2. P. 135–147.

[6] *Blume L. E.* Evolutionary equilibrium with forward-looking players. 2004. URL: `http://www.santafe.edu/media/workingpapers/05-04-015.pdf`

Методы интеллектуальной обработки качественных данных*

***И. В. Покровская*[1], *М. Д. Гольдовская*[1], *Ю. А. Дорофеюк*[1,2], *Н. Е. Киселева*[1]**
ivp750@mail.ru
[1]Москва, Институт проблем управления им. В. А. Трапезникова РАН (ИПУ РАН)
[2]Москва, Научно-исследовательский университет Высшая школа экономики (НИУ ВШЭ)

Исследуются задачи интеллектуальной обработки качественных данных. Рассмотрено два примера постановок задач и алгоритмов обработки качественных данных, представленных в виде признаков долевого типа и эмпирических графов большой размерности. Разработана методика интеллектуальной обработки признаков долевого типа, проведено тестирование на реальных данных. Исследованы возможности точного и приближенного представления графа большой размерности через его описание. На задачу агрегирования распространен оптимизационный подход к построению размытой классификации. В рамках структурно-классификационной методологии интеллектуального анализа сложно организованных данных разработаны оригинальные алгоритмы решения задачи обработки информации с помощью агрегирования графов большой размерности.

Ключевые слова: *качественные данные; интеллектуальный анализ данных; экспериментальные графы большой размерности; параметры долевого типа; размытая упорядоченная классификация*

Intellectual methods of processing qualitative data*

***I. V. Pokrovskaya*[1], *M. D. Goldovskaya*[1], *J. A. Dorofeyuk*[1,2] *and N. E. Kiseleva*[1]**
[1]ICS RAS; [2]SIU HSE

Intellectual processing of qualitative data problem is investigated. Two examples of the states of the problems and algorithms for qualitative data processing, presented in the form of the equity-type characteristics and the large dimension empirical graphs, are considered. The methodology of data mining (group) characteristics of equity-type (equivalent blurred classifications) is developed; this method was tested on real data. The possibilities of the exact and approximate representation of the large dimension graph through its description are studied. The optimization approach to the construction of the fuzzy classification is distributed to the problem of aggregation. In the framework of the structural-classification mining methodology of complex data, the original information processing algorithms by large dimension graphs aggregation methods are developed.

Keywords: *qualitative data; data mining; large dimension experimental graphs; options equity-type; fuzzy ordered classification*

Введение

В последнее время существенно возрос интерес к исследованию и моделированию слабо формализованных социально-экономических систем. Для многих систем такого рода

*Работа выполнена при частичной финансовой поддержке РФФИ, гранты № 14-07-00463-а, № 13-07-00992-а и № 12-07-00540-а.

значительная часть исходных параметров, описывающих состояние системы, имеют качественную природу. К таким параметрам относятся, например, бинарные, ранговые и номинальные признаки. Очевидно, что решение стандартных задач моделирования подобных систем, например, идентификации или структурного описания, невозможно получить методами, использующими только количественные (числовые) показатели. Здесь возможны два пути выхода из создавшейся ситуации. Первый путь — это разработка своеобразных преобразователей, позволяющих использовать алгоритмическую базу методов моделирования для количественных признаков. Типичным примером такого случая является модификация процедур расчета расстояний между объектами в многомерном пространстве бинарных признаков при решении задач структуризации множества исследуемых объектов. А именно, предлагается для таких расчетов вместо евклидовой метрики использовать метрику Хэмминга. Легко также преобразуется процедура расчета расстояний для случая, когда часть признаков — числовые, а другая часть — бинарные. Второй путь — это разработка принципиально новых алгоритмов решения стандартных задач моделирования и анализа систем, описываемых качественными параметрами. Здесь часто определяющую роль играет схема представления исходных данных или, другими словами, качественная модель порождения данных. Примером реализации этого пути является представление многих социологических данных в виде направленных бинарных или взвешенных графов. В этом случае задача структуризации, например, сводится к известной задаче декомпозиции графа на подграфы по степени связности, не требующей подсчета в явном виде каких-либо расстояний. В настоящей работе рассмотрены две задачи, на примере которых продемонстрированы особенности реализации первого и второго пути. Первая задача — структуризация специального типа качественных параметров — параметров долевого типа. Вторая задача — исследование структуры множества взаимосвязанных объектов многоагентной системы, когда сама система и взаимосвязь входящих в нее объектов характеризуется ориентированным не взвешенным графом большой размерности.

Группировка качественных параметров долевого типа

Под структуризацией (группировкой) некоторого множества параметров имеется в виду разбиение его на группы «близких», «взаимозависимых» параметров на основе выбранной меры близости (зависимости) между параметрами. Так построены алгоритмы экстремальной группировки параметров, измеряемых в количественных, ранговых и номинальных шкалах [1]. В работе предлагается подход к решению этой задачи для особого вида параметров — качественных параметров долевого типа.

Задача структуризации множества исходных параметров. Опыт использования алгоритмов структурно-классификационного анализа показывает, что классификация по всем исходным параметрам далеко не всегда приводит к желаемым результатам. Действительно, при сравнительно небольших выборках экспериментальных наблюдений и наличии помех (ошибки в определении значений параметров, сознательное искажение информации и т. д.) использование для классификации большого числа входных параметров приводит к сильному «перемешиванию» классов, а сами классы при этом плохо поддаются интерпретации. По этой причине классификацию целесообразно проводить не в исходном пространстве, а в пространстве наиболее существенных (информативных) параметров, имеющем значительно меньшую размерность. Для структуризации параметров обычно используются алгоритмы экстремальной группировки параметров [2]. При этом необходимо определить, нужна ли группировка с фоновой группой или без нее (т. е. отсекать или нет сильно шумящие параметры) [3]. Результатом группировки являются группы

параметров и факторы — обобщенные характеристики групп. На основе результатов группировки строятся интегральные показатели исследуемой структуры. В качестве таковых выбираются либо сами факторы, либо параметры в определенном смысле ближайшие к факторам. Основное условие — они должны быть легко интерпретируемы. Для удобства использования интегральных показателей по каждому из них делается одномерная классификация объектов. Благодаря этому интегральный показатель преобразуется в качественный, так как его значения можно качественно характеризовать в терминах типа «низкие», «средние», «высокие».

Другое применение метода экстремальной группировки — выбор информативных параметров для структуризации на последующих этапах. В качестве набора информативных параметров выбирается либо набор факторов, либо набор, в который входят один или небольшое число параметров из каждой группы экстремальной группировки. Обычно окончательное решение о выборе информативных параметров производится экспертом-пользователем [4].

Структуризация результатов классификации. Практически все алгоритмы структурно-классификационного анализа содержат свободные параметры, значения которых трудно выбрать заранее из теоретических соображений. Кроме того, эти алгоритмы находят лишь локальный экстремум соответствующего критерия качества структуризации, поэтому результаты их работы зависят от начальных условий (начального разбиения объектов на классы или параметров на группы). В связи с этим при решении практических задач свободные параметры алгоритмов, начальные условия, а часто и состав переменных, образующих исходное пространство, варьируются в широких пределах. Это приводит к тому, что в результате получается достаточно обширное множество различных вариантов классификации. Число классификаций часто оказывается столь большим, что для их анализа приходится применять компьютерные методы, вводя меру близости между классификациями и разбивая их на группы «похожих» классификаций. Легко показать, что размытые классификации можно рассматривать как признаки долевого типа. Следовательно, можно для структуризации множества классификаций использовать методы группировки признаков долевого типа.

Признаки долевого типа. Рассмотрим некоторый «агрегированный объект» (например, город), включающий множество «индивидуальных объектов» (например, жителей города). Пусть каждый житель характеризуется некоторым качественным показателем, измеряемым в номинальной шкале (например, уровнем образования, имеющим три значения: ниже среднего, среднее, высшее). Тогда для города этот же показатель, уровень образования, естественно характеризовать набором из трех чисел, показывающих, какую долю его населения составляют жители с соответствующими уровнями образования.

Рассмотрим множество из n агрегированных объектов, каждый из которых состоит из ряда индивидуальных объектов. Будем считать, что индивидуальные объекты описываются двумя параметрами x и y, измеренными в номинальных шкалах. Пусть параметр x принимает одно из значений $(x_1, \ldots, x_k)$, а y — одно из значений $(y_1, \ldots, y_m)$.

Рассмотрим соответствующие параметры долевого типа и , описывающие агрегированные объекты. Их значения для t-го объекта представляют собой векторы $A_t = = (\alpha_t^{(1)}, \ldots, \alpha_t^{(k)})$ и $B_t = (\beta_t^{(1)}, \ldots, \beta_t^{(m)})$. Здесь $\alpha_t^{(i)}$ — доля индивидуальных объектов, принадлежащих t-му агрегированному объекту, для которых $x = x_i$, а $\beta_t^{(j)}$ — доля индивидуальных объектов, принадлежащих t-му агрегированному объекту, для которых $y = y_j$.

Вначале предположим, что все индивидуальные данные нам известны. Тогда, кроме этих параметров, мы можем подсчитать параметр $G_t = (g_t^{(i,j)}, i = \overline{1,k},\ j = \overline{1,m})$, где $g_t^{(i,j)}$ — доля индивидуальных объектов, принадлежащих t-му агрегированному объекту, для которых $x = x_i$, а $y = y_j$.

Если интерпретировать долю $\alpha_t^{(i)}$ как вероятность i-й градации параметра x для t-го агрегированного объекта, а $\beta_t^{(j)}$ как вероятность j-й градации параметра y, то $g_t^{(i,j)}$ интерпретируется как совместная вероятность i-й градации параметра x и j-й градации параметра y.

Введем матрицы условных вероятностей

$$Q_t^{(i,j)} = P_t(y = y_j \,|\, x = x_i) = \frac{g_t^{(i,j)}}{\alpha_t^{(i)}},\ R_t^{(i,j)} = P_t(x = x_i \,|\, y = y_j) = \frac{g_t^{(i,j)}}{\beta_t^{(j)}},\quad i = \overline{1,k},\ j = \overline{1,m}.$$

Справедливы соотношения:

$$\beta_t^{(j)} = \sum_{i=1}^{k} Q_t^{(i,j)} \alpha_t^{(i)};\ \alpha_t^{(i)} = \sum_{j=1}^{m} R_t^{(i,j)} \beta_t^{(j)};\ i = \overline{1,k},\ j = \overline{1,m}. \tag{1}$$

Матрицы Q_t и R_t отражают вероятностную зависимость между долевыми параметрами A и B. Однако во многих практических задачах индивидуальные данные недоступны, имеются только значения долевых показателей A и B. В этом случае показатель G и матрицы Q_t и R_t напрямую подсчитать нельзя, тогда возникает следующая задача: восстановить матрицы Q_t и R_t по параметрам A и B.

Алгоритм восстановления матриц. Для решения указанной задачи сделаем следующее допущение: матрицы условных вероятностей Q_t и R_t не зависят от агрегированного объекта, т. е. $Q_t = Q$ и $R_t = R$. Рассмотрим алгоритм восстановления матрицы Q (восстановление матрицы R производится аналогично).

Будем считать, что соотношения (1) выполняются не точно, а с некоторой случайной погрешностью, имеющей характер аддитивного шума, в частности:

$$\beta_t^{(j)} = \sum_{i=1}^{k} Q_t^{(i,j)} \alpha_t^{(i)} + \varepsilon_t^{(j)}, i = \overline{1,k},\ j = \overline{1,m}. \tag{2}$$

Уравнение (2) имеет вид линейной регрессии и отличается от нее только ограничением:

$$\sum_{j=1}^{m} Q^{(i,j)} = 1,\ i = \overline{1,k};\ Q^{(i,j)} \geqslant 0,\ i = \overline{1,k},\ j = \overline{1,m}.$$

Оценочная матрица $\widehat{Q}$, минимизирующая суммарную дисперсию случайной погрешности, находится стандартными процедурами квадратичного программирования.

Качество линейной регрессионной модели обычно измеряют коэффициентом детерминации. В рассматриваемой задаче это:

$$D(\widehat{Q}) = \frac{T(Y) - F(\widehat{Q})}{T(Y)},$$

где $T(Y)$ — сумма квадратов отклонений компонент вектора B от своих средних значений, а $F(\widehat{Q})$ — сумма квадратов невязок в (2). Преимущество коэффициента детерминации по сравнению с некоторыми другими мерами качества модели состоит в том, что он

Таблица 1. Значения параметра X

X	$\mathbf{A_1}$	$\mathbf{A_2}$	$\mathbf{A_3}$	$\mathbf{A_4}$	$\mathbf{A_5}$	$\mathbf{A_6}$	$\mathbf{A_7}$	$\mathbf{A_8}$	$\mathbf{A_9}$	$\mathbf{A_{10}}$
α_1	0,1	0,3	0,0	0,5	0,2	0,4	0,3	0,0	1,0	0,4
α_2	0,3	0,4	0,2	0,3	0,1	0,4	0,1	0,3	0,0	0,2
α_3	0,2	0,2	0,4	0,1	0,4	0,0	0,2	0,4	0,0	0,3
α_4	0,4	0,1	0,4	0,1	0,3	0,2	0,4	0,3	0,0	0,1
X	$\mathbf{A_{11}}$	$\mathbf{A_{12}}$	$\mathbf{A_{13}}$	$\mathbf{A_{14}}$	$\mathbf{A_{15}}$	$\mathbf{A_{16}}$	$\mathbf{A_{17}}$	$\mathbf{A_{18}}$	$\mathbf{A_{19}}$	$\mathbf{A_{20}}$
α_1	0,2	0,7	0,1	0,3	0,8	0,9	0,1	0,2	0,4	0,1
α_2	0,1	0,1	0,1	0,5	0,1	0,0	0,6	0,1	0,2	0,6
α_3	0,1	0,0	0,4	0,1	0,0	0,0	0,2	0,6	0,0	0,1
α_4	0,6	0,2	0,4	0,1	0,3	0,1	0,1	0,1	0,4	0,2

достаточно нагляден и легко интерпретируем — меняется от 0 до 1, и чем он ближе к 1, тем лучше модель.

Матрица R восстанавливается аналогично. Однако в общем случае $D(\widehat{Q}) \neq D(\widehat{R})$, т. е. зависимость между долевыми параметрами A и B несимметрична (A может зависеть от B сильнее, чем B от A, и наоборот). Поэтому при структуризации множества долевых параметров в качестве меры зависимости целесообразно использовать сумму коэффициентов детерминации.

Связь с размытыми классификациями. Напомним, что четкой классификацией множества объектов на k классов называется такое разбиение объектов на классы, что каждый объект отнесен к одному и только одному классу. Будем приписывать каждому объекту номер класса, в который он попал. Тогда получается, что на множестве объектов задан номинальный признак — номер класса. Наоборот, если есть номинальный признак, то множество объектов разбивается по нему на классы эквивалентности. Следовательно, понятия номинальный признак и четкая классификация можно интерпретировать друг через друга. Соответственно, понятие рангового признака можно интерпретировать как упорядоченную классификацию, т. е. такую, у которой классы упорядочены.

Наряду с четкими классификациями широкое применение получили размытые классификации, у которых объекты с разными весами могут принадлежать сразу нескольким классам. Размытая классификация задается через вектор-функцию принадлежностей. Заметим, что формальный объект такого рода можно интерпретировать как параметр долевого типа.

Таким образом, номинальные признаки на индивидуальных данных можно интерпретировать как четкие классификации, а параметры долевого типа на агрегированных данных как размытые классификации.

Компьютерное моделирование. Для проверки работоспособности предложенной методики было проведено компьютерное моделирование как на модельных, так и на реальных данных, содержащих данные о $g_t^{(i,j)}$. Результаты моделирования показывают, что оценки матриц условных вероятностей получаются близкими к реальным матрицам, а при отсутствии шума $\varepsilon_t^{(j)}$ совпадают с ними.

Компьютерное моделирование на модельных данных. Исследовались 20 агрегированных объектов. На них был построен параметр долевого типа X, состоящий из четырех градаций, значения параметра X приведены в табл. 1.

Таблица 2. Матрица условных вероятностей Q

Q	β_1	β_2	β_3
α_1	$P_{11} = 0{,}1$	$P_{21} = 0{,}8$	$P_{31} = 0{,}1$
α_2	$P_{12} = 0{,}3$	$P_{22} = 0{,}6$	$P_{32} = 0{,}1$
α_3	$P_{13} = 0{,}5$	$P_{23} = 0{,}4$	$P_{33} = 0{,}1$
α_4	$P_{14} = 0{,}4$	$P_{24} = 0{,}2$	$P_{34} = 0{,}4$

Таблица 3. Результаты экспериментов

δ	$F_1(\widehat{Q})$	$D_1(\widehat{Q})$	$\rho(Q, \widehat{Q})$
0,0	0,000	1,000	0,000
0,1	0,010	0,731	0,011
0,2	0,038	0,465	0,042
0,3	0,077	0,393	0,105

Значения параметра долевого типа Y, состоящего из трех градаций, рассчитывались по следующей схеме. Задана матрица условных вероятностей Q, приведенная в табл. 2.

По параметру X и матрице Q строились величины:

$$\tilde{\beta}_t^{(j)} = \max\left(0, \sum_{i=1}^{4} Q_t^{(i,j)} \alpha_t^{(i)} + \delta z_t^{(j)}\right), \ j = 1, 2, 3; \ t = \overline{1, 20}.$$

где δ — константа, задающая уровень шума; $z_t^{(j)}$ — величины, полученные датчиком случайных чисел, распределенных равномерно на отрезке $[-1; 1]$.

Наконец, значения компонент (градаций) параметра Y вычислялись по формуле:

$$\beta_t^{(j)} = \frac{\tilde{\beta}_t^{(j)}}{\sum_{l=1}^{4} \tilde{\beta}_t^{(l)}}.$$

Такая схема введения шума в зависимость параметра Y от параметра X гарантирует выполнение ограничений (2) и (3).

В эксперименте строилась оценка $\widehat{Q}$ для матрицы условных вероятностей Q для четырех разных уровней шума. Величина δ равнялась последовательно 0; 0,1; 0,2 и 0,3. Результаты эксперимента приведены в табл. 3.

В табл. 3 величина $\rho(\widehat{Q}, Q) = (\widehat{Q} - Q)^2$ является мерой отличия модельной матрицы $\widehat{Q}$, полученной при разных уровнях шума, от исходной матрицы Q.

Во-первых, следует отметить, что если шума нет, то исходная матрица восстанавливается точно. Во-вторых, при возрастании уровня шума возрастает значение $F_1(\widehat{Q})$ и падает значение коэффициента детерминации $D_1(\widehat{Q})$. В-третьих, следует отметить хорошую корреляцию между вторым и четвертым столбцами табл. 3, т. е. между $F_1(\widehat{Q})$ и $\rho(Q, \widehat{Q})$. Следовательно, величина $F_1(\widehat{Q})$ достаточно хорошо отражает качество оценивания матрицы Q.

Компьютерное моделирование на реальных данных. Для моделирования использовались данные переписи населения России [5]. Анализировались некоторые демографические показатели 77 регионов России. В качестве первого параметра долевого типа X рассматривалась степень урбанизации региона ($\alpha_t^{(1)}$ — доля городских жителей в t-м

Таблица 4. Значения элементов матрицы $\widehat{Q}$, минимизирующие критерий $F_1(\widehat{Q})$

Q	$P(\beta_1)$	$P(\beta_2)$	$P(\beta_3)$
α_1	$P_{11} = 0{,}156$	$P_{21} = 0{,}638$	$P_{31} = 0{,}206$
α_2	$P_{12} = 0{,}271$	$P_{22} = 0{,}542$	$P_{32} = 0{,}187$

Таблица 5. Значения элементов выборочной матрицы $P(Y/X)$

$Q = P(Y/X)$	$P(\beta_1)$	$P(\beta_2)$	$P(\beta_3)$
α_1	$P_{11} = 0{,}165$	$P_{21} = 0{,}631$	$P_{31} = 0{,}204$
α_2	$P_{12} = 0{,}221$	$P_{22} = 0{,}560$	$P_{32} = 0{,}218$

регионе, а $\alpha_t^{(2)}$ — доля сельских). В качестве второго долевого параметра Y рассматривалась возрастная структура населения соответствующего региона ($\beta_t^{(1)}$ — доля людей в t-м регионе, чей возраст меньше трудоспособного; $\beta_t^{(2)}$ — доля людей трудоспособного возраста; $\beta_t^{(3)}$ — доля людей старше трудоспособного возраста). При рассмотрении демографических данных доля населения в регионе от суммарной численности населения во всех рассматриваемых регионах является тем масштабирующим коэффициентом d_t, который используется в критерии $F_3(\widehat{Q})$.

В табл. 4 приведены значения элементов матрицы $\widehat{Q}$, минимизирующие критерий $F_1(\widehat{Q})$. Для этой матрицы $F_1(\widehat{Q}) = 0,0025$ и $D_1(\widehat{Q}) = 0,245$. Расчеты показывают, что коэффициент детерминации получился значимым.

Из данных переписи можно извлечь также данные о пересечении рассматриваемых параметров. Отметим, что в построении матрицы $\widehat{Q}$ эти данные не использовались, поэтому их можно рассматривать как тестовый материал для модели. По этим данным была построена выборочная матрица условных вероятностей параметра Y от параметра X. Значения ее элементов приведены в табл. 5.

Сравнение табл. 4 и 5 показывает, что матрица оценок $\widehat{Q}$ достаточно хорошо соответствует реальной матрице Q (в данном случае $P(Q, \widehat{Q}) = 0,0039$).

Агрегирование графов большой размерности

Пусть задан ориентированный невзвешенный граф большой размерности, полученный как результат экспериментального исследования группы взаимосвязанных объектов (например, некоторой многоагентной системы). Задача состоит в выявлении основных пучков дуг в этом графе, т. е. в выделении пар подмножеств множества вершин графа, таких, что из первого подмножества во второе идут почти все дуги. Особый интерес представляет случай, когда совокупность всех пучков можно рассматривать как некоторый малый граф, множество вершин которого является набором подмножеств множества вершин исходного графа. Набор подмножеств некоторого множества можно интерпретировать, как кластеризацию с перекрывающимися кластерами. Задача агрегирования исходного графа заключается в нахождении такой кластеризации множества вершин исходного графа и такого малого графа построенного на кластерах, которые в некотором смысле оптимально описывают исходный граф.

Формально задача ставится следующим образом. Обозначим исходный граф через G, множество его вершин через $X = \{x_1, ..., x_n\}$, а его матрицу смежности через $M(G) = \|m_{i,j}; i = \overline{1,n}; j = \overline{1,n}\|$.

Пусть $H = \{H_1, ..., H_r\}$ $(H_i \subseteq X)$ — некоторая кластеризация множества X с перекрывающимися кластерами. Такую кластеризацию можно задавать с помощью матрицы $B(H) = \| b_{pi} \|$, элемент b_{pi} которой, находящийся на пересечении p-ой строки и i-го столбца, равен 1, если i-я вершина принадлежит p-му кластеру, а в противном случае он равен 0. Пусть на H как на множестве вершин построен малый граф , матрицу смежности которого обозначим через $M(\Gamma) = \|\mu_{i,j}; i = \overline{1,r}; j = \overline{1,r}\|$. По кластеризации H и графу Γ строится аппроксимирующий граф $G(\Gamma, H)$ с помощью следующего **алгоритма построения графа** $G(\Gamma, H)$. Выбирается дуга графа Γ, пусть она для определенности идет из вершины H_p в вершину H_q. Затем в графе $G(\Gamma, H)$ проводятся дуги из всех элементов кластера H_p во все элементы кластера H_q. Такая процедура выполняется со всеми дугами графа Γ. Матрица смежности $M(G(\Gamma, H)) = \|\hat{m}_{i,j}\|$ графа $G(\Gamma, H)$ вычисляется по формуле $M(G(\Gamma, H)) = B(H)^T * M(\Gamma) * B(H)$. Здесь знак «$*$» означает булево произведение матриц. Отсюда следует, что элементы матрицы смежности $M(G(\Gamma, H))$ определяются следующим выражением $\hat{m}_{i,j} = \bigvee_{p,q=1}^{r} b_{pi}\mu_{pq}b_{qj}$. Возможны две постановки задачи агрегирования, которые далее рассмотрены более подробно.

Первая задача агрегирования — оптимальное сужение исходного графа. Для заданного исходного графа G найти граф Γ и соответствующую кластеризацию $H = \{H_1, ..., H_r\}$ минимального размера такие, что $G = G(\Gamma, H)$. Другими словами это постановка задачи точного представления графа G через граф меньшего размера. Будем его называть сужением графа G, а сужение с минимальным числом вершин будем называть оптимальным сужением графа G. Если рассматривать эту задачу без каких либо дополнительных ограничений, то она представляет собой NP-полную задачу.

Ограничим поиск сужений графа G его подграфами. Для этого будем рассматривать пары «граф Γ — кластеризация $H = \{H_1, ..., H_r\}$» только следующего специального вида: в каждом кластере H_p выделяется один из элементов x_{ip} (напомним, что такой элемент — одна из вершин графа G). Далее считается, что на графе Γ дуга из вершины H_p идет в вершину H_q тогда и только тогда, когда из вершины x_{ip} идет дуга в вершину x_{iq} на графе G. Другими словами подграф графа G, построенный на множестве вершин $\{x_{i_1}, ..., x_{i_r}\}$ изоморфен графу Γ. Такие сужения будем называть внутренними.

Для нахождения оптимального внутреннего сужения графа G построим матрицу $D(G)$, у которой n строк и $2n$ столбцов. Первые n столбцов этой матрицы соответствуют матрице $M(G)$, а последующие n столбцов соответствуют матрице $M(G)^T$ — транспонированная матрица $M(G)$, т. е. матрица $D(G)$ имеет следующую структуру: $D(G) = (M(G) : (M(G)^T))$. Среди строк этой матрицы выделим строки, которые нельзя представить в виде булевой суммы никакого набора других строк этой же матрицы. В содержательном смысле этот набор строк составляет «независимый базис». Пусть $\{x_{i_1}, ..., x_{i_r}\}$ — подмножество вершин графа G, соответствующих выделенным строкам.

Теорема 1. Граф в оптимальном внутреннем сужении графа G — изоморфен подграфу G, построенному на множестве вершин $\{x_{i_1}, ..., x_{i_r}\}$.

Вторая задача агрегирования — аппроксимационный подход к декомпозиции графа. Для заданного исходного графа G найти такие граф Γ и соответствующую кластеризацию $H = \{H_1, ..., H_r\}$ с фиксированным числом классов r, чтобы граф $G(\Gamma, H)$ был как можно ближе к графу G в смысле заранее выбранного критерия J: $J = J(G(\Gamma, H)) = (M(G) - M(G(\Gamma, H)))^2 = \sum_{i,j=1}^{n} [m_{ij} - \hat{m}_{ij}]^2$. Такая постановка реализует аппроксимационный подход к задаче декомпозиции графа G. Алгоритм решения этой задачи

Таблица 6. Матрица смежности модельного графа

$M(G)$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
x_1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1
x_2	1	1	1	0	0	1	0	1	0	0	0	1	0	1	1	0	1	0
x_3	1	1	1	1	1	1	1	1	0	1	0	1	0	0	0	0	1	1
x_4	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1
x_5	0	1	1	1	1	1	0	0	1	0	0	0	0	1	0	0	0	0
x_6	1	0	0	0	1	1	1	1	1	0	0	1	1	0	0	0	1	1
x_7	1	1	0	1	0	0	1	1	1	0	0	1	0	1	0	1	0	0
x_8	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0
x_9	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	1
x_{10}	0	1	1	0	0	1	0	0	1	1	0	0	1	0	0	0	1	1
x_{11}	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0
x_{12}	0	1	1	0	0	1	1	1	1	0	1	1	0	1	0	1	1	0
x_{13}	1	0	1	1	1	1	0	1	0	1	0	1	1	0	0	0	0	1
x_{14}	0	1	0	0	0	1	1	1	0	0	1	1	0	1	1	1	0	0
x_{15}	0	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0
x_{16}	0	0	1	1	0	1	0	0	0	0	1	1	0	1	1	1	1	0
x_{17}	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1	1
x_{18}	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	1	1

состоит из последовательного выполнения двух процедур, которые выполняются на каждом шаге алгоритма:

1. При заданном графе Γ ищется такая кластеризация $H = \{H_1, ..., H_r\}$ с заданным числом классов r, которая позволяет уменьшить значение критерия J.
2. При фиксированном числе вершин графа Γ и при заданной кластеризации H ищется такое изменение структуры графа Γ, которое уменьшает значение критерия J.

В работе предложены эффективные алгоритмы оптимизации для обеих процедур. Предложенный подход является продвижением идей размытой классификации для анализа больших экспериментальных графов. Отметим, что кластеризация с перекрывающимися классами является важным частным случаем размытой классификации [3].

