

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

7(215)  
2014

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ  
Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Левин В. И. Интервально-дифференциальное исчисление и некоторые его применения . . . . . 3  
Варфоломеева А. А., Стрижов В. В. Алгоритм разметки библиографических списков методами структурного обучения . . . . . 11  
Скуратов А. К., Кошкин Д. Е. Сравнение 12 алгоритмов кластеризации данных применительно к задаче кластеризации текстов . . . . . 16

### МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Карпенко А. П., Сахаров М. К. Мультимемеевая глобальная оптимизация на основе алгоритма эволюции разума . . . . . 23  
Родионов А. Н. Моделирование и реализация отношения "is part of" на множествах композитных сущностей баз данных . . . . . 31

### ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Полищук Ю. В., Черных Т. А. О способах реализации концепции единого источника . . . . . 37  
Козлова Л. А., Трубочкина Н. К. Информационные технологии в английской лингвистике — визуализация правил грамматики . . . . . 42

### СОЦИАЛЬНО-ЭКОНОМИЧЕСКИЕ СИСТЕМЫ

- Залещанский Б. Д., Свиридов А. П., Павлова О. А., Шалобина Е. А. Вероятностно-статистические стратегии обеспечения качества подготовки персонала социо-технических систем путем оптимизации полных и частичных проверок . . 50

### Журнал в журнале НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Осипов В. Ю. Рекуррентная нейронная сеть со структурой слоев в виде двойной спирали . . . . . 56  
Мандрикова О. В., Полозов Ю. А. Аппроксимация и анализ ионосферных параметров на основе совмещения вейвлет-преобразования с коллективами нейронных сетей . . . . . 61  
Галушкин А. И. Метод обратного распространения ошибки и российские работы по теории нейронных сетей . . . . . 66  
Боронников Д. А., Пантюхин Д. В., Данько С. В. Нейросетевой алгоритм организации пространственных данных о рельефе местности . . . . . 77

Главный редактор:  
СТЕМПКОВСКИЙ А. Л.,  
акад. РАН, д. т. н., проф.

Зам. главного редактора:  
ИВАННИКОВ А. Д., д. т. н., проф.  
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

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

Редакционная коллегия:  
АВДОШИН С. М., к. т. н., доц.  
АНТОНОВ Б. И.  
БАРСКИЙ А. Б., д. т. н., проф.  
ВАСЕНИН В.А., д. ф.-м. н., проф.  
ГАЛУШКИН А.И., д. т. н., проф.  
ДИМИТРИЕНКО Ю.И., д. ф.-м. н., проф.  
ДОМРАЧЕВ В.Г., д. т. н., проф.  
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.  
ЗАРУБИН В. С., д. т. н., проф.  
ИСАЕНКО Р. О., к. т. н., с.н.с.  
КАРПЕНКО А. П., д. ф.-м. н., проф.  
КОЛИН К. К., д. т. н., проф.  
КУЛАГИН В. П., д. т. н., проф.  
КУРЕЙЧИК В. М., д. т. н., проф.  
КУХАРЕНКО Б. Г., к. ф.-м. н., доц.  
ЛЬВОВИЧ Я. Е., д. т. н., проф.  
МИХАЙЛОВ Б. М., д. т. н., проф.  
НЕЧАЕВ В. В., к. т. н., проф.  
РЯБОВ Г. Г., чл.-корр. РАН, д. т. н., проф.  
СОКОЛОВ Б. В., д. т. н., проф.  
УСКОВ В. Л., к. т. н. (США)  
ФОМИЧЕВ В. А., д. т. н., проф.  
ЧЕРМОШЕНЦЕВ С. Ф., д. т. н., проф.  
ШИЛОВ В. В., к. т. н., доц.

Редакция:  
БЕЗМЕНОВА М. Ю.  
ГРИГОРИН-РЯБОВА Е. В.  
ЛЫСЕНКО А. В.  
ЧУГУНОВА А. В.

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

**CONTENTS**

**INTELLIGENT SYSTEMS AND TECHNOLOGIES**

**Levin V. I.** Wiping Differential Calculus and Its Application . . . . . 3  
**Varfolomeeva A. A., Strijov V. V.** An Algorithm for Bibliographic Records Parsing Using Structure Learning Methods . . . . . 11  
**Skuratov A. K., Koshkin D. E.** Comparison of 12 Data Clustering Algorithms Applied to the Problem of Texts Clustering . . . . . 16

**MODELING AND OPTIMIZATION**

**Karpenko A. P., Sakharov M. K.** Multi-Memes Global Optimization Based on the Algorithm of Mind Evolutionary Computation . . . . . 23  
**Rodionov A. N.** Modeling and Implementation of "Is Part of" Relation at a Set of Databases Composite Entities . . . . . 31

**APPLIED INFORMATION TECHNOLOGIES**

**Polishuk Yu. V., Chernyh T. A.** On the Methods of Implementing the Concept of a Single Source . . . . . 37  
**Kozlova L. A., Trubochkina N. K.** Information Technology in English Linguistics — Visualization of Grammar Rules . . . . . 42

**SOCIAL AND ECONOMIC SYSTEMS**

**Zaleshchanskiy B. D., Sviridov A. P., Pavlova O. A., Shalobina E. A.** Probabilistic and Statistical Strategy of Quality Ensurance of Personnel Training in Sotsio-Technical Systems by Means of Full and Partial Testing Optimization . . . . . 50

**Journal-in-journal**  
**NEUROTECHNOLOGIES**

**Osipov V. Yu.** Recurrent Neural Network with Structure of Layers in the Form of the Double Spiral . . . . . 56  
**Mandrikova O. V., Polozov Yu. A.** Approximation and Analysis of Ionospheric Parameters Based on a Combination of Wavelet Transformation and Neural Networks Groups . . . . . 61  
**Galushkin A. I.** The Back Propagation Error Method and Russian Works on Neural Networks Theory . . . . . 66  
**Boronnikov D. A., Pantiukhin D. V., Danko S. V.** Neural Networks Algorithm of Spatial Relief Data Organization . . . . . 77

**Editor-in-Chief:**

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

**Deputy Editor-in-Chief:**

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

**Chairman:**

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

**Editorial Board Members:**

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.  
Antonov B. I.  
Barsky A. B., Dr. Sci. (Tech.), Prof.  
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.  
Galushkin A. I., Dr. Sci. (Tech.), Prof.  
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.  
Domrachev V. G., Dr. Sci. (Tech.), Prof.  
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.  
Zarubin V. S., Dr. Sci. (Tech.), Prof.  
Isaenko R. O., Cand. Sci. (Tech.)  
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.  
Kolin K. K., Dr. Sci. (Tech.)  
Kulagin V. P., Dr. Sci. (Tech.), Prof.  
Kureichik V. M., Dr. Sci. (Tech.), Prof.  
Kukhareenko B. G., Cand. Sci. (Phys.-Math.)  
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.  
Mikhailov B. M., Dr. Sci. (Tech.), Prof.  
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.  
Ryabov G. G., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Sokolov B. V., Dr. Sci. (Tech.)  
Uskov V. L. (USA), Dr. Sci. (Tech.)  
Fomichev V. A., Dr. Sci. (Tech.), Prof.  
Chermoshentsev S. F., Dr. Sci. (Tech.), Prof.  
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

**Editors:**

Bezmenova M. Yu.  
Grigorin-Ryabova E. V.  
Lysenko A. V.  
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.  
According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

## Интервально-дифференциальное исчисление и некоторые его применения

*Рассмотрено обобщение классического дифференциального исчисления на функции с интервальной неопределенностью. Введено понятие производной от интервальной функции. Даны формулы для вычисления интервальных производных любого порядка.*

**Ключевые слова:** интервальная функция, интервальная производная, недетерминистское дифференциальное исчисление

### Введение

Проектирование и исследование свойств разнообразных систем обычно опирается на тот или иной подходящий математический аппарат. К настоящему времени создано большое число различных видов математического аппарата. Однако при всем их различии почти все они обладают одним общим свойством — применимостью только к полностью определенным (детерминированным) системам. В то же время встречающиеся на практике системы обычно характеризуются той или иной степенью неопределенности (недетерминированы). В целях построения и исследования таких систем чаще всего применяют математический аппарат теории вероятностей [1], теории нечетких множеств [2], интервальной математики [3]. В настоящей работе впервые предлагается новый математический аппарат для исследования недетерминированных систем — недетерминистское дифференциальное исчисление. Этот аппарат, в отличие от названных выше, нацеленных на исследование статических систем, применим к изучению динамических систем, возникающих в различных сферах человеческой деятельности: экономике (производственные системы и производственные процессы), социологии (динамика социальных групп), информатике (информационные системы и процессы в них, системы и алгоритмы анализа данных) и т. д.

### 1. Вспомогательные математические сведения

Будем использовать в качестве вспомогательных сведения из алгебры интервальных чисел [3, 4]. Операнды в ней — замкнутые вещественные интервалы, определяемые как множества всех веще-

ственных чисел между нижней и верхней границами интервала, включая сами границы:

$$\tilde{a} \equiv [a_1, a_2] \equiv \{a | a_1 \leq a \leq a_2\}. \quad (1)$$

Эти операнды назовем интервальными числами. Операции  $\circ$  над такими числами  $\tilde{a} = [a_1, a_2]$ ,  $\tilde{b} = [b_1, b_2]$  можно ввести как прямые теоретико-множественные обобщения соответствующих алгебраических операций над вещественными числами  $a, b$ , т. е.

$$\tilde{a} \circ \tilde{b} = \{a \circ b | a \in \tilde{a}, b \in \tilde{b}\}. \quad (2)$$

Таким образом, основные алгебраические операции над интервальными числами определяются следующими формулами:

$$\begin{aligned} \tilde{a} + \tilde{b} &= \{a + b | a \in \tilde{a}, b \in \tilde{b}\}, \\ \tilde{a} - \tilde{b} &= \{a - b | a \in \tilde{a}, b \in \tilde{b}\}, \\ k \cdot \tilde{a} &= \{k \cdot a | a \in \tilde{a}\}, \\ \tilde{a} \cdot \tilde{b} &= \{a \cdot b | a \in \tilde{a}, b \in \tilde{b}\}, \\ \tilde{a} / \tilde{b} &= \{a / b | a \in \tilde{a}, b \in \tilde{b}\}. \end{aligned} \quad (3)$$

На основе определений (3) операций над интервальными числами можно вывести формулы для вычисления результатов этих операций [3]:

$$\begin{aligned} \tilde{a} + \tilde{b} &\equiv [a_1, a_2] + [b_1, b_2] = [a_1 + b_1, a_2 + b_2], \\ \tilde{a} - \tilde{b} &\equiv [a_1, a_2] - [b_1, b_2] = [a_1 - b_2, a_2 - b_1], \\ k \cdot \tilde{a} &\equiv k \cdot [a_1, a_2] = \begin{cases} [ka_1, ka_2], & k > 0, \\ [ka_2, ka_1], & k < 0, \end{cases} \\ \tilde{a} \cdot \tilde{b} &\equiv [a_1, a_2] \cdot [b_1, b_2] = [\min_{i,j} (a_i \cdot b_j), \max_{i,j} (a_i \cdot b_j)], \\ \tilde{a} / \tilde{b} &\equiv [a_1, a_2] / [b_1, b_2] = [a_1, a_2] \cdot [1/b_2, 1/b_1]. \end{aligned} \quad (4)$$

Также в качестве вспомогательного нам требуется еще понятие интервальной функции [5], которая вводится как однозначное отображение множества замкнутых вещественных интервалов  $\{\tilde{x}\}$ ,  $\tilde{x} = [x_1, x_2]$  вида (1) на такого же типа множество  $\{\tilde{y}\}$ ,  $\tilde{y} = [y_1, y_2]$ . Интервальная функция обозначается

$$\tilde{y} = \tilde{f}(\tilde{x}) \quad (5)$$

и, аналогично числовым функциям,  $\tilde{x}$  называется интервальной независимой переменной (интервальным аргументом),  $\tilde{y}$  — интервальной зависимой переменной,  $\tilde{f}$  — интервальной функцией.

Введем теперь понятие предела интервальной функции (5). Рассмотрим независимую переменную  $\tilde{x} = [x_1, x_2]$  этой функции. Будем говорить, что переменная  $\tilde{x}$  в процессе своего изменения неограниченно приближается к некоторому предельному интервалу  $\tilde{x}_0 = [x_{01}, x_{02}]$ , если в указанном процессе  $x_1$  неограниченно приближается к  $x_{01}$ , а  $x_2$  неограниченно приближается к  $x_{02}$ . Символически неограниченное приближение интервального аргумента  $\tilde{x}$  к пределу  $\tilde{x}_0$  показывается как

$$\begin{aligned} (\tilde{x} \rightarrow \tilde{x}_0) &\equiv (x_1 \rightarrow x_{01}, x_2 \rightarrow x_{02}) \equiv \\ &\equiv (\lim x_1 = x_{01}, \lim x_2 = x_{02}). \end{aligned} \quad (6)$$

Аналогично независимой переменной  $\tilde{x}$  зависимая переменная  $\tilde{y}$  интервальной функции (5) в процессе своего изменения может неограниченно приближаться к некоторому предельному интервалу  $\tilde{y}_0 = [y_{01}, y_{02}]$ , т. е.

$$(\tilde{y} \rightarrow \tilde{y}_0) \equiv (y_1 \rightarrow y_{01}, y_2 \rightarrow y_{02}). \quad (7)$$

При этом, если неограниченное приближение зависимой переменной  $\tilde{y}$  интервальной функции (5) к  $\tilde{y}_0$  вызвано неограниченным приближением независимой переменной  $\tilde{x}$  этой функции к  $\tilde{x}_0$ , будем говорить, что предел интервальной функции (5) при  $\tilde{x}$ , стремящемся к  $\tilde{x}_0$ , равен интервалу  $\tilde{y}_0$ . Символически это записывается в виде

$$\lim_{\tilde{x} \rightarrow \tilde{x}_0} \tilde{y} = \tilde{y}_0 \text{ или, по-другому, } \lim_{\tilde{x} \rightarrow \tilde{x}_0} \tilde{f}(\tilde{x}) = \tilde{y}_0. \quad (8)$$

Если интервальная функция (5) непрерывна, т. е. как нижняя, так и верхняя границы интервала  $\tilde{y}$  (т. е. зависимой переменной) являются непрерывными функциями нижней и верхней границ интервала  $\tilde{x}$  (независимой переменной), то предел функции (5) равен значению функции от предельного значения аргумента, или, в символической записи,

$$\lim_{\tilde{x} \rightarrow \tilde{x}_0} \tilde{f}(\tilde{x}) = \tilde{f}(\tilde{x}_0). \quad (9)$$

## 2. Интервальная производная функция

Рассмотрим произвольную интервальную функцию (5). Будем считать ее непрерывной. Зафиксируем некоторое значение  $\tilde{x}_0 = [x_{01}, x_{02}]$  независимой переменной. Этому значению, в силу непрерывности нашей функции, будет соответствовать фиксированное значение функции  $\tilde{y}_0 = \tilde{f}(\tilde{x}_0)$ . Определим теперь приращения независимой и зависимой переменных нашей функции относительно их указанных фиксированных значений

$$\Delta \tilde{x} = \tilde{x} - \tilde{x}_0, \Delta \tilde{y} = \tilde{y} - \tilde{y}_0 = \tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x}_0) \quad (10)$$

и составим отношение второго приращения к первому

$$\frac{\Delta \tilde{y}}{\Delta \tilde{x}} = \frac{\tilde{y} - \tilde{y}_0}{\tilde{x} - \tilde{x}_0} = \frac{\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x}_0)}{\tilde{x} - \tilde{x}_0}. \quad (11)$$

Возьмем предел отношения (11) при неограниченном приближении независимой переменной  $\tilde{x}$  к ее фиксированному значению  $\tilde{x}_0$ :

$$\lim_{\tilde{x} \rightarrow \tilde{x}_0} \frac{\Delta \tilde{y}}{\Delta \tilde{x}}. \quad (12)$$

Предел (12), если он существует, будем называть производной интервальной функции от исходной интервальной функции (5) в точке  $\tilde{x}_0$  или, более коротко, интервальной производной от функции (5) и обозначать  $\tilde{y}'_{\tilde{x}_0}$  или же  $\tilde{f}'_{\tilde{x}_0}(\tilde{x})$ . Таким образом,

$$\tilde{y}'_{\tilde{x}_0} = \tilde{f}'_{\tilde{x}_0}(\tilde{x}) = \lim_{\tilde{x} \rightarrow \tilde{x}_0} \frac{\Delta \tilde{y}}{\Delta \tilde{x}}. \quad (13)$$

Условие существования интервальной производной интервальной функции определяется следующей теоремой.

**Теорема 1.** Для того чтобы в точке  $\tilde{x}_0$  существовала интервальная производная от интервальной функции (5), определяемая формулами (10)—(13), необходимо и достаточно, чтобы в некоторой окрестности этой точки, включая ее саму, все значения независимой переменной  $\tilde{x}$  функции были невырожденными интервалами (т. е. интервалами с несовпадающими между собой нижней и верхней границами).

**Доказательство.** Из выражения (11) интервальной функции, предел которой есть интервальная производная в точке  $\tilde{x}_0$ , видно, что эта производная существует только тогда, когда в некоторой окрестности указанной точки, включая ее саму, все возможные значения знаменателя выражения (11) не равны нулю. Но знаменатель выражения (11), согласно формуле (4) разности интервалов, равен интервалу

$$\tilde{x} - \tilde{x}_0 = [x_1, x_2] - [x_{01}, x_{02}] = [x_1 - x_{02}, x_2 - x_{01}].$$

Правый интервал равен нулю (так называемому нулевому интервалу  $[0, 0]$ ) только при условии

$$x_1 = x_{02}, x_2 = x_{01},$$

равносильном следующему:

$$x_1 = x_{01} = x_2 = x_{02}$$

или

$$\tilde{x} = \tilde{x}_0 = [x, x],$$

что означает вырожденность интервалов  $\tilde{x}$ ,  $\tilde{x}_0$ .

Ввиду произвольности выбранной точки  $\tilde{x}_0$  последнее означает, что для существования интервальной производной в точке  $\tilde{x}_0$  необходимо и достаточно, чтобы в некоторой окрестности этой точки, включая ее саму, все возможные значения независимой переменной функции  $\tilde{x}$  были невырожденными. Что и требовалось доказать.

Интервальная производная от интервальной функции (5), определяемая для произвольной точки  $\tilde{x}_0$  формулами (10)–(12) в виде предела, может быть выражена также в конечном виде через значения независимой  $\tilde{x}$  и зависимой  $\tilde{y} = \tilde{f}(\tilde{x})$  переменных этой функции в указанной точке. Именно, справедлива следующая теорема.

**Теорема 2.** Интервальная производная от непрерывной интервальной функции (5), определяемая для произвольной точки  $\tilde{x}_0$  формулами (10)–(13) в виде предела, может быть выражена в конечном виде напрямую через значения независимой  $\tilde{x}$  и зависимой  $\tilde{y} = \tilde{f}(\tilde{x})$  переменных этой функции в указанной точке следующим образом:

$$\tilde{y}'_{\tilde{x}_0} \equiv \tilde{f}'_{\tilde{x}_0}(\tilde{x}) = \frac{\tilde{f}(\tilde{x}_0) - \tilde{f}(\tilde{x}_0)}{\tilde{x} - \tilde{x}_0}. \quad (14)$$

*Примечание 1.* При первом взгляде выражение (14) может показаться неопределенностью вида  $0/0$ . Но это впечатление неверно, поскольку, согласно теореме 1, у любой существующей в точке  $\tilde{x}_0$  интервальной производной интервал  $\tilde{x}_0$  невырожден и потому, по формуле (4) разности интервалов, разность  $\tilde{x}_0 - \tilde{x}_0$  не равна нулю (нулевому интервалу  $[0, 0]$ ). Точно также не равна нулю разность  $\tilde{f}(\tilde{x}_0) - \tilde{f}(\tilde{x}_0)$ .

**Доказательство.** Согласно определению интервальной производной в точке  $\tilde{x}_0$  (10)–(13) она может быть записана в виде предела

$$\tilde{y}'_{\tilde{x}_0} \equiv \tilde{f}'_{\tilde{x}_0}(\tilde{x}) = \lim_{\tilde{x} \rightarrow \tilde{x}_0} \frac{\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x}_0)}{\tilde{x} - \tilde{x}_0}. \quad (15)$$

В процессе предельного перехода в правой части равенства (15)  $\tilde{x}$  неограниченно приближается к  $\tilde{x}_0$ , а  $\tilde{f}(\tilde{x})$ , в силу непрерывности интервальной

функции  $\tilde{f}$ , — к  $\tilde{f}(\tilde{x}_0)$ . Таким образом, предел в правой части (15) равен

$$\lim_{\tilde{x} \rightarrow \tilde{x}_0} \frac{\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x}_0)}{\tilde{x} - \tilde{x}_0} = \frac{\tilde{f}(\tilde{x}_0) - \tilde{f}(\tilde{x}_0)}{\tilde{x}_0 - \tilde{x}_0}. \quad (16)$$

Подставив полученное значение предела из (16) в выражение (15), получим формулу (14). Что и требовалось доказать.

Итак, выражение интервальной производной функции (14) представляет ее через исходную (первообразную) интервальную функцию (5). При этом, поскольку в указанном выражении точка  $\tilde{x}_0$ , в которой определяется производная, произвольна, выражение это можно переписать в общем виде так:

$$\tilde{y}'_{\tilde{x}} \equiv \tilde{f}'_{\tilde{x}}(\tilde{x}) = \frac{\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x}_0)}{\tilde{x} - \tilde{x}_0}, \quad (17)$$

где  $\tilde{x}$  — произвольное значение независимой переменной непрерывной интервальной функции (5) из ее области определения, при котором существует интервальная производная этой функции. Как видно из формулы (17), интервальная производная выражается непосредственно через исходную (первообразную) интервальную функцию простой алгебраической формой, что упрощает процесс вычисления производной. Такой эффект отсутствует при нахождении обычных производных, фигурирующих в традиционном дифференциальном исчислении детерминированных функций [6].