Компьютерное моделирование. Было проведено компьютерное моделирование алгоритма декомпозиции исходного графа (в рамках аппроксимационного подхода) на модельном материале. В качестве исходного материала была рассмотрена группа детей (18 чел.) одного из детских садов г. Москвы. Каждого ребенка попросили ответить на вопрос, с кем из детей из предъявленного списка он хочет играть. В результате был получен ориентированный граф G с матрицей смежности, приведенной в табл. 6.

Затем с помощью описанного выше алгоритма находились малый граф Γ и соответствующая кластеризация H на два класса (т. е. аппроксимирующий граф Γ состоял из двух вершин). В итоге было получено два следующих результата:

1. Граф Γ_1 состоит из двух несвязанных вершин с петлями; это означает, что есть два основных класса детей, желающих играть с детьми из «своего» класса (отметим, что классы пересекающиеся), этот результат отражен в табл. 7.
2. В графе Γ_2 есть петли у обеих вершин, и из второй вершины идет дуга в первую. Это означает, что дети хотят играть с детьми из «своего» класса, но, кроме того, все дети

Таблица 7. Структура графа Γ_1

$M(\Gamma_1)$	H_1	H_2	Кластеризация
H_1	1	0	$H_1 = \{x_2, x_7, x_{11}, x_{12}, x_{14}, x_{15}, x_{16}\}$
H_2	0	1	$H_2 = \{x_1, x_2, x_3, x_4, x_6, x_7, x_8, x_9, x_{17}, x_{18}\}$

Таблица 8. Структура графа Γ_2

$M(\Gamma_2)$	H_1	H_2	Кластеризация
H_1	1	0	$H_1 = \{x_1, x_2, x_6, x_7, x_8, x_9, x_{17}, x_{18}\}$
H_2	1	1	$H_2 = \{x_2, x_3, x_4, x_7, x_{12}, x_{14}, x_{18}\}$

из второго класса хотят играть со всеми детьми из первого класса. Данный результат дает более детальную картину взаимоотношений между детьми. Этот результат отражен в табл. 8.

Заключение

Разработаны новые методы исследования данных качественной природы: методика структурной обработки признаков долевого типа, а также алгоритмы точного и приближенного представления графа большой размерности через его описание. В настоящее время полученные результаты распространяются на случай взвешенных ориентированных графов динамического типа, широко используемых в мультиагентных системах управления. Разрабатываются также специализированные алгоритмы интеллектуальной обработки результатов многовариантного экспертного оценивания, масштабных социологических обследований и структурного анализа информационных потоков в Интернете.

Литература

[1] *Бауман Е. В.* Структуризация номинальных признаков в задаче экспертизы // *Экспертные оценки в задачах управления.* М.: ИПУ, 1982. С. 16–23.

[2] *Браверман Э. М., Мучник И. Б.* Структурные методы обработки эмпирических данных. М.: Наука, 1983. 464 с.

[3] *Дорофеюк А. А., Бауман Е. В., Дорофеюк Ю. А.* Методы интеллектуальной обработки информации на базе алгоритмов стохастической аппроксимации // *Математические методы распознавания образов: 15-я Междунар. конф.* М: МАКС ПРЕСС, 2011. С. 108–112.

[4] *Дорофеюк Ю. А., Киселева Н. Е., Покровская И. В.* Комплекс алгоритмов интеллектуального анализа данных для исследования функционирования сложных систем // *Управление развитием крупномасштабных систем MLSD'2013): Тр. 7-й Междунар. конф.* М.: ИПУ РАН, 2013. Т. 1. С. 220–232.

[5] Итоги переписи населения 2002 г. Т. 2: Возрастно-половой состав и состояние в браке. М.: Статистика России, 2004.

References

[1] *Bauman E. V.* 1982. Strukturizatsiya nominal'nykh priznakov v zadache ekspertizy. *Ekspertnye otsenki v zadachakh upravleniya.* M.: IPU Publ. 16–23. (In Russian.)

[2] *Braverman E. M., Muchnik I. B.* 1983. *Strukturnye metody obrabotki empiricheskikh dannykh.* M.: Nauka. 464 p. (In Russian.)

[3] *Dorofeyuk A. A., Bauman E. V., Dorofeyuk Yu. A.* 2011. Metody intellektual'noy obrabotki informatsii na baze algoritmov stokhasticheskoy approksimatsii. *Matematicheskie Metody Raspoznavaniya Obrazov: 15-aya Mezhdunar. Konf.: Sb. dokladov.* Moscow. 108–112. (In Russian.)

[4] *Dorofeyuk Yu. A., Kiseleva N. E., Pokrovskaya I. V.* 2013. Kompleks algoritmov intellektual'nogo analiza dannykh dlya issledovaniya funktsionirovaniya slozhnykh sistem. *Upravlenie razvitiem krupnomasshtabnykh sistem MLSD'2013): Tr. 7-y Mezhdunar. Konf.* 1:220–232. (In Russian.)

[5] *Itogi perepisi naseleniya 2002 g. Tom 2: Vozrastno-polovoy sostav i sostoyanie v brake.* M.: Statistika Rossii, 2004. (In Russian.)

Структурные аналогии в символьных последовательностях различной языковой природы*

В. Д. Гусев, Л. А. Мирошниченко, Н. В. Саломатина
gusev@math.nsc.ru, luba@math.nsc.ru
Новосибирск, Институт математики им. С. Л. Соболева СО РАН

Изучение структуры символьных последовательностей (текстов) играет важную роль при решении многоплановых задач анализа данных, возникающих в биологии, лингвистике и других областях знания. При всем многообразии текстов их объединяет наличие повторов как элементарных структурообразующих единиц. Целью работы является систематизация повторов и их комбинаций, т.е. *структурных единиц* более высокого уровня. Для их выделения используются сложностные профили последовательности (введены авторами) и аппарат сканирующих статистик (адаптирован для текстов на естественном языке). По итогам обработки текстов различной языковой природы выделены и описаны структурные единицы, характеризующиеся «межъязыковой общностью», что и является отличительной особенностью работы.

Ключевые слова: *символьные последовательности; структурные единицы; разнотипные повторы; профили сложности; профили кластеризуемости*

Structural analogies in symbolic sequences of different nature*

V. D. Gusev, L. A. Miroshnichenko, N. V. Salomatina
Sobolev Institute of Mathematics of the Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia

Symbolic sequences (words, strings, texts) as an object of study are encountered in various areas of knowledge: informatics, biology, linguistics, music. The notion of integrated repeats as elementary structure-forming units is general on conceptual level for all symbolic sequences, despite of diversity of alphabets, lengths, and nature of the texts. The purpose of this work is the systematization of elementary repeats and their combinations, i. e., structural units of higher level. Their function in different language systems is discussed.

Too low complexity of fragments of the text is usually correlated with existence of too long repeats or their high concentration. Thus, a basis of all methods of research is a complexity profile construction and the analysis of complexity decomposition of the text in the sliding window mode. Such analysis gives a conception of the most typical structural units which can be found in texts. In the natural language texts, where repeatability is less expressed, also the profile of clustering can be used.

DNA sequences of different organisms, texts in a natural language, and also neume himns are a source material for investigation. Systematization of structural units is the result of the complexity analysis of a huge number of texts of various nature. The interlanguage community principle is a reason for selection of the illustrating structures.

The approach stated in this work and the algorithms realizing it have rather universal character in respect of its applicability to various language systems. The interlanguage analo-

*Работа выполнена при финансовой поддержке РФФИ, проект № 13-07-00400.

gies described at the level of structures can extend to formulation of substantial problems and selection of tools of their solving.

Keywords: *symbolic sequences; structural units; repeats; complexity profile; clustering profile*

Введение

Символьные последовательности (тексты) как объект исследования встречаются во многих областях знания: математике, информатике, биологии, лингвистике, музыке. Примерами могут служить тексты на естественном языке, биологические последовательности (ДНК, РНК, аминокислотные, последовательности генов и др.), знаменные песнопения (знаковые последовательности, использовавшиеся для записи древнерусского церковного пения) и др.

Изучение структуры последовательностей в целом и отдельных их фрагментов является основой для решения многочисленных задач классификации (родо-видовой, жанровой, тематической и т. п.). Так, сходство первичных структур ДНК-последовательностей обычно предполагает и функциональную близость кодируемых ими белков. Специфические тандемные повторы используются для проведения ДНК-дактилоскопии. Этот же тип структур помогает идентифицировать в текстах знаменных песнопений фрагменты, характеризующиеся нестандартным («зашифрованным») распевом (так называемые «лица»). Аномально длинные тандемные или разнесенные повторы разного типа (см. далее) в последовательностях, вырабатываемых датчиками «случайных» чисел, или в шифротекстах сигнализируют о несовершенстве схем генерации чисел и шифрования. Длинная цепочка существительных в родительном падеже (серия в частеречном алфавите) трактуется как стилистическая погрешность и т. д.

При всем многообразии языковых систем и характеризующих их текстов объединяющим началом для них является понятие *повтора в широком смысле*, выступающего в качестве *основного структурообразующего элемента* текста. В частности, аномально длинные, а также аномально частые или редкие повторы различных типов уже могут рассматриваться как потенциально возможные структурные единицы. Аномальность определяется в сопоставлении со значениями, ожидаемыми для указанных параметров в предполагаемой модели порождения последовательности. Различают повторы прямые и симметричные, следующие друг за другом (тандемные) и разнесенные, совершенные (точные) и несовершенные (с искажениями). Последние связаны с проявлениями вариативности языковых единиц, присущей всем эволюционирующим языковым системам. Возможны также повторы с переименованием элементов алфавита или повторы с точностью до фиксированного агрегирования элементов алфавита.

Комбинации позиционно близких элементарных повторов разного типа порождают структурные единицы более высокого уровня. *Целью работы* является *систематизация* такого рода структурных единиц, характерных (общих) для различных языковых систем. Обсуждается их функциональная нагрузка. Иллюстрируются возможности использования *межъязыковых аналогий* при постановке содержательных задач и выработке подходов к их решению.

Заметим, что эволюционный фактор не позволяет говорить о полноте систематизации в каком-либо строгом смысле этого слова. В процессе эволюции меняется даже алфавит обсуждаемых языковых систем. Поэтому акцент сделан не столько на полноту охвата структур, присутствующих в реальных текстах, сколько на проявления межъязыковой

общности. Мы не апеллируем к схемам порождения текста, основанным на использовании формальных грамматик, не учитывающих возможность появления искажений в процессе эволюции и больше подходящих для языков программирования. Подразумеваемая нами схема порождения основана на использовании операций копирования (см., например, [1]), фиксирующих разноплановые проявления повторности в текстах.

Используемый подход. Сложностные разложения

Некоторые типы структур, представленные в статье (например, фракталоподобные, комбинированные и др.), выявлены в реальных текстах и даже поименованы авторами данной работы. Часть структур (например, кумулятивные) была описана и систематизирована ранее другими авторами, но применительно к какой-либо одной языковой системе. Наша роль в этом случае сводилась к поиску аналогов этих структур в других языковых системах (хотя бы в одной). В связи с этим работа носит и частично обзорный характер. Для выявления структур разными авторами использовались разные подходы, в том числе не алгоритмические (см., к примеру, [2, 3]). Наш подход, кратко описанный ниже, основан на идеях А. Н. Колмогорова относительно определения понятия «количество информации» [4]. Объективным индикатором насыщенности символьной последовательности повторами может служить ее сложность. А. Н. Колмогоров предложил оценивать сложность объекта (в данном случае последовательности S) длиной кратчайшего описания $K(S)$, по которому этот объект можно восстановить однозначно. Известно, однако, что колмогоровская сложность не является вычислимой функцией. Из возможных конструктивных приближений к оцениванию $K(S)$ мы опираемся на меру сложности конечной символьной последовательности, предложенную Лемпелем и Зивом [1]. Она в явном виде *апеллирует к понятию повтора* в традиционном его понимании (прямой совершенный) и легко обобщается на случай фиксированной совокупности *разнотипных повторов*, характерных для конкретной языковой системы [5].

Пусть Σ — конечный алфавит; $|\Sigma|$ — размер алфавита; S — конечная последовательность, составленная из элементов Σ (текст); $N = |S|$ — длина текста S; $S[i]$ — элемент S, стоящий в i-й позиции ($1 \leqslant i \leqslant N$); $S[i : j]$ — фрагмент S, включающий элементы с i-го по j-й ($1 \leqslant i \leqslant j \leqslant N$); x^m — m-кратное повторение символа (символьной цепочки) x; $S = S_1S_2$ — конкатенация (сцепление) последовательностей S_1 и S_2.

Лемпель и Зив определили сложность конечной символьной последовательности S как число шагов гипотетического процесса синтеза S с использованием двух допустимых операций: «порождение нового символа», и «копирование *максимально длинного* («готового») прототипа из предыстории» , т. е. из уже синтезированной части текста. Последовательность фрагментов, отражающих процесс синтеза, мы называем «*сложностным разложением*» S:

$$H(S) = S[1 : i_1]S[i_1 + 1 : i_2] \dots S[i_{k-1} + 1 : i_k] \dots S[i_{c-1} + 1 : N],$$

где $S[i_{k-1} + 1 : i_k]$ — фрагмент S, добавляемый на k-м шаге, а c — число шагов процесса (*сложность* S). В этом разложении операция порождения символа задействована не более чем $|\Sigma|$ раз. Подавляющая же часть компонентов получена с применением операции копирования. Таким образом, сложностное разложение — это представление текста в виде конкатенации повторяющихся фрагментов в том смысле, что каждому компоненту в $H(S)$ (за исключением порождаемых) соответствует свой прототип (повтор) в предыстории. Возможны случаи наложения (со сдвигом) компонента на прототип, сигнализирующие о наличии тандемной повторности.

Проиллюстрируем, например. как выглядит сложностное разложение последовательности $S = caa(ccatgat)^5at$ $(N = 40)$, содержащей достаточно длинную периодичность:

k	1	2	3	4	5	6	7	8	9	10	
$H(S) =$	$c\cdot$	$a\cdot$	$a\cdot$	$c\cdot$	$ca\cdot$	$t\cdot$	$g\cdot$	$at\cdot$	$(ccatgat)^4\cdot$	$at;$	$c(S) = 10$
q_k	1	2	3	4	5	7	8	9	11	39	
j_k	0	0	2	1	1	0	0	6	4	37	
l_k	1	1	1	1	2	1	1	2	28	2	

Здесь компоненты разложения разделены точками, k — номер компонента, l_k — его длина, q_k — начальная позиция k-го компонента, а j_k — начальная позиция прототипа для k-го компонента (в случае, если символ встретился впервые и применяется операция порождения, полагаем $j_k = 0$). Нетрудно видеть, что прототипом для девятого компонента $S[11 : 38]$ служит фрагмент $S[4 : 31]$, т. е. имеет место наложение компонента на прототип, сигнализирующее о наличии периодичности. Первые 7 символов компонента №9 копируются из предыстории, а все последующие — с элементов, синтезированных на текущем (еще не завершенном) шаге. Следует заметить, что если бы кратность повторения периода *ccatgat* была выше, то длина S могла бы возрасти в разы, но число компонентов в разложении осталось бы прежним.Формально, о наличии периодичности свидетельствует выполнение условия $j_k + l_k \geqslant q_k$. Легко показать при этом, что длина периода $p = q_k - j_k$, а кратность повторений не меньше, чем $l_k/p + 1$. На этом свойстве и основан алгоритм обнаружения периодичностей [8].

Сложность последовательности можно оценивать как в целом, так и в окне заданного размера W, которое скользит вдоль нее. В последнем случае речь идет об отслеживании «*локальной сложности*». Кривую изменения локальной сложности вдоль последовательности S мы называем *сложностным профилем* S и обозначаем $P(S, W)$. Формально $P(S, W) = c_1 c_2 \ldots c_i \ldots c_{N-W+1}$, где c_i — сложность фрагмента из S, выделяемого окном на i -м шаге, т.е. включающего в себя позиции $i, i + 1, \ldots, i + W - 1$.

Фрагменты текста, которым соответствуют *аномально низкие* значения локальной сложности, характеризуются высокой концентрацией повторов, т. е. *высокой степенью структурированности*. Именно эти фрагменты дают представление о наиболее характерных структурах, представленных в последовательности. Параметр W при этом может меняться в широких пределах , что позволяет выделять структуры, соответствующие разным иерархическим уровням. Для иллюстрации на рис. 1 и 2 приведены сложностные профили бактериального генома микоплазмы «Mycoplasma gallisepticum str. R(low)» (AE015450) для $W = 60$ и $W = 1000$ соответственно. Средние значения сложности $c_{\exp} =$ $= 20{,}72$ при $W = 60$ и $c_{\exp} = 186{,}81$ при $W = 1000$. Пунктиром обозначены линии, соответствующие уровням $c_{\exp} \pm 3\sigma$. К аномальным можно отнести все пики, лежащие ниже нижней пунктирной линии. При малых размерах окна ($W = 60$ в нашем случае) фрагменты с низкими значениями сложности зачастую содержат достаточно длинные периодичности с небольшой длиной периода, комплементарные палиндромы, либо комбинированные структуры (см. ниже). Главный минимум на рис. 1 обусловлен наличием в позиции 497 461 периодичности $(aga)^{27}$. При больших размерах окна (см. рис. 2) низкие значения сложности обычно обусловлены крупными повторами разных типов (тандемными или разнесенными), а также их комбинациями. Главный минимум объясняется наличием в позиции 925 156 фрагмента *gttttagcactgtacaatacttgtgtaagcaataac*, регулярно (с равными промежутками) повторяющегося более 30 раз (см. ниже раздел «Периодичности со сложной струк-

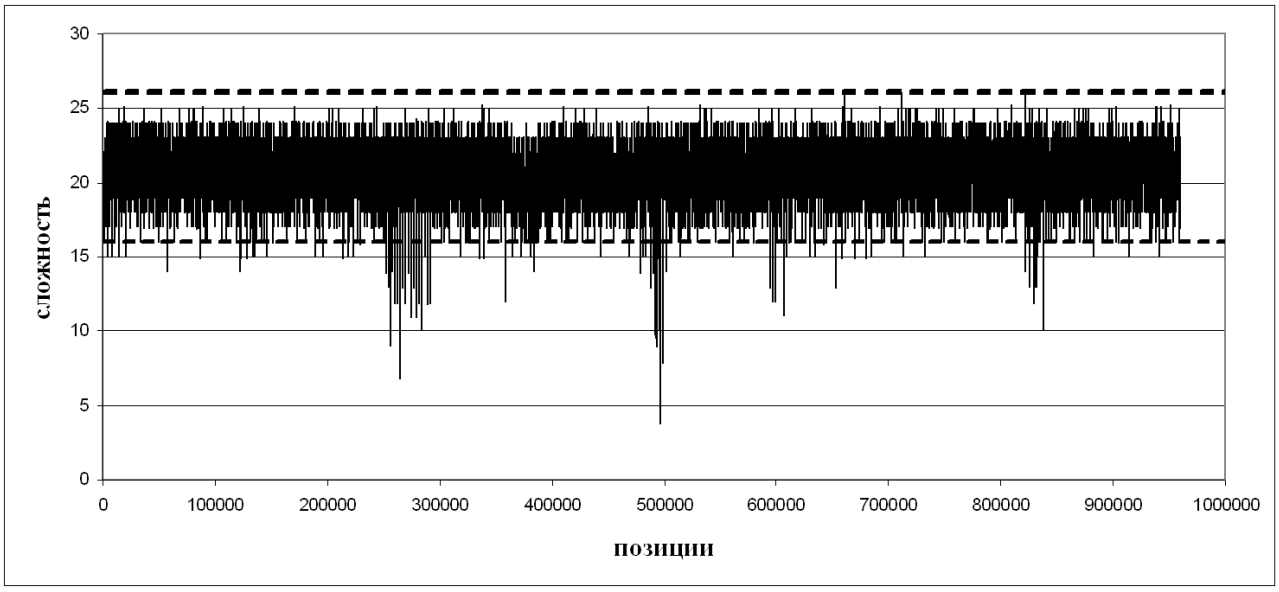


Рис. 1. Профиль сложности генома микоплазмы «R» при размере окна $W = 60$

турой») Распределение пиков на обоих кривых демонстрирует существенные различия в числе, размерах и расположении аномальных по сложности зон в геноме.

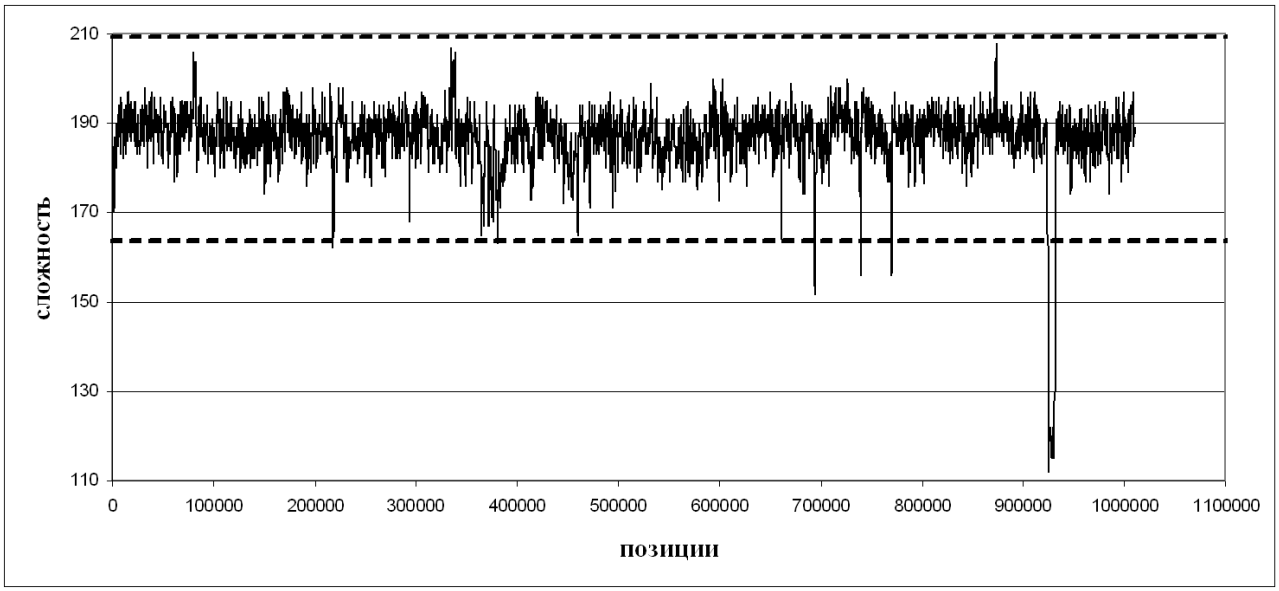


Рис. 2. Профиль сложности генома микоплазмы «R» при $W = 1000$

Обобщение подхода Лемпеля и Зива

Для конкретной языковой системы, содержащей специфические типы повторов, подход Лемпеля и Зива можно модифицировать, введя дополнительные операции копирования. Продемонстрируем это на примере ДНК-последовательностей [5]. А именно, наряду с прямым копированием, фиксирующим повторы в обычном смысле ($\ldots atcgag \ldots atcgag \ldots$), допускается симметричное копирование, выявляющее инверсии ($\ldots atcgag \ldots gagcta \ldots$), а также прямое и симметричное копирование с точностью до

подстановки $(a \leftrightarrow t)$, $(c \leftrightarrow g)$, реализующей известное отношение комплементарности на элементах ДНК-алфавита. Прямому комплементарному копированию соответствуют повторы вида $(\ldots atcgag \ldots tagctc \ldots)$, где второй фрагмент получен из первого заменой a на t, t на a, c на g, g на c. Симметричному комплементарному копированию соответствуют повторы вида $(\ldots atcgag \ldots ctcgat \ldots)$, где второй фрагмент совпадает с первым при прочтении его в обратном направлении и тех же заменах.

Расширение спектра допустимых операций копирования обусловило: (а) некоторое усложнение алгоритма вычисления $H(S)$ (добавляется перебор, связанный с выбором операции копирования, которая максимально удлиняет синтезируемую последовательность); (б) выявление значительного количества комбинированных структур, представленных разнотипными повторами (см. ниже); (в) появление «эффекта маскировки», связанного с возможностью наложения структур в анализируемом фрагменте текста [6]. Следует отметить, что сложностные разложения аномальных фрагментов лишь фиксируют потенциальное многообразие локальных структур. Для получения количественных оценок по каждому типу структур приходится строить специальные алгоритмы (см, например, [7, 8]).

В рассмотренной выше мере сложности операции копирования, использующие комплементарные подстановки, делали эту меру ДНК-ориентированной, т. е. применимой лишь к конкретной языковой системе.

В общем случае (произвольный алфавит и априори неизвестная подстановка $f : \Sigma \to \Sigma$) также нельзя исключать наличия в тексте аномально длинных f-повторов. Они могут быть прямыми и симметричными. Всего имеем $2|\Sigma|!$ типов f- повторов. При построении сложностного разложения выбор прототипа максимальной длины на каждом шаге можно проводить по всем $|\Sigma|!$ подстановкам на элементах алфавита и использовать копирование в обоих направлениях. Прямой перебор по подстановкам возможен лишь при небольших размерах алфавита. В [5] авторами предложен структурный инвариант, использование которого позволяет обойти проблему факториального перебора и выявлять произвольные аномально длинные f-повторы, если таковые присутствуют в тексте. Ориентиром для выработки критерия аномальности могут служить оценки длин максимальных f-повторов в случайных последовательностях, представленные в [9]. Важно отметить, что описанная в данном абзаце мера сложности C_f вновь приобретает свой «универсальный характер» в смысле возможности применения ее к текстам любой природы, в частности, к музыкальным текстам, где f-повторам соответствуют секвентные переносы фрагментов мелодии (звуковысотные сдвиги). Однако платой за такую универсальность будет увеличение трудоемкости алгоритма.

Завершая раздел об используемом аппарате, отметим, что:

- сложностное разложение как способ представления текста в терминах повторов применимо к текстам любой длины и произвольной языковой природы, в связи с чем широко используется для сжатия данных. Нас интересует не сжатие, как таковое, а аномально длинные компоненты в разложении всего текста и фрагменты с аномально низким значением сложности, выявляемые с помощью сложностного профиля;
- спектр и специфика повторов более ярко просматривается на неструктурированных (без разделителей) текстах с малым размером алфавита. При обработке текстов со значительным размером алфавита полезным может оказаться осмысленное агрегирование алфавита (например, переход к частеречным значениям для текстов на естественном языке);
- нами разработаны модификации сложностного подхода, ориентированные на сравнение пар и групп текстов (разложение одного текста по другому или одной группы тек-

стов по другой). При малоповторности отдельных текстов значимыми могут оказаться межтекстовые повторы

Исходные данные

При выявлении межъязыковых аналогий в качестве базовых рассматривались три языковые системы, представленные: (а) биологическими текстами; (б) текстами на естественном языке; (в) знаменными песнопениями. Обработка их производилась в разные годы в связи с решением конкретных прикладных задач, связанных с формализацией выделения структурных единиц в символьных последовательностях и многоплановой классификацией.

Биологические тексты нижнего уровня – это последовательности ДНК и РНК. Авторы обрабатывали полные геномы простейших микроорганизмов (бактериофагов $\varphi x174$, $g4$, λ [10], вирусов гриппа, клещевого энцефалита, бактерий из семейства микоплазм и др.) с длинами в диапазоне от 10^3 до 10^7 символов, а также фрагменты более крупных геномов вплоть до генома пшеницы. Последний был представлен данными частичного секвенирования хромосомы 5B (свыше 100 тысяч фрагментов длиной от 50 до 600 символов по длинному плечу и вдвое меньше по короткому [11]).

Биологические тексты верхнего уровня — это последовательности генов в геномах и еще более крупных единиц — дисков в политенных хромосомах двукрылых. Эволюция на хромосомном уровне идет уже не путем точечных замен или вставок, но путем более редких крупноблочных операций: транспозиций и инверсий. Уникальная коллекция последовательностей дисков в хромосомных плечах видов рода Chironomus (комары-звонцы) собрана в Институте цитологии и генетики СО РАН. Эти данные являют собой пример бесповторных последовательностей (перестановки длиной до 150 символов), переводимых одна в другую конечным числом инверсий. Инверсия предполагает изменение порядка следования символов на обратный в выделенном фрагменте. При сравнении двух перестановок мы использовали представление одной из них в виде конкатенации минимально возможного числа фрагментов (прямых или инвертированных) из другой. Построенная нами матрица попарного сходства последовательностей дисков из упомянутой коллекции позволила на геномном уровне уточнить филогенетические связи между видами [12].