### 3. Интервальные производные высших порядков

Производная  $\tilde{f}'_{\tilde{x}}(\tilde{x})$  от интервальной функции  $\tilde{f}(\tilde{x})$ , введенная в п. 2, также является интервальной функцией, притом зависящей от того же самого интервального аргумента  $\tilde{x}$ . Это позволяет продолжить процесс взятия интервальных производных функций, получив сначала вторую производную  $\tilde{f}''_{\tilde{x}}(\tilde{x})$  (производную от первой производной  $\tilde{f}'_{\tilde{x}}(\tilde{x})$ )

$$\tilde{y}''_{\tilde{x}} = \tilde{f}''_{\tilde{x}}(\tilde{x}) = [\tilde{f}'_{\tilde{x}}(\tilde{x})]'_{\tilde{x}}, \quad (18)$$

затем третью производную  $\tilde{f}'''_{\tilde{x}}(\tilde{x})$  (интервальную производную от интервальной второй производной  $\tilde{f}''_{\tilde{x}}(\tilde{x})$ )

$$\tilde{y}'''_{\tilde{x}} = \tilde{f}'''_{\tilde{x}}(\tilde{x}) = [\tilde{f}''_{\tilde{x}}(\tilde{x})]'_{\tilde{x}} \quad (19)$$

и т. д., вплоть до интервальной производной любого  $n$ -го порядка, определяемой как производная от производной  $(n - 1)$ -го порядка

$$\tilde{y}^{(n)}_{\tilde{x}} \equiv \tilde{f}^{(n)}_{\tilde{x}}(\tilde{x}) = [\tilde{f}^{(n-1)}_{\tilde{x}}(\tilde{x})]'_{\tilde{x}}. \quad (20)$$

Условие существования интервальной производной  $n$ -го порядка определяется следующей теоремой.

**Теорема 3.** Для того чтобы в точке  $\tilde{x}_0$  существовала интервальная производная  $n$ -го порядка ( $n = 2, 3, \dots$ ) от интервальной функции (5), определяемая формулами (18)—(20), необходимо и достаточно, чтобы в некоторой окрестности этой точки, включая ее саму, все значения независимой переменной  $\tilde{x}$  интервальной функции были невырожденными интервалами.

**Доказательство.** По теореме 1 для существования в данной точке  $\tilde{x}_0$  производной 1-го порядка от функции (5) необходимо и достаточно невырожденности всех интервалов, являющихся значениями независимой переменной  $\tilde{x}$  указанной функции в некоторой окрестности точки  $\tilde{x}_0$ . Но производная 1-го порядка от функции (5) имеет ту же независимую переменную  $\tilde{x}$ , что и сама функция. Поэтому, согласно теореме 1, для существования в точке  $\tilde{x}_0$  производной от производной 1-го порядка функции (5) (т. е. производной 2-го порядка от интервальной функции (5)) необходимо и достаточно выполнения того же самого условия, что и для существования в точке  $\tilde{x}_0$  производной 1-го порядка от функции (5), т. е. невырожденности всех интервалов, являющихся значениями независимой переменной  $\tilde{x}$  функции (5) в некоторой окрестности точки  $\tilde{x}_0$ . Продолжая наши рассуждения, придем к необходимым и достаточным условиям существования в точке  $\tilde{x}_0$  производной  $n$ -го порядка от функции (5) в той форме, в которой они сформулированы в теореме 3. Что и требовалось доказать.

Интервальная производная любого  $n$ -го порядка от интервальной функции (5), определяемая формулами (18)—(20) итеративно, может быть выражена также в конечном виде, аналогично выражению в конечном виде (17) производной 1-го порядка. Эти выражения основаны на следующей теореме.

**Теорема 4.** Интервальная производная  $n$ -го порядка ( $n = 2, 3, \dots$ ) от функции (5) может быть выражена в конечном виде через интервальную производную  $(n - 1)$ -го порядка от этой функции (если она непрерывна) и независимую переменную  $\tilde{x}$  в следующем виде:

$$\tilde{y}_{\tilde{x}}^{(n)} \equiv \tilde{f}_{\tilde{x}}^{(n)}(\tilde{x}) = \frac{\tilde{f}_{\tilde{x}}^{(n-1)}(\tilde{x}) - \tilde{f}_{\tilde{x}}^{(n-1)}(\tilde{x})}{\tilde{x} - \tilde{x}}, \quad n = 2, 3, \dots \quad (21)$$

**Доказательство** получается с помощью формулы (17), если подставить в нее в качестве функции  $\tilde{f}(\tilde{x})$  производную  $(n - 1)$ -го порядка  $\tilde{f}_{\tilde{x}}^{(n-1)}(\tilde{x})$  и учесть еще, что производная 1-го порядка от  $\tilde{f}_{\tilde{x}}^{(n-1)}(\tilde{x})$ , согласно (20), есть производная  $n$ -го порядка  $\tilde{f}_{\tilde{x}}^{(n)}(\tilde{x})$ .

С помощью теорем 2 и 4 можно последовательно получить конечные выражения для интервальной производной 2-го порядка

$$\tilde{y}_{\tilde{x}}'' \equiv \tilde{f}_{\tilde{x}}''(\tilde{x}) = \frac{\tilde{f}_{\tilde{x}}'(\tilde{x}) - \tilde{f}_{\tilde{x}}'(\tilde{x})}{\tilde{x} - \tilde{x}} = \frac{\frac{\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x})}{\tilde{x} - \tilde{x}} - \frac{\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x})}{\tilde{x} - \tilde{x}}}{\tilde{x} - \tilde{x}}; \quad (22)$$

для интервальной производной 3-го порядка

$$\tilde{y}_{\tilde{x}}''' \equiv \tilde{f}_{\tilde{x}}'''(\tilde{x}) = \frac{\tilde{f}_{\tilde{x}}''(\tilde{x}) - \tilde{f}_{\tilde{x}}''(\tilde{x})}{\tilde{x} - \tilde{x}} = \frac{\frac{\frac{\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x})}{\tilde{x} - \tilde{x}} - \frac{\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x})}{\tilde{x} - \tilde{x}}}{\tilde{x} - \tilde{x}} - \frac{\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x})}{\tilde{x} - \tilde{x}}}{\tilde{x} - \tilde{x}} \quad (23)$$

и т. д. Отметим, что эти выражения аналогичны конечному выражению (17) для интервальной производной 1-го порядка.

**Примечание 2.** Приведенные выражения (22), (23) интервальных производных высших порядков кажутся с первого взгляда суперпозициями неопределенностей вида  $0/0$ , однако, по причине, указанной в примечании 1, не являются таковыми. Заметим также, что эти выражения не могут быть преобразованы к более простому виду с помощью эквивалентных преобразований, так как в интервальной алгебре [3] не выполняются эквивалентности

$$(\tilde{a} \pm \tilde{b})\tilde{c} = \tilde{a}\tilde{c} \pm \tilde{b}\tilde{c}; \quad \frac{\tilde{a} \pm \tilde{b}}{\tilde{c}} = \frac{\tilde{a}}{\tilde{c}} \pm \frac{\tilde{b}}{\tilde{c}}. \quad (24)$$

#### 4. Вычисление интервальных производных

Доказанные выше выражения интервальных производных (17)—(23) являются аналитическими выражениями, имеющими вид суперпозиций операций над интервальными переменными. Эти выражения удобны для теоретического изучения интервальных производных.

Что касается вычисления производных, то использование здесь выражений (17)—(23) оказывается неудобным, поскольку оно предполагает объемную работу с весьма громоздкими формулами интервальной математики (4), которые и позволяют выразить в конце концов интервальную производную в виде интервального числа с явно выраженными в числовой форме нижней и верхней границами. Однако гораздо удобнее вычислять интервальные производные по формуле, выражающей в явном виде нижнюю и верхнюю границы интервального числа, служащего значением интервальной производной. Эта формула выводится ниже.

**Теорема 5.** Интервальная производная от непрерывной интервальной функции (5), выражаемая для произвольной точки  $\tilde{x}$  формулой (17) в виде суперпозиции операций над интервальными пере-

менными, может быть также представлена явно в виде интервала, нижняя и верхняя границы которого выражены через нижнюю и верхнюю границы интервальных независимой переменной  $\tilde{x} = [x_1, x_2]$  и зависимой переменной  $\tilde{y} = [y_1, y_2]$  этой функции в точке  $\tilde{x}$  следующим образом:

$$\tilde{y}'_{\tilde{x}} \equiv \tilde{f}'_{\tilde{x}}(\tilde{x}) = \left[ \frac{y_2 - y_1}{x_2 - x_1}, \frac{y_2 - y_1}{x_2 - x_1} \right]. \quad (25)$$

**Доказательство.** Будем исходить из выражения интервальной производной от интервальной функции (5) формулой (17). Представим указанную функцию в явном виде интервала:

$$\tilde{f}(\tilde{x}) = [f_1(\tilde{x}), f_2(\tilde{x})] \equiv [y_1, y_2] \equiv \tilde{y}. \quad (26)$$

Здесь

$$y_1 = f_1(\tilde{x}), y_2 = f_2(\tilde{x}) \quad (27)$$

есть соответственно нижняя и верхняя границы интервальной зависимой переменной функции (5). Так же представим независимую переменную  $\tilde{x}$ :

$$\tilde{x} = [x_1, x_2]. \quad (28)$$

После проведения необходимых подстановок выражение (17) интервальной производной от интервальной функции (5) примет вид

$$\tilde{y}'_{\tilde{x}} \equiv \tilde{f}'_{\tilde{x}}(\tilde{x}) = \frac{\tilde{y} - \tilde{y}}{\tilde{x} - \tilde{x}} = \frac{[y_1, y_2] - [y_1, y_2]}{[x_1, x_2] - [x_1, x_2]}. \quad (29)$$

Разности интервалов в числителе и знаменателе выражения (29) по формуле (4) представим в виде интервального числа

$$\begin{aligned} [y_1, y_2] - [y_1, y_2] &= [y_1 - y_2, y_2 - y_1], \\ [x_1, x_2] - [x_1, x_2] &= [x_1 - x_2, x_2 - x_1]. \end{aligned} \quad (30)$$

Подставив выражения (30) в (29), получим представление производной в виде частного двух интервалов:

$$\tilde{y}'_{\tilde{x}} \equiv \tilde{f}'_{\tilde{x}}(\tilde{x}) = \frac{[y_1 - y_2, y_2 - y_1]}{[x_1 - x_2, x_2 - x_1]}. \quad (31)$$

В свою очередь, выражение (31) можно, согласно (4), представить в виде следующего произведения двух интервалов:

$$\begin{aligned} \tilde{y}'_{\tilde{x}} &\equiv \tilde{f}'_{\tilde{x}}(\tilde{x}) = \\ &= [y_1 - y_2, y_2 - y_1] \cdot [1/(x_2 - x_1), 1/(x_1 - x_2)]. \end{aligned} \quad (32)$$

Перемножая интервалы в правой части равенства (32) по соответствующему правилу (4), получим

$$\begin{aligned} \tilde{y}'_{\tilde{x}} &= \left[ \min \left( \frac{y_1 - y_2}{x_2 - x_1}, \frac{y_1 - y_2}{x_1 - x_2}, \frac{y_2 - y_1}{x_2 - x_1}, \frac{y_2 - y_1}{x_1 - x_2} \right), \right. \\ &\left. \max \left( \frac{y_1 - y_2}{x_2 - x_1}, \frac{y_1 - y_2}{x_1 - x_2}, \frac{y_2 - y_1}{x_2 - x_1}, \frac{y_2 - y_1}{x_1 - x_2} \right) \right]. \end{aligned} \quad (33)$$

Одинаково подчеркнутые члены в выражении (33) равны. Оставив из каждых двух равных членов по одному, представим (33) в простом виде:

$$\begin{aligned} \tilde{y}'_{\tilde{x}} &\equiv \tilde{f}'_{\tilde{x}}(\tilde{x}) = \\ &= \left[ \min \left( \frac{y_2 - y_1}{x_2 - x_1}, \frac{y_2 - y_1}{x_1 - x_2} \right), \max \left( \frac{y_2 - y_1}{x_2 - x_1}, \frac{y_2 - y_1}{x_1 - x_2} \right) \right]. \end{aligned} \quad (34)$$

Два члена в круглых скобках формулы (34) различаются лишь знаком, причем (с учетом того, что  $x_2 > x_1, y_2 > y_1$ ) левый член положителен, а правый — отрицателен. Так, из (34) получаем простейшее выражение интервальной производной от интервальной функции (5):

$$\tilde{y}'_{\tilde{x}} \equiv \tilde{f}'_{\tilde{x}}(\tilde{x}) = \left[ \frac{y_2 - y_1}{x_1 - x_2}, \frac{y_2 - y_1}{x_2 - x_1} \right]. \quad (35)$$

С учетом того, что  $x_1 - x_2 = -(x_2 - x_1)$ , выражение (35) окончательно переписывается в виде

$$\tilde{y}'_{\tilde{x}} \equiv \tilde{f}'_{\tilde{x}}(\tilde{x}) = \left[ -\frac{y_2 - y_1}{x_2 - x_1}, \frac{y_2 - y_1}{x_2 - x_1} \right], \quad (36)$$

что и требовалось доказать.

Аналогичное (25) явное выражение интервальной производной функции, дающее сразу в явном виде нижнюю и верхнюю границы интервального числа — значения этой функции — справедливо и для интервальных производных высших порядков. Это выражение приведено ниже.

**Теорема 6.** Интервальная производная любого  $n$ -го порядка от непрерывной интервальной функции  $\tilde{y} = \tilde{f}(\tilde{x})$  (5), выражаемая для произвольной точки  $\tilde{x}$  соотношением (21) в виде суперпозиции операций над интервальными переменными, может быть представлена в этой точке также в явном виде интервала, нижняя и верхняя границы которого выражены через соответственно нижнюю и верхнюю границы интервальных независимой переменной  $\tilde{x} = [x_1, x_2]$  и производной  $(n - 1)$ -го порядка  $\tilde{y}^{(n-1)} = [y_1^{(n-1)}, y_2^{(n-1)}]$  от интервальной функции (5) следующим образом:

$$\begin{aligned} \tilde{y}'_{\tilde{x}}^{(n)} &\equiv \tilde{f}'_{\tilde{x}}^{(n)}(\tilde{x}) = \\ &= \left[ \frac{y_2^{(n-1)} - y_1^{(n-1)}}{x_2 - x_1}, \frac{y_2^{(n-1)} - y_1^{(n-1)}}{x_2 - x_1} \right]. \end{aligned} \quad (37)$$

**Доказательство** теоремы 6 аналогично доказательству теоремы 5, с той лишь разницей, что в качестве исходного выражения искомой интервальной производной  $n$ -го порядка от функции (5) берется формула (21), тогда как в случае теоремы 5 исходным выражением искомой интервальной производной 1-го порядка от функции (5) служит такая же по форме формула (17).

Используя формулу (37), нетрудно получить простые явные выражения типа (25) для интервальных производных 2-го, 3-го и т. д. порядка от интервальной функции (5). Для интервальной производной 2-го порядка из (37) имеем следующее выражение:

$$\tilde{y}_x'' = \tilde{f}_x''(\tilde{x}) = \left[ \frac{y_2' - y_1'}{x_2 - x_1}, \frac{y_2' - y_1'}{x_2 - x_1} \right]. \quad (38)$$

Подставляя в соотношение (38) значения нижней и верхней границ интервальной производной 1-го порядка из (25) в виде

$$y_1' = -\frac{y_2 - y_1}{x_2 - x_1}, \quad y_2' = \frac{y_2 - y_1}{x_2 - x_1}, \quad (39)$$

с учетом необходимых эквивалентных преобразований, получим

$$\tilde{y}_x'' \equiv \tilde{f}_x''(\tilde{x}) = \left[ -\frac{2(y_2 - y_1)}{(x_2 - x_1)^2}, \frac{2(y_2 - y_1)}{(x_2 - x_1)^2} \right]. \quad (40)$$

Аналогично, для интервальной производной 3-го порядка из (37) имеем

$$\tilde{y}_x''' \equiv \tilde{f}_x'''(\tilde{x}) = \left[ \frac{y_2'' - y_1''}{x_2 - x_1}, \frac{y_2'' - y_1''}{x_2 - x_1} \right]. \quad (41)$$

Как и в предыдущем случае, подставляя в формулу (41) значения границ интервальной производной 2-го порядка из (40) в виде

$$y_1'' = -\frac{2(y_2 - y_1)}{(x_2 - x_1)^2}, \quad y_2'' = \frac{2(y_2 - y_1)}{(x_2 - x_1)^2}, \quad (42)$$

после необходимых эквивалентных преобразований будем иметь

$$\tilde{y}_x''' \equiv \tilde{f}_x'''(\tilde{x}) = \left[ -\frac{4(y_2 - y_1)}{(x_2 - x_1)^3}, \frac{4(y_2 - y_1)}{(x_2 - x_1)^3} \right]. \quad (43)$$

Продолжая процесс для интервальных производных 4-го, 5-го и последующих порядков, приходим к общему результату.

**Теорема 7.** Интервальная производная любого  $n$ -го порядка от непрерывной интервальной функции  $\tilde{y} = \tilde{f}(\tilde{x})$  вида (5), выражаемая для произвольной точки  $\tilde{x}$  формулой (21) как суперпозиция операций над интервальными переменными, может быть представлена в этой точке также в виде интервала с явно выраженными границами вида (25), (40), (43):

$$\tilde{y}_x^{(n)} \equiv \tilde{f}_x^{(n)}(\tilde{x}) = \left[ -\frac{2^{n-1}(y_2 - y_1)}{(x_2 - x_1)^n}, \frac{2^{n-1}(y_2 - y_1)}{(x_2 - x_1)^n} \right]. \quad (44)$$

Здесь  $x_1, x_2$  — нижняя и верхняя границы интервальной независимой переменной  $\tilde{x} = [x_1, x_2]$  в точке взятия производной от нашей интервальной функции  $\tilde{y} = \tilde{f}(\tilde{x})$ , а  $y_1, y_2$  — нижняя и верхняя границы интервальной зависимой переменной  $\tilde{y} = [y_1, y_2]$  этой функции в той же точке.

Сводка полученных явных выражений интервальных производных различных порядков приведена в таблице.

| Функция  | Обозначение функции   | Явное выражение функции   |
|--|---|---|
| Исходная интервальная функция                    | $\tilde{y} = \tilde{f}(\tilde{x}), \tilde{x} = [x_1, x_2], \tilde{y} = [y_1, y_2]$                            | $[y_1, y_2] = \tilde{f}[x_1, x_2]$ или $[y_1, y_2] = [f_1(x_1, x_2), f_2(x_1, x_2)]$                |
| Интервальная производная функция 1-го порядка    | $\tilde{y}_x' \equiv \tilde{f}_x'(\tilde{x})$ или $\tilde{y}_x^{(1)} \equiv \tilde{f}_x^{(1)}(\tilde{x})$     | $\left[ \frac{y_2 - y_1}{x_2 - x_1}, \frac{y_2 - y_1}{x_1 - x_2} \right]$                           |
| Интервальная производная функция 2-го порядка    | $\tilde{y}_x'' \equiv \tilde{f}_x''(\tilde{x})$ или $\tilde{y}_x^{(2)} \equiv \tilde{f}_x^{(2)}(\tilde{x})$   | $\left[ -\frac{2(y_2 - y_1)}{(x_2 - x_1)^2}, \frac{2(y_2 - y_1)}{(x_2 - x_1)^2} \right]$            |
| Интервальная производная функция 3-го порядка    | $\tilde{y}_x''' \equiv \tilde{f}_x'''(\tilde{x})$ или $\tilde{y}_x^{(3)} \equiv \tilde{f}_x^{(3)}(\tilde{x})$ | $\left[ -\frac{4(y_2 - y_1)}{(x_2 - x_1)^3}, \frac{4(y_2 - y_1)}{(x_2 - x_1)^3} \right]$            |
| ...  | ...   | ...   |
| Интервальная производная функция $n$ -го порядка | $\tilde{y}_x^{(n)} \equiv \tilde{f}_x^{(n)}(\tilde{x})$   | $\left[ \frac{2^{n-1}(y_2 - y_1)}{(x_2 - x_1)^n}, \frac{2^{n-1}(y_2 - y_1)}{(x_2 - x_1)^n} \right]$ |

## 5. Возможные применения

Рассмотренный в настоящей работе математический аппарат недетерминистского дифференциального исчисления может быть успешно применен к построению и количественному изучению свойств различных недетерминированных систем динамического типа, т. е. систем, количественные характеристики которых известны с той или иной степенью неопределенности и вдобавок изменяются во времени. Такие системы широко распространены в экономике, социологии и экологии. Они также встречаются в технике и технологиях. Методика применения указанного математического аппарата к этим системам основана на предположении о том, что все параметры изучаемой системы определяются с точностью до интервалов возможных значений. При этом статические звенья характеризуются постоянными интервалами возможных значений их параметров, а динамические звенья — переменными интервалами данных значений. В соответствии с этим построение действующей математической модели изучаемой недетерминированной системы и последующее математическое модели-



рование этой системы осуществляется по следующему алгоритму:

**Шаг 1.** Взять (построить) математическую модель идеального (детерминированного) прообраза системы, которая получается в предположении, что все параметры системы заданы точно, соответственно чему статические звенья системы имеют постоянные точечные значения параметров, а динамические звенья — переменные точечные значения параметров; таким образом, в этой модели изучаемые характеристики системы принимают точечные постоянные и переменные значения и могут быть представлены в виде суперпозиции обычных алгебраических операций над точно известными постоянными и переменными значениями параметров звеньев системы.

**Шаг 2.** Заменить в полученной на шаге 1 математической модели точечные значения параметров соответствующими интервалами возможных значений параметров; обычные производные в исходной детерминированной системе заменяются соответствующими интервальными производными, в результате получаем математическую модель собственно изучаемой системы, все параметры которой определяются с точностью до интервалов возможных значений; в этой модели изучаемые характеристики системы принимают интервальные значения и могут быть представлены в виде суперпозиции интервальных операций (3) над интервалами — значениями параметров системы.

**Шаг 3.** Используя формулы (4), выражающие результаты элементарных преобразований интервалов, представляем интервальнозначные характеристики недетерминированной системы в форме интервала  $\tilde{F} = [f_1, f_2]$ , нижняя  $f_1$  и верхняя  $f_2$  границы которого есть обычные функции от нижних и верхних границ интервалов — параметров звеньев системы. После этого математическое моделирование недетерминированной системы по любой ее интервальной характеристике  $\tilde{F} = [f_1, f_2]$  сводится к анализу детерминированных функций  $f_1, f_2$ .

**Пример.** Опишем простейшее применение предложенного математического аппарата в экономике. Пусть  $y = f(x)$  — некоторая детерминированная функция, показывающая зависимость одного экономического показателя  $y$  от другого  $x$  [7]. Тогда эластичностью  $E_x(y)$  этой функции называется предел относительного приращения функции  $y$  к относительному приращению аргумента (независимой переменной)  $x$  при  $\Delta x \rightarrow 0$ , т. е.

$$E_x(y) = \lim_{\Delta x \rightarrow 0} \left( \frac{\Delta y}{y} / \frac{\Delta x}{x} \right) = \frac{x}{y} \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \frac{x}{y} y'_x. \quad (45)$$

Здесь  $y'_x$  — производная от функции  $y = f(x)$  по переменной  $x$ . Эластичность функции показывает приближенно, на сколько процентов изменится значение функции  $y = f(x)$  при изменении аргумента  $x$  на 1 %. Например, если функция  $y = f(x)$  пока-

зывает зависимость спроса  $y$  на товар от его цены  $x$ , то эластичность этой функции  $E_x(y)$  есть коэффициент, приближенно показывающий, на сколько процентов изменится спрос на товар при изменении его цены на 1 %. Аналогично, если функция  $y = f(x)$  показывает зависимость издержек производства  $y$  от объема выпускаемой продукции  $x$ , то эластичность этой функции  $E_x(y)$  показывает, как изменятся издержки производства при изменении объема выпуска продукции на 1 %. Наконец, если функция  $y = f(x)$  есть зависимость себестоимости единицы продукции  $y$  от стоимости всей продукции  $x$ , то  $E_x(y)$  показывает, как изменится себестоимость этой единицы при изменении стоимости всей продукции на 1 % и т. д.

Теперь положим, что (как это бывает на практике) независимая  $x$  и зависимая  $y$  переменные рассматриваемой экономической функции  $y = f(x)$  определяются не точно, а с точностью до интервалов возможных значений. Следовательно, эта функция из детерминированной превращается в недетерминированную (интервальную) функцию  $\tilde{y} = \tilde{f}(\tilde{x})$  вида (5), где  $\tilde{x} = [x_1, x_2]$  — независимая интервальная переменная (аргумент), а  $\tilde{y} = [y_1, y_2]$  — зависимая интервальная переменная. Построим математическую модель получившейся экономической системы, используя описанный выше трехшаговый алгоритм.

**Шаг 1.** Исходная математическая модель идеальной (детерминированной) системы уже есть, она описывается функцией эластичности системы (45) с точными значениями переменных  $x$  и  $y$ .

**Шаг 2.** Заменяем в математической модели, установленной на шаге 1, точные значения параметров  $x$  и  $y$  интервалами их значений  $\tilde{x} = [x_1, x_2]$  и  $\tilde{y} = [y_1, y_2]$ , а детерминированную производную 1-го порядка  $y'_x$  — соответствующей интервальной производной  $\tilde{y}'_{\tilde{x}}$ , явное выражение которой в виде интервала дается формулой (25). В результате получаем математическую модель искомой недетерминированной экономической системы — ее интервальную функцию эластичности — в виде такой суперпозиции интервальных операций над интервалами — параметрами системы:

$$\tilde{E}_{\tilde{x}}(\tilde{y}) = \frac{\tilde{x}}{\tilde{y}} \tilde{y}'_{\tilde{x}} = \left[ \frac{x_1, x_2}{y_1, y_2} \right] \left[ \frac{y_2 - y_1}{x_2 - x_1}, \frac{y_2 - y_1}{x_2 - x_1} \right]. \quad (46)$$

**Шаг 3.** После вычисления выражения (46) по формулам (4) элементарных преобразований интервалов, с учетом положительности значений  $x_1, x_2, y_1, y_2$  в экономических системах, получим математическую модель системы:

$$\tilde{E}_{\tilde{x}}(\tilde{y}) = \left[ \frac{x_2(y_2 - y_1)}{y_1(x_2 - x_1)}, \frac{x_2(y_2 - y_1)}{y_1(x_2 - x_1)} \right]. \quad (47)$$

## Заклучение

В настоящей статье показывается возможность обобщения классического дифференциального исчисления на случай недетерминированных функций, в которых переменные задаются с точностью до интервалов возможных значений. Новое дифференциальное исчисление идейно близко к классическому дифференциальному исчислению, в частности, производная функция показывает скорость изменения первообразной функции относительно ее аргумента. Однако форма нового исчисления существенно иная. Главные отличия состоят в том, что:

- производная любого порядка является интервальной функцией, в которой все переменные имеют вид интервалов;

- производная любого порядка выражается в явном виде через значения независимой и зависимой переменных первообразной функции.

## Список литературы

1. **Гнеденко Б. В.** Курс теории вероятностей. М.: Наука, 2004.
2. **Заде Л. А.** Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976.
3. **Алефельд Г., Херибергер Ю.** Введение в интервальные вычисления. М.: Мир, 1987.
4. **Левин В. И.** Интервальные методы оптимизации систем в условиях неопределенности. Пенза: Изд-во Пензенского технологического ин-та, 1999.
5. **Левин В. И.** Оптимизация в условиях интервальной неопределенности. Метод детерминизации // Автоматика и вычислительная техника. 2012. № 4.
6. **Фихтенгольц Г. М.** Курс дифференциального и интегрального исчисления. Т. 1. М.: Наука, 2005.
7. **Кремер Н. Ш.** Высшая математика для экономистов. М.: ЮНИТИ, 2001.

V. I. Levin, Professor, e-mail: vilevin@mail.ru  
Penza State Technological University

## Wiping Differential Calculus and its Application

*A generalization of the classical differential calculus on the Newton-Leibniz function with interval uncertainty. In these functions, independent and dependent variables are defined as intervals of possible values. Construction of the new calculus algebra essentially uses interval numbers. Additionally, we use the concept of the limit interval function that is similar to the concept introduced limit normal function. The basic concept is the concept of the proposed calculation interval derivative. It is usual for the initial interval function as well as the concept of a classical derivative for normal function. However, the properties of interval derivative significantly different from those of the classical derivative. This is due to the laws of honors algebra interval number of laws of algebra of real numbers. It is proved that the existence of an interval derivative at some point it is necessary and sufficient that in some neighborhood of all the values of the independent variable initial interval function were non-degenerate intervals. An explicit expression is derived from the interval, interval function:  $\tilde{f}'(\tilde{x}) = (\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{x})) / (\tilde{x} - \tilde{x})$ .*

*Introduced the concept of interval iteratively higher derivative (n-th) order as a derivative of the derivative of (n - 1) th order. In combination with the expression (1) the derivative of the 1st order is sequentially allows to obtain explicit expressions interval of the 2-nd, 3-rd and subsequent orders. Proved that interval derivatives 2-nd, 3-rd and all subsequent orders exist for the same necessary and sufficient condition that the interval derivative of the 1st order. Obtain estimates of interval derivatives of various orders in the form of intervals, lower and upper bounds are expressed through the lower and upper boundaries of the initial interval function and its independent variable. Possible applications of interval derivatives in economics, sociology and technology. Is an example.*

**Keywords:** interval, interval function, function calculus, interval derivative, interval computing, nondeterministic differential calculus

## References

1. **Gnedenko B. V.** Kurs teorii verojatnosti. M.: Nauka, 2004.
2. **Zade L. A.** Poniatie lingvisticheskoj peremennoi i ego primeneniye k priinaitija priblizhennin reshenij. M.: Mir, 1976.
3. **Alefeld G., Herzberger Yu.** Vvedenie v intervalnye vichuslenija. M.: Mir, 1987.
4. **Levin V. I.** Intervalnije metodi optimizazii sistem v uslovijach neopredelennosti. Penza. Izd-vo Penzenskogo tehnologicheskogo institute. 1999.
5. **Levin V. I.** Optimizacija v usloviach intervalnoj neopredelennosti. Metod determinizazii. *Avtomatika u vichislitel'naja tehnika.* 2012. N. 4.
6. **Fih tengolz G. M.** Kurs differenzialnogo i integralnogo ischislenija. T. 1. M.: Nauka, 2005.
7. **Kremer N. Sh.** Vischaja matematika dla ekonomistov. M.: YuNITI, 2001.

**А. А. Варфоломеева**, студент, Московский физико-технический институт,  
**В. В. Стрижов**, канд. физ.-мат. наук, доц., науч. сотр., Вычислительный Центр РАН,  
 e-mail: strijov@ccas.ru

## Алгоритм разметки библиографических списков методами структурного обучения

*Решается прикладная задача сегментации структурированных текстов: для каждого сегмента библиографической записи определяется его тип поля в формате BibTeX. Также для каждой записи определяется тип ее библиографического описания. В работе предлагается алгоритм разметки библиографических списков методом структурной регрессии. Решается задача выбора параметров регрессионной модели. Качество полученной модели исследуется на наборе неформатированных библиографических списков.*

**Ключевые слова:** разметка текстов, структурное обучение, сегментирование, выбор признаков, кластеризация

### Введение

Работа посвящена построению модели, прогнозирующей структуру текстовой строки. Решается прикладная задача представления неформатированных библиографических записей в виде структуры в формате BibTeX [1] — стандарта управления коллекцией библиографических записей. Требуется разметить библиографические записи: определить соответствия между текстовыми сегментами и полями записей BibTeX, а также указать тип библиографической записи. Необходимость форматирования текстовой строки вызвана наличием различных стандартов (ГОСТ 7.82—2001, MLA и др.), определяющих различный порядок следования полей библиографической записи. Кроме того, запись может быть составлена с нарушением стандартов.

Методы, предложенные ранее для решения задачи сегментирования текстов, описаны в работе [2], где граф цитирований строится с помощью полученной разметки библиографического списка. В работе [3] описана постановка и решение задачи разметки адресной строки. Используется скрытая марковская цепь, недостатком которой является не точное описание структурных зависимостей внутри исходных данных. Автоматическая разметка библиографических записей представлена в работе [4], для обучения модели которой использовалась библиотека одного из стандартов, в силу чего модель неустойчиво работает на других стандартах.

В данной работе для поиска структуры текстовой строки предлагается использовать методы структурного обучения, описанные в работах [5—7]. Эти методы используют в области анализа текстов для определения синтаксических зависимостей в предложении [8]. В предлагаемом алгоритме сегментация записей выполняется на базе метода структурной регрессии [9]. Ставится и решается задача определения соответствий между текстовыми сегментами библиографической записи и набором полей структуры BibTeX. Рассматриваются следующие типы

полей: автор; название работы; название источника; номера страниц; номер выпуска; номер тома; год; город; издательство; редакторы; ссылка на работу в Интернете. В качестве типов библиографических записей рассматриваются статья, книга, тезис конференции, диссертация, электронный источник.

В таблице приведен пример неформатированной библиографической записи и верные либо неверные соотношения между сегментами записи и набором полей BibTeX.

Решается задача выбора оптимального набора признаков модели. Для этого используется модификация ранее предложенного авторами [10, 11] алгоритма последовательного добавления и удале-

**Пример верного и неверного сегментирования библиографической записи**

| <i>Kwok T. Y., Yeung D. Y. Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems // IEEE Transactions on Neural Networks, 1997. Vol. 8. Pp. 630–645.</i> |  |  |
|--|--|--|
|  | Верно  | Неверно  |
| Type   | Article  | Book   |
| <b>Author</b>  | <i>Kwok T. Y., Yeung D. Y.</i>   | <i>Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems</i> |
| <b>Title</b>   | <i>Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems</i> | <i>IEEE Transactions on Neural Networks</i>  |
| <b>Journal</b>   | <i>IEEE Transactions on Neural Networks</i>  | —  |
| <b>Pages</b>   | 630–645  | —  |
| <b>Volume</b>  | 8  | 1997   |
| <b>Number</b>  | —  | 8  |
| <b>Year</b>  | 1997   | 630–645  |
| <b>Address</b>   | —  | —  |
| <b>Publisher</b>   | —  | —  |
| <b>Editor</b>  | —  | <i>Kwok T. Y., Yeung D. Y.</i>   |
| <b>URL</b>   | —  | —  |

ния признаков. Веса признаков оцениваются для случая логистической регрессии [12].

После построения соотношений между текстовыми сегментами и набором полей структуры BibTeX требуется определить тип библиографической записи. Для этого строится новое признаковое описание библиографической записи как подмножество полей структуры BibTeX, присутствующих в библиографической записи. С помощью кластеризации *k-means* [13] определяется тип каждой записи.

Работа содержит описания двух подзадач:

1) определение для каждого сегмента типа поля библиографической записи в структуре BibTeX;

2) определение типа библиографической записи.

Работа предложенного метода демонстрируется на наборе неформатированных библиографических списков.

### 1. Постановка задачи

Дан набор из  $m$  строк  $\{t_1, t_2, \dots, t_m\}$  — библиографических записей. Каждая запись  $t$  состоит из текстовых сегментов  $t_i = \{t_i^1, t_i^2, \dots, t_i^n\}$ . Задан набор  $G$  порождающих функций  $G = \{g\}$ , отображающих  $j$ -й текстовый сегмент  $i$ -й строки  $s_i^j$  в вектор-строку признаков  $\mathbf{x}_{ij}$ :  $g: s_i^j \mapsto \mathbf{x}_{ij}$ .

Задана трехиндексная матрица  $\mathbf{X}$  (рис. 1) "объект — сегмент — признак" размера  $m \times n \times p$ , где  $m$  — число библиографических записей;  $n$  — число сегментов;  $p$  — число признаков. Каждому объекту, заданному двухиндексной матрицей  $\mathbf{X}_i$ , где  $i$  — номер объекта, поставлена в соответствие бинарная матрица ответов  $\mathbf{Y}_i$  размера  $n \times s$ , где  $s$  — число типов полей структуры BibTeX. Элементы матрицы  $\mathbf{Y}_i$  отвечают за принадлежность  $j$ -го сегмента к  $l$ -му типу поля библиографической записи:

$$Y_i(j, l) = \begin{cases} 1, & \text{если } \mathbf{x}_{ij} \text{ принадлежит к } l\text{-му типу поля;} \\ 0, & \text{иначе,} \end{cases} \quad (1)$$

где  $j \in \{1, 2, \dots, n\}$  — индекс сегмента текстовой строки,  $l \in \{1, 2, \dots, s\}$  — индекс типа поля. Вводится двухиндексная матрица весовых параметров  $\mathbf{W}$  размером  $p \times s$ , элементы которой  $w_{kl}$  отвечают за значимость  $k$ -го признака для  $l$ -го типа поля,  $k = 1, \dots, p$ ,  $l = 1, \dots, s$ . При умножении матрицы  $\mathbf{W}$  справа на

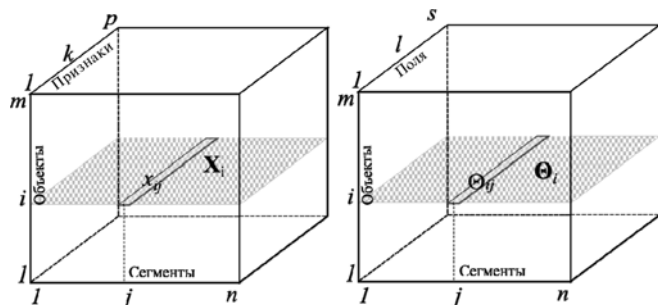


Рис. 1. Вид матриц  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\Theta$

вектор-строку  $\mathbf{x}_{ij}$  признаков  $j$ -го сегмента  $i$ -го объекта получается вектор-строка  $\theta_{ij}$ , элементы которой определяют оценку принадлежности данного сегмента к полям структуры BibTeX:

$$\theta_{ij} = \mathbf{x}_{ij} \mathbf{W} \quad (2)$$

Тогда оптимальный прогнозируемый тип поля с индексом  $\hat{l}$  для признакового описания  $\mathbf{x}_{ij}$  сегмента библиографической записи с фиксированным номером  $i$  определяется как индекс максимального элемента вектор-строки  $\theta_{ij}$ :

$$\hat{l}_j = \operatorname{argmax}_{l=1,2,\dots,s} \theta_{ij}(l).$$

Аналогично, записывая строки  $\theta_{ij}$  для каждого вектора признаков  $\mathbf{x}_{ij}$  с индексом  $j$  объекта  $\mathbf{X}_i$ , составляется матрица оценок  $\Theta_i$ , значения которой определяют тип поля каждого сегмента объекта  $\mathbf{X}_i$ :

$$\hat{Y}_i(j, l) = \begin{cases} 1, & \text{если } l = \hat{l}_j, \\ 0, & \text{иначе.} \end{cases} \quad (3)$$

Требуется найти такой набор признаков  $\mathbf{A}$  из множества  $G$  и веса  $\mathbf{W}$  этих признаков, что расстояние  $\operatorname{Dist}(\hat{\mathbf{Y}}, \mathbf{Y})$  между матрицей ответов  $\mathbf{Y}$  и прогнозируемой матрицей  $\hat{\mathbf{Y}}$  минимально:

$$\hat{\mathbf{W}} = \operatorname{argmin}_{\mathbf{W}, G} \operatorname{Dist}(\hat{\mathbf{Y}}, \mathbf{Y}) = \operatorname{argmin}_{\mathbf{W}, G} \frac{1}{2} \sum_{i,j,k=1}^{m,n,p} |\hat{Y}_i(j, k) - Y_i(j, k)|. \quad (4)$$

Искомая матрица весов  $\hat{\mathbf{W}}$  определяется минимумом аппроксимированного эмпирического риска  $Q$  для случая логистической регрессии.

Введем обозначения, необходимые для определения функции эмпирического риска  $Q$ . Матрица весов признаков  $\mathbf{W}$  разбивается на  $s$  независимых вектор-столбцов  $\mathbf{w}_l$ , соответствующих типу поля  $l$ ,  $l \in \{1, 2, \dots, s\}$ , и для каждого столбца  $\mathbf{w}_l$  векторизируется соответствующая ему часть матрицы  $\mathbf{Y}$ :

$$\mathbf{u}_l = \begin{pmatrix} \mathbf{Y}_1(l) \\ \vdots \\ \mathbf{Y}_i(l) \\ \vdots \\ \mathbf{Y}_m(l) \end{pmatrix},$$

где  $\mathbf{Y}_i(l)$  —  $l$ -й столбец матрицы  $\mathbf{Y}_i$ .

Матрица  $\mathbf{X}$  записывается в двумерном виде:

$$\mathbf{Z} = \begin{pmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_i \\ \vdots \\ \mathbf{X}_m \end{pmatrix},$$

где  $\mathbf{X}_i$  — матрица признаков  $i$ -го объекта библиографической записи.

Обозначим  $\mathbf{z}_i$  —  $i$ -ю строку матрицы  $\mathbf{Z}$ . Тогда функция эмпирического риска записывается в виде

$$Q = \sum_{i=1}^{m \times n} \sum_{l=1}^s L(\mathbf{w}_l, \mathbf{z}_i, u_l(i), A),$$

где функция потерь

$$L(\mathbf{w}_l, \mathbf{z}_i, u_l(i), A) = \log(1 + \exp(-\langle \mathbf{w}_l^A, \mathbf{z}_i^A \rangle u_l(i))).$$

При этом  $\mathbf{w}_l^A$  и  $\mathbf{z}_i^A$  содержат только подмножество  $A \subset G$  своих элементов индексов признаков:

$$\hat{A} = \operatorname{argmax}_{A \subset G} Q(\mathbf{W}, \mathbf{Z}, \mathbf{u}, A),$$

где  $\hat{A}$  — решение оптимизационной задачи.

## 2. Описание алгоритма выбора признаков

Требуется решить задачу выбора оптимального подмножества индексов признаков  $A \subset G$  и оценки  $\hat{\mathbf{W}}(A)$  матрицы весов признаков (4). Для этого используется следующая процедура последовательного добавления и удаления признаков.

Зададим начальное множество признаков  $A_0 = \emptyset$ , параметр останова процедуры  $d$  и начальные значения функции эмпирического риска  $\hat{Q} = Q(\emptyset)$  и номера итерации  $t = 0$ .

*Шаг 1.* Пока мощность набора признаков  $|A_t|$  меньше общего числа признаков  $p$  выполняем следующие действия:

а) увеличиваем номер итерации  $t = t + 1$ ;  
 б) находим оптимальный для добавления признак с индексом  $\hat{g} = \operatorname{argmin}_{g \in G \setminus A_{t-1}} Q(A_{t-1} \cup \{g\})$ , и добавляем его к набору:  $A_t = A_{t-1} \cup \{\hat{g}\}$ ;

в) если  $Q(A_t) < \hat{Q}$ , то текущее минимальное значение эмпирического риска  $\hat{Q} = Q$ , номер оптимальной итерации  $\hat{t} = t$ ;

г) если значение функционала не улучшалось на протяжении  $d$  шагов  $t - \hat{t} \geq d$ , то прервать цикл.

*Шаг 2.* Пока мощность набора признаков  $|A_t|$  ненулевая выполняем следующие действия:

а) увеличиваем номер итерации  $t = t + 1$ ;  
 б) находим оптимальный для удаления признак  $\hat{g} = \operatorname{argmin}_{g \in A_{t-1}} Q(A_{t-1} \setminus \{g\})$  и удаляем его из набора:  $A_t = A_{t-1} \setminus \{\hat{g}\}$ ;

в) если  $Q(A_t) < \hat{Q}$ , то  $\hat{Q} = Q$ ,  $\hat{t} = t$ ;  
 г) если значение функционала не улучшалось на протяжении  $d$  шагов  $t - \hat{t} \geq d$ , то прервать цикл.

*Шаг 3.* Повторять шаги 1 и 2 пока значения  $Q(A_t)$  убывают.

Алгоритм выбора признаков определяет их оптимальный набор  $\hat{A} = A_{\hat{t}}$  с одновременным оцениванием матрицы весов  $\hat{\mathbf{W}}(\hat{A})$ .

## 3. Определение типа библиографической записи

Матрица  $\hat{\mathbf{Y}}_i$  (3) содержит полную информацию о типах полей  $l$ , содержащихся в  $i$ -й библиографической записи. Для решения подзадачи об определении типа записи BibTeX составляется матрица  $\mathbf{B}$  размера  $m \times s$  по правилу

$$B(i, l) = \begin{cases} 1, & \text{если } l\text{-я строка матрицы } \hat{\mathbf{Y}}_i \text{ ненулевая,} \\ 0, & \text{иначе.} \end{cases} \quad (5)$$

Таким образом, элемент  $B(i, l)$  матрицы  $\mathbf{B}$  определяет присутствие в  $i$ -й библиографической записи  $l$ -го типа поля BibTeX, а строка  $\mathbf{b}_i$  является новым признаковым описанием объекта — библиографической записи. Поставим задачу разбиения объектов на фиксированное число  $r$  кластеров — типов записи в структуре BibTeX.

Обозначим  $k(i)$  — номер кластера, к которому отнесена  $i$ -я библиографическая запись, и введем следующие функции качества:

- минимизацию среднего внутрикластерного расстояния

$$F_0 = \frac{\sum_{i < j} [k(i) = k(j)] \rho(\mathbf{b}_i, \mathbf{b}_j)}{\sum_{i < j} [k(i) \neq k(j)]} \rightarrow \min,$$

- максимизацию среднего межкластерного расстояния

$$F_1 = \frac{\sum_{i < j} [k(i) \neq k(j)] \rho(\mathbf{b}_i, \mathbf{b}_j)}{\sum_{i < j} [k(i) \neq k(j)]} \rightarrow \max,$$

где индикаторная функция

$$[k(i) = k(j)] = \begin{cases} 1, & \text{если } k(i) = k(j), \\ 0, & \text{иначе.} \end{cases}$$

Задача кластеризации сводится к минимизации функции качества  $F$ :

$$F = \frac{F_0}{F_1} \rightarrow \min.$$

В качестве метрики  $\rho(\mathbf{b}_i, \mathbf{b}_j)$  используется диагонально взвешенная Евклидова метрика:

$$\rho(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Lambda^2 (\mathbf{x} - \mathbf{y})}, \text{ где } \Lambda = \operatorname{diag}(\lambda). \quad (6)$$

Диагональная матрица  $\Lambda$  задает веса, соответствующие признакам описания библиографической записи. Ее значения определяются частотой вхождения поля в библиографические записи.