Тексты на естественном языке уже частично структурированы (разбивка на слова, предложения, абзацы...), однако часто возникает потребность в выделении промежуточных уровней иерархии, фиксирующих, например, устойчивые словосочетания или сверхфразовые единства. Первые доминируют в терминологических словарях различных предметных областей, вторые могут быть использованы для построения квазирефератов текста. Для выделения устойчивых словосочетаний могут быть использованы сложностные разложения значительных по объему предварительно нормализованных текстов с традиционной операцией прямого копирования. Для выделения сверхфразовых единств приходится строить аналог сложностного профиля (см. ниже) и использовать другую технику выявления скоплений значимых языковых единиц [13]. Материалом для указанных разработок послужили весьма объемные труды конференций по компьютерной лингвистике на русском языке (Диалог-2002 и др.) и по катализу на английском языке (EuropaCat-2005 и др.).

Уникальный материал по знаменным песнопениям собран авторами данной работы. Это певческие книги конца XVII – начала XVIII вв., в которых песнопения представлены параллельно в знаменной и нотолинейной форме (так называемые двознаменники — своего рода билингвы знаменного распева, положенные нами в основу дешифровки знаменной

нотации [14, 15, 16]). Кодирование материала проводилось вручную, поскольку программ распознавания рукописного знаменного текста не существует. На данный момент закодированы и анализируются четыре двознаменника, примерно по 200 песнопений в каждом. Длины песнопений колеблются в диапазоне от нескольких десятков до нескольких сот знамен, каждое знамя интерпретируется цепочкой от 1 до 6 нотных знаков. Текст песнопения (старославянский) синхронизован со знаменным и нотолинейным.

Систематизация локальных структур

Как уже упоминалось во введении, фиксируются локальные структуры, встречающиеся во многих эволюционирующих языковых системах. Некоторые из этих структур могут быть описаны и на «языке образцов» [17].

Совершенные периодичности (тандемные повторы) Под ними мы понимаем фрагменты текста, представимые в виде $P = x^m$, где x — произвольная цепочка символов из Σ (период), $|x| \geqslant 1$ — длина периода, $m \geqslant 2$ — кратность повторения. Этот класс структур чрезвычайно распространен в ДНК-последовательностях. Они достаточно детально описаны и систематизированы. Механизмы их возникновения известны. Длины периодов могут меняться от 1 до 10^3 и более символов, а кратность повторений доходит до сотен раз и выше. Считается, что насыщенность ДНК-последовательностей повторами способствует повышению помехоустойчивости генетического языка. Разнесенные повторы, в частности, симметричные комплементарные, часто образуют палиндромно-шпилечные конструкции, играющие важную роль в регуляции генетических процессов. Ниже приведена для иллюстрации структурная единица шпилечного типа, выявленная в позиции 27 724 генома фага λ (выделена сходящимися стрелками):

$$\ldots \overrightarrow{gctttttta}\overrightarrow{tactaagttggcat}\overleftarrow{tataaaaaagc}\ldots$$

Она, предположительно, участвует в связывании int-белка с ДНК фага λ. Легко видеть, что основу выделенных повторов составляют периодичности t^6 и a^6.

Совершенные тандемные повторы в повествовательных текстах естественного языка встречаются редко и обычно сигнализируют об ошибке редактирования (повтор слова, строки и т. д.). Повтор осмысленный представляет собой специальную конструкцию, которую хорошо проиллюстрировал Б. Заходер: «В *чаще чаще* меньше пищи, значит в *чаще чаще* чище». Здесь тандем «чаще чаще» — это омографы — слова, которые пишутся и звучат одинаково только в определенной форме (числе, падеже, времени, лице).

В стихотворных текстах тандемные повторы слов и строк встречаются довольно часто и далеко не всегда с целью усиления значения, а скорее как элемент формообразования («Однажды *вечером, вечером, вечером*, когда пилотам, прямо скажем, делать нечего. . . »). Здесь имеет место своего рода заполнение повторами стихотворной строки. Применительно к музыкальным текстам термин «заполнение интервала» используется в ситуациях, когда значительный звуковысотный скачок, например на 4 ступени, заменяется серией шагов на одну ступень, образующих в интервальном представлении тандемный повтор с длиной периода 1 и кратностью 4 ($4 = 1 + 1 + 1 + 1$).

Тандемные повторы в знаменных песнопениях претендуют, по нашему мнению [16], на роль самостоятельных структурных единиц. Они важны в плане дешифровки, поскольку тандемному повтору на знаменном уровне не всегда соответствует нотолинейный повтор и наоборот. Их функциональная нагрузка разнообразна. Серии «стопиц» $\left(\text{Г}\right)^m$ соответствуют речитативным участкам. По насыщенности ими песнопений можно судить о датировке певческих рукописей [14]. Тандемы вида $\left(\text{Ґ}\ \text{Ґ}\right)^2$, $\left(\text{Ґ}\ \text{Ґ}\ \text{Г}\right)^2$, $\left(\text{Г}\ \text{Ґ}\ \text{Ґ}\right)$

 $)^2$, ...с длиной периода от 2 до 4 и кратностью повторения 2 или 3, составленные из простых по распеву высокочастотных знамен, обычно встречаются на стыках попевок (основных структурных единиц знаменного распева), регулируя расстояния между ними путем изменения длины периода и кратности повторений. Здесь явная аналогия с заполнением интервалов, но уже не на высотном, а на позиционном уровне. Тандемы «статей» в попевках или цепочек знамен, заканчивающихся статьей, например,  $)^2$,  $)^2$,  $)^2$ и др. усиливают кадансовую структуру попевок. И, наконец, короткие тандемы из достаточно сложных по распеву и редко используемых знамен являются индикаторами начертаний лиц и фит — наиболее ярких и нестандартных структурных единиц знаменного распева, служащих для его украшения. Такие индикаторы помогают вычленять эти структурные единицы (в первую очередь, лица) из текстов песнопений. К сожалению, этот признак носит факультативный характер: не все лица и фиты снабжены им.

Несовершенные периодичности (тандемные повторы с искажениями). Практически во всех эволюционирующих языковых системах наряду с совершенными повторами встречаются и несовершенные. Доля последних как минимум сопоставима с совершенными повторами, а чаще всего превалирует. Характер искажений на нижних уровнях языковой иерархии в большинстве случаев точечный: одиночные замены, короткие вставки и делеции (устранения) символов. На более высоких уровнях те же операции приобретают «блочный» характер, т. е. применяются к цепочкам символов (примером в русском языке может служить замена корней слов при сохранении аффиксального окружения [18]). Характерной для верхних уровней является операция транспозиции, связанная с переносом языковых форм из одного места текста в другое или (в музыкальных текстах) с одного звуковысотного уровня на другой (секвентные переносы). Последовательности дисков в политенных хромосомах искажаются путем инверсий [12] и т. д.

Единого определения, как уже отмечалось, и, соответственно, алгоритма отыскания несовершенной периодичности не существует. В немалой степени это обусловлено многообразием возможных способов искажения последовательностей. Для простейшей модели порождения ДНК-последовательностей, путем дупликации фрагментов произвольной длины с последующими их точечными искажениями, предложено несколько достаточно эффективных алгоритмов [19, 20], дающих близкие, но не тождественные результаты.

В сложностных разложениях несовершенные периодичности проявляют себя при наличии в периодах «совершенных» ядер, что имеет место достаточно часто. Важно отметить, что точечные мутации , накладывающиеся на совершенную периодичность, зачастую способствуют формированию регуляторных структур. Так, приводимый ниже фрагмент генома бактериофага λ содержит несовершенную периодичность с периодом длины 13 (выделен скобками):

$$\begin{array}{c} \text{поз. } 6112 \downarrow \qquad \text{I} \qquad\quad RBS \qquad \text{II} \quad \text{начало гена } E \\ \ldots(g\ g\,c\,t\,t\,t\,t\,t\,t\,t\,a\,c\ \underline{g)(g\,g\,a}\ t\,t\,t\,t\,t\,t\,t\ \overrightarrow{\underline{a\,t\,g)}}\,t\,c\,g\ldots \end{array}$$

Две замены, которыми II отличается от I формируют рибосомный сайт связывания RBS (подчеркнут) и инициирующий кодон atg, т.е. элементы, обеспечивающие начало трансляции гена E. В [10] приводятся и другие примеры на эту тему.

Интересным частным случаем совершенных и несовершенных периодичностей являются фрактальные и фракталоподобные структуры. Так мы называем периодичности, образованные повторением палиндрома, например ($aga\,aga\,aga\,\ldots$) или комплементарного палиндрома ($acgt\,acgt\,acgt\ldots$). Применительно к совершенным периодичностям такого

рода используется термин фрактальная структура, а к несовершенным — фракталоподобная. Это связано с проявлениями самоподобия в том смысле, что повторение палиндрома любого типа приводит к образованию аналогичной структуры вдвое большей длины, т.е. имеет место «усиление закономерности» (см. примеры «а» и «б»)

$$\text{(а)}\ \ldots \overbrace{tac\,cat}\,\overbrace{tac\,cat}\ldots;\ \text{б)}\ \overrightarrow{actg}\,\overleftarrow{cagt}\,\overrightarrow{actg}\,\overleftarrow{cagt}.$$

Здесь расходящиеся стрелки соответствуют палиндромам (случай «а»), а сходящиеся — комплементарным палиндромам (случай «б»).

В [8] авторами описан алгоритм отыскания фрактальных и фракталоподобных структур в режиме скользящего окна для случая, когда искажение самих палиндромов не допускается, но возможны вставки между ними, размер которых не превышает заданного порога r. Фрагмент *agagaagactagattcaagatcaga*, например, при $r = 4$ относится к категории фракталоподобных структур с повторяющимся (базовым) палиндромом «aga» .

Периодичности со сложной структурой . Многие периодичности (как совершенные, так и несовершенные) имеют иерархическую структуру в том смысле, что внутри большого периода могут, в свою очередь, присутствовать периодичности с меньшей длиной периода или другие регулярности. Так, в [21, 22], например, рассматриваются структуры вида $(Xx^n)^m$, где X и x — цепочки символов, $X \neq x$, n и m — целые, большие 1, а также структуры с переменным значением $n : x^{n_0}Xx^{n_1}X\ldots x^{n_{l-1}}Xx^{n_l}$, где $l > 2$, $n_i \geqslant 1$ для $i = 1, \ldots, l-1$ и хотя бы одно из $n_i \geqslant 2$. Допускается наличие ограниченных искажений в X и x. Фактически речь идет об обнаружении тандемных повторений цепочки x, прерываемых одиночными вставками цепочки X, причем количество таких вставок должно быть не менее трех. Представляет интерес то, что расстояния между вставками X регулируются количеством и длиной тандемных повторов x. О таком способе разнесения значимых структурных единиц (в данном случае цепочек X) на «нужное расстояние» мы упоминали в связи с обсуждением функциональной нагрузки тандемных повторов в знаменных песнопениях.

Другой интересный случай вставок теперь уже неидентичных цепочек в периодическую структуру выявлен нами с помощью сложностного разложения в геноме микоплазмы «Mycoplasma synoviae 53» (ID AE017245) . Соответствующий фрагмент текста (начальная позиция 690229) представлен в виде выравнивания:

$$\begin{array}{cc} X & Y \\ gttttggggttgtacaattattttgttaagtaaaac & aaatgataaataacgcttaacttgcttact \\ gttttggggttgtacaattattttgttaagtaaaac & cctataaacaaatcaggattatatgtacta \\ gttttggggttgtacaattattttgttaagtaaaac & ttaagtcaagatttttaataccagggtgca \\ gttttggggttgtacaattattttgttaagtaaaac & tccatatttttccttactattactatgct \text{ и т. д.} \end{array}$$

Структура имеет вид $XY_1XY_2XY_3\ldots$ (свыше 10 повторений), где $X = gt^4g^4t^2gtaca^2t^2at^4gt^2a^2gta^4c$ ($|X| = 36$) — регулярно повторяющийся фрагмент, а вставки Y_i, $i = 1, 2\ldots$, $|Y_i| \approx 30$, не обнаруживают значимого сходства. Информация об этой структуре в разметке генома отсутствует. Аналогичная, но более сильная структура указана в разметке другого представителя этого семейства «Mycoplasma gallisepticum str. R(low)» (ID AE015450). По-видимому, она выявлена с помощью инструмента CRISPFinder [23].

Подводя итог, отметим, что граница между совершенными и несовершенными периодичностями может быть размытой, если искажения носят регулярный характер. Так, приведенная в [21] структура $((cagta)(cagca)(cagta)(caaca))^3$, представленная здесь как совершенная периодичность с длиной периода 20, может рассматриваться и как несовершенная

периодичность с длиной периода 5, отличающаяся от $(cagta)^{12}$ девятью заменами, и как несовершенная периодичность с длиной периода 10, отличающаяся от $((cagta)(cagca))^6$ тремя заменами. Структуры, допускающие многозначную трактовку, выделены авторами [21] в отдельный класс.

Кумулятивные структуры. Это интересный класс структур, характеризующийся специфическими проявлениями повторности. Длина повторяющейся единицы нарастает или убывает на каждом шаге. Схема формирования кумулятивной структуры пошаговая. Например, в известном переводе С. Маршака английской песенки «The house that Jack built» эта схема имеет вид: $A + BA + CBA + \cdots$

$A =$ Вот дом, который построил Джек.
$BA =$ А это пшеница, которая в темном чулане хранится в доме, который построил Джек.
$CBA =$ А это веселая птица — синица, которая часто ворует пшеницу, которая в темном чулане хранится в доме, который построил Джек ...

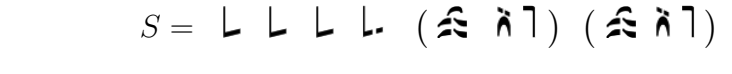
Подобные схемы встречаются и в ДНК-последовательностях. Приведем одну из них, в которой дуплицируются уже последовательно уменьшающиеся фрагменты из генома фага λ, представленные для наглядности в виде выравнивания:

```
Позиции
21479:  g a t t t g - - g g g a c g a a a a c c a c a  t  c g t c g t t t
21509:  g a t t a c - - g g g a c g a a a a c a a c a - g g c a g t t t c
21539:  g a t t a c - - g g c a c c a a a t c g a c g
21560:  a a t a a c a  c g g g g
```

В [10] подобного рода схемы названы «иерархической дупликацией», а в музыке — «вариационностью типа прорастания» [2], характеризующейся тем, что тематическое ядро при повторении получает новое продолжение. И, наконец, стоит упомянуть целый пласт детских сказок («Репка», «Колобок» и др.), классифицируемых как цепочные, кумулятивные или сказки с «нанизыванием», т. е. присоединением новой информации при повторении уже известной [3]. Считается, что такие сказки способствуют развитию устной речи у детей.

Комбинированные структуры. К ним мы относим позиционно сближенные (или даже налагающиеся одна на другую) комбинации элементарных структур (разнотипных повторов) и периодичностей. Простейшим примером могут служить «компаунды» — примыкающие друг к другу периодичности, отличающиеся составом элементов в периодах или даже длиной периода. Так, в упомянутой выше подборке секвенированных фрагментов генома пшеницы мы наблюдали компаунды $(ca)^{19}(at)^{14}$; $(ag)^7(agat)^{10}$; $(tc)^{11}(ta)^{12}(ac)^7$ и много других. Обращает на себя внимание наличие общего элемента в разнотипных периодах каждого компаунда: чаще всего на нем и осуществляется стыковка соседних периодичностей. На данный момент не совсем понятно, несет ли какую-то смысловую нагрузку позиционное сближение периодичностей, образующих компаунд. Применительно к естественному языку этот вопрос звучал бы так: образует ли какая-то пара соседних слов в тексте устойчивое словосочетание или нет?

Компаунды встречаются и в других языковых системах. Песня «В темном лесе» представляет собой совершенный компаунд. Приведем один пример из знаменных песнопений:



Здесь компаунд образован серией «стопиц» $\left(\text{┗}\right)^3$ и тандемным повторением цепочек длины 3 (в скобках). За компаундом следует «статья мрачная» $\left(\text{}\right)$, которая интерпретируется целой нотой, обычно завершающей структурные единицы знаменного распева. С большой вероятностью фрагмент $S[4:11]$ относится к категории «лиц», что подтверждается индикатором в виде тандемного повтора и ритмическим остановом, реализуемым статьей. Распевам лиц и фит нередко предшествует речитативный участок, называемый «разбегом стопиц». Здесь он представлен серией из трех стопиц. Таким образом, позиционное сближение двух видов тандемных повторов выглядит закономерным.

Пример наложения разнотипных ДНК-повторов иллюстрирует фрагмент эукариотического промотора (регуляторной структуры, ответственной за начало транскрипции) [7]:

$$\ldots ag\,\overset{1}{\overline{g}}\,c\,\overrightarrow{cgggc}^{\,1}\,g\,\overleftarrow{ccgcct}^{\,2}\,\overrightarrow{tccgcc}^{\,2}\,t\,\overleftarrow{gcccg}^{\,1}\,c\,\overset{1}{\overline{c}}\,t\ldots$$

Здесь симметричный повтор (2) фланкирован разнесенными симметричными комплементарными повторами (1).

Межъязыковые аналогии на уровне содержательных задач и подходов к их решению

Описанные выше образцы структурного сходства, обнаруживаемого в текстах различных языковых систем, позволяют предположить, что межъязыковые аналогии распространяются также на постановку содержательных задач и выработку подходов к их решению. Возможны переносы идей и технологий из одной языковой системы в другую. Упомянем в связи с этим сходные по постановкам и используемым алгоритмам задачи выявления гомологий в ДНК-последовательностях, неосознанных заимствований в музыкальных произведениях, плагиатов в научных текстах. Другим примером может служить формализация понятия «структурная единица» и разработка алгоритмов автоматического выделения структурных единиц из текстов конкретных языковых систем. В частности, сходные подходы использованы для выделения: (а) морфем из текста на русском языке, записанного без пробелов и знаков препинания [24]; (б) попевок — основных структурных единиц знаменного распева [15]; (в) устойчивых словосочетаний (коллокаций) из русскоязычных текстов [25].

Чуть подробнее осветим возможность переноса понятийного аппарата из одной языковой системы в другую. Так, весьма плодотворным в плане выявления локальных структур в биологических текстах оказалось понятие «сложностного профиля». Возникает вопрос, что могло бы служить его аналогом для текстов на естественном языке, где проявления повторности не так заметны? Напомним, что аномальные по сложности фрагменты в биологических текстах — это участки с высокой концентрацией разнотипных повторов. Поэтому в текстах на естественном языке следует обратить внимание на фрагменты, характеризующиеся аномально высокой (по сравнению с оставшейся частью текста) концентрацией вхождений какой-либо лексической единицы (слова или словосочетания). Такого рода закономерности будем называть позиционной кластеризацией лексических

единиц. Выделенные фрагменты, соответствующие разным лексическим единицам, могут пересекаться, оказаться вложенными один в другой или разнесенными. Одно и то же предложение текста может покрываться разными фрагментами. Имея эту информацию, можно определить профиль кластеризуемости лексических единиц в тексте как ступенчатую функцию, аргументом которой является порядковый номер предложения в тексте, а значение равно числу различных фрагментов, включающих в себя данное предложение. Поскольку каждый фрагмент связан с конкретной лексической единицей, то в каждой точке профиля фиксируется совокупность лексических единиц, определяющих локальное содержание данного участка текста.

В основе построения профиля кластеризуемости лежит отбор лексических единиц, демонстрирующих аномалии в позиционном распределении. Для этого используется аппарат сканирующих статистик . В частности, статистика $d(n, x)$ фиксирует длину минимального интервала, содержащего ровно n последовательных вхождений лексической единицы x $(2 \leqslant n \leqslant F(x))$, где $F(x)$ — частота встречаемости x в тексте. Распределение этой статистики при случайной расстановке x вдоль текста известно. Если наблюдаемое значение $d(n, x)$ при каком-то значении n аномально мало по сравнению с ожидаемым при равномерном распределении, фиксируется наличие позиционного кластера. В [13] описанный подход рассмотрен более детально. Пикам профиля кластеризуемости сопоставлены структурные единицы более высокого уровня, чем предложение — так называемые «сверхфразовые единства», определяющие макроструктуру текста. Предложен способ построения квазиреферата текста на основе профиля кластеризуемости.

Заключение

Основными структурообразующими элементами в текстах, представляющих различные эволюционирующие языковые системы, являются повторы. Номенклатура повторов очень широка. Существуют повторы, типичные для всех языковых систем и присущие лишь отдельным языковым системам. Повторяющиеся цепочки символов отличаются своей длиной, составом элементов алфавита, частотой встречаемости в тексте и характером распределения по длине текста. Наиболее значимыми считаются цепочки, распределенные неравномерно. Одним из наиболее типичных проявлений неравномерности позиционного распределения является высокая концентрация повторов разного типа в ограниченном участке текста. В работе проведена типизация таких участков, сопровождаемая примерами из различных языковых систем (биологические последовательности, тексты на естественном языке, знаменные песнопения). Обсуждается их функциональная нагрузка в разных языковых системах.

Особое внимание уделено межъязыковым аналогиям. Предполагается, что структурам, наблюдаемым в одной языковой системе, могут быть найдены аналоги и в других языковых системах. Более того, межъязыковые аналогии могут распространяться и на постановку различных содержательных задач, а также отыскание способов их решения.

Нами разработаны в рамках сложностного подхода и апробированы на реальном материале эффективные (квазилинейные) алгоритмы отыскания описанных выше структур (см. [5, 7, 8, 11, 12, 13, 16, 25]). Из решенных (и решаемых) прикладных задач отметим разработку формальных методов для выделения (и дешифровки) структурных единиц знаменного распева [15, 16], сверхфразовых единств в текстах на естественном языке, используемых для автоматического построения квазирефератов текста [13], фракталоподобных и комбинированных структур в частично секвенированном геноме пшеницы [11]. Последние могут выполнять роль регуляторов основных генетических процессов — тран-

скрипции, трансляции и др. Разработаны модификации метода для выявления сходства (различия) пар или групп текстов, допускающих нестандартные редакционные операции, в частности, инверсии (в перестановках) и транспозиции. Они использованы для структурного (и филогенетического) анализа уникальных данных — последовательностей дисков политенных хромосом [12]. Эти же методы применимы для установления авторства литературного или музыкального произведения, но результаты существенно зависят от объема предоставленного материала. В качестве удачного примера можно указать на работу [26], близкую в идейном плане к описываемому подходу.

Литература

[1] *Lempel A., Ziv J.* On the complexity of finite sequences // *IEEE Trans. Inf. Theor.*, 1976. Vol. IT-22, no. 1. P. 75–81.

[2] *Протопопов В.* Вариационные процессы в музыкальной форме. М.: Музыка, 1967. 150 с.

[3] *Пропп В. Я.* Кумулятивная сказка // *Фольклор и действительность. Избр. статьи.* М.: Наука, 1976. С. 242–249.

[4] *Колмогоров А. Н.* Три подхода к определению понятия «количество информации» // *Проблемы передачи информации*, 1965. Т. 1, № 1. С. 3–11.

[5] *Gusev V. D., Nemytikova L. A., Chuzhanova N. A.* On the complexity measures of genetic sequences // *Bioinformatics*, 1999. Vol. 15, no. 12. P. 994–999.

[6] *Гусев В. Д., Мирошниченко Л. А.* Использование сложностных разложений в задачах анализа символьных последовательностей // *Докл. 8-й Междунар. конф. «Интеллектуализация обработки информации» (ИОИ-2010).* Кипр, Пафос, 2010. С. 469–472.

[7] *Гусев В. Д., Мирошниченко Л. А.* Поиск комбинированных структур в ДНК-последовательностях // *Докл. Всеросс. конф. ММРО-13 «Математические методы распознавания образов».* М.: Макс-Пресс, 2007. С. 473–476.

[8] *Гусев В. Д., Мирошниченко Л. А., Чужанова Н. А.* Выявление фракталоподобных структур в ДНК-последовательностях // *Classification, forecasting, data mining. Information science and computing international book ser.* Sofia: ITHEA, 2009. No. 8. P. 117–123.

[9] *Михайлов В. Г., Шойтов А. М.* О числах множеств эквивалентных цепочек в последовательности независимых случайных величин // *Математические вопросы криптографии*, 2013. Т. 4. № 1. С. 77–86.

[10] *Гусев В. Д., Куличков В. А., Чупахина О. М.* Сложностной анализ генетических текстов (на примере фага λ). Новосибирск, 1989. Препринт № 20 ИМ СО РАН. 41 с.

[11] *Sergeeva E. M., Afonnikov D. A. , Koltunova M. K. , Gusev V. D., Miroshnichenko L. A., Vrána J., Kubaláková M., Poncet C. , Sourdille P. , Feuillet C., Doležel J., Salina E. A.* Common wheat chromosome 5B composition analysis using low-coverage 454 sequencing // *Plant Genome*, 2014. Vol. 7, no. 2. doi: 10.3835/plantgenome2013.10.0031.

[12] *Gunderina L. I., Kiknadze I. I., Istomina A. G., Gusev V. D., Miroshnichenko L. A.* Divergence of polytene chromosome band sequences as a reflection of evolutionary reorganization of the linear structure of the genome // *Rus. J. Genet.*, 2005. Vol. 41, no. 2. P. 130–137.

[13] *Гусев В. Д., Мирошниченко Л. А., Саломатина Н. В.* Тематический анализ и квазиреферирование текста с использованием сканирующих статистик // *Компьютерная лингвистика и интеллектуальные технологии. Тр. Междунар. конф. Диалог'2005.* М.: Наука, 2005. С. 121–125.

[14] *Бражников М. В.* Пути развития и задачи расшифровки знаменного распева XII–VIII веков. Л., М.: Гос. муз. изд., 1949. 103 с.

[15] *Бахмутова И. В., Гусев В. Д., Титкова Т. Н.* L-граммные азбуки для дешифровки знаменных песнопений // *Сибирский журнал индустриальной математики*, 1998. Т. 1, № 2. С. 51–66.

[16] *Бахмутова И. В., Гусев В. Д., Мирошниченко Л. А., Титкова Т. Н.* Тандемные повторы в знаменных песнопениях // *Анализ структурных закономерностей: Вычислительные системы*, 2005. Вып. 174. С. 13–28.

[17] *Matescu A., Salomaa A.* Aspects of classical laguage theory // *Handbook of formal languages*, 1996. Vol. 1. P. 230–242.

[18] *Саломатина Н. В.* Количественные исследования морфемной структуры слов русского языка (на базе электронного словаря Д. Уорта) // *Обнаружение эмпирических закономерностей: Вычислительные системы*, 1999. Вып. 166. С. 104–118.

[19] *Benson G.* Tandem repeats finder: A program to analyze DNA sequences // *Nucleic Acids Res.*, 1999. Vol. 27, no. 2. P. 573–580.

[20] *Sokol D., Benson G., Tojeira J.* Tandem repeats over the edit distance // *Bioinformatics*, 2007. Vol. 23, no. 2. P. e30–e35.

[21] *Hauth A. M, Joseph D. A.* Beyond tandem repeats: Complex pattern structures and distant regions of similarity // *Bioinformatics*, 2002. Vol. 18. Suppl. 1. P. s31–s37.

[22] *Matroud A. A., Hendy M. D, Tuffley C. P.* NTRFinder: a software tool to find nested tandem repeats // *Nucleic Acids Res.*, 2012. Vol. 40, no. 3. P. e17.

[23] *Grissa I., Vergnaud G., Pourcel C.* CRISPRFinder: A web tool to identify clustered regularly interspaced short palindromic repeats // *Nucl. Acids Res.*, 2007. Vol. 35. Suppl. 2. P. W52–W57.

[24] *Сухотин Б. В.* Морфологический анализ текста без пробелов // *Оптимизационные методы исследования языка.* М.: Наука, 1976. С. 73–169.

[25] *Гусев В. Д, Саломатина Н. В.* Алгоритм выявления устойчивых словосочетаний с учетом их вариативности (морфологической и комбинаторной) // *Компьютерная лингвистика и интеллектуальные технологии. Тр. Междунар. конф. Диалог'2004.* М.: Наука, 2004. С. 530–535.

[26] *Кукушкина О В., Поликарпов А. А., Хмелев Д. В.* Опреление авторства текста с использованием буквенной и грамматической информации // *Проблемы передачи информации*, 2001. Т. 37. Вып. 2. С. 96–109.

References

[1] *Lempel A., Ziv J.* 1976. On the complexity of finite sequences. *IEEE Trans. Inf. Theor.* IT-22(1):75–81.

[2] *Protopopov V.* 1967. Variation processes in a musical form. Moscow: Music. 150 p.

[3] *Propp V. Ia.* 1976. Cumulative fairy tale. *Folklore and Reality.* Moscow. 242–249.

[4] *Kolmogorov A. N.* 1965. Three approaches to the quantitative definition of information *Problemy Peredachi Informatsii [Problems of Information Transmission]* 1(1):3–11.