Кластеризация выполняется с помощью метода  $k$ -means. Метод состоит из двух основных шагов:

*Шаг 1.* Для каждого элемента  $\mathbf{b}_i$  находится ближайший к нему центр кластеров, к которому и отнесется данный элемент:

$$k(i) = \operatorname{argmin}_{j=1, \dots, r} \rho(\mathbf{b}_i, \boldsymbol{\mu}_j).$$

**Шаг 2.** Положение центра кластера  $\mu_j$  есть центр масс объектов, принадлежащих кластеру:

$$\mu_j = \frac{\sum_{i=1}^{|D_j|} [k(i) = j] \mathbf{b}_i}{\sum_{i=1}^{|D_j|} [k(i) = j]},$$

где  $|D_j|$  — мощность  $j$ -го кластера,  $j \in \{1, \dots, r\}$ ,

где  $r$  — заданное число кластеров.

Шаги алгоритма повторяются, пока кластеризация объектов меняется.

Начальное приближение центров кластеров  $\mu_j$ ,  $j = 1, \dots, r$ , считается заданным. Результатом работы алгоритма является вектор-столбец  $\mathbf{k}$ , элементы которого  $k(i) = j$  отвечают за принадлежность  $i$ -й библиографической записи к  $j$ -му типу записи.

#### 4. Вычислительный эксперимент

Алгоритм выбора признаков протестирован на выборке из 100 библиографических записей. Задано число используемых типов полей структуры BibTeX для данной выборки,  $s = 11$ : автор; название работы; название источника; номера страниц; номер выпуска; номер тома; год; город; издательство; редакторы; ссылка на работу в Интернете. Максимальное число текстовых сегментов  $m = 9$ . Для каждого сегмента генерировался столбец из  $p = 18$  признаков: длина сегмента; порядковый номер; число различных знаков препинания (точки, запятые, тире, кавычки, скобки, двоеточия, точки с запятой); общее число сегментов объекта; число заглавных букв; число цифр; число слов; наличие инициалов; наличие подряд идущих цифр (числа); наличие подряд идущих заглавных букв (аббревиатуры); общая длина всех сегментов в записи; общее число слов в записи.

Таким образом задана матрица  $\mathbf{X}$  размера  $m \times n \times p$ , где  $m = 100$ ,  $n = 9$ ,  $p = 18$ . Матрица ответов  $\mathbf{Y}$  задана в виде (1). Множество объектов  $\{\mathbf{X}_i, \mathbf{Y}_i\}$  разбито на обучающую и контрольную выборки.

На рис. 2, *a* (см. четвертую сторону обложки) показан набор признаков на каждой итерации алгоритма выбора признаков. Видно, что оптимальный набор был найден за небольшое число итераций ( $t = 10 \dots 15$ ). Показано, что в данном случае некоторые признаки оказались неинформативны и были удалены сразу же после их добавления в набор. На рис. 2, *b* (см. четвертую сторону обложки) показано среднее число ошибок на обучающей выборке (сплошная линия) и контрольной выборке (штриховая линия).

На рис. 3 (см. четвертую сторону обложки) показано количество совпадений экспертной сегментации с сегментацией, проведенной предложенным алгоритмом: красный цвет означает нахождение в

данном квадрате максимального числа сегментов, синий — минимального. Нахождение сегмента на диагонали означает совпадение экспертной и полученной сегментации. Видно, что первые по порядку поля имеют хорошее качество сегментации. Последующие поля имеют заметно худшее качество сегментации. Разница возникает в силу того, что первые поля (автор, название, год и др.) чаще присутствуют в библиографических записях и выборка достаточно велика, чтобы получить адекватную оценку вектора параметров логистической регрессии. Последние же поля присутствовали лишь в небольшом количестве из выборки 100 записей (некоторые — меньше, чем в 10).

Библиографические записи были разделены на шесть заданных кластеров — типов библиографических записей: статья, книга, тезис конференции, диссертация, электронный источник, либо ни один из указанных типов. Параметры  $\lambda_i$  метрики  $\rho$  заданы в зависимости от частоты вхождения  $i$ -го поля в библиографические записи. Чем чаще поле присутствует в библиографической записи, тем менее оно важно:

$$\lambda_i^2 = \left( 1 - \frac{\sum_{i=1}^n B(i, j)}{n} \right)^2, \quad 0 \leq \lambda_i < 1, \quad i = 1, \dots, s.$$

Например, поля "название" или "автор" присутствуют почти во всех типах записей, а "число страниц", "название журнала" — в заметно меньшем числе.

Начальное положение центра кластера  $\mu_j$  задано теми векторами  $\mathbf{b}_i = \mu_j$ , которые соответствуют записям, содержащим редко встречающиеся типы полей и при этом находящимся на удалении друг от друга больше заданного значения  $\rho_{\min}$ .

Диаграмма на рис. 4, *a* (см. четвертую сторону обложки) отражает число записей, относящихся к каждому из типов BibTeX по мнению эксперта. На диаграмме рис. 4, *b* (см. четвертую сторону обложки) представлен результат кластеризации. Типы записей, представленные малым числом примеров, не определились: как видно из рис. 4, *b*, столбцы 5 и 6 имеют нулевые значения. На тех типах записей, которые были представлены в выборке большим числом примеров, предложенный метод показал результат, слабо отличающийся от экспертной кластеризации.

#### Заключение

В работе решена задача разметки библиографических записей. Предлагается новая постановка задачи структурного обучения для прогнозирования структуры библиографической записи. Поставленная задача прогнозирования требует использования алгоритма выбора оптимального набора признаков.

Использован алгоритм последовательного выбора признаков, исследованы его свойства. Алгоритм проиллюстрирован выборкой из неформатированных библиографических записей, для которых каждому сегменту ставилось в соответствие поле в структуре BibTeX. Для каждой записи определялся ее тип в данной структуре. Представлены результаты работы предложенного метода сегментирования и кластеризации библиографических записей.

Работа выполнена при поддержке РФФИ, грант № 13-07-00709.

#### Список литературы

1. **Библиографические** записи в формате BibTeX. URL: <http://www.bibtex.org> (дата обращения: 20.12.2012).
2. **Полежаев В.** Задачи и методы автоматического построения графа цитирований по коллекции научных документов // Труды МФТИ. 2012. Т. 4. С. 1–12.
3. **Borkar V., Deshmukh K., Saravagi S.** Automatic segmentation of text into structured records // Proc. of the 2001 ACM SIGMOD international conference on management of data. 2001. V. 30, N. 2. P. 175–186.
4. **Cititation Parser.** URL: <http://freecite.library.brown.edu/> (дата обращения: 20.12.2012).

5. **Lampert C. H.** Maximum Margin Multi-Label Structured Prediction // Advances in Neural Information Processing Systems. 2011. V. 24. P. 289–297.
6. **Jaakkola T., Sontag D.** Learning Bayesian Network Structure using LP Relaxations // Proc. of the 13th International Conference on Artificial Intelligence and Statistics. 2010. V. 9, N. 1. P. 358–365.
7. **Kwok T. Y., Yeung D. Y.** Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems // IEEE Transactions on Neural Networks. 1997. V. 8. P. 630–645.
8. **Martins A. F. T., Dipanjan D.** Stacking Dependency Parsers // Conference on Empirical Methods in Natural Language Processing. 2008. P. 157–166.
9. **Jaakkola T.** Scaled structured prediction. URL: [http://video.yandex.ru/users/ya-events/view/486/user-tag/научный %20семинар/](http://video.yandex.ru/users/ya-events/view/486/user-tag/научный_%20семинар/) (дата обращения: 20.12.2012).
10. **Strijov V. V., Krymova E. A., Weber G. W.** Evidence optimization for consequently generated models // Mathematical and Computer Modelling. 2013. V. 57 (1–2). P. 50–56.
11. **Стрижов В. В., Крымова Е. А.** Выбор моделей в линейном регрессионном анализе // Информационные технологии. 2011. № 10. С. 21–26.
12. **Bishop C. M.** Pattern Recognition and Machine Learning. LLC: Springer Science, 2006. 638 p.
13. **Адуенко А. А., Кузьмин А. А., Стрижов В. В.** Выбор признаков и оптимизация метрики при кластеризации коллекции документов // Известия Тульского государственного университета. Естественные науки. 2012. Вып. 4. С. 119–131.

A. A. Varfolomeeva, Student, V. V. Strijov, Researcher, Computer Centre PAS, e-mail: [strijov@gmail.com](mailto:strijov@gmail.com)

## An Algorithm for Bibliographic Records Parsing Using Structure Learning Methods

*The paper solves the application problem of structured texts segmentation, namely each segment of a bibliographic record must correspond to its filed type of the BibTeX format and each record must correspond to its bibliographic type.*

*This problem arises due to the existence of different standards for bibliographic records: an algorithm for determining the types of fields of bibliographic records, which is independent of the specific standards of their composition, should be proposed.*

*To solve the problem of determining the field type the method of constructing matrix "objects" and matrices "answers" is proposed. The authors offer an algorithm of a bibliography lists parsing using the structure regression method, and the optimization problem of regression model's parameters is also solved. According to the results of fields' segmentation bibliographic types of the records are clustered. The quality of the constructed model is investigated using a collection of non-parsed bibliography lists. In the paper it is shown the proposed algorithm has good quality of segmentation and clustering, if it has sufficient training sample.*

**Keywords:** text parsing, structure learning, structure regression, segmentation, features selection, clustering

#### References

1. **Библиографические** записи в формате BibTeX. URL: <http://www.bibtex.org> (reference date: 20.12.2012).
2. **Polezhaev V.** Automated citation graph building from a corpora of scientific documents. *Trudy MFTI*. 2012. V. 4. P. 1–12.
3. **Borkar V., Deshmukh K., Saravagi S.** Automatic segmentation of text into structured records. *Proc. of the 2001 ACM SIGMOD international conference on management of data*. 2001. V. 30, N. 2. P. 175–186.
4. **Cititation Parser.** URL: <http://freecite.library.brown.edu/> (reference date: 20.12.2012).
5. **Lampert C. H.** Maximum Margin Multi-Label Structured Prediction. *Advances in Neural Information Processing Systems*. 2011. V. 24. P. 289–297.
6. **Jaakkola T., Sontag D.** Learning Bayesian Network Structure using LP Relaxations. *Proc. of the 13th International Conference on Artificial Intelligence and Statistics*. 2010. V. 9, N. 1. P. 358–365.

7. **Kwok T. Y., Yeung D. Y.** Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems. *IEEE Transactions on Neural Networks*. 1997. V. 8. P. 630–645.
8. **Martins A. F. T., Dipanjan D.** Stacking Dependency Parsers. *Proc. of Conference on Empirical Methods in Natural Language Processing*. 2008. P. 157–166.
9. **Jaakkola T.** Scaled structured prediction. URL: [http://video.yandex.ru/users/ya-events/view/486/user-tag/научный %20семинар/](http://video.yandex.ru/users/ya-events/view/486/user-tag/научный_%20семинар/) (reference date: 20.12.2012).
10. **Strijov V. V., Krymova E. A., Weber G. W.** Evidence optimization for consequently generated models. *Mathematical and Computer Modelling*. 2013. V. 57 (1–2). P. 50–56.
11. **Strizhov V. V., Krymova E. A.** Model Selection in Linear Regression Analysis. *Informacionnye tehnologii*. 2011. N. 10. P. 21–26.
12. **Bishop C. M.** Pattern Recognition and Machine Learning. LLC: Springer Science, 2006. 638 p.
13. **Aduenko A. A., Kuzmin A. A., Strizhov V. V.** Feature selection and metrics' optimisation when clustering documents collection. *Izvestija Tul'skogo gosudarstvennogo universiteta. Estestvennye nauki*. 2012. N. 4. P. 119–131.

**А. К. Скуратов**, д-р техн. наук, проф., ст. науч. сотр.,

Федеральное государственное бюджетное учреждение "Дирекция научно-технических программ",  
**Д. Е. Кошкин**, ассистент, Московский государственный технический университет радиотехники,  
 электроники и автоматики, e-mail: munin89@mail.ru

## Сравнение 12 алгоритмов кластеризации данных применительно к задаче кластеризации текстов

*Рассматриваются 12 алгоритмов кластеризации в целях анализа их особенностей, достоинств и недостатков применительно к задаче кластеризации текстов. Для справки приводятся метрики схожести, наиболее часто используемые в алгоритмах кластеризации. В конце приведена таблица со сравнением алгоритмов по параметру вычислительной сложности, формы итоговых кластеров, достоинствах и недостатках.*

**Ключевые слова:** кластеризация данных, алгоритм кластеризации метрики схожести

### Введение

Задача кластерного анализа состоит в том, чтобы разделить множество объектов на некоторое число подмножеств (кластеров) так, чтобы каждый объект принадлежал одному и только одному подмножеству и чтобы объекты, принадлежащие одному и тому же кластеру, были сходными, в то время как объекты, принадлежащие разным кластерам, были разнородными (несходными). Решением этой задачи является разбиение, удовлетворяющее некоторому критерию оптимальности. Цели кластеризации могут быть различными в зависимости от особенностей решаемой задачи:

- понять структуру множества объектов, разбив его на группы схожих объектов. В этом случае стараются уменьшить число кластеров, упростить дальнейшую обработку данных и принятия решений, работать с каждым кластером по отдельности;
- сократить объем хранимых данных в случае сверхбольшой выборки, оставив по одному наиболее типичному представителю от каждого кластера. В этом случае не так важно число кластеров, как высокая степень сходства объектов внутри каждого кластера;
- выделить нетипичные объекты, которые не подходят ни к одному из кластеров. При этом подходе интерес представляют именно нетипичные объекты, выбивающиеся из общего строя.

В настоящей работе проводится сравнительный анализ 12 современных алгоритмов кластеризации данных применительно к задаче кластеризации текстового контента и сравнение на основе вычислительной сложности и характеристик алгоритмов. Рассматриваются такие алгоритмы, как *k*-средних, алгоритмы, образующие "Метод опорных векторов" (SVM), EM-алгоритм, алгоритм клики (CLIQUE), алгоритм кластеризации с наклоном (CLOPE), параллельный алгоритм слияния конечных адаптивных интервалов (pMAFIA), алгоритм нечетких *s*-сред-

них (fuzzy *s*-means), алгоритм минимального покрывающего дерева (MST), алгоритм ROCK, алгоритм CURE, алгоритм WaveCluster, алгоритм DBSCAN. Из русскоязычных работ, связанных с анализом и сравнением алгоритмов кластеризации, можно отметить материалы [1, 2]. Иностранные авторы достаточно часто публикуют сравнения алгоритмов и по запросу "clustering algorithms comparison" в поисковике Google выдается 138 млн результатов, таким образом, отметить какую-либо отдельную работу не представляется возможным.

### Алгоритм *k*-средних (*k*-means)

Метод *k*-средних — наиболее популярный метод кластеризации. Был изобретен в 1950-х годах математиком Гуго Штейнгаузом [3] и почти одновременно Стюартом Ллойдом, который опубликовал свою работу только 25 лет спустя в [4]. Особую популярность метод приобрел после работы Маккуина [5].

Алгоритм *k*-means является примером четкой кластеризации, в основе которого лежит итеративный процесс стабилизации центроидов кластеров. Основной характеристикой кластера является его центроид, и вся работа алгоритма направлена на стабилизацию или, в лучшем случае, на полное прекращение изменения центроида кластера. Алгоритм состоит из следующих шагов.

**Шаг 1.** Выбираются начальные центроиды для множества документов. (Из множества документов случайным образом выбираются *k* точек, где *k* равно требуемому числу кластеров, или через эмпирические оценки числа кластеров и их центроидов.)

**Шаг 2.** Все документы множества распределяются среди кластеров. Документ попадает только в тот кластер, к которому он наиболее близко расположен, в зависимости от функции, через которую считается расстояние между объектами.



**Шаг 3.** Пересчитываются центроиды кластеров исходя из нового множества документов в каждом кластере. Зачастую координаты центроидов рассчитываются как простое математическое среднее по каждой координате пространства. Если центроид кластера переместился, то цикл вычислений повторяется с шага 2. Иначе, если центроид стабилизировался в некоторой окрестности или полностью, процесс кластеризации завершается.

В качестве критерия останова работы алгоритма выбирают минимальное изменение среднеквадратической ошибки. Также возможно останавливать работу алгоритма, если на шаге 2 не было объектов, переместившихся из кластера в кластер.

К недостаткам данного алгоритма можно отнести необходимость априорно задавать число кластеров для разбиения [6].

### Набор алгоритмов "Метод опорных векторов" (SVM)

Оригинальная версия метода опорных векторов предложена Владимиром Вапником (Vladimir N. Vapnik); в настоящем виде его предложили Коринна Кортес (Corinna Cortes) и Вл. Вапник в 1993 г., а опубликовали в 1995 [7].

Метод опорных векторов (англ. SVM, *support vector machine*) — набор схожих алгоритмов вида "обучение с учителем", использующихся для задач классификации и регрессионного анализа. Этот метод принадлежит к семейству линейных классификаторов. Особым свойством метода опорных векторов является непрерывное уменьшение эмпирической ошибки классификации и увеличение зазора. Поэтому данный метод также известен как метод классификатора с максимальным зазором.

Цель метода опорных векторов — найти плоскость, разделяющую два множества объектов, заключается в поиске некоторой линейной функции, которая правильно разделяет набор данных на два класса. Функция такова, что для одной группы она должна принимать значения меньше нуля, а для другой группы — больше нуля (рис. 1). Метод отыскивает образцы (они же опорные векторы), находящиеся на границах между двумя классами. Классификация считается хорошей, если область между границами пуста.

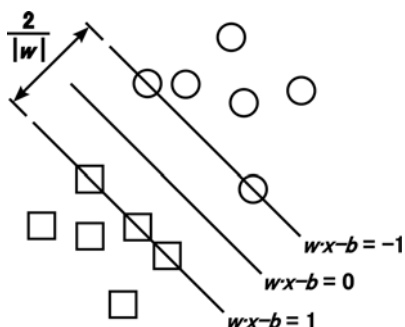


Рис. 1. Разделение групп методом опорных векторов

Основная идея метода опорных векторов — перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Алгоритм работает в предположении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора [8].

### EM-алгоритм

EM-алгоритм был представлен, изложен и получил свое имя в работе Демпстера, Лайрд и Рубин (Dempster, Laird, Rubin) [9]. В работе авторы также отметили, что в разных вариантах для разных случаев алгоритм уже появлялся ранее, но именно в работе 1977 г. он был обобщен для широкого круга решаемых задач.

Одним из широко известных в аналитическом сообществе алгоритмов кластеризации, позволяющих эффективно работать с большими объемами данных, является EM-алгоритм. Его название происходит от слов *expectation—maximization*, что переводится как "ожидание—максимизация". Это связано с тем, что каждая итерация содержит два шага — вычисление математических ожиданий (*expectation*) и максимизацию (*maximisation*). Алгоритм основан на методике итеративного вычисления оценок максимального правдоподобия. В основе идеи EM-алгоритма лежит предположение, что исследуемое множество данных может быть смоделировано с помощью линейной комбинации многомерных нормальных распределений, а целью является оценка параметров распределения, которые максимизируют логарифмическую функцию правдоподобия, используемую в качестве меры качества модели. Иными словами, предполагается, что данные в каждом кластере подчиняются определенному закону распределения, а именно, нормальному распределению. С учетом этого предположения можно определить параметры — математическое ожидание и дисперсию, которые соответствуют закону распределения элементов в кластере, наилучшим образом "подходящему" к наблюдаемым данным.

### Алгоритм Клики (CLIQUE)

Алгоритм CLIQUE был опубликован коллективом авторов из Альмаденского исследовательского центра IBM в 2005 г. [10]. Общая идея алгоритма заключается в том, что пространство объектов разбивается на конечное число ячеек, образующих сетевую структуру, в рамках которой выполняются все операции кластеризации. Алгоритм CLIQUE, адаптированный под кластеризацию данных высокой размерности, является одним из классических сетевых алгоритмов. Алгоритм основан на том предположении, что если в многомерном пространстве данных распределение объектов не равномерно —

встречаются регионы плотности и разрежения, то проекция региона плотности в подпространство с меньшей размерностью будет частью региона плотности в этом подпространстве. Главное достоинство в малом времени выполнения, которое обычно не зависит от числа объектов данных, а зависит только от числа ячеек в каждом измерении пространства.

### Алгоритм кластеризации с наклоном (CLOPE)

CLOPE предложен в 2002 г. группой китайских ученых [11]. Назначение алгоритма CLOPE (*Clustering with sLOPE*) — кластеризация огромных наборов категориальных данных. К достоинствам данного алгоритма можно отнести высокие масштабируемость и скорость работы, а также качество кластеризации, что достигается использованием глобального критерия оптимизации на основе максимизации градиента высоты гистограммы кластера. Во время работы алгоритм хранит в оперативной памяти небольшое количество информации по каждому кластеру и требует минимальное число сканирований набора данных. CLOPE автоматически подбирает число кластеров, причем это регулируется одним единственным параметром — коэффициентом отталкивания. Недостатком алгоритма является то, что использование гистограмм может привести к неоптимальным кластерам для некоторых наборов данных [12].

В основе алгоритма кластеризации CLOPE лежит идея максимизации глобальной функции стоимости, которая повышает близость транзакций в кластерах с помощью увеличения параметра кластерной гистограммы. Для примера рассмотрим пять транзакций:  $\{(a, b), (a, b, c), (a, c, d), (d, e), (d, e, f)\}$ .

Далее сравниваются два разных возможных разбиения:

- (1)  $\{\{ab, abc, acd\}, \{de, def\}\}$
- (2)  $\{\{ab, abc\}, \{acd, de, def\}\}$

Для первого и второго вариантов разбиения в каждом кластере рассчитывается число вхождений в него каждого элемента транзакции, а затем вычисляется средняя высота ( $H$ ) и ширина ( $W$ ) кластера. Например, кластер  $\{ab, abc, acd\}$  имеет вхождения  $a:3, b:2, c:2, d:1$  с  $H = 2$  и  $W = 4$ . На рис. 2 эти результаты показаны геометрически в виде гистограмм.

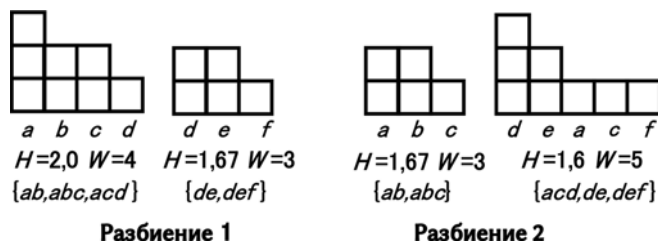


Рис. 2. Гистограммы двух разбиений

Качество двух разбиений оценивается через их высоту  $H$  и ширину  $W$ . Кластеры  $\{de, def\}$  и  $\{ab, abc\}$  имеют одинаковые гистограммы, следовательно, они равноценны. Гистограмма для кластера  $\{ab, abc, acd\}$  содержит четыре различных элемента и имеет площадь восьми блоков ( $H = 2,0, H/W = 0,5$ ), а кластер  $\{acd, de, def\}$  — пять различных элементов с такой же площадью ( $H = 1,6, H/W = 0,32$ ). Очевидно, что разбиение (1) лучше, поскольку обеспечивает большее наложение транзакций друг на друга (соответственно, параметр  $H$  там выше) [13].