[5] *Gusev V. D., Nemytikova L. A., Chuzhanova N. A.* 1999. On the complexity measures of genetic sequences. *Bioinformatics* 15(12):994–999.

[6] *Gusev V. D., Miroshnichenko L. A.* 2010. Complexity decompositions in problems of symbolic sequences analysis. *8th Conference (International) "Intelligent Information Processing" (IIP-2010).* Cyprus, Paphos. 469–472.

[7] *Gusev V. D., Miroshnichenko L. A.* 2007. Detection of the combined structures in DNA sequences. *13th All-Russian Conference "Mathematical methods of pattern recognition".* Zelenogorsk (Leningrad Region). 473–476.

[8] *Gusev V. D., Miroshnichenko L. A., Chuzhanova N. A.* 2009. Identification of the fractal-like structures in DNA sequences. *Classification, forecasting, data mining.* Information science and computing international book ser. Sofia: ITHEA. 8:117–123.

[9] *Mikhailov V. G., Shoitov A. M.* 2013. About numbers of sets of equivalent chains in sequence of independent random variables. *Mathematical Questions Cryptography* 4(1):77–86.

[10] *Gusev V. D., Kulichkov V. A., Chupakhina O. M.* 1989. *Complexity analysis of genetic texts (on the example of a phage λ).* Novosibirsk. Preprint No. 20 IM SB RAS. 41 p.

[11] *Sergeeva E. M., Afonnikov D. A., Koltunova M. K., Gusev V. D., Miroshnichenko L. A., Vrána J., Kubaláková M., Poncet C. , Sourdille P. , Feuillet C., Doležel J., and Salina E. A.* 2014. Common wheat chromosome 5B composition analysis using low-coverage 454 sequencing. *Plant Genome* 7(2). doi: 10.3835/plantgenome2013.10.0031.

[12] *Gunderina L. I., Kiknadze I. I., Istomina A. G., Gusev V. D., Miroshnichenko L. A.* 2005. Divergence of polytene chromosome band sequences as a reflection of evolutionary reorganization of the linear structure of the genome. *Rus. J. Genet.* 41(2):130–137.

[13] *Gusev V. D., Miroshnichenko L. A., SalomatinaN. V.* 2005. The thematic analysis and quasiabstracting of the text with the scan statistics using. *Conference (International) "Computational Linguistics and Intellectual Technologies" (Dialogue-2005).* Moscow. 121–125.

[14] *Brazhnikov M. V.* 1949. *Puti razvitiia i zadachi rasshifrovki znamennogo rospeva XII-XVIII vekov.* Leningrad; Moscow: Gos. Muz. Izd-vo. 103 p. (In Russian).

[15] *Bakhmutova I. V., Gusev V. D., Titkova T. N.* 1998. *L*-gramm alphabet for deciphering the neume hymns. *J. Appl. Ind. Math.* 1(2):51–66.

[16] *Bakhmutova I. V., Gusev V. D., Miroshnichenko L. A., Titkova T. N.* 2005. Tandem repeats in the neume hymns. *Computing Syst. Analysis of structural regularities* 174:13–28.

[17] *Matescu A., and Salomaa A.* 1996. Aspects of classical laguage theory. Handbook of Formal Languages. 1:230–242.

[18] *Salomatina N. V.* 1999. Quantitative researches of morphemic structure of words of Russian (on the basis of the electronic dictionary of D. Worth) *Computing systems. Detection of empirical regularities* 166:104–118.

[19] *Benson G.* 1999. Tandem repeats finder: A program to analyze DNA sequences *Nucleic Acids Res.* 27(2):573–580.

[20] *Sokol D., Benson G., Tojeira J.* 2007. Tandem repeats over the edit distance. *Bioinformatics* 23(2):e30–e35.

[21] *Hauth A. M., Joseph D. A.* 2002. Beyond tandem repeats: Complex pattern structures and distant regions of similarity. *Bioinformatics* 18(1):s31–s37.

[22] *Matroud A. A., Hendy M. D., Tuffley C. P.* 2012. NTRFinder: A software tool to find nested tandem repeats. *Nucleic Acids Res.* 40(3):e17.

[23] *Grissa I., Vergnaud G., Pourcel C.* 2007. CRISPRFinder: A web tool to identify clustered regularly interspaced short palindromic repeats. *Nucl. Acids Res.* 35(2):W52–W57.

[24] *Sukhotin B. V.* 1976. Morphological analysis of the text without gaps. *Optimization methods in language research.* Moscow: Nauka. 73–169.

[25] *Gusev V. D., Salomatina N. V.* 2004. Algorithm of identification of stable word combination taking into account their variability (morphological and combinatory). *Conference (International) "Computational Linguistics and Intellectual Technologies" (Dialogue-2004).* Moscow. 530–535.

[26] *Kukushkina O V., Polikarpov A. A., Khmelev D. V.* 2001. Using literal and grammatical statistics for authorship attribution. *Problems Information Transmission* 37(2):172–184.

Комплекс алгоритмов интеллектуального анализа сложно организованных данных при исследовании слабо формализованных систем управления*

Ю. А. Дорофеюк[1,2]***, И. В. Покровская***[1]***, Н. Е. Киселева***[1]
dorofeyuk_julia@mail.ru
[1]Москва, Институт проблем управления им. В. А. Трапезникова РАН (ИПУ РАН);
[2]Москва, Научно исследовательский университет Высшая школа экономики (НИУ ВШЭ)

Рассматривается задача исследования системы управления заданного множества объектов, каждый из которых характеризуется фиксированным (исходным) набором разнородных параметров. Для решения этой задачи предлагается исследовать структуру взаиморасположения управляемых объектов в пространстве информативных параметров. Это позволяет существенно повысить эффективность анализа функционирования системы, а также устойчивость процедур принятия управленческих решений. Для выявления такой структуры разработан специальный комплекс алгоритмов интеллектуального анализа сложно организованных данных, а также процедур экспертной коррекции. Проведен теоретический анализ различных вариантов алгоритма СКАД (структурно-классификационного анализа данных), доказаны теоремы о сходимости алгоритма к локальному экстремуму соответствующего критерия качества.

Ключевые слова: *интеллектуальный структурно-классификационный анализ данных; информативные параметры; начальное разбиение, выбор числа классов; заполнение пропущенных наблюдений; процедуры экспертной коррекции*

The complicated data mining algorithms complex in the study of weakly formalized management systems*

Yu. A. Dorofeyuk[1,2]***, I. V. Pokrovskaya***[1]***, and N. E. Kiseleva***[1]
[1]ICS RAS; [2]SIU HSE

The problem of the large-scale management system study is considered. The system consists of a large number of objects, each of which is characterized by a heterogeneous set of parameters. To solve the set of problems, it is proposed to investigate the structure of the relative location of these objects in the informative parameters space. This allows to significantly increase the analysis efficiency of the system functioning and the stability of the procedures for making management decisions. To identify such patterns, special mining complicated data algorithms complex and expert correction procedures were designed. The theoretical analysis of various types of SCDA algorithm was carried out, the algorithm convergence to the local extremum of the appropriate quality criterion theorems were proved.

Keywords: *intellectual structural-classification data analysis; informative parameters; initial partitioning; the number of classes selection; filling in missing observations expert correction; procedures*

*Работа выполнена при частичной финансовой поддержке РФФИ, гранты № 14-07-00463-а, № 13-07-00992-а, № 13-07-12201-офи, № 12-07-00540-а.

Введение

В последнее время для исследования сложных систем управления стали широко использоваться структурно-классификационные методы интеллектуального анализа данных, базирующиеся на алгоритмах классификационного анализа [1, 2]. Это объясняется тем, что многие системы управления, в первую очередь организационно-административные, функционируют в условиях большой информационной размытости и неопределенности.

В работе рассматривается задача анализа функционирования системы управления заданного множества объектов, каждый из которых характеризуется фиксированным (исходным) набором разнородных параметров. Основная идея предлагаемого метода решения подобных задач состоит в следующем. В работе предлагается исследовать не точные значения параметров, описывающих состояние каждого объекта системы, а лишь структуру взаиморасположения этих объектов в пространстве параметров. Такое интегральное описание управляемых объектов, позволяет существенно повысить эффективность анализа поведения системы, а также устойчивость и робастность процедур принятия управленческих решений. Для формализации такой задачи используется методология классификационного анализа данных [1, 2].

Пусть исследуемая система состоит из n объектов, каждый из которых характеризуется набором из k параметров. Вводится в рассмотрение k-мерное пространство параметров X, в котором каждый объект представляется точкой $x_j = (x_j^{(1)}, ..., x_j^{(k)})$, $j = 1, \ldots, n$. Предполагается, что вектор значений параметров x_j достаточно полно характеризует состояние j-го объекта, т. е. взаиморасположение множества точек $x_1, \ldots, x_n$ в пространстве параметров X отражает реальную структуру исследуемого множества объектов. Для выявления такой структуры был разработан комплекс алгоритмов интеллектуального анализа данных и процедур экспертной коррекции, включающий алгоритмы: структурно-классификационного анализа данных (СКАД), выбора информативных параметров, выбора начального разбиения, выбора числа классов, заполнения пропущенных наблюдений, а также процедуры экспертной коррекции результатов работы этих алгоритмов. Далее каждый из этих алгоритмов рассматривается отдельно.

Алгоритм структурно-классификационного анализа данных

Пусть задано R_0 — некоторое начальное разбиение (классификация) точек классифицируемой выборки $x_1, \ldots, x_n$, на r классов A_i, $i = 1, \ldots, r$. Алгоритм итерационный, циклический, многоэтапный — на j-м шаге l-го этапа рассматривается некоторый набор из l точек X_j^l из исходной последовательности $x_1, \ldots, x_n$, принадлежащих одному и тому же классу, где j — номер этого набора. Номер этапа l равен мощности множества точек, которые «перебрасываются» на каждом шаге этого этапа из класса в класс, т. е. числу точек в наборе. На j-м шаге происходит пробная «переброска» из класса в класс множества точек X_j^l. Тогда X_j^l относится к тому классу A_s, значение критерия качества классификации J для которого будет наибольшим, т. е. $X_j^l \in A_s$, для которого $A_s = \arg\max\limits_{A_l} J(X_j^l \in A_i)$, $i = 1, \ldots, r$, $j = 1, \ldots, N_l$, где N_l — число различных наборов из l точек в исходной выборке, принадлежащих одному и тому же классу. На следующем шаге l-го этапа процедура повторяется для множества X_{j+1}^l. Число шагов (итераций) на l-м этапе равно N_l и определяется выражением $N_l = \sum\limits_{i=1}^{r} C_{n_i}^l$ для $n_i \geqslant (l+2)$, где C_m^k – число сочетаний из m по k. Из этого выражения следует, что для всех итераций l-го этапа процедура не применяется

для таких классов A_i, число точек n_i в которых меньше, чем $(l+2)$. Число этапов (глубина перебора) $l_{\max}$ либо фиксируется заранее, либо выбирается из условия: в классификации, полученной после $(l-1)$-го этапа, должен быть хотя бы один класс, число точек в котором не меньше $(l+2)$. Это правило обеспечивает автоматический выбор максимально возможной глубины перебора $l_{\max}$. Для повышения эффективности алгоритма СКАД используется следующая циклическая процедура. После завершения последнего этапа (либо m-й, либо $l_{\max}$-й) весь описанный выше цикл повторяется заново, только в качестве начальной классификации используется не R_0, а классификация, полученная на последнем этапе первого цикла. Алгоритм СКАД заканчивает работу, если на некотором цикле среди точек $x_1, \ldots, x_n$ не будет сделано ни одной «переброски» из класса в класс, т. е. для этого цикла начальная классификация совпадает с конечной. Доказана следующая теорема о сходимости этого алгоритма.

Теорема 1. *Алгоритм СКАД сходится за конечное число шагов к локальному максимуму критерия J.*

Доказательство. Без ограничения общности дается для случая двух классов. По процедуре работы алгоритма СКАД значения критерия J образуют монотонно неубывающую, ограниченную сверху последовательность. Величина ограничения $C_1 \geqslant J_i$, $i \in D$, где D — множество номеров всех возможных дихотомий исходной выборки $x_1, ..., x_n$, зависит от вида выбранного критерия J. С другой стороны, в силу конечности исходной выборки существует такая константа C_2, что $C_2 \leqslant \mid J_i - J_j \mid$, $i, j \in D$, $i \neq j$. Таким образом, может быть только конечное число шагов N, на которых критерий J возрастает: $N \leqslant C_1/C_2$. На некоторых шагах работы алгоритма СКАД возможны ситуации, когда при равенстве значений критерия J до и после «переброски» l точек происходит изменение принадлежности этих точек к классу. Для того чтобы не допустить возможности циклической последовательности таких «перебросок» (что приводит к бесконечному числу таких шагов), в алгоритме введено специальное правило для таких случаев: при равенстве значений критерия J до и после «переброски» соответствующие l точек относятся к классу с меньшим номером. Это означает, что таких перебросок с нулевым приращением значения критерия J также будет конечное число. Достижимость локального максимума непосредственно следует из самой процедуры «переброски» точек. Действительно, предположим противное — после останова значение критерия J не является l-локальным максимумом. А это по определению локального экстремума означает, что существует, по крайней мере, один набор из l точек исходной последовательности, для которого изменение принадлежности к классу приведет к увеличению значения критерия J. Однако правило останова гарантирует, что такого набора в исходной последовательности не существует, поскольку на последнем перед остановом цикле не было ни одной «переброски» l точек. Теорема доказана. ■

Алгоритм сокращенного перебора — эвристический вариант выбора множеств X_j^l. На каждом шаге алгоритма для пробной «переброски» используются точки в определенном смысле ближайшие к границе между классами. Иллюстрация идеи работы алгоритма представлена на рис. 1. Четыре точки, обведенные кружочками — это как раз и есть те l точек (рассматривается случай $l = 4$), которые на j-ом шаге ближе всего расположены к границе (квадратиками обозначены центры классов на j-м шаге). Если уравнение границы в явном виде неизвестно, то выбираются l точек, ближайших к эталону другого класса. Обычно в качестве эталона выбирается точка «центра тяжести» всех точек

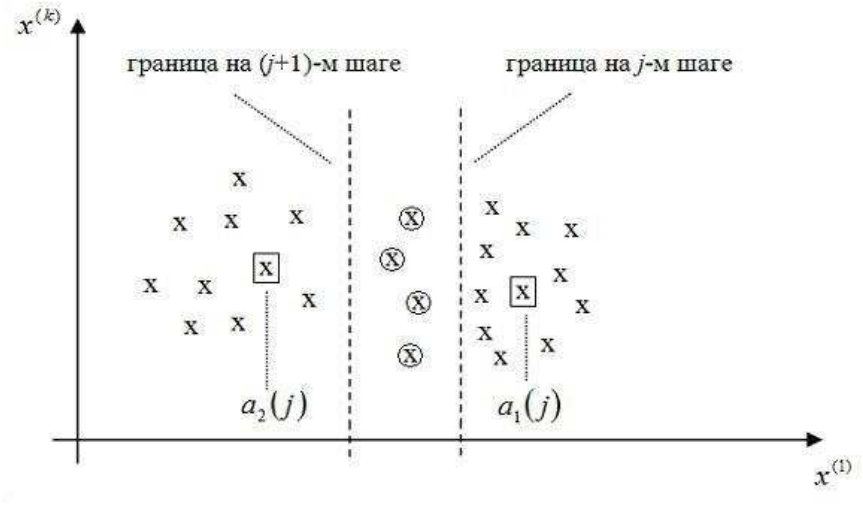


Рис. 1. Иллюстрация идеи сокращенного перебора

исходной выборки, принадлежащих на j-ом шаге соответствующему классу (т. е. среднему этого класса):

$$a_s(j) = \frac{1}{n_s(j)} \sum_{x_i \in A_s(j)} x_i, \tag{1}$$

где s — номер класса.

Алгоритм СКАД для одномерного случая. Необходимо специально отметить этот частный, но весьма распростаненный в прикладных задачах случай, связанный со структурным анализом временных рядов [3]. Дело в том, что одномерный случай имеет уникальное свойство, существенно упрощающее процедуру целенаправленного перебора, используемую при структурном анализе. А именно: ввиду одномерной упорядоченности классов границей между двумя классами (в детерминированном случае) служит только одна точка, и таких границ может быть не более двух (для крайних правого и левого классов — только одна).

Теорема 2. *Одномерный вариант алгоритма СКАД сходится за конечное число шагов к глобальному максимуму критерия J.*

Доказательство. Одномерный случай существенно отличается от многомерного тем, что классы упорядочены на оси X. Это, в свою очередь, позволяет декомпозировать процесс минимизации функционала J для всей выборки на независимые процедуры его минимизации для подвыборок, каждая из которых составляет одну из всех соседних пар классов. Таким образом, этот алгоритм фактически является реализацией схемы динамического программирования, обеспечивающей нахождение глобального экстремума функционала J [4].

Действительно, предположим противное, после завершения работы одномерного алгоритма СКАД получена классификация на r классов (обозначим ее через $H_{\text{лок}}$), доставляющая не глобальный, а лишь локальный экстремум $J_{\text{лок}}$ функционала J. Это означает, что существует такая классификация $H_{\text{глоб}}$, для которой значение функционала $J_{\text{глоб}}$ будет больше, чем $J_{\text{лок}}$.

Введем в рассмотрение пересечение двух классификаций $H_1 = \{A_{11}, ..., A_{1r}\}$ и $H_2 = \{A_{21}, ..., A_{2r}\}$: это множество $H_1 \cap H_2 = \{A_{1i} \cap A_{2i}, i = 1, \ldots, r\}$; а также их разность $H_1 \setminus H_2 = \{A_{1i} \setminus A_{2i}, i = 1, \ldots, r\}$.

Рассмотрим пересечение классификаций $H_{\text{глоб}} \cap H_{\text{лок}}$, а затем вычтем его из классификации $H_{\text{лок}}$. Обозначим получившийся в результате набор множеств точек через $B_1, \ldots, B_r$. Далее рассматриваются только непустые множества такого вида. Рассмотрим для примера множество B_1 (пусть оно содержит m_1 точек) из первого класса классификации $H_{\text{лок}}$. Рассмотрим этап алгоритма СКАД для одномерного случая, на котором анализируются точки только первого и второго классов, остальные границы считаются фиксированными. В качестве начальных условий выберем границу между первым и вторым классами из классификации $H_{\text{глоб}}$. В соответствии с работой алгоритма, точки множества B_1 должны быть «переброшены» в первый класс, так как по построению такой переброске будет соответствовать большее значение функционала J. Аналогичные рассуждения проводятся для всех множеств $B_i, i = 1, \ldots, r$. Следует подчеркнуть, что на каждом цикле рассмотрения пары соседних классов используется правило выбора максимально возможной глубины перебора $l_{\max}$, обеспечивающее глобальный экстремум критерия J для рассматриваемой пары классов (при фиксированных остальных классах).

Из вышеизложенного можно сделать вывод, что предположение о существовании классификации $H_{\text{глоб}}$, доставляющее большее значение функционалу J, чем классификация $H_{\text{лок}}$, неверно. Таким образом, полученная в результате работы алгоритма классификация доставляет глобальный экстремум функционалу J. ■

При моделировании и в приложениях в качестве критерия качества классификации J использовался функционал средней близости точек в классах, определяемый через потенциальную функцию $K(x, y)$ близости точек x и y [5]:

$$K(x, y) = \frac{1}{1 + \alpha R^p(x, y)}, \tag{2}$$

где α и p — настраиваемые параметры алгоритма. Средняя близость точек в классе определяется как число точек в классе A_1:

$$K(A_i, A_i) = \frac{2}{n_i(n_i - 1)} \sum_{i=1}^{n_i} \sum_{j>i} K(x_i, x_j), \tag{3}$$

где $K(x_i, x_j)$ определяется формулой 2. Тогда критерий J_1 определяется как

$$J_1 = \sum_{i=1}^{r} \frac{n_i}{n} K(A_i, A_i). \tag{4}$$

Во многих задачах структурно-классификационного анализа объекты по самой постановке задачи могут относиться к разным классам с различной степенью «достоверности». Для таких случаев была разработана постановка задачи размытого классификационного анализа [1, 2].

Вариант алгоритма СКАД в размытой постановке. Размытой классификацией множества X на r классов называется r-мерная вектор-функция $H(x) = (h_1(x), ..., h_r(x))$, где $h_i(x)$ — функция принадлежности объекта x к i-му классу, удовлетворяющая условию нормировки: $\sum_{i=1}^{r} h_i(x) = 1$, $0 \leqslant h_i(x) \leqslant 1$ [2].

Критерий оценки качества классификации содержательно остается прежним, только видоизменяется процедура подсчета его значений. А именно: функционал принимает следующий вид: $J_1 = \sum_{i=1}^{r} B_i K(A_i, A_i)$, где $B_i = \frac{1}{n}\sum_{j=1}^{n} h_i(x_j)$ — нормирующий множитель, аналогичный n_i/n для детерминированного случая. Величина $K(A_i, A_i)$ средней близости точек в классе A_i для размытого случая определяется по формуле:

$$K(A_i, A_i) = c_i \sum_{j=1}^{n} \sum_{l>j} K(x_j, x_l) h_i(x_j) h_i(x_l), \tag{5}$$

где $c_i = 2/\sum_{j=1}^{n} (h_i(x_j))^2$ — нормирующий множитель, аналогичный $2/n_i(n_i - 1)$ для детерминированного случая.

Рассмотрим вкратце работу ***размытого алгоритма СКАД***. Для простоты изложения и без ограничения общности рассмотрим случай двух классов ($r = 2$). Пусть задано начальное размытое разбиение $H_0 = \{h_i(x_j), i = 1, 2, j = 1, \ldots, n\}$ точек классифицируемой выборки $x_1, \ldots, x_n$, которое может задаваться либо изначально, либо при помощи специального алгоритма выбора начального разбиения. Для начального размытого разбиения H_0 подсчитывается значение критерия $J(H_0)$. Как и в детерминированном случае размытый алгоритм является циклическим, многоэтапным, итерационным – на j-м шаге l-го этапа рассматривается некоторый набор из l точек X_j^l из исходной последовательности $x_1, \ldots, x_n$. При этом для всех точек множества X_j^l справедливо неравенство $h_l(x_s) > h_k(x_s)$, $l \neq k$, что соответствует требованию для детерминированного алгоритма: все точки множества X_j^l принадлежат одному и тому же классу. Для множества точек ^l_j выполняется следующая операция, аналогичная «переброске» точек из класса в класс. Вводится понятие «старой» и «новой» функций принадлежности $h_i(x_s)_{\text{стар}}$ и $h_i(x_s)_{\text{нов}}$ соответственно, $x_s \in X_j^l$. Тогда, если для всех $x_s \in X_j^l$ выполняется $h_1(x_s)_{\text{стар}} > h_2(x_s)_{\text{стар}}$ (аналог того, что набор точек ^l_j принадлежит первому классу), то $h_1(x_s)_{\text{нов}} = h_2(x_s)_{\text{стар}}$ и $h_2(x_s) = h_1(x_s)$ (аналог того, что набор точек ^l_j «переброшен» во второй класс). Далее подсчитывается значение критерия $J(H_1)$ с «переброшенным» набором точек ^l_j. Если значение $J(H_1) > J(H_0)$, то изменения функций принадлежности для набора точек ^l_j остается в силе, в противном случае произведенные изменения значений функций принадлежности отменяются. Аналогичная операция выполняется для случая, когда $h_1(x_s)_{\text{стар}} < h_2(x_s)_{\text{стар}}$ аналог того, что набор точек ^l_j принадлежит второму классу). Алгоритм прекращает работу, если на каком-то цикле не было произведено изменений функций принадлежности ни для одной из точек исходной выборки.

Алгоритм построения начального разбиения

В составе комплекса алгоритмов интеллектуального анализа данных был разработан алгоритм построения начального разбиения как для детерминированного, так и для размытого случая. Рассмотрим эти алгоритмы более подробно.

Детерминированный случай. Для простоты изложения без ограничения общности, алгоритм описан для случая двух классов — A_1 и A_2. На первом шаге из всех точек выборки $x_1, \ldots, x_n$ находится пара наиболее удаленных друг от друга точек, x_l и x_p, одна из которых — x_l, относится к классу A_1, а другая — x_p, относится к классу A_2. Если n достаточно велико, то используется усеченный вариант первого шага, а именно: x_l выбирается случайно, а x_p ищется как точка, наиболее от нее удаленная.

Затем последовательно рассматриваются все точки выборки, за исключением точек x_l и x_p. А именно: на втором шаге рассматривается точка x_1 (при условии, что $x_1 \neq x_l$, $x_1 \neq$

$= x_p$), которая относится к первому классу, если она ближе к x_l, чем к x_p, и ко второму классу в противном случае. Если $x_1 = x_l$ или $x_1 = x_p$, тогда переходим к рассмотрению следующей точки. На j-м шаге рассматривается точка x_j (при условии, что $x_j \neq x_l$, $x_j \neq$ $= x_p$), которая относится к одному из двух классов в соответствии с правилом:

$$x_j \in \begin{cases} A_1, & \text{если } K(x_j, A_1) \geqslant K(x_j, A_2)\,; \\ A_2, & \text{если } K(x_j, A_1) < K(x_j, A_2)\,, \end{cases} \quad j = 1, \ldots, n,\ x_j \neq x_l, x_j \neq x_p$$

Такая процедура повторяется до тех пор, пока не будут исчерпаны все точки выборки. Полученное разбиение принимается в качестве начального разбиения R_0.

Размытый случай. На первом шаге алгоритма находится начальное разбиение R_0 выборки $x_1, \ldots, x_n$ на r классов в детерминированном случае. На втором шаге определяются $a_1, \ldots, a_r$ — центры всех классов в полученном разбиении R_0. Далее для каждой точки x_j рассчитываются функции принадлежности $h_i(x_j)$, $i = 1, \ldots, r$, $j = 1, \ldots, n$, значения которых обратно пропорциональны расстояниям до центров соответствующих классов: $h_i(x_j) = K(a_i, x_j) / \sum\limits_{i=1}^{r} K(a_i, x_j)$, где $K(a_i, x_j)$ — потенциальная функция вида 2, а $\sum_{i=1}^{r} K(a_i, x_j)$ — нормирующий множитель, обеспечивающий выполнения условия нормировки функций принадлежности: $\sum\limits_{i=1}^{r} h_i(x) = 1$. Полученный набор функций принадлежности и определяет размытое начальное разбиение $H_0 = \{h_i(x_j),\, i = 1, \ldots, r,\, j = 1, \ldots, n\}$.

Алгоритм выбора числа классов

Одна из основных проблем использования структурно-классификационных методов при решении задач исследования сложных систем управления — это выбор числа классов. Дело в том, что чрезвычайно важным фактором в прикладных исследованиях является содержательная интерпретация элементов получаемых в результате анализа структуры объектов (например, классов объектов). В работе описан специально разработанный алгоритм оптимального выбора числа классов. При этом оптимальность понимается в смысле максимизации содержательно обоснованного критерия качества классификации. Алгоритм, по сути, представляет собой экспертно-компьютерную процедуру, которая работает следующим образом. Сначала эксперт оценивает диапазон $(r_{\min}, r_{\max})$, в пределах которого заведомо находится искомое число классов. Далее, используя алгоритм СКАД, проводится разбиение анализируемого множества объектов на $(r_{\min}, r_{\max})$ классов. Качество каждой из полученных классификаций оценивалось с помощью критерия

$$J_3 = J_1 - qJ_2. \tag{6}$$

В формуле (6) величина J_1 — это средняя (по классам) мера близости точек, принадлежащих одному и тому же классу, вычисляется по формуле (4); q — свободный (настраиваемый) параметр алгоритма; J_2 — средняя мера близости классов, определяемая соотношением:

$$J_2 = \frac{1}{r-1} \sum_{i=1}^{r} \sum_{j>i} \frac{n_i + n_j}{n} K(A_i, A_j). \tag{7}$$

В формуле (7) величина n_i — это число точек в классе A_i, а величина $K(A_i, A_j)$ — мера близости классов A_i, A_j — вычисляется по формуле:

$$K(A_i, A_j) = \frac{1}{n_i n_j} \sum_{x_l \in A_i} \sum_{x_p \in A_j} K(x_l, x_p). \quad (8)$$

В формуле (8) потенциальная функция $K(x_l, x_p)$ определяется формулой (2). Параметр q является масштабирующим параметром, приводящим значения функционалов J_1 и J_2 к соизмеримым величинам; на практике q имеет значения порядка $2, \ldots, 7$ (во столько раз обычно отличается средняя близость внутри классов от средней близости между самими классами). Более подробно вопрос выбора значений настраиваемых параметров рассмотрен далее в специальном разделе.