### Параллельный алгоритм слияния конечных адаптивных интервалов (pMAFIA)

Алгоритм pMAFIA разработан Харшей Нагешем (Harsha S. Nagesh), Санждем Гойлом (Sanja Goil) и Алоком Шудхари (Alok Choudhary). Алгоритм является продолжением алгоритма MAFIA, добавив к нему параллельность выполнения расчетов. Оригинальный алгоритм опубликован в 1999 г. [14], параллельная версия в июне 2000 г. [15].

Алгоритм pMAFIA (от *Merging of Adaptive Finite Intervals*, объединение адаптивных конечных интервалов) — масштабируемый параллельный алгоритм кластеризации с использованием адаптивного подпространственного вычисления конечных интервалов в каждом измерении, которые объединяются для изучения кластеров в более высоких измерениях. Алгоритм в подпространствах строит сеть и считает, что кластер — это зона большей плотности объектов в ячейках сети, нежели в окружении этих ячеек.

Качество результатов и вычислительные требования в сильной степени зависят от числа элементов в каждом измерении и от размера ячеек сети. Большие размеры ячеек приводят к плохому качеству кластеризации, в то время как мелкие ячейки будут требовать огромный объем вычислений.

### Алгоритм нечетких с-средних (fuzzy c-means)

Первая версия алгоритма была опубликована в 1973 г. Данном (J. C. Dunn). Далее, в 1981 г., он был модифицирован математиком Джеймсом Бездеком (James C. Bezdek) [16, 17]. Fuzzy Classifier Means (Fuzzy c-Means, FCM, Нечеткий с-средних) — алгоритм нечеткой кластеризации. Целью алгоритма является кластеризация множества объектов, которые задаются векторами признаков в пространстве. Алгоритм определяет кластеры, ориентируясь на число кластеров, которое ему было задано на старте, и соответственно классифицирует объекты. Кластеры и границы между кластерами представляются нечеткими множествами. Предполагается, что объекты принадлежат всем кластерам с определенной степенью принадлежности. Степень принадлежности определяется расстоянием от объекта до соответствующих кластерных центров. Данный

алгоритм итерационно вычисляет центры кластеров и новые степени принадлежности объектов. К основным недостаткам можно отнести: вычислительную сложность; необходимость задавать вручную число кластеров; неопределенность с объектами, которые удалены от центров всех кластеров. Достоинства: нечеткость при определении объекта в кластер позволяет определять в кластеры объекты, которые находятся на границе.

### Алгоритм минимального покрывающего дерева (MST)

Одной из наиболее ранних публикаций о применении графового алгоритма минимального покрывающего (остовного) дерева можно назвать работу Говера и Росса (J. C. Gower, G. J. S. Ross) [18]. Алгоритм предполагает построение на графе минимального покрывающего дерева с последовательным удалением ребер с наибольшим весом.

На рис. 3 изображено минимальное покрывающее дерево, полученное для девяти объектов.

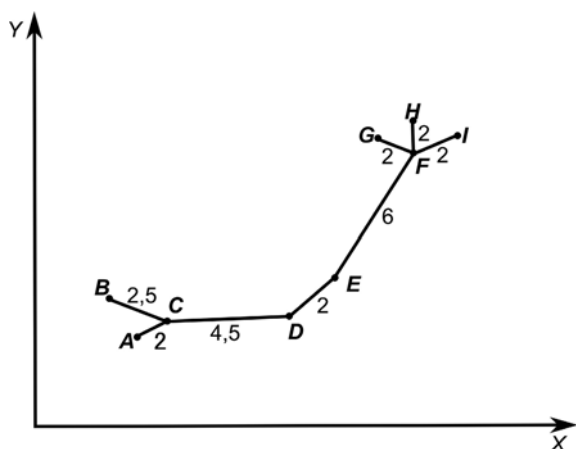


Рис. 3. Пример минимального покрывающего дерева

Путем удаления связи с максимальным расстоянием, равным 6 единицам, помеченной  $EF$ , получаем два кластера:  $\{A, B, C, D, E\}$  и  $\{F, G, H, I\}$ . Первый кластер в дальнейшем может быть разделен еще на два кластера путем удаления ребра  $CD$ , которое имеет длину, равную 4,5 единицам.

Недостатком алгоритма можно считать сложность в определении максимально допустимой длины связи между двумя объектами [19].

### Алгоритм ROCK

В 2000 г. была опубликована работа [20], где описывался алгоритм ROCK (Robust Clustering using Links). Алгоритм — это иерархический агломеративный алгоритм, основанный на  $k$ -means и алгоритме ближайшего соседа, сочетающий их достоинства, но лишенный их недостатков. (При этом  $k$ -means продолжает использоваться как быстрый не иерархический алгоритм.) Основная идея алго-

ритма заключается во введении в каждую пару объектов нового параметра — число общих соседей (ссылок). Этот параметр получается из матрицы соседства путем выяснения в цикле по каждому объекту, является ли он соседом двух рассматриваемых. Благодаря учету ссылок алгоритм становится нечувствительным к выбросам и не требует четкого разделения объектов на кластеры. Также, работая в цикле кластеризации со ссылками, объединяя кластеры, легко получить новые значения параметров ссылок с другими кластерами; для этого достаточно сложить соответствующие значения параметра ссылок двух объединяемых кластеров. Алгоритм был разработан для кластеризации объектов данных с большим количеством числовых и номинальных атрибутов. Типичными задачами кластеризации алгоритмом ROCK является прогнозирование свойств сложных объектов по неполным наборам атрибутов у некоторых из них. При всех своих достоинствах и решении проблем предшественников алгоритм ROCK имеет один серьезный недостаток: увеличение вычислительной сложности. Процесс подсчета ссылок является самым долгим во всей кластеризации.

### Алгоритм CURE

Алгоритм CURE — второй алгоритм кластеризации, от авторов алгоритма ROCK — Судипто, Раджива и Кьюсека (Sudipto; Rajeev; Kyuseok). Описание было опубликовано в работе [21]. Алгоритм CURE (Clustering Using REpresentatives) выполняет иерархическую кластеризацию с использованием набора определяющих точек для определения объекта в кластер, т. е. самые близкие точки в пространстве заменяются представителями. Применяется для очень больших наборов числовых данных (несколько сот тысяч записей базы данных), но эффективен лишь для данных в пространстве малой размерности. Достоинства: выполняет кластеризацию на высоком уровне даже при наличии выбросов, выделяет кластеры сложной формы и различных размеров, обладает линейно зависящими требованиями к месту хранения данных и временную сложность для данных высокой размерности. Недостатки: есть необходимость в задании пороговых значений и числа кластеров. Еще одно описание алгоритма приведено в работе [22].

### Алгоритм WaveCluster

Алгоритм был представлен в 1998 г. [23] Шейхо-леслами, Шаттержи и Жангом (Sheikholeslami, Chatterjee and Zhang), далее, в 2011 г., Ахмет Йилдрим и Цем Оздоган (Ahmet Artu Yildirim, Cem Ozdogan) представили его модифицированную, параллельную версию [24]. WaveCluster представляет собой алгоритм кластеризации на основе волновых преобразований. В начале работы алгоритма данные обобщаются

путем наложения на пространство данных многомерной решетки. На дальнейших шагах алгоритма анализируются не отдельные точки, а обобщенные характеристики точек, попавших в одну ячейку решетки. В результате такого обобщения необходимая информация уместается в оперативной памяти. На последующих шагах для определения кластеров алгоритм применяет волновое преобразование к обобщенным данным. Достоинствами WaveCluster

является то, что алгоритм может обнаруживать кластеры произвольных форм и не чувствителен к шумам. В качестве недостатков можно указать сложность реализации, а также то, что алгоритм применим только к данным низкой размерности.

### Алгоритм DBSCAN

Алгоритм DBSCAN был предложен Мартином Эстером, Гансом-Питером Кригель и коллегами

Сравнение 12 алгоритмов кластеризации

| Алгоритм кластеризации               | Вычислительная сложность  | Форма кластеров              | Достоинства  | Ограничения   |
|--------------------------------------|---|------------------------------|--|---|
| k-средних                            | $O(nkl)$ , где $k$ — число кластеров, $l$ — число итераций        | Гиперсфера                   | Скорость работы  | Ограничения по объему данных, чувствительность к выбросам, необходимость задавать число кластеров   |
| Метод опорных векторов (SVM)         | лучшая — $O(n^2)$ , худшая — $O(n^3)$                             | Произвольная                 | Работает при любом числе измерений, нет проблем с локальными минимумами (одинаковые решения на одних наборах данных) | Неустойчив к шумам, для каждой задачи требуется индивидуальное решение, данные должны быть линейно разделимы  |
| EM-алгоритм                          | $O(nkl)$ , где $k$ — число кластеров, $l$ — число итераций        | Эллиптическая, настраиваемая | Устойчив к шумам, быстрая сходимость (при удачной инициализации)   | Не всегда выполняется предположение о нормальности всех измерений, может дать квазиоптимальное решение  |
| CLIQUE                               | $O(2^n)$  | Произвольная                 | Поддержка данных в пространствах большой размерности   | Низкая скорость работы (высокая вычислительная сложность) Может потерпеть неудачу, если кластеры имеют сильно различающиеся плотности                               |
| CLOPE                                | $O(nla)$ , где $k$ — число кластеров, $a$ — средняя длина объекта | Произвольная                 | Скорость работы, масштабируемость  | Для некоторых наборов данных вероятно квазиоптимальное разбиение  |
| pMAFIA                               | Порядка $O(2^k)$ , где $k$ — объект наибольшей размерности        | Произвольная                 | Хорошо справляется с многомерными данными  | Вычислительная сложность при увеличении размерности данных  |
| Нечетких c-средних (Fuzzy c-means)   | $O(nkl)$ , где $k$ — число кластеров, $l$ — число итераций        | Гиперсфера                   | Позволяет определять объекты, не принадлежащие ни к одному кластеру  | Ограничения по объему данных  |
| Минимальное покрывающее дерево (MST) | $O(n^2 \log(n))$  | Произвольная                 | Выбирает наиболее оптимальное решение из ряда возможных  | Ограниченная применимость (хорош на кластерах в форме сгущений или лент, для других может показать неадекватные результаты). Вычислительная сложность               |
| ROCK                                 | Порядка $O(n^2 \log(n))$  | Произвольная                 | Для принятия решения о близости использует не информацию о расстоянии, а представление о самом разбиении             | Использует разреженную матрицу для хранения информации о близости объектов, из-за чего страдает общая скорость. Функция близости зависит от длины вектора документа |
| CURE                                 | $O(n^2 \log(n))$  | Произвольная                 | Не чувствителен к выбросам и шумам   | Требуется задавать пороговые значения для расчетов  |
| WaveCluster                          | $O(K)$ , где $K$ — число ячеек, на которое разделено пространство | Произвольная                 | Высокая скорость работы, определяет кластеры разной формы  | Показывает хорошие результаты на данных не очень большой размерности  |
| DBSCAN                               | $O(n \log(n))$  | Произвольная                 | Не надо задавать число кластеров, не чувствителен к выбросам и шумам   | Результат зависит от выбранной метрики близости, испытывает сложности, если в пространстве сильно различается плотность кластеров                                   |

Примечание. В таблице  $n$  — число обрабатываемых (кластеризуемых) объектов, если не указано иное.

(Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu) в 1996 г. [25] как решение проблемы разбиения (изначально пространственных) данных на кластеры произвольной формы. DBSCAN (Density Based Spatial Clustering of Applications with Noise) — алгоритм кластеризации пространственных данных с присутствием шума. Большинство алгоритмов, проводящих плоское разбиение, создают кластеры, по форме близкие к сферическим, так как минимизируют расстояние документов до центра кластера. Авторы DBSCAN экспериментально показали, что их алгоритм способен распознать кластеры различной формы. Идея, положенная в основу алгоритма, заключается в том, что внутри каждого кластера наблюдается типичная плотность точек (объектов), которая заметно выше, чем плотность снаружи кластера, а также плотность в областях с шумом ниже плотности любого из кластеров. Еще точнее, что для каждой точки кластера ее соседство заданного радиуса должно содержать не менее некоторого числа точек, это число точек задается пороговым значением [26, 27].

В таблице приведено сравнение характеристик перечисленных алгоритмов по критериям вычислительной сложности, формы кластеров, достоинств и ограничений, приписываемых авторами своим алгоритмам, или относящихся ко всем алгоритмам одного класса.

## Заключение

Сравнительный анализ алгоритмов кластеризации показал, что кластеризация является затратным по времени процессом, продолжительность которого часто нелинейно зависит, в том числе и от количества обрабатываемого материала и от точности желаемого результата. Для каждого алгоритма есть ограничения на объем данных, которые он мог бы эффективно обрабатывать. В результате кластеризация в общем случае является инструментом анализа данных, применение которого требует некоторой подготовки исследователя, а интерпретация результатов — опыта.

## Список литературы

1. **Любимов Н., Михеев Е., Лукин А. С.** Сравнение алгоритмов кластеризации в задаче идентификации диктора // Тр. 13-й междунар. конф. "Цифровая обработка сигналов и ее применение" (DSPA'2011). 2011. Т. 1. С. 204—207.
2. **Нейский И. М.** Классификация и сравнение методов кластеризации. [Электронный ресурс]. URL: [http://it-claim.ru/Persons/Neyskiy/Article2\\_Neyskiy.pdf](http://it-claim.ru/Persons/Neyskiy/Article2_Neyskiy.pdf)
3. **Steinhaus H.** Sur la division des corps materiels en parties // Bulletin de l'Académie Polonaise des Sciences. 1956. Classe III. V. IV, N. 12. P. 801—804.
4. **Lloyd S. P.** Least squares quantization in PCM // IEEE Transactions on Information Theory. March 1982. V. IT-28. N. 2. P. 129—137.
5. **MacQueen J.** Some methods for classification and analysis of multivariate observations // Proc. of the 5<sup>th</sup> Berkeley Symp. on Math. Statistics and Probability. 1967. V. 1. P. 281—297.
6. **Андреев А. М., Березкин Д. В., Морозов В. В., Симаков К. В.** Метод кластеризации документов текстовых коллекций и синтеза аннотаций кластеров // Тр. Десятой Всероссийской научной конф. "Электронные библиотеки: перспективные методы и технологии, электронные коллекции". Дубна. 7—11 октября 2008 [Электронный ресурс]. URL: [http://rcdl2008.jinr.ru/pdf/220\\_229\\_paper26.pdf](http://rcdl2008.jinr.ru/pdf/220_229_paper26.pdf)
7. **Cortes C., Vapnik V.** Support-vector networks // Machine Learning. 1995. N 20 (3). P. 273—297.
8. **Википедия.** Свободная энциклопедия [Электронный ресурс]. URL: <http://ru.wikipedia.org>.
9. **Dempster A. P., Laird N. M., Rubin D. B.** Maximum likelihood from incomplete data via the EM algorithm // Journal of the Royal Statistical Society. Series B. 1977. V. 39, N. 1. P. 1—38.
10. **Agrawal R., Gehrke J., Gunopulos D., Raghavan P.** Automatic Subspace Clustering of High Dimensional Data // Data Mining and Knowledge Discovery (Springer Netherlands). 2005. N. 11 (1). P. 5—33.
11. **Yang Y., Guan H., You J.** CLOPE: A fast and Effective Clustering Algorithm for Transactional Data // Proc. of SIGKDD'02, July 23—26, 2002, Edmonton, Alberta, Canada.
12. **Aggarwal C. C., Reddy C. K.** Data Clustering: Algorithms and Applications // CRC Press. 2013. 652 p.
13. **Кластеризация** категорийных данных: масштабируемый алгоритм CLOPE. BaseGroup Labs. [Электронный ресурс]. URL: <http://www.basegroup.ru/library/analysis/clusterization/clope/> (дата обращения 17.01.14).
14. **Nagesh H., Goil S., Choudhary A.** MAFIA: Efficient and Scalable Subspace Clustering for Very Large Data Sets // Technical Report Number CPDC-TR-9906-019, Center for Parallel and Distributed Computing, Northwestern University, 1999. 20 p.
15. **Nagesh H. S., Goil S., Choudhary A.** A scalable parallel subspace clustering algorithm for massive data sets // Proceedings of the 2000 International Conference on Parallel Processing, 2000. P. 447—454.
16. **Dunn J. C.** A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters // Journal of Cybernetics. 1973. N. 3. P. 32—57.
17. **Bezdek J. C.** Pattern Recognition with Fuzzy Objective Function Algorithms // Plenum Press, New York, 1981.
18. **Gower J. C., Ross G. J. S.** Minimum Spanning Trees and Single Linkage Cluster Analysis // Journal of the Royal Statistical Society. C (Applied Statistics), 1969. N. 18 (1). P. 54—64.
19. **Liu Y.** Minimum Spanning Trees, LDCSEE, West Virginia University, Morgantown, WV. [Электронный ресурс]. URL: <http://www.csee.wvu.edu/~ksmani/courses/fa01/random/lecnotes/lecture11.pdf> (дата обращения 17.01.14).
20. **Sudipto G., Rajeev R., Kyuseok S.** ROCK: A Robust Clustering Algorithm for Categorical Attributes // Information Systems — IS. 2000. V. 25, N 5. P. 345—366.
21. **Sudipto G., Rajeev R., Kyuseok S.** CURE: An Efficient Clustering Algorithm for Large Databases // Information Systems — IS. 2001. V. 26, N. 1. P. 35—58.
22. **Дорофеев А. А.** Алгоритмы автоматической классификации: Обзор // Автоматика и телемеханика. 1971. № 12. С. 78—113.
23. **Sheikholeslami G., Chatterjee S., Zhang A.** WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases // Proc. of 24th International Conference Very Large Data Bases, 1998, Morgan Kaufmann, San Francisco. 1998. P. 428—439.
24. **Yildirim A. A., Ozdogan C.** Parallel WaveCluster: A linear scaling parallel clustering algorithm implementation with application to very large datasets // Journal of Parallel and Distributed Computing archive. Academic Press, Inc. Orlando, FL, USA. 2011. V. 71, Is. 7. P. 955—962.
25. **Ester M., Kriegel H. S. J., Xu X.** A density-based algorithm for discovering clusters in large spatial databases with noise // Proc. of the Second International Conference on Knowledge Discovery and Data Mining (KDD—96). AAAI Press. 1996. P. 226—231.
26. **Большакова Е. И., Кльшинский Э. С., Ландэ Д. В., Носков А. А., Пескова О. В., Ягунова Е. В.** Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие. М.: МИЭМ, 2011. 272 с.

## Comparison of 12 Data Clustering Algorithms Applied to the Problem of Texts Clustering

*In this paper made a comparative analysis of 12 data clustering algorithms applied to the problem of text clustering. Comparison based on the computational complexity of algorithms and their features and limitations. Article contents description of such algorithms as *k*-means, "Support Vector Machine" (SVM), Expectation-Minimization (EM-algorithm), CLIQUE, clustering with slope algorithm (CLOPE), parallel merging algorithm for adaptive finite intervals (pMAFIA), fuzzy *c*-means, the minimum spanning tree algorithm (MST), ROCK algorithm, CURE algorithm, WaveCluster algorithm, DBSCAN algorithm. For every algorithm was made little historic introduction, then described logic and idea of algorithm, and then gave features and limitations of algorithm. At the end of the article was made comparison of algorithms and gave table with compute complexity, form of clusters, features and limitations of algorithms.*

**Keywords:** clustering data, clustering algorithm, similarity metrics

### References

1. **Ljubimov N., Miheev E., Lukin A. S.** Svrnenie algoritmov klasterizacii v zadache identifikacii diktora. *Trudy 13 mezhdunarodnoj konferencii "Cifrovaja obrabotka signalov i ejo primenenie" (DSPA'2011)*. 2011. V. 1. P. 204—207.
2. **Nejskij I. M.** Klassifikacija i svrnenie metodov klasterizacii. [Elektronnyj resurs]. URL: [http://it-claim.ru/Persons/Neyskiy/Article2\\_Neiskiy.pdf](http://it-claim.ru/Persons/Neyskiy/Article2_Neiskiy.pdf)
3. **Steinhaus H.** Sur la division des corps materiels en parties. *Bulletin de l'Académie Polonaise des Sciences*. 1956. Classe III. V. IV, N. 12. P. 801—804.
4. **Lloyd S. P.** Least squares quantization in PCM. *IEEE Transactions on Information Theory*. March 1982. V. IT-28, N. 2. P. 129—137.
5. **MacQueen J.** Some methods for classification and analysis of multivariate observations. *Proc. 5<sup>th</sup> Berkeley Symp. on Math. Statistics and Probability*. 1967. V. 1. P. 281—297.
6. **Andreev A. M., Berezkin D. V., Morozov V. V., Simakov K. V.** Metod klasterizacii dokumentov tekstovyh kollekcij i sinteza annotacij klasterov. *Tr. Desjatoj Vserossijskoj nauchnoj konferencii "Elektronnye biblioteki: perspektivnye metody i tehnologii, jelektronnye kollekcii"*. Dubna. 7—11 oktjabrja 2008. URL: [http://rcdl2008.jinr.ru/pdf/220\\_229\\_paper26.pdf](http://rcdl2008.jinr.ru/pdf/220_229_paper26.pdf)
7. **Cortes C., Vapnik V.** Support-vector networks. *Machine Learning*. 1995. N. 20 (3). P. 273—297.
8. **Vikipedija.** Svobodnaja jenciklopedija [Elektronnyj resurs]. URL: <http://ru.wikipedia.org>
9. **Dempster A. P., Laird N. M., Rubin D. B.** Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*. 1977. V. 39, N. 1. P. 1—38.
10. **Agrawal R., Gehrke J., Gunopulos D., Raghavan P.** Automatic Subspace Clustering of High Dimensional Data. *Data Mining and Knowledge Discovery (Springer Netherlands)*. 2005. N. 11 (1). P. 5—33.
11. **Yang Y., Guan H., You J.** CLOPE: A fast and Effective Clustering Algorithm for Transactional Data. *Proc. of SIGKDD'02, July 23—26, 2002*, Edmonton, Alberta, Canada.
12. **Aggarwal C. C., Reddy C. K.** Data Clustering: Algorithms and Applications. *CRC Press*. 2013. 652 p.
13. **Klasterizacija kategorijnyh dannyh: masshtabiruemyj algoritm CLOPE.** BaseGroup Labs. URL: <http://www.basegroup.ru/library/analysis/clusterization/clope/> (data obrashhenija 17.01.14).
14. **Nagesh H., Goil S., Choudhary A.** MAFIA: Efficient and Scalable Subspace Clustering for Very Large Data Sets. *Technical Report Number CPDC-TR-9906-019*, Center for Parallel and Distributed Computing, Northwestern University, 1999. 20 p.
15. **Nagesh H. S., Goil S., Choudhary A.** A scalable parallel subspace clustering algorithm for massive data sets. *Proc. of the 2000 International Conference on Parallel Processing*. 2000. P. 447—454.
16. **Dunn J. C.** A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*. 1973. N. 3. P. 32—57.
17. **Bezdek J. C.** Pattern Recognition with Fuzzy Objective Function Algorithms. *Plenum Press*. New York. 1981.
18. **Gower J. C., Ross G. J. S.** Minimum Spanning Trees and Single Linkage Cluster Analysis. *Journal of the Royal Statistical Society. C (Applied Statistics)*. 1969. N. 18 (1). P. 54—64.
19. **Liu Y.** Minimum Spanning Trees, LDCSEE, West Virginia University, Morgantown, WV. [Elektronnyj resurs]. URL: <http://www.csee.wvu.edu/~ksmani/courses/fa01/random/lecnotes/lecture11.pdf> (data obrashhenija 17.01.14).
20. **Sudipto G., Rajeev R., Kyuseok S.** ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Information Systems — IS*. 2000. V. 25, N. 5. P. 345—366.
21. **Sudipto G., Rajeev R., Kyuseok S.** CURE: An Efficient Clustering Algorithm for Large Databases. *Information Systems — IS*. 2001. V. 26, N. 1. P. 35—58.
22. **Dorofejuk A. A.** Algoritmy avtomaticheskoy klassifikacii: Obzor. *Avtomatika i telemekhanika*. 1971. N. 12. P. 78—113.
23. **Sheikholeslami G., Chatterjee S., Zhang A.** WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases. *Proceedings of 24th International Conference Very Large Data Bases*. 1998. Morgan Kaufmann, San Francisco. 1998. P. 428—439.
24. **Yildirim A. A., Ozdogan C.** Parallel WaveCluster: A linear scaling parallel clustering algorithm implementation with application to very large datasets. *Journal of Parallel and Distributed Computing archive. Academic Press, Inc. Orlando, FL, USA*. 2011. V. 71, Is. 7. P. 955—962.
25. **Ester M., Kriegel H. S. J., Xu X.** A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press. 1996. P. 226—231.
26. **Bol'shakova E. I., Klyshinskij Je. S., Landje D. V., Noskov A. A., Peskova O. V., Jagunova E. V.** Avtomaticheskaja obrabotka tekstov na estestvennom jazyke i komp'juternaja lingvistika: ucheb. posobie. M.: MIJeM, 2011. 272 p.

# МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 519.6

А. П. Карпенко, д-р физ.-мат. наук, проф., e-mail: apkarpenko@mail.ru,  
М. К. Сахаров, аспирант, e-mail: max.sfn90@gmail.com, МГТУ им. Н. Э. Баумана

## Мультимемеевая глобальная оптимизация на основе алгоритма эволюции разума

*Предметом работы являются гибридные алгоритмы глобальной оптимизации на основе концепции мемов — так называемые мемеевые метаэвристические алгоритмы поисковой оптимизации. В качестве базового алгоритма для мультимемеевой гибридизации используется алгоритм эволюции разума (Mind Evolutionary Computation, MEC).*

*На основе алгоритма MEC разработан мультимемеевый алгоритм, ориентированный на класс слабосвязанных распределенных вычислительных систем. Исследована эффективность его последовательной реализации на ряде тестовых задач глобальной оптимизации.*

*Дана постановка задачи глобальной безусловной оптимизации, изложена концепция мультимемеевых алгоритмов поисковой оптимизации, представлен алгоритм MEC как базовый алгоритм для гибридизации, предложен гибридный алгоритм HMEC, рассмотрена последовательная программная реализация алгоритма HMEC и представлены результаты исследования его эффективности.*

**Ключевые слова:** глобальная оптимизация, гибридизация, мем, алгоритм эволюции разума

### Введение

В последние десятилетия интенсивно развивается класс стохастических метаэвристических алгоритмов поисковой оптимизации, называемых популяционными [1]. Эти алгоритмы зарекомендовали себя как эффективное средство решения задач глобальной оптимизации. Отличительными чертами таких задач часто являются высокая размерность пространства поиска, сложная топология области допустимых значений, а также нелинейность, недифференцируемость, многоэкстремальность, овражность, отсутствие аналитического выражения и высокая вычислительная сложность оптимизируемых функций.

В настоящее время основные усилия исследователей в области популяционных алгоритмов глобальной оптимизации направлены на повышение их эффективности путем гибридизации и/или метаоптимизации. Гибридизация подразумевает интеграцию двух и более алгоритмов в целях устранения их недостатков и объединения достоинств. Метаоптимизация заключается в настройке или самонастройке значений свободных параметров алгоритма с тем, чтобы обеспечить его максимальную эффективность [2].

Предметом работы являются гибридные алгоритмы глобальной оптимизации на основе концепции мемов — так называемые мемеевые метаэвристические алгоритмы поисковой оптимизации [3, 4].

В контексте работы под *мемом* понимаем какой-либо алгоритм локальной оптимизации, уточняющий текущее решение исходной задачи оптимизации [5]. В широком смысле мемеевые алгоритмы представляют собой гибридизацию какого-либо популяционного алгоритма глобального поиска и одного или нескольких алгоритмов локальной оптимизации.

Концепция мемеевых алгоритмов предоставляет широкие возможности для разработки различных модификаций этих алгоритмов, которые могут различаться частотой выполнения локального поиска, условиями его окончания и т. д. [6, 7]. Практически значимые модификации мемеевых алгоритмов предполагают одновременное использование различных мемов. Такие алгоритмы называют *мультимемеевыми*.

В качестве базового алгоритма для мультимемеевой гибридизации используем алгоритм эволюции разума (*Mind Evolutionary Computation, MEC*). Концепция алгоритма *MEC* предложена в 1998 г. [8]. Алгоритм моделирует некоторые аспекты поведения человека в обществе, а не, как можно было бы предположить, работу человеческого мозга. В алгоритме *MEC* каждый индивид рассматривается как разумный агент, функционирующий в некоторой группе индивидов. При принятии решений он ощущает влияние как со стороны членов своей группы, так и со стороны членов других групп. Точнее говоря, чтобы достичь высокого положения в группе, индивиду приходится учиться у наиболее успешных индивидов в своей группе. В то же время, для того

чтобы группа, которой принадлежит данный индивид, становилась более успешной по сравнению с другими группами, этот индивид, как и все индивиды его группы, должны руководствоваться тем же самым принципом в межгрупповой конкуренции.

Выбор в качестве объекта исследования мультимемеевых алгоритмов обусловлен следующими соображениями. Как отмечалось выше, одной из особенностей современных практических задач глобальной оптимизации является высокая вычислительная сложность оптимизируемых функций. Для решения таких задач за приемлемое для исследователя время необходимо использовать параллельные вычислительные системы. Среди доступных в настоящее время инженерному и научному сообществам вычислительных ресурсов особое место занимают грид-системы, состоящие из персональных компьютеров и рабочих станций [9]. Их широкое распространение обусловлено относительно невысокой стоимостью и простотой масштабирования. Такие системы относятся к классу слабосвязанных распределенных вычислительных систем. Авторы полагают, что таким системам адекватны мультимемеевые алгоритмы.

Существенным преимуществом мультимемеевого подхода является возможность использования в его программной реализации широкого набора мемов, так что эта реализация превращается в универсальный программный комплекс с возможностью адаптивного выбора наилучшего алгоритма для решения каждой данной задачи глобальной оптимизации.

Выбор в качестве базового для мультимемеевой гибридизации алгоритма МЕС также обусловлен ориентацией на слабосвязанные распределенные вычислительные системы. Дело в том, что для эффективного функционирования на таких системах базовый алгоритм должен иметь минимальное число связей между субпопуляциями, эволюционирующими на отдельных вычислительных узлах. Из известных в настоящее время популяционных алгоритмов лишь немногие, в том числе алгоритм МЕС, отвечают данному требованию.

Целью работы является разработка на основе алгоритма МЕС мультимемеевого алгоритма, ориентированного на класс слабосвязанных распределенных вычислительных систем, а также исследование эффективности его последовательной реализации на ряде тестовых задач глобальной оптимизации.

### Постановка задачи

Рассматриваем детерминированную задачу глобальной безусловной минимизации

$$\max_{X \in R^{|X|}} f(X) = f(X^*) = f^*, \quad (1)$$

где  $f(X)$  — скалярная целевая функция;  $f(X^*) = f^*$  — ее искомое максимальное значение;  $X = (x_1, x_2, \dots, x_{|X|})$  —  $|X|$ -мерный вектор варьируемых параметров;  $R^{|X|}$  —  $|X|$ -мерное арифметическое пространство.

Задана область

$$D = \{X | x_i^{\min} \leq x_i \leq x_i^{\max}, i \in \overline{1, |X|} \subset R^{|X|}, \quad (2)$$

в которой отыскивается решение задачи (1).

Заметим, что в вычислительной практике задачу глобальной условной оптимизации обычно сводят к задаче безусловной оптимизации, например, методом штрафных функций.

### Концепция мультимемеевых алгоритмов оптимизации

Основной проблемой синтеза мультимемеевых алгоритмов является определение целесообразной стратегии использования того или иного мема из роя доступных мемов  $M = (m_j, j \in \overline{1, |M|})$ . Указанные стратегии иногда называют *гиперэвристиками* [6].

Выбор мемов из *роя мемов*  $M$  может происходить в зависимости от значений их некоторых характеристик и/или исследуемого в данный момент фрагмента области поиска. Обычно в качестве характеристики мемов, на основе которой происходит их выбор, используют некоторую оценку эффективности мемов.

Наиболее известны три следующие категории гиперэвристик:

- случайные гиперэвристики (*random hyper heuristics*);
- жадные гиперэвристики (*greedy hyper heuristics*);
- гиперэвристики с функцией выбора (*choice-function hyper heuristics*).

К первой категории относятся стратегии, согласно которым в каждой точке принятия решений выбор мемов происходит случайным образом. При этом вероятность выбора мема может быть как постоянной, так и динамически изменяющейся в процессе оптимизации.

Жадная гиперэвристика предполагает, что сначала проводится запуск и оценка эффективности всех доступных мемов  $m_j, j \in \overline{1, |M|}$ , а затем выбирается мем, показавший наилучший результат.

Гиперэвристики с функцией выбора основаны на использовании некоторой *функции выбора*  $\phi(m_j)$ , которая может быть построена на основе одной или нескольких метрик, отражающих различные аспекты качества мемов. Данная функция может представлять собой аддитивную скалярную свертку, которая учитывает, к примеру, величину последних улучшений, сделанных мемом, возможный эффект от последовательного применения двух и более мемов, а также частоту выбора того или иного мема. Основываясь на значениях данной функции, можно использовать различные стратегии выбора мемов. Это может быть, например, прямой выбор, предполагающий на каждом шаге принятия решения выбор мема с наилучшим значением функции выбора. Возможно также использование стратегий, в которых мемов предварительно ранжируют со-



гласно значению их функции выбора, а затем экспериментально определяют лучший из них.

В данной работе выбор сделан в пользу модифицированной жадной гиперэвристики.

### Базовый алгоритм глобальной оптимизации

Алгоритм *MEC* удобно интерпретировать как мультипопуляционный. Мультипопуляция состоит из лидирующих групп  $\mathbf{S}^b = (S_1^b, S_2^b, \dots, S_{|\mathbf{S}^b|}^b)$  и отстающих групп  $\mathbf{S}^w = (S_1^w, S_2^w, \dots, S_{|\mathbf{S}^w|}^w)$ , включающих

в себя  $|\mathbf{S}^b|$  и  $|\mathbf{S}^w|$  субпопуляций соответственно. В исходном алгоритме *MEC* число индивидов в каждой из указанных субпопуляций полагается одинаковым и равным  $|\mathbf{S}|$ .

Каждая из субпопуляций  $S_i^b, S_j^w$  имеет свою коммуникационную среду, названную авторами алгоритма *MEC* *локальной доской объявлений*. Обозначим эти доски  $C_i^b, C_j^w$  соответственно. Кроме того, вся мультипопуляция  $\mathbf{S} = \{\mathbf{S}^b, \mathbf{S}^w\}$  имеет общую глобальную доску объявлений  $C^g$ . Если далее в обозначениях субпопуляций и соответствующих локальных досок объявлений индексы  $b, w$  опущены, то имеется в виду произвольная из этих субпопуляций и их досок объявлений.

Исходная версия алгоритма *MEC*, названная в работе [10] простым алгоритмом эволюции разума (*Simple MEC, SMEC*), построена на основе *операций инициализации групп, локальных состязаний (similar-taxis) и диссимилиации (dissimilation)*. Операции локальных состязаний и диссимилиации итерационно повторяются до тех пор, пока имеет место увеличение максимального *счета (scores)* лидирующих групп. При прекращении роста этого показателя решение задачи, соответствующее победителю лучшей из лидирующих групп, объявляется точкой глобального минимума. Под счетом индивида понимаем значение целевой функции  $f(X)$  в его текущем положении.

Известны различные модификации простого алгоритма эволюции разума — расширенный *MEC*, улучшенный *MEC*, хаотический *MEC* и т. д. [10, 11]. В данной работе использована модификация алгоритма *SMEC*, заключающаяся в добавлении к числу перечисленных выше операций этого алгоритма *операции декомпозиции* области поиска на подобласти. Наши исследования показывают, что такая модификация позволяет во многих случаях избежать преждевременной стагнации алгоритма.

В соответствии с концепцией мемеевой гибридизации нами также модифицирована операция локальных состязаний. Выбор оптимального для данной подобласти поиска мема осуществляется в каждой группе через фиксированное число итераций. Возможны различные критерии оценки оптимальности мемов. Используем в качестве этого критерия

максимальное найденное мемом значение целевой функции  $f(X)$ . Такой подход позволяет алгоритму подстроиться под специфику различных участков области поиска. Например, если на данном участке функция  $f(X)$  является овражной и среди доступных мемов имеется алгоритм, ориентированный на поиск максимума овражных функций, то с высокой вероятностью в качестве оптимального будет выбран именно этот мем.

Общую схему предлагаемого модифицированного алгоритма, названного нами *HMES*, представляет следующая последовательность шагов.

#### 1. *Выполняем операцию инициализации групп D.*

1а. Разбиваем область  $D$  на подобласти  $D_1, D_2, \dots, D_\eta$  путем деления интервала  $[x_\alpha^{\min}; x_\alpha^{\max}]$  на  $\eta$  равных подынтервалов. Здесь  $\alpha \in \overline{1, |X|}$ ,  $\eta$  — свободные параметры алгоритма.

1б. В каждой из подобластей  $D_k, k \in \overline{1, \eta}$ , генерируем заданное число  $\gamma$  групп  $S_{k,i}, i \in [1, \gamma]$ , где  $\gamma$  — еще один свободный параметр алгоритма:

$$\bigcup_{k=1}^{\eta} \bigcup_{i=1}^{\gamma} S_{k,i} = \mathbf{S}.$$

1в. В каждой группе  $S_{k,i}$  генерируем случайный вектор  $X_{k,i,1}$ , компоненты которого равномерно распределены в соответствующей подобласти поиска. Отождествляем этот вектор с индивидом  $s_{k,i,1}$  группы  $S_{k,i}$ .

1г. Определяем начальные координаты остальных индивидов группы  $s_{k,i,j}, j \in \overline{2, |\mathbf{S}|}$  по формуле

$$X_{k,i,j} = X_{k,i,1} + N_{|X|}(0, \sigma), \quad (3)$$

т. е. размещаем их случайным образом вокруг индивида  $s_{k,i,1}$  в соответствии с  $|X|$ -мерным нормальным законом распределения  $N_{|X|}(0, \sigma)$ , имеющим по всем  $|X|$  координатам нулевое математическое ожидание и среднее квадратичное отклонение  $\sigma$  (свободный параметр алгоритма).

1д. Вычисляем *счета* всех индивидов популяции  $\mathbf{S}$  и заносим эту информацию на соответствующие локальные доски объявлений.

1е. Формируем на основе указанной информации лидирующие группы  $\mathbf{S}^b$  и отстающие группы  $\mathbf{S}^w$ .

#### 2. *Проводим операцию локальных состязаний в каждой группе.*

2а. С доски объявлений  $C_{k,i}$  берем информацию о текущем индивиде-победителе группы  $S_{k,i}$ . Пусть это будет индивид  $s_{k,i,j^*}, j^* \in [1, |\mathbf{S}|]$ .

2б. Из текущего положения главного индивида  $s_{k,i,j^*}$  по очереди запускаем все доступные мемы. Определяем лучший мем.

2в. Последовательно запускаем из текущих положений каждого индивида группы, кроме главного, лучший мем данной группы.

2г. На основе результатов предыдущего шага определяем нового победителя каждой из групп

$s_{k, i, l^*}, l^* \in \overline{1, |S|}$ , т. е. индивида данной группы, который имеет максимальное значение счета:

$$\max_{l \in \overline{1, |S|}} f(X_{k, i, l}) = f(X_{k, i, l^*}) = f_{k, i, l^*} \quad (4)$$

2д. Формируем на основе указанной информации лидирующие группы и отстающие группы.

2е. Заносим информацию о новых победителях всех групп популяции на соответствующие локальные и глобальную доски объявлений.

### 3. Выполняем операцию диссимилиации.

3а. С глобальной доски объявлений  $S^g$  считываем счета  $f_i^b, f_j^b, i \in \overline{1, |S^b|}, j \in \overline{1, |S^w|}$  всех групп (текущие счета победителей этих групп).

3б. Выполняем сравнение указанных счетов между собой. Если счет некоторой лидирующей группы  $S_i^b$  меньше счета одной из отстающих групп  $S_j^w$ , то последняя занимает место группы  $S_i^b$  во множестве лидирующих групп, а группа  $S_i^b$  — место группы  $S_j^w$  среди отстающих групп. Если счет группы  $S_k^w$  ниже счетов всех лидирующих групп, то удаляем группу  $S_k^w$  из популяции.

3в. С помощью операции инициализации по формуле вида (3) взамен каждой из удаленных групп инициализируем новую группу.

4. **Проверяем условие останова** — превышено ли число итераций стагнации  $\lambda_{stop}$  либо максимально допустимое число итераций  $\lambda_{max}$ . Если условие останова выполнено, то объявляем решение, соответствующее текущему победителю популяции, искомым оптимальным решением  $X^*$ , иначе переходим к п. 2.

Рис. 1 (см. третью сторону обложки) на примере двумерной функции, обратной функции Растригина [12], иллюстрирует одну из итераций алгоритма *HMEC*. Красные точки на рисунке соответствуют главным индивидам групп, вертикальные линии представляют собой границы подобластей декомпозиции;  $\eta = 10$ ;  $x_i^{\min} = -10$ ,  $x_i^{\max} = 10$ ,  $i = 1, 2$ .

### Программная реализация и исследование эффективности алгоритма *HMEC*

Алгоритм *HMEC* реализован на языке C++ с использованием библиотеки *BNB-Solver* [9, 13]. Программная реализация получила наименование *HMEC-Optimizer*. Отличительными особенностями программного комплекса *HMEC-Optimizer* являются единый интерфейс подключаемых мемов, основанный на интерфейсе библиотеки *BNB-Solver*, и возможность одновременного использования различных классов мемов.

В качестве мемов в исследовании использованы алгоритмы на основе следующих методов:

- метод Солиса—Ветса [14];
- метод Нелдера—Мида [15];
- метод *LBFSGS (Limited Memory Broyden Fletcher Goldfarb Shanno)* [16].

Поскольку эффективность алгоритма существенно зависит от случайного начального расположения индивидов, во всех исследованиях каждый вычислительный эксперимент повторялся 100 раз. В качестве основных критериев качества алгоритма и его программной реализации использованы среднее число итераций  $\bar{\lambda}$  и средняя достигнутая точность  $\bar{\epsilon}$  локализации максимума целевой функции.

Вычислительные эксперименты показали, что мем на основе метода *LBFSGS* является слишком "мощным" для рассматриваемого набора тестовых функций — во всех запусках программы *HMEC-Optimizer* метод обеспечил локализацию глобального максимума исследуемых функций с высокой точностью. Поэтому ниже рассматриваем только мем, соответствующие методам Солиса—Ветса и Нелдера—Мида.

Все вычислительные эксперименты выполнены на персональном компьютере с двухъядерным процессором *Intel Core 2 Duo 2,53 ГГц* и оперативной памятью объемом 4 Гбайт.

**Классические тестовые функции.** Рассматриваем пятимерные ( $n = 5$ ) многоэкстремальные функции, обратные функциям Растригина

$$f(X) = \sum_{i=1}^{|X|} (10 + x_i^2 - 10 \cos(2\pi x_i))$$

и Гриванка

$$f(X) = \sum_{i=1}^{|X|} \left( \frac{x_i^2}{4000} \right) - \prod_{i=1}^{|X|} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1.$$

Исследование ограничено параллелепипедом

$$D = \{X \mid -10 \leq x_i \leq 10, i \in \overline{1, |X|}\}, \quad (5)$$

в котором глобальные максимумы указанных функций достигаются в точке  $X^* = (0, 0, \dots, 0)$  и равны нулю [12].

Также рассматриваем пятимерные овражные функции, обратные функциям Розенброка

$$f(X) = - \sum_{i=1}^{|X|-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$$

и Захарова

$$f(X) = \sum_{i=1}^{|X|} x_i^2 + \left( \sum_{i=1}^{|X|} 0,5ix_i \right)^2 + \left( \sum_{i=1}^{|X|} 0,5ix_i \right)^4,$$

в параллелепипеде (5). Глобальные максимумы этих функций в указанном параллелепипеде достигаются в точках  $X^* = (1, 1, \dots, 1)$ ,  $X^* = (0, 0, \dots, 0)$  соответственно и равны нулю [12].

Вычислительный эксперимент выполнен при неблагоприятных для мультимемеевого алгоритма значениях его свободных параметров:

- число индивидов в группе  $|S| = 5$ ;
- число групп в каждой из подобластей равно двум;

## Результаты экспериментов

| Критерий эффективности   | Функция Растригина   | Функция Гриванка     | Функция Розенброка   | Функция Захарова     |
|--|----------------------|----------------------|----------------------|----------------------|
| Среднее число итераций $\bar{\lambda}$   | 166                  | 102                  | 170                  | 100                  |
| Оценка математического ожидания ошибки локализации глобального минимума $\bar{\epsilon}$ | $7,41 \cdot 10^{-5}$ | $1,99 \cdot 10^{-3}$ | $1,34 \cdot 10^{-4}$ | $4,34 \cdot 10^{-7}$ |
| Минимальная ошибка локализации $\epsilon_{\min}$   | $3,38 \cdot 10^{-5}$ | $2,49 \cdot 10^{-8}$ | $2,53 \cdot 10^{-5}$ | $1,74 \cdot 10^{-7}$ |
| Оценка среднего квадратичного отклонения величины $\epsilon$                             | $1,72 \cdot 10^{-5}$ | $3,64 \cdot 10^{-3}$ | $7,29 \cdot 10^{-5}$ | $6,27 \cdot 10^{-8}$ |
| Число удачных запусков   | 100                  | 100                  | 100                  | 100                  |
| Среднее число обращений к целевой функции  | $1,18 \cdot 10^6$    | $0,8 \cdot 10^6$     | $1,12 \cdot 10^6$    | $0,8 \cdot 10^6$     |

- число подобластей  $\eta = 10$ ;
- число лидирующих групп  $|S^b| = 5$ .