В итоге получается последовательность $J_3(r_{\min}), ..., J_3(r_{\max})$. Формально в качестве наилучшего (оптимального) можно выбрать такое число классов r_{opt}, которое соответствует экстремальному значению критерия (6):

$$r_{\text{opt}} = r_j \mid \max J_3(r_j), r_j = r_{\min}, \ldots, r_{\max}.$$

Однако наличие существенной, но неиспользованной при классификации информации (например, ввиду отсутствия данных) может привести к тому, что полученное таким способом r_{opt} не будет наилучшим с точки зрения эксперта. Для компенсации этого недостатка предлагается использовать следующую экспертную процедуру. Экспертам представляются значения $J_3(r_j)$, $r_j = r_{\min}, \ldots, r_{\max}$, представленные для удобства в виде графика, на котором отмечается значение r_{opt} (оно соответствует максимальной точке на графике). Используя эту информацию, эксперты могут корректировать выбираемое число классов. В подавляющем числе случаев экспертное число классов либо совпадает с r_{opt}, либо незначительно (± 1) отличается от него.

При классификации многомерных объектов во время такой экспертизы анализируется также классификация каждого объекта. Для этой цели экспертам сообщается информация о мере близости $K(x_i, c_j)$ каждой точки x_i до центров классов c_j, $j = 1, \ldots, r$, в оптимальной классификации, т. е. матрица близости $\|K(x_i, c_j)\|$, $i = 1, \ldots, n$, $j = 1, \ldots, r_{\text{opt}}$. Перенесение точки (объекта) x_i из j-го класса в l-й считается допустимым, если величины $K(x_i, c_j)$ и $K(x_i, c_l)$ отличаются незначительно. Другими словами, содержательно обоснованное перенесение допустимо для точек, расположенных вблизи границы между соответствующими классами.

Алгоритм выбора информативных параметров

Опыт использования алгоритмов структурно-классификационного анализа показывает, что классификация по всем исходным параметрам далеко не всегда приводит к желаемым результатам [1]. Действительно, при сравнительно небольших выборках экспериментальных наблюдений и наличии помех (ошибки в определении значений параметров, сознательное искажение информации и т. д.) использование для классификации большого числа входных параметров приводит к сильному «перемешиванию» классов, а сами классы при этом плохо поддаются интерпретации. По этой причине классификацию объектов целесообразно проводить не в исходном пространстве, а в пространстве наиболее существенных (информативных) параметров, имеющем значительно меньшую размерность.

Для выбора информативных параметров в работе предлагается использовать результаты структуризации параметров. Далее, для того чтобы отличать структуризацию объектов и параметров, будем говорить о классификации объектов, но о группировке параметров.

Алгоритм СКАД в задаче группировки параметров. Для группировки параметров, как и в случае классификации объектов, предлагается использовать алгоритм СКАД. Формальная постановка задачи группировки параметров подразумевает определение: множества параметров, подлежащих группировке; множества решающих правил и критерия качества группировки [2].

Группируемое множество параметров — это конечный набор параметров $\{x^{(1)}, \ldots, x^{(1)}\}$, полученный из исходного набора после нормировки дисперсии каждого параметра на 1. Здесь $x_j^{(i)}$, $i = 1, \ldots, k$, $j = 1, \ldots, n$ определены как реализации случайной величины $x^{(i)}$ на множестве исследуемых объектов. ***Множество решающих правил***, как и в случае классификации объектов — единичный симплекс [2].

Для формулировки критерия качества группировки необходимо ввести меру близости между параметрами (случайными величинами) x и y. В качестве такой меры используется коэффициент ковариации (совпадающий с коэффициентом корреляции для нормированных параметров x и y), который будем обозначать через $\mathrm{cov}_{x,y} = (x, y)$, понимая его как скалярное произведение случайных величин x и y. Для дисперсии $\mathrm{cov}_{x,x}$ случайной величины x используется обозначение $\mathrm{cov}_{x,x} = (x, x) = x^2$. Критерий качества группировки используется в виде следующего функционала:

$$J^* = \sum_{j=1}^{s} \sum_{\substack{x^{(i)}, x^{(l)} \in A_j \\ x^{(i)} \neq x^{(l)}}} \mathrm{cov}^2_{x^{(i)}, x^{(l)}} = \sum_{j=1}^{s} \sum_{\substack{x^{(i)}, x^{(l)} \in A_j \\ x^{(i)} \neq x^{(l)}}} (x^{(i)}, x^{(l)})^2, \tag{9}$$

где s — число групп. Максимизация функционала (9) соответствует интуитивному представлению о «хорошем» разбиении параметров, когда в одну и ту же группу попадают наиболее близкие (в определенном выше смысле) параметры. В этом смысле функционал (9) полностью аналогичен функционалу (4), который используется как критерий качества классификации объектов.

Для выбора информативных параметров чрезвычайно важно знать интегральные характеристики (эталоны) полученных групп. Для классификации объектов такими эталонами обычно являются «центры тяжести» точек, попавших в один и тот же класс, которые вычисляются по формуле (9). Для группировки параметров такого типа эталонами являются «средние» (в определенном выше смысле) виртуальные нормированные параметры (случайные величины) $f_1, \ldots, f_s$ такие, что $f_j^2 = 1, j = 1, \ldots, s$, которые будем называть факторами. Факторы (эталоны) некоторой группировки на s групп $A_1, \ldots, A_s$ определяются соотношением (10), являющемся, в определенном смысле, аналогом (10):

$$f_j = \arg\max_j \sum_{x^{(i)} \in A_j} (x^{(i)}, f)^2, \qquad f^2 = 1. \tag{10}$$

При решении прикладных задач критерий качества группировки (9) иногда удобнее представить в эквивалентном виде:

$$J^* = \sum_{j=1}^{s} \sum_{x^{(i)} \in A_j} (x^{(i)}, f_j)^2. \tag{11}$$

Таким образом, задача группировки набора k параметров на заданное число групп s состоит в максимизации функционала (11) как по разбиению параметров на группы A_j,

так и по выбору факторов f_j, $j = 1, \ldots, s$, $f_j^2 = 1$ определяемых из соотношения (10) при фиксированной группировке.

Легко показать [5], что для фиксированной группировки на непересекающиеся группы $A_1, ..., A_s$ (детерминированная постановка задачи) факторы (эталоны групп) определяются по формуле:

$$f_j = \frac{\sum\limits_{x^{(i)} \in A_j} \alpha_i x^{(i)}}{\sqrt{\left(\sum\limits_{x^{(i)} \in A_j} \alpha_i x^{(i)} \right)^2}} = \frac{\sum\limits_{x^{(i)} \in A_j} \alpha_i x^{(i)}}{\sqrt{\sum\limits_{x^{(i)} \in A_j, x^{(l)} \in A_j} \alpha_i \alpha_j \left(x^{(i)}, x^{(l)} \right)}}, \; j = 1, \ldots, s, \tag{12}$$

где α_i — компоненты собственного вектора матрицы $R_j = \|x^{(i)}, x^{(l)}\|$, соответствующего ее наибольшему собственному значению. Из (12) непосредственно следует, что фактор группы — это линейная комбинация параметров, отнесенных к этой группе (знаменатель в (12) необходим для нормировки $f^2 = 1$), причем коэффициентами в этой комбинации являются компоненты «максимального» собственного вектора ковариационной матрицы параметров из этой группы.

Для одновременного определения групп $A_1, ..., A_s$ и факторов $A_1, ..., A_s$, удовлетворяющих этим условиям, используется описанный выше алгоритм СКАД, который в данном случае работает следующим образом (для простоты описан алгоритм СКАД в детерминированном случае для $l = 1$, $s = 2$).

Пусть задано некоторое начальное разбиение R_0^* группируемого множества параметров $\{x^{(1)}, \ldots, x^{(k)}\}$. Обозначим через $x^{(j)} \in A_1^*$ параметры, относящиеся к первой группе, а через $x^{(j)} \in A_2^*$ — ко второй. Алгоритм итерационный, на каждом шаге рассматривается один параметр из последовательности $x^{(1)}, \ldots, x^{(k)}, x^{(1)}, \ldots$ («зацикленная» исходная последовательность параметров). Как и в случае классификации объектов, отнесение параметра $x^{(j)}$ к одной из двух групп обозначается с помощью индекса $\rho(x^{(j)})$, который равен 1, если $x^{(j)} \in A_1^*$, и -1, в противном случае. Тогда этот вариант алгоритма группировки записывается в виде:

$$\rho(x^{(j)}) = \operatorname{sign}\Big[J^*(x^{(j)} \in A_1^*) - J^*(x^{(j)} \in A_2^*) \Big] j = 1, \ldots, k, 1, \ldots, k, 1, \ldots. \tag{13}$$

Таким образом, на каждом шаге текущий параметр $x^{(j)}$ относится к той группе, при отнесении к которой значение критерия J^* будет больше (если эти значения равны, то он относится к группе с наименьшим номером). Алгоритм (13) заканчивает работу, если на некотором цикле среди параметров $x^{(1)}, \ldots, x^{(k)}$ не будет сделано ни одной «переброски» параметра из группы в группу. При этом, если критерий качества имеет вид(9), то факторы групп f_1 и f_2 определяются с помощью (12) по завершении процедуры группировки. Если же используется критерий в виде (11), то для подсчета значений $J^* = (k^{(j)} \in A_i^*) =$ в (13) факторы групп необходимо определять с помощью (12) на каждом шаге.

Теорема 3. *Алгоритм СКАД в задаче группировки параметров сходится к локальному максимуму функционала J^* за конечное число шагов (итераций).*

Доказательство этого утверждения аналогично доказательству теоремы 1.

Выбор информативных параметров. В результате применения алгоритма СКАД к исходным k параметрам будет получено их разбиение на заданное число групп s (в прикладных задачах значение s колеблется в диапазоне $3, \ldots, 10$), а также значения факторов

для полученных групп. При решении прикладных задач в дальнейшем используются либо новые интегральные параметры — факторы групп (если удается получить их удовлетворительное содержательное описание), либо такой набор параметров из исходного множества параметров (число которых равно числу групп), каждый из которых является ближайшим (в определенном выше смысле) к фактору в соответствующей группе. В некоторых случаях (например, когда нет такого параметра в группе, значения коэффициента корреляции которого с фактором значимо больше, чем для других параметров этой группы) в отдельных группах может быть отобрано по 2, а для особо многочисленных групп — по 3 и более параметров, ближайших к соответствующему фактору и максимально удаленных друг от друга. Иногда при формировании набора информативных параметров используются процедуры экспертной коррекции [1, 6].

В большинстве приложений исходные или выделенные информативные параметры имеют неравнозначную важность при анализе структуры объектов. Для формирования коэффициентов важности (весов) в работе предлагается использовать процедуры экспертного оценивания. Хорошие результаты дает процедура многовариантной экспертизы [7], когда для оценки таких весов используется несколько групп экспертов — специалистов в различных аспектах исследуемой проблемы. В результате экспертизы каждому параметру присваивается определенный вес (важности) при исследовании структуры объектов.

Выбор «оптимального» числа классов в задаче группировки параметров. Для выбора числа классов в задаче группировки параметров используется специальная экспертно-компьютерная процедура оптимизации критерия, аналогичного критерию выбора «оптимального» числа классов в задаче классификации объектов. Опишем вкратце эту процедуру.

Сначала эксперт-пользователь оценивает диапазон $(s_{\min}, s_{\max})$, в пределах которого заведомо находится искомое число групп. Далее, используя алгоритм СКАД, проводится разбиение группируемого множества параметров на $s_{\min}, s_{\min+1}, \ldots, s_{\max}$ групп. Качество каждой из полученных группировок оценивается с помощью критерия:

$$J_3^*(s) = J_1^*(s) - qJ_2^*(s), \tag{14}$$

где $J_1^*(s)$ — величина средней по группам меры близости параметров в группе, а $J_2^*(s)$ — величина средней меры близости между группами. Величина q в (14) является масштабирующим параметром, приводящим к одному масштабу средние значения функционалов $J_1^*(s)$ и $J_2^*(s)$. На практике величина q выбирается в диапазоне значений 2–5 (обычно во столько раз отличается средняя близость параметров внутри групп от средней близости самих групп).

В качестве «оптимального» можно выбрать такое число классов $r_{\text{opt}} = r_j$, которое соответствует максимальному значению критерия (14) для $r_j = r_{\min}, \ldots, r_{\max}$. Однако наличие существенной, но неиспользованной при классификации информации может привести к тому, что так полученное r_{opt} не будет «<истинно оптимальным». Для компенсации этого недостатка используется процедура экспертной коррекции [1, 6].

Особенности реализации разработанного комплекса алгоритмов

В процессе реализации разработанного комплекса алгоритма интеллектуального анализа данных возникает целый ряд проблем, для разрешения которых приходится разрабатывать специальные процедуры или использовать уже известные алгоритмы. В большинстве приложений, особенно связанных с социально-экономическими системами, пользова-

тель сталкивается с проблемой качества исходных данных. Здесь, прежде всего, необходимо выявлять ошибки в исходных данных, в том числе имеющие случайный характер. Для этой цели используются разнообразные алгоритмы фильтрации. Например, для выявления существенных «выбросов» в значениях параметров строится гистограмма распределения значений каждого из параметров и в зависимости от содержательной модели исследуемого объекта выбирается тот или иной тип функции распределения. Для структурно-классификационных алгоритмов наиболее адекватной моделью является смесь нормальных распределений. Существуют стандартные статистические методы для определения того, является ли анализируемое значение выбросом или согласуется с выбранной моделью порождения данных [8]. В любом случае, по виду гистограммы экспертным путем всегда можно определить, какое из значений параметра заведомо является «выбросом». Так, например, во многих приложениях широко используется так называемое «правило 3σ». Правило действует следующим образом: для каждого числового параметра по имеющейся выборке определяется среднее значение $\hat{x}^{(i)} = (1/n)\sum\limits_{j=1}^{n} x_j^{(i)}$ и стандартное отклонение $\sigma^{(i)} = \sqrt{(1/(n-1))\sum\limits_{j=1}^{n}(x_j^{(i)} - \hat{x}^{(i)})^2}$. Все значения $x_j^{(i)}$, превосходящие $\hat{x}^{(i)} \pm 3\sigma^{(i)}$, считаются «выбросами». Обычно «выбросы» заменяются либо на среднее значение этого параметра, либо на соответствующую границу диапазона $\hat{x}^{(i)} \pm 3\sigma^{(i)}$. В работе предлагается выбросы считать пропущенными наблюдениями и использовать для их заполнения специально разработанную процедуру.

Процедура заполнения пропущенных наблюдений. Как уже говорилось выше, во многих приложениях имеются пропуски в данных, кроме того, в процессе фильтрации «выбросы» часто рассматриваются как пропущенные наблюдения. В этой ситуации нужно либо использовать специальные процедуры подсчета расстояний между объектами, в параметрах которых имеются пропуски, либо разрабатывать специальные процедуры заполнения таких пропусков. В подавляющем большинстве работ, пропуски по каждому параметру предлагается заполнять средним известных значений соответствующего параметра (для исходной выборки). В настоящей работе была разработана специальная процедура заполнения пропусков в исходных данных с использованием алгоритмов автоматической классификации. Основная идея процедуры состоит в следующем. Если множество изучаемых объектов структурировано (т. е. их можно разделить на классы, достаточно компактно расположенные в пространстве параметров X), то дисперсия (диапазон) изменения каждого параметра в пределах каждой группы, как правило, будет существенно меньше чем этот показатель для значения этого параметра на всей выборке. Таким образом, если по данным с пропусками удастся определить реальную структуру взаиморасположения точек (т. е. провести классификацию, адекватную этой структуре), то заполнять пропущенное значение l-го параметра для объекта из i-го класса можно средним этого параметра по его известным значениям для всех объектов, попавших в i-й класс. Исходя из сделанного предположения, отклонение полученного значения от «истинного» должно быть существенно меньше (в среднем), чем обычная схема заполнения по общему среднему.

Опишем процедуру более подробно. На первом шаге все пропуски заполняются средними значениями каждого параметра по всей выборке. Далее проводится классификация выборки с заполненными пропусками на r_0 классов, где r_0 выбирается из следующих соображений. В каждом классе число объектов должно быть достаточным для статистически

значимой оценки среднего значения параметра, т. е. не меньше чем 8–10 точек. Поэтому $r_0^{\text{нач}} = n/15$ (с учетом неоднородности распределения числа точек по классам). Если в полученной классификации для некоторого класса число входящих точек будет меньше 8, то такой класс присоединяется к ближайшему классу. Некоторые из таких классов могут объединиться между собой, тогда дальнейшее их объединение не производится, если число точек в образованном классе больше или равно 8. В качестве меры близости двух классов A_i, A_j используется величина $K(A_i, A_j)$, определяемая формулой (8). В итоге, получается разбиение на r_1 классов. Затем в каждом из полученных классов ранее заполненные пропущенные наблюдения заполняются новыми значениями. А именно: пропущенное значение i-го параметра для j-го объекта заменяется средним известных значений i-го параметра для всех объектов из l-го класса (к которому принадлежит j-й объект). Такое заполнение производится для всех значений параметров, пропущенных в исходной выборке. На втором шаге происходит точно такая же процедура для матрицы данных, полученной после первого шага. Процедура заканчивается на таком шаге, на котором классификация точек осталась неизменной относительно предыдущего шага.

Выбор свободных параметров алгоритма. Комплекс алгоритмов имеет несколько настраиваемых параметров, которые должны быть выбраны либо до его использования на конкретном материале с помощью экспертов, либо в процессе такого использования с привлечением экспертных процедур. Таковыми параметрами являются: α и p в формуле (2), определяющей значение потенциальной функции $K(x, y)$, а также параметр q в формуле (6), определяющей критерий J_3 выбора оптимального числа классов. При выборе α и p в (2) воспользуемся следующими соображениями. Введем в рассмотрение величину $R_{\text{ср}}$ (расстояние «среза»), определяемую равенством:

$$\left.\frac{d^2K[R(x,y)]}{dR^2(x,y)}\right|_{R_{\text{ср}}} = 0. \tag{15}$$

Значение величины $R_{\text{ср}}$ в (15) определяет точку перегиба функции $K[R(x, y)]$, т. е. точку максимальной крутизны этой функции. На рис. 2 изображен график функции для различных значений p при одном и том же значении величины $R_{\text{ср}}$. Параметр p при фиксированном $R_{\text{ср}}$ характеризует крутизну функции $K[R(x, y)]$ в районе точки перегиба. Для удобства счета в качестве p выбирают числа кратные двум (2, 4, 6,. . .).

Параметр α в выражении (2) при известном $R_{\text{ср}}$ и фиксированном p определяется из выражения: $\alpha = (p-1)/((p+1)R_{\text{ср}})$. Обычно $R_{\text{ср}}$ выбирается из следующих соображений. По определению, точки, входящие в одну и ту же группу (класс), имеют высокие значения функции близости (значения потенциальной функции). А это означает, что расстояние между точками одной и той же группы в большинстве случаев меньше $R_{\text{ср}}$. И, наоборот, значения потенциальной функции (меры близости) между точками из разных групп существенно меньше, чем аналогичные значения для точек из одной и той же группы, т. е. соответствующее значение расстояния будет больше, чем $R_{\text{ср}}$. Это означает, что $R_{\text{ср}}$ должно равняться расстоянию от границы группы до центра группы (в среднем по всем группам). Поскольку до самой группировки определить это значение невозможно, то обычно делается 2–4 пробных расчета для различных значений этого параметра. Начальное $R_{\text{ср}}$ обычно выбирается как функция от размерности k пространства X, числа классов r и «характерного» размера множества точек выборки, например, диаметр сферы, описывающей все точки исходной выборки. В работе для этой цели используется выражение $R_{\text{ср}}^{\text{нач}} = \sqrt{k}R_{\max}(x_i, x_j)/r$, где $R_{\max}(x_i, x_j)$ — расстояние между максимально удаленной

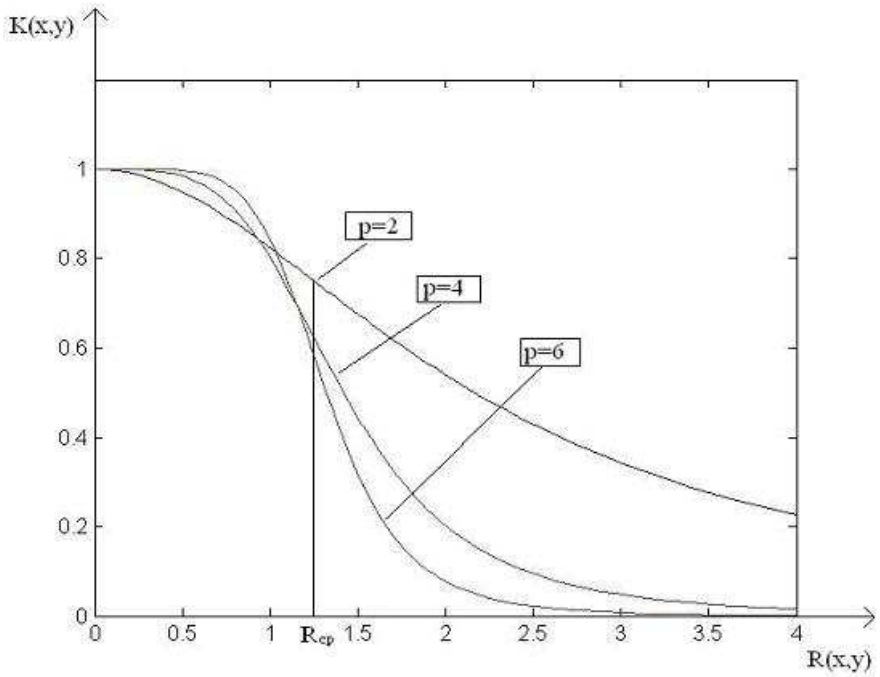


Рис. 2. График функции $K[R(x,y)]$ для различных значений p при одном и том же $R_{\text{ср}} = 1{,}25$

пары точек исходной выборки (после фильтрации и замены «выбросов», о которых говорилось выше).

При выборе p необходимо иметь в виду следующее обстоятельство. Если исследуемый материал достаточно хорошо структурирован, т. е. в пространстве X имеются хорошо обособленные друг от друга группы точек, то крутизна функции $K(x,y)$ в районе $R_{\text{ср}}$ может быть не очень большой, так как влияние далеких точек, находящихся на расстоянии существенно большем, чем $R_{\text{ср}}$, будет не существенно. С другой стороны, если такой явной структурированности нет (например, в случае сильной зашумленности данных), то в «промежутках» между группами будет достаточное количество точек (так называемые «мосты»). В этом случае, крутизна потенциальной функции в районе границы, т. е. в районе $R = R_{\text{ср}}$, должна быть достаточно высокой, чтобы минимизировать влияние точек в районе границы на процесс кластеризации. Для сильно зашумленных данных разработаны алгоритмы, в которых вводится специальный, так называемый, «фоновый» класс [2]. К фоновому классу относятся точки, которые расположены достаточно далеко от центров всех классов. В прикладных исследованиях величина p подбирается экспериментально: начальное значение $p = 2$ выбирается для случая хорошей структурированности, а $p = 4, \ldots, 8$ — для случаев слабой структуризации.

Следует отметить, что в прикладных задачах могут использоваться как числовые, так и качественные переменные. В первом случае, в качестве $R(x,y)$ в формуле (2) используется евклидово расстояние $R(x,y) = R_e(x,y) = \sqrt{\sum_{i=1}^{k}(x^{(i)} - y^{(i)})^2}$. В приложениях различные типы качественных параметров в подавляющем числе случаев приводятся к набору логических переменных, для них используется расстояние по Хеммингу: $R(x,y) = R_h(x,y)$, т. е. число несовпадающих разрядов в двоичных кодах векторов. Заметим, что для логических переменных x и y: $R_h(x,y) = R_e^2(x,y)$. Если среди входных параметров

есть как числовые, так и логические переменные, то в качестве квадрата расстояния можно использовать величину $R^2(x,y) = R_h(\tilde{x},\tilde{y}) + R_e^2(\hat{x},\hat{y})$, где $\tilde{x},\tilde{y}$ — логические переменные, $\hat{x},\hat{y}$ — числовые.

При выборе масштабирующего параметра q в формуле (6) обычно руководствуются следующими соображениями. Из формул (3) и (4), определяющих J_1, и формул (7) и (8), определяющих J_2, непосредственно следует, что величина J_1 существенно больше, чем J_2. Значения J_1 и J_2 определяются конкретной структурой расположения точек в пространстве X, и выбранными значениями параметров R_{cp} и p. Моделирование разработанных алгоритмов, а также решение некоторых прикладных задач показало, что характер поведения функции $J_3 = J_1 - qJ_2$ мало меняется в широком диапазоне значений q. В зависимости от структурированности пространства X хорошие результаты получаются для значений q в диапазоне $2,\ldots,7$.

Заключение

Разработанный комплекс алгоритмов интеллектуального анализа данных использовался для анализа сложноорганизованных данных в рамках исследования сложных систем управления, а также при совершенствовании процедур принятия решений для нескольких крупных систем управления, в основном регионального характера. Во всех приложениях, а также при машинном моделировании [9], была подтверждена высокая эффективность разработанного комплекса.

Литература

[1] *Дорофеюк А. А.* Методология экспертно-классификационного анализа в задачах управления и обработки сложноорганизованных данных (история и перспективы развития) // *Проблемы управления*, 2009. № 3.1. С. 19–28.

[2] *Дорофеюк А. А., Бауман Е. В., Дорофеюк Ю. А.* Методы интеллектуальной обработки информации на базе алгоритмов стохастической аппроксимации // *Математические методы распознавания образов: 15-ая Междунар. конф.* М.: МАКС ПРЕСС, 2011. С. 108–112.

[3] *Гольдовская М. Д., Дорофеюк Ю. А., Киселева Н. Е.* Методы структурного анализа в прикладных задачах исследования временных рядов // *Проблемы управления*, 2013. № 3. С. 33–41.

[4] *Bellman R.* Dynamic programming and lagrange multipliers // *Proc. Nat. Acad. Sci. USA*, 1956. Vol. 42, no. 10. P. 767–769.

[5] *Браверман Э. М., Мучник И. Б.* Структурные методы обработки эмпирических данных. М.: Наука, 1983. 464 с.

[6] *Дорофеюк А. А., Покровская И. В., Чернявский А. Л.* Экспертные методы анализа и совершенствования систем управления // *Автоматика и телемеханика*, 2004. № 10. С. 172–188.

[7] *Дорофеюк А. А., Дорофеюк Ю. А., Покровская И. В., Чернявский А. Л.* Метод независимой многовариантной экспертизы и его использование при решении прикладных задач // *Управление развитием крупномасштабных систем MLSD'2013): Тр. Седьмой Междунар. конф.* М.: ИПУ РАН, 2013. Т. 38. С. 260–271.

[8] *Крамер Г.* Математические методы статистики. М.: Мир, 1975. 648 с.

[9] *Дорофеюк Ю. А., Гольдовская М. Д., Спиро А. Г.* Особенности компьютерной реализации и моделирования алгоритмов интеллектуального анализа сложноорганизованных данных // *Управление развитием крупномасштабных систем (MLSD'2013): Мат-лы Седьмой Междунар. конф.* М.: ИПУ РАН, 2013. Т. 2. С. 328–331.

References

[1] *Dorofeyuk A. A.* 2009. Metodologiya ekspertno-klassifikatsionnogo analiza v zadachakh upravleniya i obrabotki slozhnoorganizovannykh dannykh (istoriya i perspektivy razvitiya). *Problemy Upravleniya* 3.1:19–28. (In Russian.)

[2] *Dorofeyuk A. A., Bauman E. V., Dorofeyuk Yu. A.* 2011. Metody intellektual'noy obrabotki informatsii na baze algoritmov stokhasticheskoy approksimatsii. *Matematicheskie Metody Raspoznavaniya Obrazov: 15-ya Mezhdunar. Konf.* Moscow. 108–112. (In Russian.)

[3] *Gol'dovskaya M. D., Dorofeyuk Yu. A., Kiseleva N. E.* 2013. Metody strukturnogo analiza v prikladnykh zadachakh issledovaniya vremennykh ryadov. *Problemy Upravleniya* 3:33–41. (In Russian.)

[4] *Bellman R.* 1956. Dynamic programming and lagrange multipliers. *Proc. Nat. Acad. of Sc. USA* 42(10):767–769.

[5] *Braverman E. M., Muchnik I. B.* 1983. *Strukturnye metody obrabotki empiricheskikh dannykh.* Moscow: Nauka. 464 p. (In Russian.)

[6] *Dorofeyuk A. A., Pokrovskaya I. V., Chernyavskiy A. L.* 2004. Ekspertnye metody analiza i sovershenstvovaniya sistem upravleniya. *Avtomatika i Telemekhanika* 10:172–188. (In Russian.)