Таким образом, представленные ниже результаты являются, в некотором смысле, гарантирующими. Принято, что максимально допустимое число итераций  $\lambda_{\max} = 1000$ , а период стагнации  $\lambda_{stop} = 100$  итераций.

Результаты экспериментов представлены в табл. 1 и на иллюстрирующих ее рис. 2, 3.

Как видно из табл. 1, глобальный максимум был успешно локализован с заданной точностью для всех тестовых функций в каждом из 100 запусков. Общее число итераций даже для сложных функций

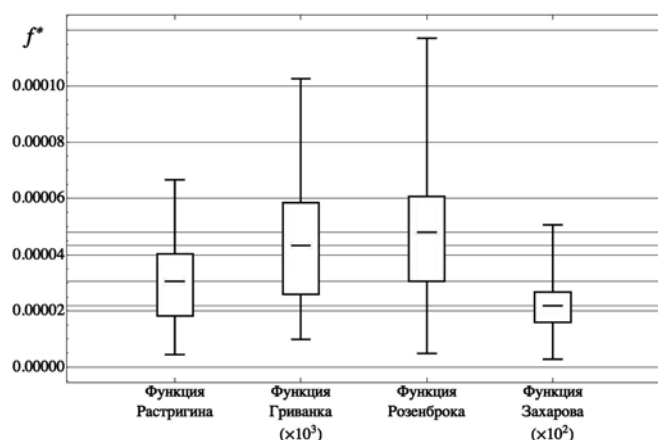


Рис. 2. Распределение максимальных достигнутых значений целевой функции по результатам мультистарта

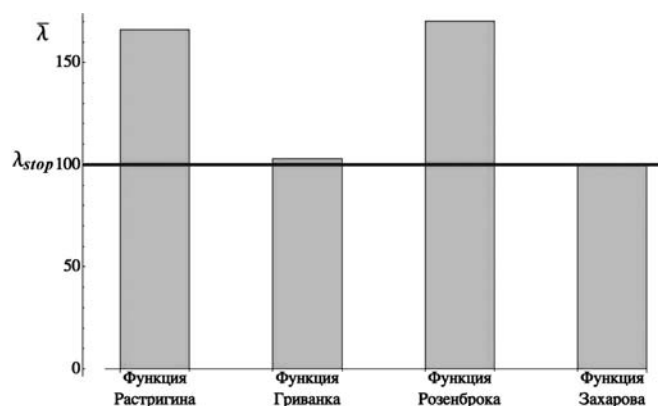


Рис. 3. Среднее число итераций  $\bar{\lambda}$

незначительно превышало период стагнации  $\lambda_{stop}$ , что свидетельствует о быстрой сходимости алгоритма к глобальному оптимуму.

Рис. 2 показывает распределение достигнутых максимальных значений тестовых функций по результатам 100 запусков. Используются ящичковые диаграммы, отображающие среднее полученное значение целевой функции и соответствующий интерквартильный размах.

На рис. 3 представлено среднее число итераций работы алгоритма для каждой тестовой функции. Жирная линия соответствует числу итераций стагнации вычислительного процесса  $\lambda_{stop}$ .

#### Минимизация энергии молекулярного кластера.

Задача минимизации энергии взаимодействия молекулярного кластера [13, 17] часто используется в качестве тестовой для алгоритмов глобальной оптимизации, что обусловлено ее высокой размерностью и многоэкстремальностью.

Модель энергии взаимодействия молекул в кластере основана на *парном потенциале*  $v(r)$ , который представляет собой функцию энергии взаимодействия двух атомов от расстояния  $r$  между ними. Энергия взаимодействия атомов кластера определяется как сумма энергий парных взаимодействий всех входящих в него  $N$  атомов:

$$E(X) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N v(r_{i,j}). \quad (6)$$

Здесь  $r_{i,j}$  — расстояние между атомами  $i$  и  $j$ ; компонентами  $(3N \times 1)$  вектора  $X$  являются декартовы координаты частиц.

Используем три наиболее распространенных потенциала: Леннард-Джонса, Морзе, Дзугутова.

Потенциал Леннард-Джонса  $v_{LJ}(r)$  представляет собой наиболее простую модель парного взаимодействия неполярных молекул и имеет вид [18]

$$v_{LJ}(r) = \delta \left( \left( \frac{r_m}{r} \right)^{12} - 2 \left( \frac{r_m}{r} \right)^6 \right). \quad (7)$$

Здесь  $\delta$  — глубина потенциальной ямы (глубина глобального минимума), а  $r_m$  — расстояние, на котором достигается значение потенциала, равное  $\delta$ . Принимаем эти параметры равными единице.

Выражение для потенциала Морзе  $v_M(r)$  записывается в виде

$$v_M(r) = e^{\rho(1-r)}(e^{\rho(1-r)} - 2), \quad (8)$$

где  $\rho$  — приведенный параметр для моделирования различных веществ.

Потенциал Дзугутова  $v_D(r)$  определяет формула

$$v_D(r) = A(r^{-m} - B)\exp\left(\frac{c}{r-a}\right)\theta(a-r) + B\exp\left(\frac{d}{r-b}\right)\theta(b-r), \quad (9)$$

где  $\theta(x)$  — функция Хевисайда:

$$\theta(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0. \end{cases}$$

Значения параметров в формуле (9) равны:

$$A = 5,82; a = 1,87; B = 1,28; b = 1,94; c = 1,1; d = 0,27; m = 16.$$

Задаче поиска глобального минимума функции (6) посвящено значительное число работ. Существует база данных [19], в которой перечислены наименьшие найденные значения энергии и соответствующие атомные конфигурации для различного числа атомов и разных потенциалов взаимодействия.

Вычислительный эксперимент выполнен в следующих условиях. Размер каждой группы фиксирован и равен  $|S| = 10$ . Общее число групп  $|S|$  варьировалось от 40 до 150. Используются следующие значения остальных свободных параметров алгоритма *НМЕС*:

- число групп в каждой из областей декомпозиции равно двум;
- среднее квадратичное отклонение при генерации популяции  $\sigma = 0,7$ ;
- максимально допустимое число итераций  $\lambda_{\max} = 1000$ ;
- период стагнации  $\lambda_{\text{stop}} = 100$ .

Начальные значения компонентов вектора варьируемых параметров  $X$  представляли собой случайные величины, равномерно распределенные в интервале  $[0; 1]$  [13]. В качестве критериев эффективности алгоритма и его программной реализации использованы точность  $\varepsilon$  найденного оптимального решения и время  $t$  расчета в секундах. Из-за высокой вычислительной сложности целевой функции (6) каждый эксперимент проводился лишь по одному разу.

Выполнено четыре серии вычислительных экспериментов (табл. 2):

- кластеры Леннард-Джонса (15, 20, 25 и 38 атомов);
- кластеры Морзе (15, 20 и 25 атомов;  $\rho = 6$ );
- кластеры Морзе (15, 20 и 25 атомов;  $\rho = 14$ );
- кластеры Дзугутова (15, 20 и 25 атомов).

Максимальное время отыскания минимума для потенциала Леннард-Джонса (строки 1—4 табл. 2) составило 3253 с. Во всех четырех случаях была найдена известная оптимальная конфигурация атомов [19]. В работе [20] при минимизации энергии 38-атомного кластера Леннард-Джонса удалось получить лишь решение, отличающееся от известного оптимального на 1,7 %. Использование алгоритма *НМЕС* позволило найти точное значение глобального минимума. При этом вычислительные затраты оказались меньше в 15 раз! Пример оптимальной конфигурации кластера Леннард-Джонса, состоящего из 38 атомов, представлен на рис. 4 (см. третью сторону обложки).

Результаты второй серии экспериментов (потенциал Морзе при  $\rho = 6$ ) представлены в строках 5—7 табл. 2. В данном случае точное значение глобального минимума также удалось отыскать для всех кластеров. Однако для этого потребовалось большее число групп, чем в первой серии экспериментов, и, как следствие, более продолжительное время расчета. Так, для кластера из 25 атомов это время составило 1769 с.

Третьей серии экспериментов (строки 8—10 табл. 2) соответствует наиболее сложный ландшафт

Таблица 2

Результаты вычислительных экспериментов

| №  | Потенциал и число атомов      | Размерность вектора $X$ | Время расчета/число атомов, с | Найденный минимум | Наилучший известный минимум | Относительная ошибка $\varepsilon$ , % |
|----|-------------------------------|-------------------------|-------------------------------|-------------------|-----------------------------|--|
| 1  | Леннард-Джонс, 15 атомов      | 45                      | 122                           | -52,322627        | -52,322627                  | 0                                      |
| 2  | Леннард-Джонс, 20 атомов      | 60                      | 163                           | -77,177043        | -77,177043                  | 0                                      |
| 3  | Леннард-Джонс, 25 атомов      | 75                      | 267                           | -102,372663       | -102,372663                 | 0                                      |
| 4  | Леннард-Джонс, 38 атомов      | 114                     | 3253                          | -173,928427       | -173,928427                 | 0                                      |
| 5  | Морзе $\rho = 6$ , 15 атомов  | 45                      | 429                           | -49,748409        | -49,748409                  | 0                                      |
| 6  | Морзе $\rho = 6$ , 20 атомов  | 60                      | 645                           | -72,507782        | -72,507782                  | 0                                      |
| 7  | Морзе $\rho = 6$ , 25 атомов  | 75                      | 1769                          | -95,127899        | -95,127899                  | 0                                      |
| 8  | Морзе $\rho = 14$ , 15 атомов | 45                      | 725                           | -44,806437        | -44,806437                  | 0                                      |
| 9  | Морзе $\rho = 14$ , 20 атомов | 60                      | 2010                          | -64,791953        | -64,791953                  | 0                                      |
| 10 | Морзе $\rho = 14$ , 25 атомов | 75                      | 8491                          | -85,316366        | -85,477376                  | 0,2                                    |
| 11 | Дзугутов, 15 атомов           | 45                      | 372                           | -26,892331        | -26,892331                  | 0                                      |
| 12 | Дзугутов, 20 атомов           | 60                      | 1150                          | -38,185980        | -38,185980                  | 0                                      |
| 13 | Дзугутов, 25 атомов           | 75                      | 4313                          | -48,633366        | -49,351951                  | 1,5                                    |

целевой функции (9) для потенциала Морзе (по сравнению с ландшафтами этой функции при других значениях параметра  $\rho$ ). Результаты экспериментов показывают, что для первых двух кластеров в этом случае удалось отыскать известный глобальный минимум. Для кластера, состоящего из 25 атомов, лучшее найденное значение отличается от известного минимума лишь на 0,2 %. Этот эксперимент оказался самым затратным с вычислительной точки зрения — время вычислений составило 8491 с.

Результаты последней серии экспериментов представлены в строках 11–13 табл. 2. Так же, как в предыдущем случае, для первых двух кластеров была найдена известная минимальная энергия взаимодействия. Для кластера, состоящего из 25 атомов, достигнута относительная ошибка  $\epsilon$ , равная 1,5 %.

Результаты экспериментов показали, что алгоритм *НМЕС* и его программная реализация позволяют в большинстве случаев получать оптимальные либо близкие к таковым конфигурации молекулярных кластеров для различных потенциалов межатомного взаимодействия. Ожидаемо, вычислительные затраты на проведение экспериментов оказались достаточно высокими, что подтверждает необходимость использования параллельных вычислений.

### Заключение

Для решения задач глобальной условной оптимизации в работе предложен алгоритм *НМЕС*, который основан на концепции мемеевых алгоритмов и использует в качестве базового алгоритм эволюции разума *МЕС*. Алгоритм *НМЕС* ориентирован на использование слабосвязанных вычислительных систем, которые на сегодняшний день являются одними из самых доступных и мощных вычислительных ресурсов.

Алгоритм *НМЕС* использует предложенную в работе модификацию канонического алгоритма *МЕС*, заключающуюся в декомпозиции области поиска в целях предотвращения преждевременной сходимости алгоритма. Гибридизация модифицированного указанным образом алгоритма *МЕС* и мемеевых алгоритмов выполнена по схеме высокоуровневого вложения [21].

Представлена реализация последовательной версии алгоритма *НМЕС* на языке C++.

Выполнено широкое исследование эффективности алгоритма *НМЕС* и его программной реализации с использованием пятимерных многоэкстремальных тестовых функций Растригина, Гриванка и аналогичных овражных функций Розенброка и Захарова. Во всех случаях алгоритм *НМЕС* и его программная реализация продемонстрировали высокие точность и вероятность локализации глобального экстремума целевых функций.

Также эффективность алгоритма *НМЕС* и его программной реализации исследована при реше-

нии задачи минимизации энергии молекулярного кластера. Рассмотрены основные потенциалы парного взаимодействия частиц кластера — потенциалы Леннард-Джонса, Морзе и Дзугутова. Проведены эксперименты для кластеров, содержащих от 15 до 38 атомов (так что размерность вектора варьируемых параметров  $X$  менялась от 45 до 114). Полученные результаты продемонстрировали высокую вероятность локализации глобального минимума и сравнительно невысокие вычислительные затраты.

На основе проведенных исследований можно сделать вывод о том, что алгоритм *НМЕС* является эффективным по критериям точности и вероятности локализации глобального оптимума.

В развитие работы планируется параллельная реализация алгоритма для слабосвязанных грид-систем.

*Авторы выражают признательность М. А. Посыпкину, благодаря сотрудничеству с которым нам стала доступной библиотека BNB-Solver.*

### Список литературы

1. Weise T. Global Optimization Algorithms. Theory and Application. University of Kassel, 2008. 758 p.
2. Karpenko A., Posypkin M., Rubtsov A., Sakharov M. Multimemetic Global Optimization based on the Mind Evolutionary Computation // Proceedings of the IV International Conference on Optimization Methods and Application "Optimization and applications" OPTIMA-2013, Moscow. 2013. P. 83–84.
3. Dawkins R. The Selfish Gene. Oxford University Press. 1976. 384 p.
4. Krasnogor N., Blackburne B., Hirst J. D., Burke E. K. Multimemetic Algorithms for the Structure Prediction and Structure Comparison of Proteins // Parallel Problem Solving From Nature. Lecture Notes in Computer Science. 2002. P. 769–778.
5. Nguyen Q. H., Ong Y. S., Krasnogor N. A Study on the Design Issues of Memetic Algorithm // IEEE Congress on Evolutionary Computation, 2007. P. 2390–2397.
6. Ong Y. S., Lim M. H., Zhu N., Wong K. W. Classification of adaptive memetic algorithms: A comparative study // IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics. 2006. P. 141–152.
7. Ong Y. S., Keane A. J. Meta-Lamarckian in Memetic Algorithm // IEEE Transactions On Evolutionary Computation, April 2004. Vol. 8, N. 2. P. 99–110.
8. Chengyi S., Yan S., Wanzhen W. A Survey of MEC: 1998–2001. 2002 IEEE International Conference on Systems, Man and Cybernetics IEEE SMC2002, Hammamet, Tunisia. October 6–9. Institute of Electrical and Electronics Engineers Inc., 2002. Vol. 6. P. 445–453.
9. Evtushenko Y., Posypkin M., Sigal I. A framework for parallel large-scale global optimization // Computer Science — Research and Development. 2009. P. 211–215.
10. Jie J., Zeng J. Improved Mind Evolutionary Computation for Optimizations // Proceedings of 5<sup>th</sup> World Congress on Intelligent Control and Automation, Hang Zhou, China. 2004. P. 2200–2204.
11. Jie J., Han C., Zeng J. An Extended Mind Evolutionary Computation Model for Optimizations // Applied Mathematics and Computation. 2007. N. 185 (2). P. 1038–1049.
12. Floudas A. A., Pardalos P. M., Adjiman C., Esposito W. R., Gilmits Z. H., Harding S. T., Klepeis J. L., Meyer C. A., Schweiger C. A. Handbook of Test Problems in Local and Global Optimization. Dordrecht: Kluwer, 1999. 441 p.
13. Посыпкин М. А. Методы и распределенная программная инфраструктура для численного решения задачи поиска молекулярных кластеров с минимальной энергией // Параллельные вычислительные технологии (ПаВТ'2009): Тр. междунар. науч. конф. Челябинск: Изд. ЮУрГУ, 2009. С. 269–281.

14. **Solis F. J., Wets R. J.-B.** Minimization by Random Search Techniques // *Mathematics of Operations Research*. 1981. Vol. 6. P. 19–30.

15. **Nelder J. A., Meade R.** A Simplex Method for Function Minimization // *Computer Journal*. 1965. V. 7. P. 308–313.

16. **Broyden C. G.** The convergence of a class of double-rank minimization algorithms // *Journal of the Institute of Mathematics and Its Applications*. 1970. V. 6. P. 76–90.

17. **Maranas C. D., Floudas C. A.** Global Minimum Potential Energy Conformations of Small Molecules // *Journal of Global Optimization*. 1994. N. 4. P. 135–170.

18. **Cai W.** Optimization of Lennard-Jones atomic clusters // *Journal of Molecular Structure (Theochem)*. 2002. N. 579. P. 229–234.

19. **The Cambridge Cluster Database.** URL: <http://www-wales.ch.cam.ac.uk/CCD.html>

20. **Карпенко А. П., Антух А. Э.** Глобальная оптимизация на основе гибридизации методов роя частиц, эволюции разума и клональной селекции // *Наука и Образование: электронное научно-техническое издание*, 2012. Вып. 8. DOI: <http://dx.doi.org/10.7463/0812.0431723>.

21. **Talbi E.** A Taxonomy of Hybrid Metaheuristics // *Journal of Heuristics*. 2002. Vol. 8. P. 541–564.

**A. P. Karpaenko**, Prof., **M. K. Sakharov**, Graduate Student, e-mail: [max.sfn90@gmail.com](mailto:max.sfn90@gmail.com)  
Bauman Moscow State Technical University

## Multi-Memes Global Optimization Based on the Algorithm of Mind Evolutionary Computation

*A subject of the paper is the hybrid global optimization algorithms based on the concept of so-called meme meta-heuristic algorithms of search engine optimization. As the basic algorithm for multi-memes hybridization we use the algorithm of Mind Evolutionary Computation (MEC). Aim of the paper is to develop a multi-memes algorithm based on MEC algorithm for the class of loosely coupled distributed computing systems, as well as research the effectiveness of its consistent implementation on number of test problems of global optimization. We state the problem of global unconstrained optimization, present the algorithm MEC as the basic algorithm for hybridization, offer a hybrid algorithm HMEC, consider consistent software of this algorithm, present the results of a study of its effectiveness.*

**Keywords:** global optimization, hybridization, meme, mind evolutionary computation

### References

- Weise T.** Global Optimization Algorithms. Theory and Application. University of Kassel, 2008. 758 p.
- Karpenko A., Posypkin M., Rubtsov A., Sakharov M.** Multimemetic Global Optimization based on the Mind Evolutionary Computation. *Proceedings of the IV International Conference on Optimization Methods and Application "Optimization and applications" OPTIMA-2013*. Moscow. 2013. P. 83–84.
- Dawkins R.** The Selfish Gene. Oxford University Press, 1976. 384 p.
- Krasnogor N., Blackburne B., Hirst J. D., Burke E. K.** Multimemetic Algorithms for the Structure Prediction and Structure Comparison of Proteins. *Parallel Problem Solving From Nature, Lecture Notes in Computer Science*. 2002. P. 769–778.
- Nguyen Q. H., Ong Y. S., Krasnogor N.** A Study on the Design Issues of Memetic Algorithm. *IEEE Congress on Evolutionary Computation*. 2007. P. 2390–2397.
- Ong Y. S., Lim M. H., Zhu N., Wong K. W.** Classification of adaptive memetic algorithms: A comparative study. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 2006. P. 141–152.
- Ong Y. S., Keane A. J.** Meta-Lamarckian in Memetic Algorithm. *IEEE Transactions on Evolutionary Computation*, April 2004. V. 8, N. 2. P. 99–110.
- Chengyi S., Yan S., Wanzhen W.** A Survey of MEC: 1998–2001. 2002. *IEEE International Conference on Systems, Man and Cybernetics IEEE SMC2002*, Hammamet, Tunisia. October 6–9. Institute of Electrical and Electronics Engineers Inc. 2002. V. 6. P. 445–453.
- Evtushenko Y., Posypkin M., Sigal I.** A framework for parallel large-scale global optimization. *Computer Science — Research and Development*. 2009. P. 211–215.
- Jie J., Zeng J.** Improved Mind Evolutionary Computation for Optimizations. *Proceedings of 5<sup>th</sup> World Congress on Intelligent Control and Automation*, Hang Zhou, China. 2004. P. 2200–2204.
- Jie J., Han C., Zeng J.** An Extended Mind Evolutionary Computation Model for Optimizations. *Applied Mathematics and Computation*. 2007. N. 185. P. 1038–1049.
- Floudas A. A., Pardalos P. M., Adjiman C., Esposito W. R., Gfirtis Z. H., Harding S. T., Klepeis J. L., Meyer C. A., Schweiger C. A.** Handbook of Test Problems in Local and Global Optimization. Dordrecht: Kluwer, 1999. 441 p.
- Posypkin M. A.** Methody i raspredelennaya programmnaya infrastruktura dlya chislennogo resheniya zadachi poiska molekulyarnykh klasterov s minimal'noy energiyey. *Parallel'nye vychislitel'nye tekhnologii (PaVT'2009): Tr. mezhdunar. nauch. konf.* Chelyabinsk: Izd. YuUrGU, 2009. P. 269–281.
- Solis F. J., Wets R. J.-B.** Minimization by Random Search Techniques. *Mathematics of Operations Research*. 1981. V. 6. P. 19–30.
- Nelder J. A., Meade R.** A Simplex Method for Function Minimization. *Computer Journal*. 1965. V. 7. P. 308–313.
- Broyden C. G.** The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*. 1970. V. 6. P. 76–90.
- Maranas C. D., Floudas C. A.** Global Minimum Potential Energy Conformations of Small Molecules. *Journal of Global Optimization*. 1994. N. 4. P. 135–170.
- Cai W.** Optimization of Lennard-Jones atomic clusters. *Journal of Molecular Structure (Theochem)*. 2002. N. 579. P. 229–234.
- The Cambridge Cluster Database.** URL: <http://www-wales.ch.cam.ac.uk/CCD.html>
- Karpenko A. P., Antukh A. E.** Global'naya optimizatsiya na osnove gibridizatsii metodov roya chastits, evolutsii razuma i klonal'noy selektsii. *Nauka i obrazovanie: elektronno-tekhnicheskoe izdanie*. 2012. Iss. 8.
- Talbi E.** A Taxonomy of Hybrid Metaheuristics. *Journal of Heuristics*. 2002. V. 8. P. 541–564.