[7] *Dorofeyuk A. A., Dorofeyuk Yu. A., Pokrovskaya I. V., Chernyavskiy A. L.* 2013. Metod nezavisimoy mnogovariantnoy ekspertizy i ego ispol'zovanie pri reshenii prikladnykh zadach. *Upravlenie razvitiem krupnomasshtabnykh sistem MLSD'2013): Tr. Sed'moy Mezhdunar. Konf.* Moscow. 38: 260–271. (In Russian.)

[8] *Kramer G.* 1975. *Matematicheskie metody statistiki.* Moscow: Mir Publ. 648 p. (In Russian.)

[9] *Dorofeyuk Yu. A., Gol'dovskaya M. D., Spiro A. G.* 2013. Osobennosti komp'yuternoy realizatsii i modelirovaniya algoritmov intellektual'nogo analiza slozhnoorganizovannykh dannykh. *Upravlenie razvitiem krupnomasshtabnykh sistem (MLSD'2013): Mat-ly Sed'moy Mezhdunar. Konf.* Moscow. 2:328–331. (In Russian.)

Математическое моделирование универсальной характеристики поворотно-лопастной гидротурбины*

***Ю. С. Волков*[1], *В. Л. Мирошниченко*[1], *А. Е. Салиенко*[2]**
volkov@math.nsc.ru, miroshn@math.nsc.ru, sa_cae@yahoo.com
[1]Институт математики им. С. Л. Соболева СО РАН, Новосибирск; [2]ОАО «ТЯЖМАШ», Сызрань

Рассматривается задача о построении универсальной характеристики рабочего колеса поворотно-лопастной гидротурбины по результатам энергетических испытаний модельной турбины. Универсальная характеристика является основным документом для выбора параметров натурной гидравлической турбины (диаметр рабочего колеса, частота вращения и др.), которые гарантируют наиболее эффективную работу турбины при всех режимах ее эксплуатации на конкретной ГЭС. Дается описание математического аппарата, примененного для создания математической модели универсальной характеристики рабочего колеса поворотно-лопастной гидротурбины по результатам стендовых энергетических испытаний модельной турбины. В основе предложенного подхода лежат методы аппроксимации многомерных функций по хаотически разбросанным данным, созданные авторами путем модификации и обощения D^m-сплайнов и мультквадриков Харди. Приводится пример моделирования по реальным данным на основе созданного комплекса программ.

Ключевые слова: *универсальная характеристика; поворотно-лопастная гидротурбина; КПД; сплайн; аппроксимация*

Mathematical modeling of hill diagram for Kaplan turbine*

***Yu. S. Volkov*[1], *V. L. Miroshnichenko*[1], *A. E. Salienko*[2]**
[1]Sobolev Institute of Mathematics, Novosibirsk; [2]JSC Tyazhmash, Syzran

Background: The problem of constructing of a hill diagram for the Kaplan turbine wheel on the power test results of the model turbine is considered. The hill diagram is the basic document for selection of full-scale hydraulic turbine parameters (turbine wheel diameter, rotating frequency, etc.) that ensure the most efficient perfomance of the turbine at all modes of its operation in a particular hydropower station.

Methods: Building a description of the mathematical formalism applied to mathematical modeling of the hill diagram of the Kaplan turbine based on the power test results of the model turbine.

Results: The basis of the proposed approach is the approximation methods for multidimensional functions at scattered data. The methods are modifications and generalizations of D^m-splines and Hardy's multiquadrics. An example of modeling for real data on the basis of the program complex is given.

Concluding Remarks: The software package for mathematical modeling of hill diagram for the Kaplan turbine was created. In the future, it is planned to use this software package for main full-scale hydraulic turbine parameters selection in a hydropower station.

Keywords: *hill diagram; Kaplan turbine; coefficient of efficiency; spline; approximation*

*Работа выполнена при финансовой поддержке программы фундаментальных исследований совместных интеграционных проектов СО РАН и УрО РАН, проект №32, и при поддержке РФФИ, грант №15-07-07530.

Введение

Одной из важнейших задач в области энергетики является оптимальное проектирование гидротурбин при строительстве гидроэлектрических станций. На современном этапе развития науки построить адекватную математическую модель работы гидротурбины с тем, чтобы провести оптимизацию, пока не представляется возможным. Большой накопленный опыт, как правило, позволяет определиться с типом гидротурбин для каждого конкретного случая. В настоящее время нашли распространение три основные системы турбин: радиально-осевые (РО), поворотно-лопастные (ПЛ) и ковшевые. Гидротурбины одной системы могут отличаться размерами, конструкцией механизмов, конфигурацией и относительными размерами проточного тракта, определяющих тип турбин. Все эти различия определяют индивидуальные свойства, главными из которых являются коэффициент полезного действия (КПД), быстроходность, приведенные параметры и кавитационная характеристика. Основными элементами, определяющими эти свойства, являются рабочее колесо, направляющий аппарат и отсасывающая труба. Однако экспериментально установленный факт [1], что гидротурбины одного типа, имеющие разные размеры, но геометрически подобный тракт, мало отличаются по индивидуальным свойствам, позволяет все исследование при проектировании натурных ГЭС перенести на малые модельные турбины, изучение которых можно проводить с использованием стендовых испытаний в заводских лабораториях. В гидротурбинах одного типа, имеющих разные размеры и геометрически подобный проточный тракт, перечисленные свойства могут несколько отличаться из-за влияния масштабного эффекта.

Параметры гидротурбин являются их количественными и качественными характеристиками. При проектировании гидротурбин заданными параметрами являются: напор, расход и мощность. Напор H (м) определяется при проектировании установки и представляет собой энергию, которой располагает турбина. Расход Q ($м^3/с$) определяется также при проектировании ГЭС. Мощность турбины N (кВт) при заданных значениях H и Q называют номинальной. Параметрами, определяемыми для выбора турбины, являются: частота вращения n (об/мин) и диаметр рабочего колеса D (м).

Диаметр рабочего колеса турбины D является основным размером, определяющим при заданных напоре и пропускной способности мощность и массу турбины. Гидродинамические качества рабочего колеса в основном определяют такие характеристики турбины, как КПД, приведенные расход, частоту вращения, кавитационный коэффициент и коэффициент быстроходности. Данные о турбине представляются в форме характеристик, определяющих все необходимые показатели турбины для различных условий ее работы, различных режимов. Поскольку такие данные для натурной гидротурбины, устанавливаемой на ГЭС, получить заранее невозможно, то их получают для подобной модельной турбины при ее испытаниях и исследованиях на гидротурбинных стендах. Затем, используя формулы пересчета [1, 2, 3], учитывающие масштабный эффект, получают энергетические, кавитационные и другие характеристики натурной турбины, выражающие зависимости КПД, кавитационного коэффициента и других величин на различных режимах работы от основных параметров (D, n, Q, H).

Основным результатом лабораторных испытаний модельной гидротурбины является главная универсальная характеристика или просто универсальная характеристика, представляющая зависимости КПД η, величины открытия направляющего аппарата a_0 и кавитационного коэффициента σ от приведенных величин частоты вращения n'_I и расхода Q'_I. Если турбина поворотно-лопастная, то присутствует еще зависимость от угла поворота ло-

пастей φ. Нахождение таких зависимостей требуется существующими методиками выбора основных параметров натурной гидротурбины.

Коэффициент полезного действия турбины η определяется отношением мощности на валу турбины к мощности потока, т.е. $N = \rho g Q H n$. Здесь ρ — плотность воды, g — ускорение свободного падения. Приведенные частота вращения и расход определяются формулами

$$Q'_I = \frac{Q}{D^2\sqrt{H}}; \qquad n'_I = \frac{nD}{\sqrt{H}}$$

и выражают соответственно расход и частоту вращения условной турбины-эталона, имеющей диаметр 1 м и работающей при напоре 1 м. Значения приведенных параметров n'_I и Q'_I в подобных режимах практически сохраняются неизменными.

В работе рассматривается задача о построении универсальной характеристики рабочего колеса поворотно-лопастной или ПЛ гидротурбины по результатам энергетических испытаний модельной гидротурбины на стенде.

В результате энергетических испытаний модельной турбины получается таблица чисел, состоящая из величин угла поворота лопастей φ, открытия направляющего аппарата a_0, приведенной частоты вращения турбины n'_I, приведенного расхода воды Q'_I и КПД турбины η, дополнительно могут содержаться значения кавитационного коэффициента σ и каких-либо других функций. Эти экспериментальные данные характеризуют зависимость КПД η от угла поворота лопастей φ и приведенных частоты n'_I и расхода Q'_I.

Задача состоит в создании по имеющимся данным математической модели, позволяющей строить пропеллерную и комбинаторную (универсальную) характеристики поворотно-лопастной гидравлической турбины и выполнять типовые расчеты, связанные с использованием этих характеристик.

Насколько нам известно, данное исследование является первым в нашей стране по построению математической модели универсальной характеристики ПЛ-гидротурбины (ранее авторами была разработана математическая модель универсальной характеристики РО-турбины [4]). До недавнего времени единственным заводом в России по проектированию и производству гидротурбин был ОАО «Ленинградский металлический завод», в сотрудничестве с инженерами которого и начиналось наше исследование. Отметим, что попытки решения данной задачи с помощью известных программных систем геометрического моделирования не увенчались успехом.

Специфика данных задачи состоит в том, что при проведении испытаний на стенде установочными параметрами являются угол поворота лопастей, величина открытия направляющего аппарата и частота вращения турбины, остальные измеряются или вычисляются. Как правило, имеются данные для небольшого количества углов установки лопастей (3–6). Для каждого угла φ точки с координатами (Q'_I, n'_I) лежат на нескольких линиях, соответствующих одинаковым значениям a_0, называемых линиями открытий. Хотя данные и лежат на линиях открытий, однако в плоскости координат Q'_I и n'_I они представляют нерегулярный (разбросанный) набор точек, в которых известны (с погрешностью) значения функции η и, возможно, σ или других функций. Задача состоит в восстановлении функции η как функции от переменных Q'_I, n'_I и φ (это пропеллерная характеристика) и функции η как функции от переменных Q'_I и n'_I, являющейся огибающей семейства η как функции от переменных Q'_I, n'_I и φ, т. е. пропеллерных характеристик, рассматриваемых как семейство по параметру φ функций от Q'_I и n'_I (комбинаторная характеристика). Ограничимся рассмотрением восстановления функции η.

В рассматриваемой задаче восстановления функции КПД гидротурбины присутствует сильная неравномерность расположения точек с данными, т.е. точки сконцентрированы вблизи некоторых линий и имеются точки, очень близко лежащие друг к другу (из-за погрешности, возможно, даже с существенно различными значениями КПД), и в то же время на достаточно больших участках (между линиями) точек нет совсем. Такая специфика задачи указывает на неприемлемость использования методов аппроксимации по хаотически расположенным данным, основанных на триангуляции области. «Хорошими» методами для такой задачи нам представляются глобальные методы типа аппроксимации D^m-сплайнами [5, 6, 7] или мультиквадриками Харди [8]. А поскольку подобные аппроксиманты довольны сложны и трудно вычислимы (с большими затратами), то решение последующих задач — типовых расчетов, связанных с использованием полученных характеристик, — становится достаточно ресурснозатратным занятием. Поэтому далее такой аппроксимант, построенный по хаотическим данным, имеет смысл заменить с нужной точностью кубическим сплайном на регулярной (прямоугольной) сетке. Использование «регулярного» кубического сплайна вместо «хаотического» сплайна, при сохранении всех достоинств последнего, позволяет значительно увеличить скорость проведения всех типовых расчетов, связанных с использованием характеристик.

Такой двухшаговый метод мы применили для создания математической модели пропеллерной и комбинаторной характеристики ПЛ-гидротурбины по результатам лабораторных испытаний модельных турбин на стенде. Для реализации первого этапа аппроксимации мы использовали модифицированные D^m-сплайны, называемые нами DMM-сплайнами, которые уже использовались для решения ряда других задач [6, 9] и моделирования универсальной характеристики РО-гидротурбины [4].

Постановка задачи

Рассмотрим математическую постановку задачи аппроксимации. Имеются следующие исходные данные. Для каждого угла поворота лопастей ПЛ-турбины φ_k, $k = 1, \ldots, L$, есть таблица чисел, строки которой образуют наборы чисел

$$a_i, n_i, q_i, n_i, \quad i = 1, \ldots, N_k,$$

где a_i — значение открытия направляющего аппарата a_0 (в дальнейшем мы опускаем для краткости общепринятый индекс 0), n_i — приведенная частота вращения n'_I, q_i — приведенный расход Q'_I (у приведенных величин также для краткости опущены двойные индексы), η_i — КПД турбины η. Данные известны с некоторой погрешностью.

Первым этапом моделирования универсальной характеристики является построение пропеллерной характеристики $\eta(q, n, \varphi)$. Пропеллерную характеристику мы строим в параметрическом виде

$$\begin{cases} \eta(a, n, \varphi), \\ q(a, n, \varphi). \end{cases} \tag{1}$$

Каждая из функций η и q на первом шаге представляет собой сглаживающий DMM-сплайн от трех переменных a, n, φ, построенный по дискретным данным, известным в, вообще говоря, хаотически расположенных точках. Степень сглаживания (точность восстановления исходных данных) регулируется параметрами сглаживания, которые, в принципе, можно задавать в каждой точке.

DMM-сплайны

Приведем определения для интерполяционных и сглаживающих DMM-сплайнов $S(x)$ трех переменных $\boldsymbol{x} = (x, y, z)$. Пусть значения $f_i = f(x_i)$ некоторой функции $f(\boldsymbol{x}) =$

$= f(x, y, z)$ известны в точках $\boldsymbol{x}_i = (x_i, y_i, z_i)$, $i = 1, \ldots, N$, некоторой области $\mathbb{R}^3$. Мы предполагаем, что все точки $\boldsymbol{x}_i$ различны. Обозначим $\mathcal{P}_k$ пространство многочленов

$$\pi(\boldsymbol{x}) = \sum_{0 \leqslant i+j+l \leqslant k} b_{ijl} x^i y^j z^l \tag{2}$$

степени k. Напомним, что размерность p пространства $\mathcal{P}_k$ определяется формулой

$$p = \dim \mathcal{P}_k = \frac{(k+1)(k+2)(k+3)}{6}.$$

Функция

$$S(\boldsymbol{x}) = \sum_{i=1}^{N} \lambda_i r_i^m \ln^{(1+(-1)^m)/2} r_i + \pi(\boldsymbol{x}), \tag{3}$$

где $m \geqslant 1$ целое число,

$$r_i = r(\boldsymbol{x}, \boldsymbol{x}_i) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 + R^2},$$

$\pi \in \mathcal{P}_k$, $k \geqslant 0$, многочлен степени k, R — вещественное число (параметр Харди) и коэффициенты λ_i удовлетворяют условиям

$$\sum_{i=1}^{N} \lambda_i \pi(\boldsymbol{x}_i) = 0 \quad \text{для всех} \quad \pi \in \mathcal{P}_k, \tag{4}$$

называется *DMM-сплайном степени* m.

Под степенью DMM-сплайна мы понимаем показатель степени «расстояния» r_i в формуле (3). Такое определение согласуется со степенью сплайна для обычного одномерного сплайна. Для четного m формула DMM-сплайна содержит логарифм, а при нечетном — квадратный корень. DMM-сплайны степени m могут отличаться степенью полиномиального слагаемого. Условия (4) могут быть переписаны в эквивалентном виде

$$\sum_{i=1}^{N} \lambda_i u_j(\boldsymbol{x}_i) = 0, \quad j = 1, \ldots, p, \tag{5}$$

где u_j некоторый базис пространства $\mathcal{P}_k$, т.е.

$$\pi(\boldsymbol{x}) = \sum_{j=1}^{p} \alpha_j u_j(\boldsymbol{x}).$$

Заметим, что D^m-сплайны и сплайны Дюшона [5], а также мультиквадрики Харди и их обобщения [8] являются частными случаями общей конструкции DMM-сплайна.

Сплайн $S(\boldsymbol{x})$ называется интерполяционным сплайном, если он удовлетворяет условиям интерполяции

$$S(\boldsymbol{x}_i) = f_i, \quad i = 1, \ldots, N. \tag{6}$$

Коэффициенты λ_i, α_j интерполяционного сплайна определяются из системы линейных $N + p$ уравнений, получаемой из условий (5), (6),

$$\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{F}, \tag{7}$$

где

$$\boldsymbol{\Lambda} = (\lambda_1, \ldots, \lambda_N, \alpha_1, \ldots, \alpha_p)^T, \quad \boldsymbol{F} = (f_1, \ldots, f_N, 0, \ldots, 0)^T.$$

Матрица этой системы имеет вид:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{C}^T & \mathbf{0} \end{pmatrix}, \tag{8}$$

где $\boldsymbol{B} = (b_{ij})$ — квадратная матрица размерности $N \times N$, $\boldsymbol{C}$ прямоугольная матрица размерности $N \times p$, символом $\mathbf{0}$ обозначена нулевая матрица размерности $p \times p$. Элементы матрицы $\boldsymbol{B}$ полностью определены условиями интерполяции (6), а именно:

$$b_{ij} = \begin{cases} [r(\boldsymbol{x}_i, \boldsymbol{x}_j)]^m [\ln r(\boldsymbol{x}_i, \boldsymbol{x}_j)]^{(1+(-1)^m)/2} & \text{при } i \neq j, \\ |R| & \text{при } i = j, \end{cases} \quad i, j = 1, \ldots, N.$$

Элементы матрицы $\boldsymbol{C}$ определяются условиями интерполяции (6) и выбранным базисом пространства многочленов $\mathcal{P}_k$.

Вопросы существования и единственности сплайнов общего вида исследовались в работе [7].

Построение интерполяционного DMM-сплайна сводится к решению системы линейных уравнений с практически плотной симметрической матрицей, размер которой в основном определяется количеством точек интерполяции $\boldsymbol{x}_i$. Для достаточно больших значений N такая система может быть плохо обусловлена. Хороший способ улучшения обусловленности состоит в предварительном преобразовании системы координат, путем отображения области с заданными точками интерполяции на единичный куб в $\mathbb{R}^3$. С этой же целью вместо представления многочленов $\pi(\boldsymbol{x})$ в форме (2) более удобно использовать формулу

$$\pi(\boldsymbol{x}) = \sum_{0 \leqslant i+j+l \leqslant k} b_{ijl} (x - x_0)^i (y - y_0)^j (z - z_0)^l,$$

где

$$x_0 = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad y_0 = \frac{1}{N} \sum_{i=1}^{N} y_i, \quad z_0 = \frac{1}{N} \sum_{i=1}^{N} z_i.$$

В этом случае мономы

$$(x - x_0)^i (y - y_0)^j (z - z_0)^l, \quad 0 \leqslant i + j + l \leqslant k,$$

образуют базис $u_j(\boldsymbol{x})$ в пространстве многочленов $\mathcal{P}_k$. Несмотря на то, что матрица (8) системы (7) симметрическая, метод Холецкого использовать нельзя, так как матрица (8) не является положительно определенной. Мы используем некоторую модификацию метода Аазена [10] для вычисления коэффициентов сплайна, что позволило вдвое уменьшить время построения сплайна в сравнении с [9]. К тому же эта модификация позволяет контролировать обусловленность при решении и автоматически определять ситуации когда система (7) близка к вырожденной, что будет происходить в случаях не существования и не единственности DMM-сплайна.

Если в заданных значениях f_i присутствуют погрешности, то, как правило, интерполяционные сплайны практически бесполезны. В таких случаях необходимо строить сглаживающие сплайны. Сглаживающий DMM-сплайн $S_\rho(\boldsymbol{x})$ отличается от интерполяционного $S(\boldsymbol{x})$ тем, что при его определении условия интерполяции (6) заменены следующими

$$(-1)^{\tilde{m}} \rho \lambda_i + S_\rho(\boldsymbol{x}_i) = f_i, \quad i = 1, \ldots, N, \tag{9}$$

где $\tilde{m} = \lfloor m/2 + 1 \rfloor$ (здесь использован знак целой части числа), $\rho \geqslant 0$ — параметр сглаживания. Тогда коэффициенты сглаживающего сплайна будут находиться из системы (7), но с измененной матрицей

$$\boldsymbol{A} = \begin{pmatrix} \tilde{\boldsymbol{B}} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{B} + (-1)^{\tilde{m}}\rho\boldsymbol{I} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{0} \end{pmatrix}, \tag{10}$$

где $\boldsymbol{I}$ — единичная матрица. Эта измененная матрица отличается от матрицы (8) для случая интерполяции только элементами главной диагонали, а именно

$$\tilde{b}_{ii} = |R| + (-1)^{\tilde{m}}\rho, \quad i = 1, \ldots, N.$$

Отметим, хотя сложности и трудности построения интерполяционных и сглаживающих DMM-сплайнов подобны, однако ненулевой параметр сглаживания улучшает обусловленность системы уравнений. Мы не обсуждаем вопросы существования и единственности DMM-сплайнов. Отметим лишь, что степень сплайна m и степень полиномиальной добавки k в случае интерполяции должны удовлетворять неравенству $k \geqslant \lfloor m/2 \rfloor$, но при сглаживании такое ограничение отсутствует.

Параметр сглаживания регулирует уровень сглаживания данных. Значение $\rho = 0$ соответствует интерполяционному сплайну, т. е. $S(\boldsymbol{x}) = S_0(\boldsymbol{x})$. С увеличением ρ поведение DMM-сплайна становится более гладким, уменьшаются или даже исчезают осцилляции, но вместе с тем, вообще говоря, растет отклонение от исходных данных, сплайн стремится к некоторому многочлену степени k.

Наш опыт использования DMM-сплайнов позволяет утверждать, что DMM-сплайны являются мощным инструментом приближения функций, особенно заданных на хаотическом множестве точек. Успех применения DMM-сплайнов зависит от корректного выбора параметров m, k, R, ρ. Приведем некоторые рекомендации.

В случае $R = 0$ дифференциальные свойства DMM-сплайнов полностью определены их степенью m. При $m = 1$ DMM-сплайн непрерывен, но его первые производные не существуют в точках $\boldsymbol{x}_i$. Для сплайнов степени 2 первые производные непрерывны, но вторые не существуют в точках $\boldsymbol{x}_i$, и т.п. Поэтому, если необходимо приближать первые производные, то степень DMM-сплайнов должна быть не меньше 2. Вместе с тем не рекомендуется использовать сплайны высоких степеней (больше 4), так как могут возникать значительные осцилляции. Любое ненулевое значение параметра Харди R делает DMM-сплайн бесконечно дифференцируемым, однако для очень маленьких значений R этот эффект становится чисто формальным. Тем не менее выбором параметра Харди можно существенным образом изменить поведение сплайна, в то же время нам кажется невозможным в настоящий момент дать какие-либо конкретные рекомендации по выбору этого параметра.

Степень полиномиальной части k существенно влияет на свойства DMM-сплайн, особенно на краях области данных и в подобластях с разреженными данными. Если поведение аппроксимируемой функции в некотором смысле близко к поведению полинома некоторой конкретной степени k, то это значение и нужно выбрать в качестве степени полиномиальной части. За пределами области данных сплайн будет вести себя как некоторый полином степени k.

Выше уже обсуждался вопрос влияния параметра сглаживания на поведение DMM-сплайна. Добавим, что можно регулировать степень сглаживания индивидуально в каждой точке $\boldsymbol{x}_i$. Для этого достаточно заменить общий параметр ρ в соотношениях (9) на

индивидуальные параметры ρ_i для каждого i. Ясно, что это не приводит ни к каким усложнениям алгоритма построения сплайна.

Универсальная характеристика

Вернемся к построению математической модели универсальной характеристики. По имеющимся данным мы строим по описанной методике два сглаживающих трехмерных DMM-сплайна $\eta(a, n, \varphi)$ и $q(a, n, \varphi)$. Построение пропеллерной характеристики — это самый трудоемкий и одновременно самый ответственный этап математического моделирования. После завершения этого этапа можно находить промежуточные характеристики при любых углах поворота лопастей φ и решать весь комплекс задач, относящихся к таким характеристикам. Кроме того, пропеллерная характеристика служит основой для построения универсальной (комбинаторной) характеристики.

Для ускорения и упрощения работы с полученными функциями $\eta(a, n, \varphi)$ и $q(a, n, \varphi)$ мы аппроксимируем их регулярными трехмерными кубическими сплайнами $S_\eta(a, n, \varphi)$ и $S_q(a, n, \varphi)$ соответственно на прямоугольной сетке $\Delta = \Delta_a \times \Delta_n \times \Delta_\varphi$. Используются различные варианты интерполяционных и локально-аппроксимационных сплайнов в разложении по базису из B-сплайнов [11, 12]. Переход к кубическим сплайнам позволяет без потери точности приближения исходных данных значительно увеличить скорость дальнейших вычислений. Отметим, что для нахождения одного значения DMM-сплайна требуется вычислить сумму достаточно сложных слагаемых, количество которых не меньше числа точек с исходными данными, и поэтому вычислительные затраты здесь растут при увеличении числа точек, в принципе, неограниченно. Напротив, затраты на вычисление одного значения кубического сплайна не зависят ни от количества узлов сплайна, ни от числа точек с исходными данными и сводятся к выполнению нескольких десятков арифметических операций.

Пропеллерная характеристика (1) представляет собой семейство поверхностей, зависящих от параметра $\varphi \in [\varphi_1, \varphi_L]$. *Универсальной (комбинаторной) характеристикой* ПЛ турбины называется огибающая семейства поверхностей (1) с параметром φ.

Комбинаторная характеристика является огибающей семейства поверхностей (1) по параметру φ. Следовательно, комбинаторная характеристика является поверхностью в плоскости координат q и n.

Универсальная характеристика находится [13] из уравнения

$$\frac{\partial q}{\partial a}\frac{\partial \eta}{\partial \varphi} - \frac{\partial q}{\partial \varphi}\frac{\partial \eta}{\partial a} = 0, \quad \varphi \in [\varphi_1, \varphi_L], \quad n \in [n_{\min}, n_{\max}]. \tag{11}$$

Решение уравнения (11) строится в виде набора бикубических сплайнов

$$\eta_k(\varphi, n), \quad q_k(\varphi, n), \quad a_k(\varphi, n),$$

где η_k, q_k, a_k — соответственно комбинаторный КПД, комбинаторный расход и комбинаторные открытия. Еще раз подчеркнем, качество комбинаторной характеристики определяется качеством построенной ранее пропеллерной характеристики.

Пример моделирования

Авторами создан комплекс программ для построения пропеллерной и комбинаторной характеристик и опробован на реальных данных энергетических испытаний модельных ПЛ-гидротурбин.

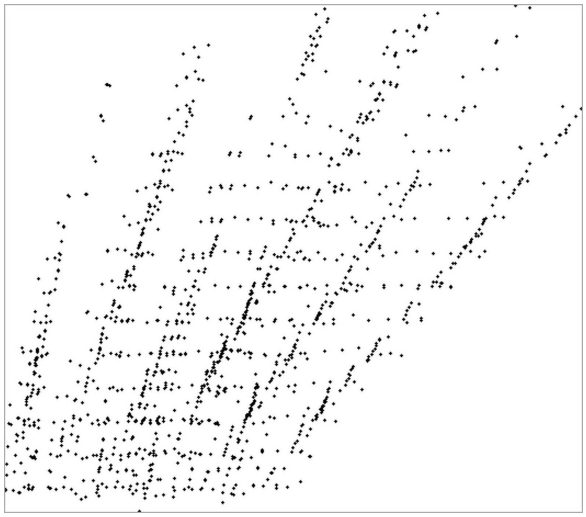


Рис. 1. Пример исходных данных

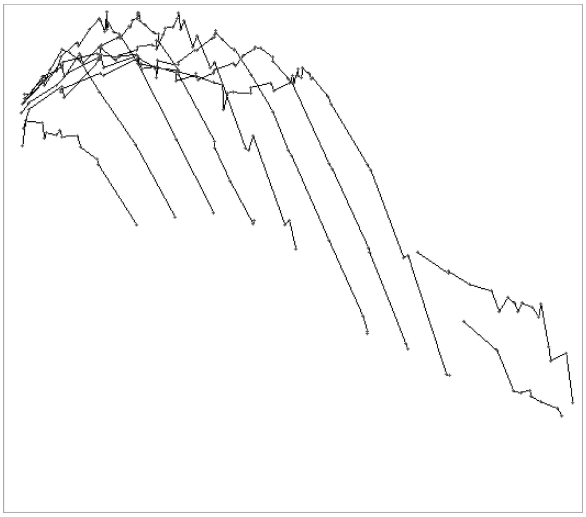
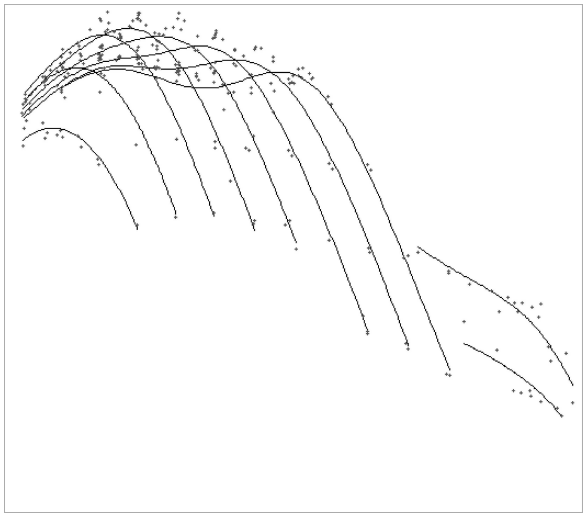
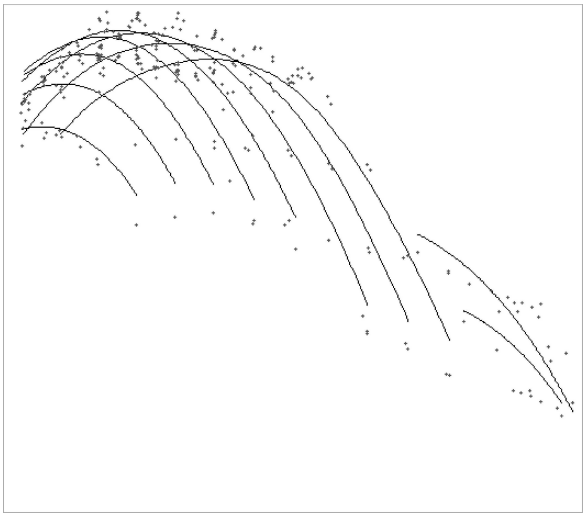


Рис. 2. Пример части исходных данных, $\varphi = 0$



(а) $\rho = 0{,}001$



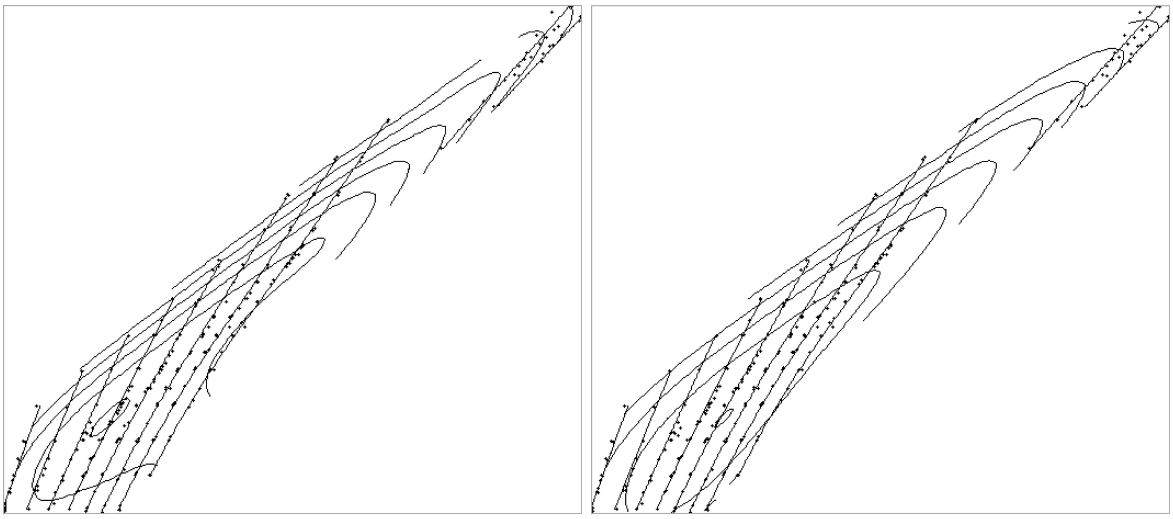
(б) $\rho = 0{,}1$

Рис. 3. Сглаживание исходных данных, $\varphi = 0$

На рис. 1 в плоскости переменных $(Q'_I,\, n'_I)$ приведены данные испытаний для углов поворота лопастей $-10°$, $-5°$, $0°$, $5°$, $10°$, $15°$. Исходный набор данных состоит из 1479 точек. На рис. 2 приведены результаты испытаний для одного угла $\varphi = 0$. Точки соответствующие одному значению открытия направляющего аппарата последовательно соединены отрезками и изображены в плоскости (n'_I,η).

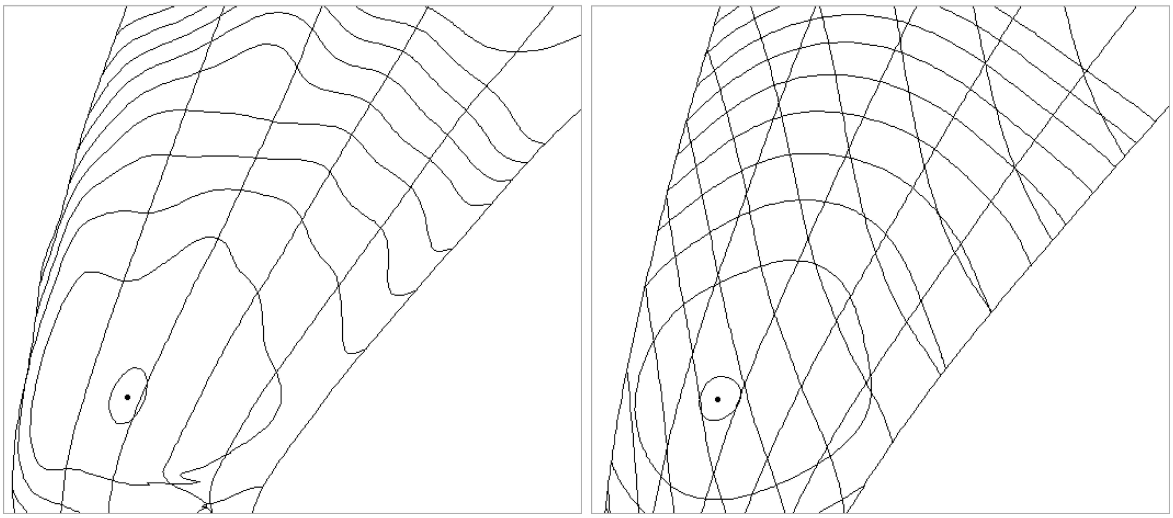
На рис. 3 приведены примеры сглаживания исходных данных для разных значений параметра сглаживания $\rho = 0{,}001$ и $\rho = 0{,}1$. В плоскости $(n'_I,\, \eta)$ изображены линии одинаковых открытий, значений соответствующих данным для одного угла $\varphi = 0$.

На рис. 4 приведен результат моделирования, пропеллерная характеристика для $\varphi = 0$ с параметрами сглаживания $\rho = 0{,}001$ и $\rho = 0{,}1$. Изображены изолинии КПД и линии открытий в плоскости переменных $(Q'_I,\, n'_I)$.



(а) $\rho = 0{,}001$ (б) $\rho = 0{,}1$

Рис. 4. Пропеллерная характеристика, $\varphi = 0$



(а) $\rho = 0{,}001$ (б) $\rho = 0{,}1$

Рис. 5. Универсальная характеристика

На рис. 5 приведен результат моделирования, универсальная (комбинаторная) характеристика, для параметров сглаживания $\rho = 0.001$ и $\rho = 0.1$. Изображены изолинии КПД и линии открытий в плоскости переменных $(Q'_I,\, n'_I)$.

Заключение

Предложена методика создания математической модели универсальной характеристики поворотно-лопастной гидротурбины по результатам стендовых энергетических испытаний модельных турбин. Результаты испытаний, как правило, представляют сильно нерегулярные хаотически расположенные и зашумленные данные. Использование DMM-сплайнов позволило получить достаточно хорошую аппроксимацию пропеллерной универсальной характеристики, что в свою очередь позволило рассчитать комбинаторную универсальную (главную) характеристику. Проведено тестирование созданного комплек-

са программ, в т. ч. на данных, являющимися реальными результатами энергетических испытаний. Получаемые математические модели универсальных характеристик были признаны специалистами достаточно адекватными. В дальнейшем планируется использовать данный комплекс программ для выбора основных параметров создаваемых натурных рабочих колес гидротурбин для гидроэлектростанций.

Литература

[1] *Орго В. М.* Основы конструирования и расчета на прочность гидротурбин. Л.: Машиностроение, 1978. 224 с.

[2] *Барлит В. В.* Гидравлические турбины. Киев: Вища школа, 1977. 360 с.

[3] *Кривченко Г. И.* Гидравлические машины: турбины и насосы. М.: Энергоатомиздат, 1983. 320 с.

[4] *Волков Ю. С., Мирошниченко В. Л.* Построение математической модели универсальной характеристики радиально-осевой гидротурбины // *Сибирский журнал индустриальной математики*, 1998. Т. 1, № 1. С. 77–88.

[5] *Bezhaev A. Yu., Vasilenko V. A.* Variational theory of splines. N. Y.: Kluwer Academic Publishers, 2001. 280 p.

[6] *Bogdanov V. V., Karsten W. V., Miroshnichenko V. L., Volkov Yu. S.* Application of splines for determining the velocity characteristic of a medium from a vertical seismic survey // *Central Eur. J. Math.*, 2013. Vol. 11, no. 4. P. 779–786.

[7] *Роженко А. И., Шайдоров Т. С.* О построении сплайнов методом воспроизводящих ядер // *Сибирский журнал вычислительной математики*, 2013. Т. 16, № 4. С. 365–376.

[8] *Hardy R. L.* Theory and applications of the multiquadric-biharmonic method. 20 years of discovery 1968–1988 // *Comput. Math. Appl.*, 1990. Vol. 19, no. 8-9. P. 163–208.

[9] *Anikonov Yu. E., Bogdanov V. V., Derevtsov E. Yu., Miroshnichenko V. L., Pivovarova N. B., Slavina L. B.* Some approaches to a numerical solution for the multidimensional inverse kinematic problem of seismics with inner sources // J. Inverse Ill-Posed Problems, 2009. Vol. 17, no. 3. P. 209–238.

[10] *Aasen J. O.* On the reduction of a symmetric matrix to tridiagonal form // *BIT*, 1971. Vol. 11. P. 233–242.

[11] *Завьялов Ю. С., Квасов Б. И., Мирошниченко В. А.* Методы сплайн-функций. М.: Наука, 1980. 352 с.

[12] *Завьялов Ю. С., Леус В. А., Скороспелов В. А.* Сплайны в инженерной геометрии. М.: Машиностроение, 1985. 221 с.

[13] *Залгаллер В. А.* Теория огибающих. М.: Наука, 1975. 104 с.

References

[1] *Orgo V. M.* 1978. *Design principles and strength design of hydraulic turbines.* Leningrad: Mashinostroenie. 224 p. (In Russian)

[2] *Barlit V. V.* 1977. *Hydraulic turbines.* Kiev: Vishcha Shkola. 360 p. (In Russian.)

[3] *Krivchenko G. I.* 1983. *Hydraulic machines: Turbines and pumps.* M.: Energoatomizdat. 320 p. (In Russian.)

[4] *Volkov Yu. S., Miroshnichenko V. L.* 1988. Constructing a mathematical model of a universal characteristic for a radial-axial hydroturbine. *Sibirskiy Zh. Industrial'noy Matematiki* 1(1):77–88. (In Russian.)

[5] *Bezhaev A. Yu., Vasilenko V. A.* 2001. *Variational theory of splines.* N. Y.: Kluwer Academic Publs. 280 p.

[6] *Bogdanov V. V., Karsten W. V., Miroshnichenko V. L., Volkov Yu. S.* 2013. Application of splines for determining the velocity characteristic of a medium from a vertical seismic survey. *Central Eur. J. Math.* 11(4):779–786.

[7] *Rozhenko A. I., Shaidorov T. S.* 2013. On spline approximation with a reproducing kernel method. *Numerical Analysis Appl.* 6(4):314–323.

[8] *Hardy R. L.* 1990. Theory and applications of the multiquadric-biharmonic method. 20 years of discovery 1968–1988. *Comput. Math. Appl.* 19(8-9):163–208.

[9] *Anikonov Yu. E., Bogdanov V. V., Derevtsov E. Yu., Miroshnichenko V. L., Pivovarova N. B., Slavina L. B.* 2009. Some approaches to a numerical solution for the multidimensional inverse kinematic problem of seismics with inner sources. *J. Inverse Ill-Posed Problems* 17(3):209–238.

[10] *Aasen J. O.* 1971. On the reduction of a symmetric matrix to tridiagonal form. *BIT* 11:233–242.

[11] *Завьялов Ю. С., Квасов Б. И., Мирошниченко В. Л.* 1980. *Методы сплайн-функций.* М.: Наука. 352 p. (In Russian.)

[12] *Zav'yalov Yu. S., Leus V. A., Skorospelov V. A.* 1985. *Splines in engineering geometry.* M.: Mashinostroenie. 221 p. (In Russian.)

[13] *Zalgaller V. A.* 1975. *Theory of envelopes.* M.: Nauka. 104 p. (In Russian.)

Отображения параллельных алгоритмов на суперкомпьютеры экзафлопсной производительности на основе имитационного моделирования*

Б. М. Глинский[1,2]***, М. А. Марченко***[1,2]***, А. С. Родионов***[1,2]***, Д. А. Караваев***[1]***, Д. И. Подкорытов***[1]

gbm@sscc.ru

[1]ФБГУН Институт вычислительной математики и математической геофизики СО РАН, Новосибирск, Россия; [2]Новосибирский государственный университет, Новосибирск, Россия

Целью работы является исследование возможности отображения параллельных алгоритмов на архитектуру суперЭВМ экзафлопсной производительности с использованием метода имитационного моделирования. Авторами предложена система AGent NEtwork Simulator (AGNES) для исследования масштабируемости алгоритмов и программного обеспечения на предполагаемых архитектурах экзафлопсных суперкомпьютеров. Приведены результаты моделирования алгоритмов различного класса: алгоритмы прямого статистического моделирования, сеточные методы.

Ключевые слова: *агентно-ориентированная система; имитационное моделирование; масштабируемые параллельные алгоритмы*

Mappings of parallel algorithms on supercomputers with exaflops performance on the basis of simulation*

B. M. Glinsky[1,2]***, M. A. Marchenko***[1,2]***, A. S. Rodionov***[1,2]***, D. A. Karavaev***[1]***, and D. I. Podkorytov***[1]

[1]Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia; [2]Novosibirsk State University, Novosibirsk, Russia

The main objective of this research is a possibility of representation of parallel algorithms on different architectures of exaflops supercomputers based on simulation. The authors have proposed AGent NEtwork Simulator(AGNES) for investigating the scalability of algorithms and program software on admissible architectures of exaflops supercomputers. In this paper, the results of the simulation of different class algorithms are presented. These are algorithms of the forward statistical modeling and grid method. The problem of investigating the properties of scalability of parallel algorithms for implementing them on supercomputers of the future with exaflops performance goes beyond the scope of technological problems. In this paper, the authors show that it is possible to estimate the behavior of algorithms and to develop a modified computation scheme by implementing them on a simulation model. The imitating model allows one to identify the bottlenecks in algorithms and to find out how to modify an algorithm and what parameters need to be configured to scale this algorithm to a greater amount of cores. In the ICMMG, the simulation system AGNES has been developed, which was used for studying the scalability of distributed statistical modeling and for solving the problem of numerical three-dimensional modeling of seismic wave propagation. Real calculations have shown that Monte-Carlo method is linearly parallelized up to 1,000 computing cores. The

*Работа выполнена при финансовой поддержке РФФИ, проекты №№ 12-01-00727, 13-07-00589, 14-07-00832 и 14-05-00867, МИП 130 СО РАН, МИП 39 СО РАН, Программы РАН 4.9.

behavior of the method proposed was investigated with simulation up to $5 \cdot 10^5$ cores. The dependence of parallelization on the number of collector cores was shown and the modified computing scheme for a great number of cores was proposed. For the other problem, a good compliance between experimental and model results up to 32768 cores is shown. The results were obtained on the simulation model for 1,124,864 cores. The calculations were performed on clusters of The Siberian Supercomputer Center.

Keywords: *agent oriented system; scalable parallel algorithms; exaflops supercomputers*

Введение

Проблема исследования свойств масштабируемости параллельных алгоритмов при их реализации на будущих суперЭВМ экзафлопсной производительности выходит за уровень технологических задач и требует научно-исследовательского подхода к ее решению. Вычислительные алгоритмы, как правило, являются более консервативными по сравнению с развитием средств вычислительной техники. Оценить поведение алгоритмов, разработать модифицированные схемы вычислений можно уже сейчас путем реализации их на имитационной модели, отображающей тысячи и миллионы вычислительных ядер. Имитационная модель позволяет выявить узкие места в алгоритмах, понять, как нужно модифицировать алгоритм, какие параметры необходимо настраивать при его масштабировании на большое количество ядер. Задача моделирования алгоритмов для исследования их масштабируемости не является новой, ею занимаются многие группы исследователей во всем мире, ктически с начала «эры параллельного программирования», один из ранних обзоров приведен в [1]. Хорошим примером ранних проектов по моделированию исполнения параллельных программ в изменяемом вычислительном окружении является проект PARSIT [2]. Идеи этого проекта актуальны и сейчас, но он, естественно, не был ориентирован на крупномасштабные вычисления, количество ядер ограничивалось несколькими тысячами. Имеется много исследований, ориентированных на моделирование исполнения конкретного алгоритма или узкого класса алгоритмов (например, матричной алгебры) [3,4]. Построение и исследование подобных моделей, очевидно, существенно проще, чем построение моделей универсальных. Наиболее известным из подобных проектов является BigSim (http://charm.cs.uiuc.edu/research/bigsim), проект проводимый в США (Университет Урбана-Шампань, Иллинойс), руководитель проекта Kale Laxmikant). Проект направлен на создание имитационного окружения, позволяющего разработку, тестирование и настройку посредством моделирования ЭВМ будущих поколений, одновременно позволяя разработчикам ЭВМ улучшать их проектные решения с учетом специального набора приложений [5]. Однако этот проект для целей исследования масштабирования слишком глобален, он требует детального описания вычислительной архитектуры и профилирования исполнения программы на низком уровне, что зачастую излишне для простой сравнительной проверки решений, когда интересны лишь относительные, а не абсолютные значения времен. В Институте системного программирования РАН (г. Москва) под руководством академика В. П. Иванникова разработана модель параллельной программы, которая может эффективно интерпретироваться на инструментальном компьютере, обеспечивая возможность достаточно точного предсказания времени реального выполнения параллельной программы на заданном параллельном вычислительном комплексе. Модель разработана для параллельных программ с явным обменом сообщениями, написанных на языке Java с обращениями к библиотеке MPI, и включена в состав среды ParJava [6, 7]. Модель получается преобразованием дерева управления программы, которое для Java-

программ может быть построено путем модификации абстрактного синтаксического дерева. Для моделирования коммуникационных функций используется модель LogGP, что позволяет учитывать специфику распределенной вычислительной системы. Предсказание времени счета отдельных участков параллельной программы производится с учетом затрат, связанных с управлением MPI, т. е. производится корректировка модельных часов с учетом средней доли процессорного времени, которую занимает нить RTS (Run Time System). Таким образом, проект ParJava, с одной стороны, позволяет решать широкий круг задач по оценке эффективности исполнения параллельных программ на перспективных вычислительных системах, но, с другой стороны, привязан к конкретному языку программирования, что существенно сужает его возможности. Стоит отметить, что ParJava и BIGSIM не учитывают, по крайней мере явно, вопросы отказоустойчивости при исполнении больших программ, в то время как использование в вычислениях одновременно десятков и сотен тысяч, а для отдельных задач и миллионов вычислительных ядер не может их не поставить. В ИВМиМГ СО РАН развивается мультиагентный подход, который органично подходит для задачи имитации вычислений. В качестве атомарной, независимой частицы в модели вычислений выбран вычислительный узел и исполняемый на нем код алгоритма. Каждый функциональный агент эмулирует поведение вычислительного узла кластера, и программу вычислений, работающую на этом узле. Вычисления представляются в виде набора примитивных операций (вычисление на ядре; запись/чтение данных в память; парный обмен данными; синхронизация данных между вычислителями) и временных характеристик каждой операции [8].

Система моделирования AGNES

Первоначально разработанная для моделирования телекоммуникационных и информационных сетей [8, 9], система AGNES показала свою эффективность и для моделирования исполнения высокопроизводительных параллельных программ [10]. Пакет AGNES базируется на Java Agent Development Framework (JADE) [4]. JADE - это мощный инструмент для создание мультиагентных систем, и он состоит из трех частей: среда исполнения агентов; библиотека базовых классов, необходимых для разработки агентной системы; набор утилит, позволяющих наблюдать и администрировать MAC. JADE приложение обладает рядом важных свойств для агентных систем. Распределенность, JADE позволяет создавать приложения, запускаемые на локальной сети. Универсальность, поддержка стандарта FIPA обеспечивает легкость взаимодействия агентов JADE с другими программными, аппаратными или комплексными агентами, поддерживающими этот стандарт. Благодаря этому инструменту, разработчик имеет ряд готовых средств, для управления агентами: создание, удаление, регистрация и миграция между вычислителями агентов; регистрация функций агентов в единой базе; взаимодействие между агентами; отслеживание сообщений внутри MAC; графическая поддержка отладки во время разработки агентов. Основные преимущества, использования платформы JADE:

1. FIPA-совместимая агентная платформа, которая основана на рекомендациях FIPA и включает в себя три обязательных типа системных агентов: сервис управления агентами (AMS), канал связи агентов (ACC) и «Желтые страницы» (DF).
2. Распределенная агентная платформа, которая может использовать один или несколько компьютеров (узлов сети), на каждом из которых должна работать только одна виртуальная JAVA машина.
3. Наличие многопоточной среды исполнения с двухуровневым распределением.

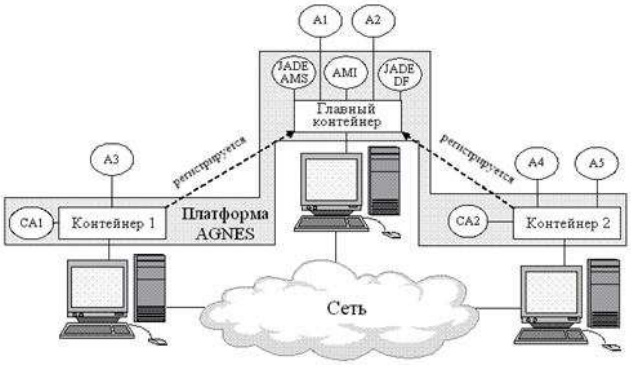


Рис. 1. Платформа AGNES

4. Наличие библиотеки со стандартными протоколами взаимодействия fipa-request и fipa-contract-net.
5. Наличие графических утилит для администрирования.

AGNES использует все эти возможности, а также расширяет мультиагентную систему до системы моделирования.

Среда моделирования AGNES состоит из двух типов агентов: управляющие агенты (УА), которые создают среду моделирования; функциональные агенты (ФА), которые образуют модель, работающую в среде моделирования. Приложение AGNES — это распределенная МАС, называемая платформой. Платформа AGNES состоит из системы контейнеров, распределенных в сети (рис. 1).

Обычно на каждом хосте находится по одному контейнеру (но при необходимости их может быть несколько). Агенты существуют внутри контейнеров. В системе может быть только один главный контейнер, который представляет собой точку начальной загрузки платформы. Главный контейнер создается первым, все созданные позже контейнеры должны быть зарегистрированы в главном контейнере. На главном контейнере обязательно работают управляющие агенты (УА) AMS (Agent Management System) и DF (Directory Facilitator) [5].

Управляющие агенты AGNES

Основные задачи УА: инициализация и запуск модели; сбор и хранение информации о ходе моделирования; синхронизация модельного времени; перераспределение нагрузки между вычислителями, участвующими в моделировании; взаимодействие с пользователем (вывод отчетов и возможность влияния на ход моделирования); обеспечение отказоустойчивости, восстановление модели. При запуске модели все ФА разделяются на виртуальные кластеры, и над каждым таким кластером назначается контролирующий агент (КА), который является разновидностью УА. Основные функции КА: идентификация отказа агентов в среде моделирования; пересылка всех управляющих команд функциональным агентам; хранение информации для восстановления агентов; восстановление модели при сбое. Инициализация модели. Во время запуска AGNES, первой стартует платформа JADE. Затем запускается агент AMI (AGNES Model Initializer) — агент, инициализирующий среду моделирования и запускающий в этой среде подготовленную модель. AMI действует по следующему алгоритму. Инициализация AGNES, запуск необходимых УА: ARM (AGNES resource manager) — агент, наблюдающий за состоянием платформы (отключением, подключением контейнеров). При изменении состояния он оповещает об этом

балансировщиков нагрузки. ALB (AGNES load balancer) — агент, наблюдающий за состоянием группы контейнеров, характеризующимися количеством ФА каждого типа, интенсивностью обмена сообщениями, средней скоростью доставки сообщения. В масштабных моделях, запущенных на больших количествах вычислителей (контейнерах), одновременно работают несколько ALB агентов, отвечающих за разные контейнеры. Обмениваясь информацией о состояниях всех контейнеров в платформе, агенты инициируют процессы миграции ФА между контейнерами, для выравнивания характеристик загруженности контейнеров. ADS (AGNES data storage) — агенты, собирающие всю информацию о характеристиках модели — модельных параметрах ФА. Каждый ФА агент через сервис JADE DF находит ADS агентов и в течение всего времени работы оповещает их о своем модельном состоянии. Инициализация модели. AMI получает конфигурационный файл модели с описанием типов, количеством и параметрами ФА, необходимых для моделирования. При запуске ФА разбиваются на «кластеры» и создается УА — ACA (AGNES controlling agent), отвечающий за этот «кластер ФА». Для уменьшения трафика управляющих команд все команды ФА получает только от своего ACA, а управляющие агенты обмениваются сообщениями между собой напрямую. После инициализации ФА, объединенные в кластеры, «расползаются» по доступным контейнерам и начинается моделирование, по команде от AMI. При необходимости дополнительно могут быть запущены агенты-утилиты с графическим интерфейсом, для взаимодействия AGNES с пользователем. В настоящее время пользователь может: контролировать состояние платформы; видеть количество контейнеров, список агентов на каждом из контейнеров, создавать или удалять агенты, подключать или отключать контейнеры; обмениваться сообщениями с агентами; пользователь может отправлять любые сообщения любому агенту, зная его AID, а так же он может получать ответные сообщения от агента как реакцию на свой запрос; наблюдать структуру сети, если модель можно представить в виде графа, узлами которого являются агенты, а ребрами информационные связи между ним; имеет механизмы глобального управления моделированием — приостановка, возобновление, преждевременное прекращение моделирования, получение промежуточных результатов.

Отказоустойчивость

Чтобы обеспечить высокий уровень отказоустойчивости, AGNES реализует несколько механизмов: отсутствие централизованного хранения данных для восстановления; хранение необходимой информации ведется подобно peer-to-peer сетям, т. е. информация располагается частями на разных агентах среды моделирования, и эта информация хранится с избытком, для гарантии ее восстановления; динамическое изменение хранилищ информации во время работы среды моделирования. То есть основные принципы улучшенной отказоустойчивости среды моделирования - это децентрализация хранилищ и избыточность информации. Рассмотрим более подробно, как происходит сохранение резервных данных и восстановление модели. В AGNES реализована служебная команда PING для проверки работоспособности агента. И все агенты должны корректно обрабатывать этот запрос. Большинство УА запускаются в нескольких экземплярах, чтобы гарантировать работы системе при отказе одного из них. КА следят за остальными УА, чтобы обнаружить отказ. Каждый агент AGNES должен реализовывать два метода SaveBackup() and RestoreBackup(), т. е. уметь сохранить свое состояние и восстановить его из сохранных данных соответственно. Все КА выбирают себе группу ФА, за которыми будет осуществляться контроль некоторое время, т. е. у КА есть список ФА, у которых он контролирует размещение backup-данных. С определенной периодичностью КА обмениваются агентами,

входящими в их кластер. Во время своей жизни ФА получают команду от КА о том, что нужно сделать backup: модельный момент времени, когда сохранить состояние, и список соседей, у кого следует разместить эти данные. В соответствующий момент агент сохраняет свое состояние, запоминает его у себя, для возможности дальнейшего отката, и отправляет его указанным агентам. Когда агент получает backup-данные соседа, он оповещает об этом своего КА. Контролирующий агент запоминает в таблицу, у кого хранятся чьи данные и за какой момент. Контролирующий агент каждый раз пытается хранить данные у разных агентов, чтобы уменьшить зависимость между агентами. Контролирующий агент регулярно опрашивает своих УА и ФА ping командами, чтобы обнаружить отказ. Помимо этого, каждый КА выбирает несколько соседних КА, состояние которых он также проверяет. При обнаружении отказа (не получит ответ на ping запрос) КА оповещает всех о необходимости прекращения моделирования, команда pause. Затем КА совместно определяют последнее стабильное состояние системы, т. е. такое состояние, в котором известна информация обо всех агентах, и местоположение данных о состоянии отказавших агентов в этот модельный момент времени. Информация об отказавших агентах запрашивается у работоспособных агентов и из нее восстанавливаются дубликаты. Затем в среде моделирования командой RestoreFrom инициализируется откат до стабильного состояния, и моделирование продолжается.

Сбор и хранение информации

Внутри AGNES циркулируют два типа сообщений: управляющие команды и информационные сообщения внутри модели. Для моделирования важно собирать и хранить информационные сообщения, т. е. все обмены данными внутри самой модели. Эту задачу выполняют агенты логгеры. Они подписываются на все сообщения определенного типа и получают их копии. В зависимости от специфики модели существует необходимость сбора сообщений определенного типа, для этого у логгера можно настроить фильтр и собирать только значимую информацию. Эта возможность реализована за счет структуры FIPA сообщений и удобных механизмов классификации сообщений. Также при моделировании важно иметь информацию о состоянии ФА. Благодаря сервису «Желтых страниц» JADE УА могут найти всех интересующих агентов модели и опрашивать их, сохраняя у себя нужную информацию.

Балансировка нагрузки

JADE приложения — это распределенные приложения, запускаемые на сети из вычислителей. JADE позволяет динамически менять среду исполнения MAC, т. е. подключаться к уже существующей программе или исключать работающие контейнеры. Для того чтобы обеспечить наилучшую производительность среды моделирования, AGNES следит за этими процессами и при обнаружении изменений в среде исполнения старается перераспределить агентов равномерно по всем доступным ресурсам, т. е. инициирует миграцию агентов из одного контейнера на другой средствами JADE. Так как агенты отличаются по своим функциям и задачам, то AGNES старается перераспределить равномерно агентов всех типов.

Взаимодействие с пользователем

Помимо доступных GUI tools среды JADE, AGNES предоставляет дополнительные средства взаимодействия с пользователем: вывод графа модели (где узлами графа являются ФА модели, а ребра каналы связи между агентами); вывод таблиц логов модели,

данные накопленные логгерами; механизмы изменения модели, добавление и удаление ФА.

Функциональные агенты AGNES

Функциональные агенты моделируют поведение исследуемой модели. AGNES ориентирована на создание моделей систем, которые легко декомпозируются на простые элементы, взаимодействующие друг с другом. Примерами подобных систем могут служить: сенсорные сети, пчелиный улей или дорожное движение. За исключением проблемно-ориентированных задач, каждый агент должен выполнять некоторые функции, которые обеспечивают процесс моделирование (периодическое резервное копирования своих данных, синхронизация, передачи сообщений между агентами и т. д.). Эти функции реализуются следующими методами: [Sleep()] — остановка моделирования; [Wakeup()] — продолжение моделирования после остановки; [Start (array InitParameters)] — инициализация и запуск моделирования; [RestoreFrom(time Moment)] — восстановление состояния агента в заданный момент времени; [CreateBackup(time Moment, agent Receiver)] — создание резервной копии состояния агента и отправка этих данных для хранения указанному агенту; [RestoreBackup(array BackupParameters)] — восстановление состояния агента из резервных данных; [GetStatus()] — получение статуса состояния модели. Следующие методы должны быть реализованы у AGNES агента независимо от модели: [SaveBackup(agent Sender, time BackupMoment, array BackupData)] — сохранение резервных данных другого агента в своем хранилище; [SendBackup(agent Receiver, time BackupMoment, agent BackupAgent)] — отправка резервных данных из своего хранилища указанному агенту; [SetTime(time ModelTime)] — синхронизация модельного времени у агента; [GetAgentTime()] — получить данные о внутреннем времени агента; [SendLog()] — широковещательная рассылка своего состояния всем подписанным на это событие агентам; [Ping()] — команда для проверки работоспособности агента. Преимуществами AGNES являются: малый объем кода имитационных программ; балансировка нагрузки при исполнении, доступность проблемно-ориентированных библиотек, возможность динамического изменения модели в ходе эксперимента.

Результаты моделирования

Рассмотрим примеры реализации отображения алгоритма на архитектуру экзафлопсной ЭВМ с использованием системы AGNES. Первая задача связана с изучением возможности масштабирования распределенного статистического моделирования на большое число вычислительных ядер. Это задачи, требующие моделирования экстремально большого количества независимых реализаций [11]. К числу таких проблем относятся задачи моделирования с использованием прямого статистического моделирования (ПСМ) течений разреженного газа с учетом химических реакций, задачи переноса излучения и теории дисперсных систем. Общая схема вычислений по методу Монте Карло (см. рис. 2):

Шаг 1: Подготовка к моделированию независимых реализаций на группах ядер.
Шаг 2: Моделирование реализаций, вычисление выборочных средних для группы.
Шаг 3: Сбор и осреднение данных.

Имитационное моделирование проводилось с использованием мультиагентной системы AGNES. Для имитации вычислений методов Монте Карло созданы два класса функциональных агентов: DataAgregator: ядро-сборщик, собирает информацию о вычислениях, обрабатывает и агрегирует ее. Возможно иерархическое построение сборщиков, которые

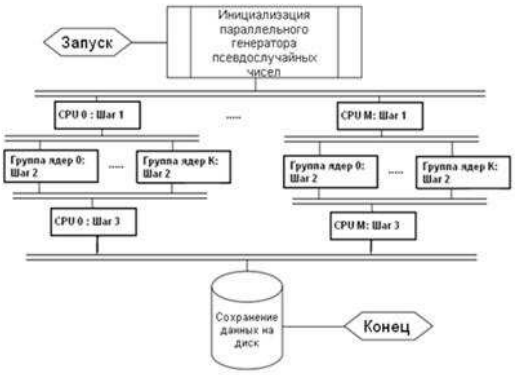


Рис. 2. Схема параллельных вычислений методов Монте Карло

на нижнем уровне обрабатывают данные непосредственно вычислителей, а затем передают их вышестоящему агенту DataAgregator. На вершине этой пирамиды всегда стоит одно главное ядро-сборщик, подготавливающее итоговые данные обо всех вычислениях и сохраняющее их на жесткий диск. MonteCarlo: агент, имитирующий расчет независимых реализаций методов Монте Карло на нескольких вычислительных узлах, группа ядер-вычислителей. Каждый агент проводит независимые вычисления согласно схеме вычислений и взаимодействует только с соответствующим DataAgregator. Основными характеристиками агента являются временные и статистические свойства, оценки которых получены на основе реальных вычислений. В результате работы модели собираются следующие отчеты.

• Набор времен, потраченных на каждую итерацию вычислений каждым агентом. Эти времена позволяют получить статистические характеристики протекающих в модели вычислений, для оценки правдоподобия модели.

• Информация о количестве итераций вычислений, совершенных каждым агентом MonteCarlo. При помощи данной статистики можно, например, отследить, как влияет количество вычислителей на скорость расчетов.

• Информация об интенсивности получения данных агентами DataAgregator от вычислителей либо нижестоящих DataAgregator, в данном случае регистрируется количество полученных за равные промежутки времени пакетов.

Исходные данные для имитационного моделирования получены с использованием библиотеки PARMONC, предназначенной для использования на современных суперкомпьютерах тера- и петафлопсного уровня [11]. Область применения библиотеки: «большие» задачи статистического моделирования в естественных и гуманитарных науках (физика, химия, биология, медицина, экономика и финансы, социология и др.). Библиотека PARMONC установлена на кластерах Сибирского суперкомпьютерного центра (ЦКП ССКЦ СО РАН) и может использоваться на вычислительных системах с аналогичной архитектурой. При этом использование библиотеки не привязано к каким-то определенным компиляторам языков C и FORTRAN или MPI. Инструкции по использованию библиотеки с примерами можно найти по ссылкам [12]. Как известно, теоретическое ускорение при распараллеливании для методов статистического моделирования практически идеальное, что подтверждается численными расчетами при числе вычислительных ядер порядка нескольких тысяч [13]. Тем не менее при числе ядер порядка сотен тысяч или нескольких миллионов вопросы организации счета требуют серьезного исследования, поскольку при этом возникают проблемы с большой загрузкой ядер-сборщиков, которые периодически

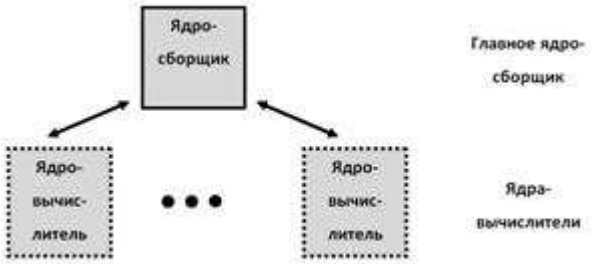


Рис. 3. Вариант организации связей между ядрами: одно главное ядро-сборщик

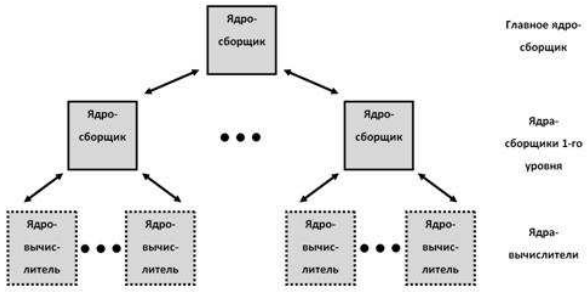


Рис. 4. Вариант организации связей между ядрами: один уровень промежуточных ядер-сборщиков

собирают статистику с ядер-вычислителей. А именно: проведенное имитационное моделирование показало, что при большом числе используемых вычислительных ядер (больше 10 000) реальное ускорение от распараллеливания существенно отличается от теоретического, что связано с большой загрузкой выделенных ядер-сборщиков, которые обрабатывают поступающие пакеты данных с ядер-вычислителей. При этом до 1000 ядер ускорение в модели совпадает с ускорением в реальных расчетах. С целью повышения эффективности распараллеливания исследовались различные варианты организации обмена данными между ядрами. А именно: целесообразно осуществлять периодическую пересылку результатов промежуточного осреднения реализаций, независимо полученных на загруженных ядрах (ядрах-вычислителях), на выделенные ядра (ядра-сборщики), объединенные в многоуровневую структуру. Ядра-сборщики будут периодически получать переданные им данные и осреднять их, передавая затем результаты на ядро (с номером 0), соответствующее вершине многоуровневой структуры (рис. 3 и 4). Будем называть такое ядро главным ядром-сборщиком; в числе его задач — сохранение осредненных данных на диск.

Рассчитанные на главном ядре-сборщике осредненные значения будут соответствовать выборке, полученной совокупно на всех ядрах-вычислителях. Распределенное статистическое моделирование на разных вычислительных ядрах-вычислителях производится в асинхронном режиме. Отправка и получение результатов статистического моделирования также осуществляется в асинхронном режиме [8]. На кластере НКС-30Т Сибирского суперкомпьютерного центра с использованием библиотеки PARMONC был произведен ряд расчетов для общего числа ядер, примерно равного 1000. Реальные затраты машинного времени на независимое моделирование реализаций на ядрах-вычислителях и обмен данными (выборочными средними) с главным ядром-сборщиком были использованы для

калибровки имитационной модели в AGNES. По результатам расчетов был сделан вывод, что требуемый уровень относительной статистической погрешности в 0,1% достигается при объеме выборки равном $L = 240\,000$. Среднее время моделирования одной реализации — примерно 12 с. Для ядер-вычислителей обмен данными с главным ядром-сборщиком происходил после каждой смоделированной на них реализации. Отображение модели на вычислительный кластер выглядит следующим образом: на каждом сервере запускается JVM и отдельный контейнер JADE. На каждом контейнере запускаются агенты имитирующие расчет методов Монте Карло. Каждый агент MonteCarlo содержит внутри себя группу циклических поведений, каждое из которых имитирует расчет независимых реализаций на одном вычислительном узле. Подобное представление позволяет моделировать расчеты по методу Монте Карло на 10^7 вычислительных узлах с использованием 10^4 агентов. При имитационном моделировании расчетов по методу Монте Карло за основу взята MPP-архитектура кластера НКС-30Т. При этом исследовалась величина относительного ускорения от распараллеливания при расчетах на M ядрах, определенная следующим образом:

$$S_L(M) = \frac{T_L(M_{\min})}{T_L(M)},$$

где $T_L(M)$ — машинное время на главном ядре-сборщике, затраченное на моделирование и сохранение выборочных средних для задачи, в которой моделируется L реализаций случайной оценки; $M_{\min}$ — наименьшее число ядер, использованных при расчетах. В первой серии экспериментов рассматривались два варианта организации обмена данными между ядрами-вычислителями и главным ядром-сборщиком:

— без использования промежуточных ядер-сборщиков (рис. 3);
— с использованием одного уровня промежуточных ядер-сборщиков (рис. 4).

В первом варианте ядра-вычислители были поделены на M_1 равных частей ($M_1 = 10$, 20, 100), для каждой из которых данные с ядер-вычислителей всегда отправлялись на «свое» ядро-сборщик. В свою очередь, M_1 ядер-сборщиков отправляли данные на главное ядро-сборщик. Во втором варианте для определенности будем считать, что параметр $M_1 = 0$. На рис. 5 приведена зависимость относительного ускорения $S_L(M)$ от общего числа моделируемых ядер M. Моделирование проводилось до $M = 5 \cdot 10^5$ ядер, но для показательности на рисунке приведены данные до $M = 10^5$. На рисунке ясно видна закономерность: увеличение числа ядер-сборщиков приводит к увеличению относительного ускорения. На начальном участке до 1000 ядер модельные данные хорошо совпадают с фактическими расчетами на гибридном кластере НСК-30+GPU, однако с увеличением количества ядер и в зависимости от количества ядер-сборщиков происходит отклонение от теоретической кривой. Следует отметить, что чем больше ядер-сборщиков, тем ближе модельная кривая к теоретической кривой. Подробнее эти вычислительные эксперименты описаны в работе [10]. В этой связи интересно исследовать вопрос об оптимальном (в смысле максимального значения относительного ускорения) числе ядер-сборщиков при фиксированном общем числе моделируемых ядер. Во второй серии экспериментов при фиксированном числе моделируемых ядер $M = 10^6$ варьировалось число ядер-сборщиков M_1, а также соответствующее число ядер-вычислителей в каждой группе, связанной со «своим» ядром-сборщиком. На рис. 6 приведен график зависимости относительного ускорения S_L от числа ядер-сборщиков M_1. Из рисунка видно, что максимальная величина относительного ускорения достигается при M_1, приближенно равном 1000. Это говорит о том, что при меньшем значении M_1 ядра-сборщики перегружены обработкой данных,

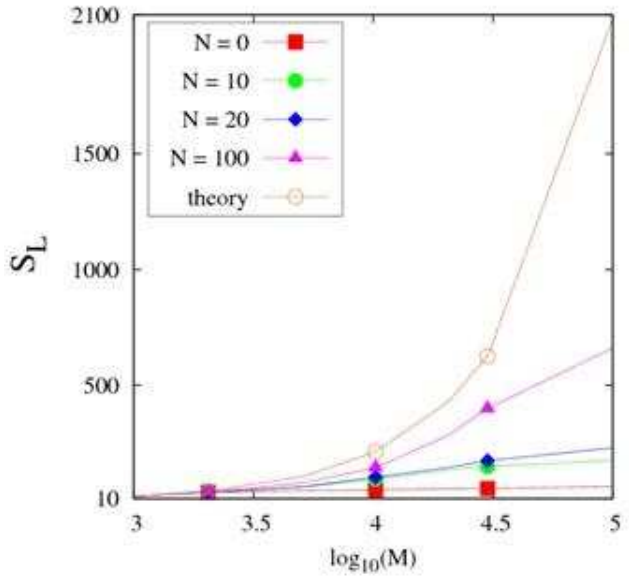


Рис. 5. Зависимость относительного ускорения S_L от общего числа моделируемых ядер M при разном числе ядер-сборщиков M_1 (горизонтальная ось — в логарифмическом масштабе)

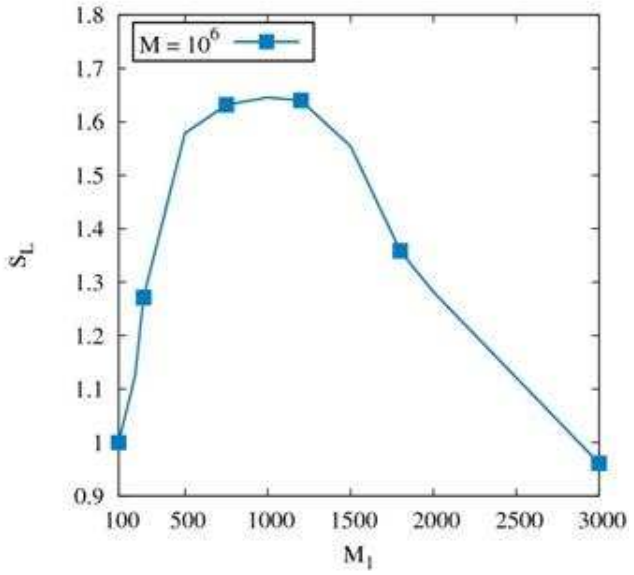


Рис. 6. Зависимость относительного ускорения S_L от числа ядер-сборщиков M_1 при общем числе моделируемых ядер $M = 10^6$

поступающих от ядер-вычислителей, а при большем числе – перегружено главное ядро-сборщик, занятое обработкой поступающих данных от ядер-сборщиков.

Аналогичные расчеты были проведены для другого класса алгоритмов, связанного с сеточными методами. Решалась задача численного моделирования распространения сейсмических полей в 3D изотропной неоднородной упругой среде [13]. В этом случае предполагаем, что архитектура гипотетического кластера является гибридной, вычислительные узлы состоят из нескольких CPU и GPU. Под масштабируемостью понимаем следующее: время счета алгоритма меняется незначительно при следующих допущениях: размер 3D модели увеличивается пропорционально количеству вычислительных узлов; каждый вы-

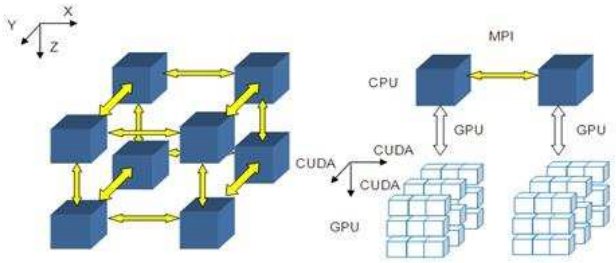


Рис. 7. Схема организации параллельных вычислений на гибридном кластере

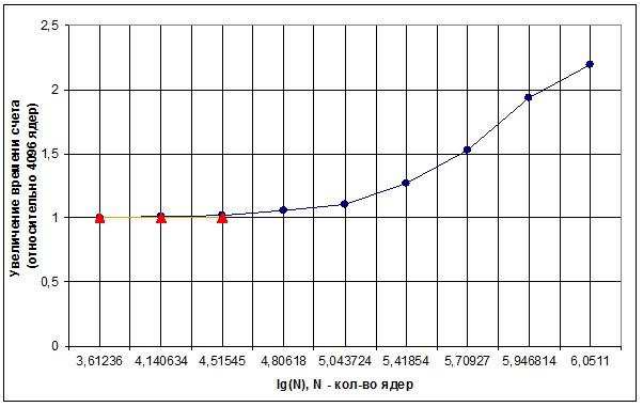


Рис. 8. Изменение времени расчета алгоритма численного моделирования в зависимости от числа вычислительных ядер (горизонтальная ось — в логарифмическом масштабе), треугольник — при проведении расчетов на суперкомпьютере, овал — при исследовании алгоритма с помощью имитационного моделирования

числительный узел совершает одно и то же количество итераций для своей подобласти. Разработана программа на основе масштабируемого параллельного алгоритма численного моделирования при использовании комбинации CUDA и MPI. Для проведения расчетов различных 3D моделей была рассмотрена следующая организация параллельного алгоритма и программы: 3D область моделирования разделяется на трехмерные подобласти по направлениям координатных осей; каждая из подобластей рассчитывается независимо на выделенном GPU, а обмены данными, между соседними GPU проводятся посредством CPU с использованием MPI (рис. 7). При этом вычисления для подобластей на GPU производятся посредством CUDA в 3D.

Для имитации сеточных методов реализован класс функциональных агентов Grid — узел-вычислитель, имитирующий расчет сеточных методов на одном вычислителе. Моделируются вычисления, когда область исследования делится вдоль осей на 3D подобласти, и полученные области загружаются на вычислители. Таким образом, получается, что у каждого вычислителя есть пересечение по данным максимум с двумя вычислителями по каждой из осей. Общие результаты изменения времени счета в зависимости от количества доступных ядер GPU (при пропорциональном увеличении размера 3D модели) в логарифмическом масштабе приведены на рис. 8. Показано хорошее соответствие экспериментальных и модельных результатов на начальном участке кривой (до 32 768 ядер).

Время работы алгоритма существенно увеличивается при увеличении количества ядер (удалось получить результаты для 1 124 864 ядер). Это объясняется характерными осо-

бенностями данного алгоритма — увеличением числа обменов каждого узла с соседями на каждой итерации, таким образом, число обменов в системе растет более стремительно. Уже на 500 000 ядер время выполнения алгоритма увеличилось в 1,5 раза, а для 1 000 000 ядер почти в 2,1 раза. Видно, что эффективное использование этого алгоритма на гибридных суперкомпьютерах с количеством ядер около 1 млн требует его модификации. Проведенные численные эксперименты по имитационному моделированию показали возможность масштабирования алгоритмов на большое число (сотни тысяч и даже миллионы) вычислительных ядер предполагаемого экзафлопсного суперкомпьютера, а также возможность исследования поведения алгоритмов при таком большом масштабировании.

Заключение

В работе представлена мультиагентная система имитационного моделирования AGNES. Система может быть использована для исследования масштабируемости различных алгоритмов для суперкомпьютеров с учетом особенностей архитектуры суперкомпьютера. Таким образом, с помощью AGNES можно проводить исследования связанные с разработкой алгоритмов для суперкомпьютеров экзафлопсного класса с учетом различных предполагаемых архитектур данных суперкомпьютеров. Приведенные тесты имитационного моделирования хорошо соотносятся с реальными данными, полученными при запуске программ на кластерах ЦКП ССКЦ СО РАН.

Литература

[1] *Sivasubramaniam A., Singla A., Ramachandran U., Venkateswaran H.* A simulation-based scalability study of parallel systems // *J. Parallel Distributed Computing*, 1994. Vol. 22, no. 3. P. 411–426.

[2] *Racherla G., Killian S., Fife L., Lehmann M., Parekh R.* Parsit — a parallel algorithm reconfiguration simulation tool // *Conference (International) on High Performance Computing Proceedings*, 1995.

[3] *D'Souza R. M., Lysenko M., Marino S., Kirschner D.* Data parallel algorithms for agent-based model simulation of tuberculosis on graphics processing units // *2009 Spring Simulation Multiconference (SpringSim'09) Proceedings*. San Diego, CA, USA: Society for Computer Simulation International, 2009.

[4] *Goodrich M. T.* Simulating parallel algorithms in the MapReduce framework with applications to parallel computational geometry CoRR. URL: `http://dblp.uni-trier.de/db/journals/corr/corr1004.html\#abs-1004-4708`.

[5] *Avetisyan A. I., Gaysaryan S. S., Ivannikov V. P., Padaryan V. A.* Productivity prediction of MPI programs based on models // *Automation and remote control*, Vol. 68, no. 5. P. 750–759 .

[6] *Ivannikov V., Avetisyan A., Padaryan V.* Evaluation of dynamic characteristics of a parallel program on a model // *Programming*, 2006. Vol. 4. P. 21–37. (In Russian.)

[7] *Глинский Б. М., Родионов А. С., Марченко М. А., Подкорытов Д. И., Винс Д. В.* Агентно-ориентированный подход к имитационному моделированию суперЭВМ экзафлопсной производительности в приложении к распределенному статистическому моделированию // *Вестник ЮУрГУ*, 2012. Т. 18(277), № 12. С. 94–99.

[8] *Podkorytov D., Rodionov A., Sokolova O., Yurgenson A.* Using agent-oriented simulation system AGNES for evaluation of sensor networks // *LNCS*. Heidelberg: Springer, 2010. Vol. 6235. P. 247–250.

[9] *Podkorytov D., Rodionov A., Choo H.* Agent-based simulation system AGNES for networks modeling: Review and researching // *6th Conference (International) on Ubiquitous Information Management and Communication (ACM ICUIMC 2012) Proceedings*, 2012. Paper 115. 4 p.

[10] *Glinsky B., Rodionov A., Marchenko M., Podkorytov D., Weins D.* Scaling the distributed stochastic simulation to exaflop supercomputers // *2012 IEEE 9th Conference (International) on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th Conference (International) on High Performance Computing and Communication Proceedings*, 2012. IEEE. P. 1131–1136.

[11] *Марченко М. А., Михайлов Г. А.* Распределенные вычисления по методу Монте-Карло // *Автоматика и телемеханика*, 2007. № 5. С. 157–170.

[12] *Marchenko M. A.* PARMONC — A software library for massively parallel stochastic simulation // *LNCS*, 2011. Vol. 6873. P. 302–315.

[13] *Глинский Б. М., Караваев Д. А., Ковалевский В. В., Мартынов В. Н.* Численное моделирование и экспериментальные исследования грязевого вулкана «Гора Карабетова» вибросейсмическими методами // *Вычислительные методы и программирование*, 2010. Т. 11, № 1. С. 99–108.

References

[1] *Sivasubramaniam A., Singla A., Ramachandran U., Venkateswaran H.* 1994. A simulation-based scalability study of parallel systems. *J. Parallel Distributed Computing* 22(3):411–426.

[2] *Racherla G., Killian S., Fife L., Lehmann M., Parekh R.* 1995. Parsit — a parallel algorithm reconfiguration simulation tool. *International Conference on High Performance Computing Proceedings*.

[3] *D'Souza R. M., Lysenko M., Marino S., Kirschner D.* 2009. Data parallel algorithms for agent-based model simulation of tuberculosis on graphics processing units. *2009 Spring Simulation Multiconference (SpringSim'09) Proceedings*. San Diego, CA, USA: Society for Computer Simulation International.

[4] *Goodrich M. T.* Simulating parallel algorithms in the MapReduce framework with applications to parallel computational geometry CoRR. Available at: `http://dblp.uni-trier.de/db/journals/corr/corr1004.html\#abs-1004-4708`.

[5] *Avetisyan A. I., Gaysaryan S. S., Ivannikov V. P., Padaryan V. A.* Productivity prediction of MPI programs based on models *Automation Remote Control* 68(5):750–759 .

[6] *Ivannikov V., Avetisyan A., Padaryan V.* 2006. Evaluation of dynamic characteristics of a parallel program on a model. *Programming* 4:21–37. (In Russian.)

[7] *Glinskiy B. M., Rodionov A. S., Marchenko M. A., Podkorytov D. I., Vins D. V.* 2012. Agent oriented approach to imitation of distributed statistical modeling in application of exaflops supercomputer. *Vestnik YURGU* 12(18):94–99.

[8] *Podkorytov D., Rodionov A., Sokolova O., Yurgenson A.* 2010. Using agent-oriented simulation system AGNES for evaluation of sensor networks. *LNCS*. Heidelberg: Springer. 6235:247–250.

[9] *Podkorytov D., Rodionov A., Choo H.* 2012 Agent-based simulation system AGNES for networks modeling: review and researching // *6th Conference (International) on Ubiquitous Information Management and Communication (ACM ICUIMC 2012) Proceedings*. Paper 115. 4 p.

[10] *Glinsky B., Rodionov A., Marchenko M., Podkorytov D., Weins D.* 2012. Scaling the distributed stochastic simulation to Exaflop supercomputers. *2012 IEEE 9th Conference (International) on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th Conference (International) on High Performance Computing and Communication Proceedings*. IEEE. 1131–1136.

[11] *Marchenko M. A., Mikhailov G. A.* 2007. Distributed computing of Monte Carlo method. *Automation Remote Control* 5:157–170.

[12] *Marchenko M. A.* 2011. PARMONC — A software library for massively parallel stochastic simulation. *LNCS* 6873:302–315.

[13] *Glinskiy B. M., Karavaev D. A., Kovalevskiy V. V., Martynov V. N.* 2010. Numerical modeling and experimental studies of "Karabetova Mount" salse by vibroseismical methods. *Computational Methods Programming* 11(1):99–108.