А. Н. Родионов, д-р техн. наук, вед. науч. сотр., Вычислительный центр ДВО РАН, г. Хабаровск  
E-mail: ran@newmail.ru

## Моделирование и реализация отношения "is part of" на множестве композитных сущностей баз данных

*Исследованы свойства графа, моделирующего состав макетов композитных сущностей. Сформирован набор ограничений, которым должны удовлетворять реляционные структуры баз данных, содержащие сведения о составе композитных типов. Разработан алгоритм проверки полученных ограничений.*

**Ключевые слова:** сущность, модель данных, композитный тип, реляционное отношение, ограничения целостности, концептуальная и логическая модели

### Введение

Данные, сосредоточенные в базах данных, должны быть согласованными. Это означает, что они должны удовлетворять всем ограничениям, которые на них накладываются. Принято различать внутренние ограничения, обусловленные природой логических моделей (реляционных, сетевых, иерархических и пр.), и явные ограничения вытекающие из бизнес-логики, действующей в конкретных предметных областях. Если первые поддаются формализации и унификации и поэтому поддерживаются всеми системами управления базами данных (СУБД) по умолчанию, то в отношении вторых этого сказать нельзя. Единственное исключение — явные ограничения, задаваемые посредством функциональных зависимостей (ФЗ) и расширений ФЗ: функциональных зависимостей, содержащих условия [1], и "совпадающих" (*matching*) ФЗ [2].

Есть еще одна группа ограничений, которая фактически выпадает из поля зрения исследователей. Ограничения, образующие эту группу, как и внутренние ограничения, также могут быть формализованы и унифицированы, а их нарушение будет всегда приводить к *рассогласованности* хранящейся в базах данных информации.

В работе рассматривается подмножество таких ограничений, обусловленных природой композитных сущностей (сущностей, состоящих из других сущностей) и отношения "is part of" (*IPO*), возникающего на множестве композитных экземпляров, с одной стороны, и конфигурацией структур, посредством которых состав композитов может быть представлен в схемах организации данных, с другой стороны.

### 1. Используемые категории и понятия

Классический подход к построению схем организации данных (моделей данных) предусматривает последовательное выполнение как минимум двух этапов: концептуального и логического проектиро-

вания. Последний, по сути, сводится к трансляции концептуальной схемы (КС) в логическую модель (в нашем случае в реляционную модель данных — РМД). Мы также будем придерживаться этой последовательности.

На концептуальном уровне в обязательном порядке назначаются типы сущностей, выявляются типы связей (взаимодействий) между ними, устанавливаются кардинальности связей и наборы актуальных свойств, характеризующие каждый отдельно взятый тип. Здесь же проводится классификация всех типов сущностей на "сильные" (экземпляры которых реально существуют) и "слабые, ассоциативные". Вторые отражают результаты "сложных" взаимодействий первых между собой. Такие взаимодействия согласно [3] приводят к появлению так называемых "данных пересечения". В концептуальную модель включаются все перечисленные компоненты.

Трансформация КС в РМД проводится посредством удаления всех связей и замены их атрибутами внешних или составных первичных ключей. РМД манипулирует отличными от КС категориями. Таблицы, соответствующие типам сущностей в КС, в РМД рассматриваются как математические отношения с заданной именованной схемой. Для каждой схемы перечисляются все образующие ее атрибуты с указанием доменов, на которых определен каждый из атрибутов. Отношение представляет собой подмножество декартового произведения доменных множеств, а элементы таких множеств (кортежи) ассоциируются с отдельными экземплярами "сильных" и "слабых" сущностей.

В процессе даталогического моделирования на концептуальном уровне типов сущностей будем проводить различие между *Pattern*-структурой, предназначенной для фиксации макетов (образцов) сущностей, принадлежащих моделируемому типу, и *Sample/Instance*-структурами, в которых размещаются реальные экземпляры [4].

## 2. Цели математического моделирования и математические модели композитных сущностей

Перечислим цели математического моделирования состава композитных типов, достижение которых позволит установить искомые ограничения.

Первая цель — формирование набора ФЗ  $F_N$ , которые могут устанавливаться на множестве  $\Omega$ , состоящем из самих композитных типов и образующих их элементов.

Вторая цель — нахождение группы ФЗ  $F_R$ , которые не могут существовать на  $\Omega$ , но допустимы в схемах отношений, используемых для представления КТ в моделях данных. (Именно эти ФЗ выступают в роли ограничений целостности КТ и должны всегда соблюдаться в соответствующих отношениях баз данных.)

Третья цель состоит в разработке алгоритмов, реализующих полный набор ограничений целостности КТ — ограничений, не допускающих появления  $F_R$ .

Природа композитных типов такова, что каждый их элемент (за исключением непосредственно экземпляров самих КТ) находится в отношении (ассоциации) IPO (*есть часть чего-либо*) с каким-нибудь другим элементом. В этом случае топологическая структурная модель в виде дерева общего вида дает исчерпывающее представление о составе отдельных экземпляров реальных КТ (рис. 1, а).

Но одной этой модели недостаточно, чтобы составить исчерпывающее представление обо всех разновидностях композитов (а точнее макетов композитов), отнесенных к некоторому типу. Эту цель реализует граф, изображенный на рис. 1, б. Заметим, что дерево экземпляра КТ (рис. 1, а) строится на основе графа КТ. Принимая во внимание первичность графа КТ по отношению к дереву экземпляра КТ, дальнейшее изложение будет касаться главным образом вопросов моделирования макетов КТ.

Введем ряд терминов, которые понадобятся в дальнейшем в нашей работе.

*Детали (Details)* —  $D$  — элементарные, неделимые сущности, которые могут быть частью других сущностей.

*Компоненты (Subassemblies)* —  $K$  — полуфабрикаты, или сущности, для которых справедливы следующие ассоциации: " $D$  is part of  $K$ ", " $K$  is part of  $K$ ", " $K$  is part of  $I$ ".

*Изделия (Assemblies)* —  $I$  — собственно композитный тип, участвующий в ассоциациях: " $D$  is part of  $I$ " и " $K$  is part of  $I$ ".

Верхний индекс  $m$  в обозначениях  $D_n^m$ ,  $K_n^m$ ,  $I_n^m$  ссылается, соответственно, на макеты *деталей*, *компонентов* и *изделий*, нижний индекс  $n$  — на их экземпляры.

Отношение IPO обладает рядом важных свойств, которые задают структуру графа КТ. Это антисимметричность (если какой-то элемент является частью другого элемента, то последний не может быть частью первого), отсутствие рефлексивности (элемент не может быть частью самого себя) и транзитивности (если один элемент часть другого, а другой часть третьего, то первый часть третьего).

Применительно к графу КТ антисимметричность говорит об отсутствии параллельных ребер между вершинами, а антирефлексивность не допускает появления ребер, замыкающихся на одну и ту же вершину (в графе не могут появиться петли). Кроме того, антисимметричность как свойство свидетельствует и об установлении определенного порядка между элементами множества  $\Omega$ . Этот порядок позволяет различать в любых парах связанных вершин исходную и конечную.

В графе КТ могут быть выделены подграфы, обладающие теми же свойствами, что и граф КТ.

Подмножество  $\Psi$ , образованное "*компонентами*" ( $\Psi \subset \Omega$ ), в реальных предметных областях, как правило, разбивается на  $N$  непересекающихся подмножеств  $\psi_k$ ,  $\Psi = \{\psi_1, \psi_2, \psi_k, \dots, \psi_N\}$ . Основанием для подобного разбиения служит то обстоятельство, что различные группы (подмножества) компонентов характеризуются различными наборами свойств. Справедливо, что для любых двух  $\psi_i, \psi_j \in \Psi$ ,  $P_{\psi_i} \neq P_{\psi_j}$ , где  $P_{\psi_i}$  и  $P_{\psi_j}$  — множества свойств компонентов, принадлежащих соответственно подмножествам  $\psi_i$  и  $\psi_j$ .

Отношение "*is part of*" действует и между элементами, принадлежащими разным  $\psi_k$ , что заставляет принимать в расчет и свойства отдельных подграфов графа КТ.

Все обозначенное выше дает основание получить полное множество функциональных зависимостей  $F_N$ , которые должны соблюдаться в процессе моделирования макетов композитных типов. Оно включает многозначные зависимости следующих видов:

$$D \rightarrow I, D \rightarrow K_{\psi_i},$$

$$K_{\psi_i} \rightarrow K_{\psi_j}, K_{\psi_i} \rightarrow I.$$

Заметим, что применительно к дереву экземпляров все перечисленные многозначные ФЗ преобразуются к однозначным ФЗ.

В другое множество — множество избыточных функциональных зависимостей  $F_R$  — войдут те зависимости, которые порождаются

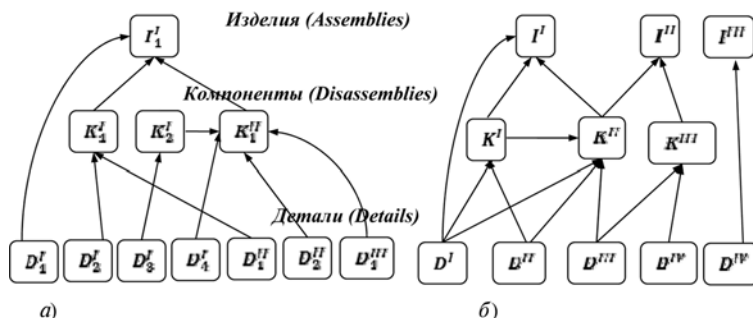


Рис. 1. Примеры графа макета КТ (б) и построенного на его основе дерева экземпляра КТ (а)



схемой реляционного отношения и непосредственно самим реляционным отношением, представляющими граф КТ в модели данных. При проведении анализа этого отношения и его схемы будем руководствоваться такими характерными свойствами графа КТ как недопустимость циклов, отсутствие петель и параллельных ребер. (Все перечисленные свойства — еще одно очевидное следствие отношения "is part of", действующего на множестве  $\Omega$ .)

Задачу по определению элементов множества  $F_R$  логично отложить до рассмотрения вопроса представления графа КТ посредством реляционного отношения. Пока же отметим, что, если конкретное реляционное отношение как структура, используемая для хранения графа КТ в базе данных, допускает присутствие кортежей, интерпретируемых как петли, параллельные и обратные ребра (последние — признак цикла), то все это является основанием для существования избыточных функциональных зависимостей.

### 3. Принципиальные подходы к представлению композитных типов

Поскольку любой граф состоит из множества узлов и множества дуг, достаточно всего двух взаимосвязанных структур, чтобы представить его в модели данных. Первая структура (в терминологии баз данных — "сильная сущность") предназначена для размещения всех узлов (в том числе и тех, которые не входят в состав графа), вторая ("слабая сущность") — для хранения дуг (а если более точно, то самого графа, поскольку перечисление дуг адекватно заданию самого графа). Таким образом, состав КТ будет всегда моделироваться посредством кластера, включающего связку "сильная" — "слабая" сущности.

Две принципиальные подходы, моделирующие КТ, показаны на рис. 2. На нем "слабые сущности" именованы как *Composition*-структуры (*C*-отношения), а "сильные сущности" — это объекты, помеченные как "Изделия", "Компоненты" и "Детали".

$A$  обозначает атрибут, идентифицирующий узел;  $A_1, A_2$  — наследуемые из  $A$  атрибуты, совместно идентифицирующие дугу. Заметим, что  $A$  и  $A_1, A_2$  — также выполняют функцию первичных ключей отношений, в которые они входят.

Рекурсивные связи в виде петель, которые на рис. 2 имеются у "сильных сущностей" (за исключением, по очевидным причинам, у "Изделий" и "Деталей"), преобразованы в *C*-отношения. Штриховые линии означают необязательность присутствия в подсхеме данных помеченных подобным образом связей и структур.

Очевидно, что данные, касающиеся состава сущностей, размещаются в *Composition*-структурах, а сами экзем-

пляры сущностей и их характеристики — в "Изделиях", "Компонентах" и "Деталях".

На рис. 2, б *Composition*-объекты классифицированы на обязательные (*mandatory*) и необязательные (*optional*). Введение *Composition\_optional* — следствие установления связей между сущностями одного уровня (о чем было сказано выше). Такие связи могут как присутствовать, так и отсутствовать между элементами КТ, что отражает реалии конкретной предметной области.

Поскольку подсхем, моделирующих состав КТ, две — нельзя оставить без внимания вопрос о предпочтении одной схемы (рис. 2, а) по отношению к другой (рис. 2, б), несмотря на то что это не предмет настоящей статьи.

Без проведения всестороннего исследования однозначно утверждать, что одна схема лучше другой достаточно сложно. Поэтому ограничимся только общими моментами.

Очевидно, что добавление каждого нового компонентного уровня повлечет за собой рост общего числа *C*-объектов, по  $k + 1$  объект на каждый очередной  $k$ -й уровень, начиная со второго. (Все объекты "Компоненты" должны быть попарно связаны друг с другом, а также с "Изделиями" и "Деталями".) Как следствие, будет увеличиваться и продолжительность выполнения любых запросов, как на считывание данных из *C*-отношений, так и на запись в них данных. (Это верно в случае, если продолжительность использовать в качестве единственного критерия сопоставимости двух подсхем, в действительности, весьма вероятно, что это не единственный критерий). Но и для подсхемы также будет наблюдаться увеличение времени ожидания на получение данных для некоторых категорий запросов (рис. 2, а), например, запросов, связанных с формированием графа КТ в целях его визуализации.

В связи с перечисленным констатируем лишь сам факт наличия проблемы — проблемы выбора оптимальной подсхемы, представляющей КТ.

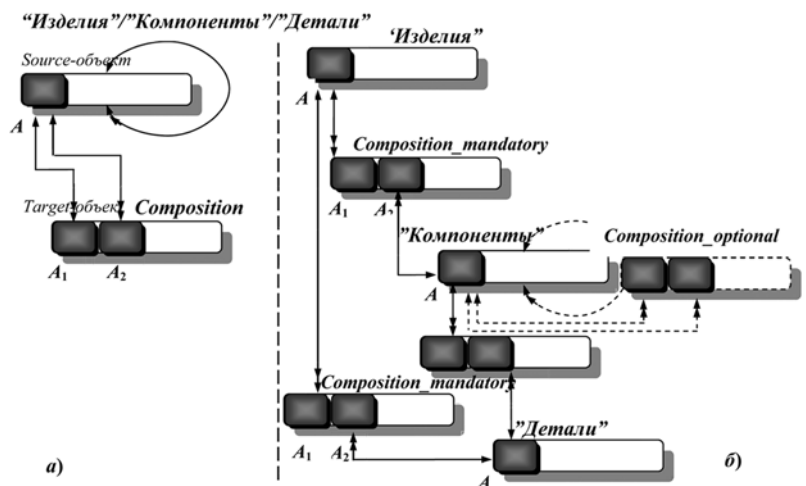


Рис. 2. Альтернативные подходы к представлению графа КТ в модели данных

#### 4. Ограничения C-отношений Pattern-структур

В данном разделе решаются две задачи. Первая задача — формирование полного набора ограничений, которым должен удовлетворять каждый экземпляр C-отношения, соответствующий C-схеме. (Любая транзакция, изменяющая кортежи в C-отношении, генерирует очередной, текущий экземпляр C-отношения.)

Вторая задача — разработка алгоритмов проверки полученных ограничений после выполнения "изменяющих" транзакций. Такие транзакции всегда включают либо единичные операции манипулирования данными — вставки  $I$ , удаления  $D$ , обновления  $U$ , либо их комбинации.

За основу возьмем подсхему, приведенную на рис. 2, а, поскольку она интегрирует в себе все объекты и связи, содержащиеся в подсхеме (рис. 2, б).

Преобразуем Composition-структуру в схему отношения, которая будет ей соответствовать в концептуальной модели данных, и рассмотрим ее реализацию в виде совокупности кортежей (рис. 3). Схема отношения задается как именованное множество с перечислением всех его элементов. Для Composition-структуры схема будет выглядеть как  $C = \{A_1^*, A_2^*, A_c\}$ . В состав C-отношения вошел количественный атрибут  $A_c$ , показывающий, какое число экземпляров  $A_2$  входит в состав  $A_1$ . Индекс "\*" обозначает атрибут первичного ключа.

Ранее было отмечено, что структуры (C-отношения), в которых хранятся данные, описывающие состав макетов КТ, поддерживают расширенный набор функциональных зависимостей, включая как допустимые  $F_N$ , так и те, которые не должны выполняться ( $F_R$ ). Кроме того, отношение, полученное на основе C-схемы, содержит одну аномалию (неопределенность) семантического рода, на устранении которой сосредоточимся в первую очередь.

| $A_1$ | $A_2$ |       |
|-------|-------|-------|
| 1     | 1     | $t_1$ |
| 1     | 2     | $t_2$ |
| 2     | 3     | $t_3$ |
| 3     | 2     | $t_4$ |
| 5     | 7     | $t_5$ |
| 8     | 5     | $t_6$ |
| 7     | 8     | $t_7$ |

Рис. 3. Фрагмент C-отношения

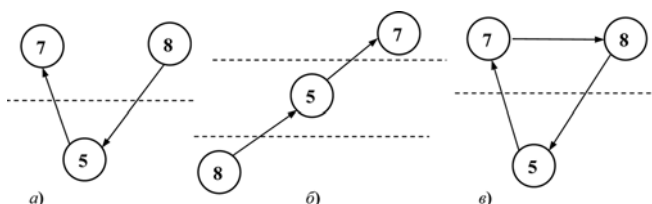


Рис. 4. Интерпретация некоторых групп кортежей C-отношений

Любой кортеж, входящий в C-отношение, может трактоваться двояко. Возьмем для примера кортеж  $t_2$  (рис. 3). В этом кортеже атрибут  $A_1$  принимает значение 1, а атрибут  $A_2$  — значение 2. Если заранее не условиться, какой из атрибутов целевой, т. е. обозначает объект, в который входит другой объект, а какой исходный, т. е. является объектом, входящим в состав целевого объекта, то нельзя составить однозначное представление о том, какой объект частью какого является: 1-й входит во 2-й или, напротив, 2-й является частью 1-го.

Чтобы исключить возникшую неопределенность, следует по умолчанию принять для всех C-отношений соглашение относительно роли  $A_1$  и  $A_2$  атрибутов: какой из них всегда является "целым", а какой выполняет функцию "части". Формально, следует задать отношение строгого порядка на множестве, состоящем из двух элементов:  $A_1$  и  $A_2$ . Правда это противоречит постулату, что схема отношения представляет собой множество атрибутов, а множества неупорядочены по определению. Но без задания порядка в данном конкретном случае обойтись нельзя.

Далее перейдем к анализу C-отношения на предмет выявления в нем нежелательных функциональных зависимостей — зависимостей, которые не могут присутствовать на ориентированном, несвязном, простом графе.

Кортеж  $t_1$  недопустим в C-отношении, поскольку ФЗ вида  $D \rightarrow D$ ,  $I \rightarrow I$ ,  $K_{\psi_i} \rightarrow K_{\psi_i}$ , не входят в состав  $F_N$ .

Ограничение целостности (присвоим ему условный номер — № 1) для этого случая можно сформулировать следующим образом: *после выполнения операций  $I$  и  $U$  в C-отношении не должны присутствовать кортежи  $t_i$ , для которых  $t_i(A_1) = t_i(A_2)$ . Здесь  $t_i(A_j)$  обозначает значение, которое принимает атрибут  $A_j$  в  $i$ -м кортеже C-отношения.*

Следующие два кортежа  $t_3$  и  $t_4$  допустимы для C-отношения, но в то же время они являются взаимоисключающими, так как граф КТ не должен содержать параллельных ребер. Таким образом, в C-отношение может входить только один из двух кортежей: или  $t_3$ , или  $t_4$ .

Запишем ограничение целостности (№ 2) для данного случая: *после выполнения операций обновления  $U$  или добавления  $I$  в C-отношение кортежа  $t_j$  в C-отношении не может отыскаться кортеж  $t_i$ , удовлетворяющий условию  $t_i(A_1) = t_j(A_2) \wedge t_i(A_2) = t_j(A_1)$ .*

Интерпретация следующих двух кортежей  $t_5$  и  $t_6$  может быть только такой, как это показано на рис. 4, б. Первая трактовка (рис. 4, а) исключена, так как между элементами, принадлежащими различным уровням, действует отношение IPO.

Несколько усложним задачу, добавив в C-отношение кортеж  $t_7$ . Получим цикл, тогда как граф, моделируемый C-отношением должен быть ациклическим. Следовательно, любая операция  $I$  или  $U$  должна инициировать процедуру, которая ищет

циклы в графе КТ. Сформулируем еще одно ограничение (№ 3), которое всегда должно выполняться для данных, размещенных в  $S$ -отношении.

*Вставка, удаление и модификация любого кортежа  $t$ ,  $t = \langle t(A_1), t(A_2) \rangle$  в  $S$ -отношении не должны приводить к появлению циклов в графе, который моделируется этим  $S$ -отношением. Формально это означает появление пути из вершины  $t(A_1)$  в  $t(A_1)$ , в котором следующей после вершины  $t(A_1)$  должна следовать вершина  $t(A_2)$ .*

До сих пор мы рассматривали только "единичные" транзакции — транзакции, выполняющие манипуляцию над одним кортежем. Но процедуры, изменяющие  $S$ -отношение, особенно когда входной поток транзакций формируется на основе "отсоединенных" данных, могут затрагивать больше чем один кортеж. В этом случае проверка на корректность  $S$ -отношения (соответствие всем ограничениям целостности) должна выполняться или после обработки всех кортежей входного потока, или после каждой элементарной  $I$ -транзакции, при условии, что перед этим успешно были завершены все  $D$ -операции. (При этом будем полагать, что любая  $u_i$ -операция будет реализована как  $\langle d_i, i_i \rangle$  — сначала удаление, а затем вставка. Ведь по сути  $u = dUi$ .)

Задание и следование только что сформулированному условию вполне оправданы, если принять во внимание тот факт, что в противном случае, появившиеся в графе после вставки новых ребер циклы, петли и параллельные ребра, могут исчезнуть после выполнения  $D$ -операций.

## 5. Вопросы реализации и стандартизации

Вне зависимости от того, какая схема работы с  $S$ -отношением будет задействована в прикладной программе (схема "тонкого" или "толстого" клиентов [5]), в составе базы данных обязательно должны присутствовать  $I$ ,  $U$ ,  $D$ -триггеры для каждой  $S$ -таблицы, функция которых — контроль над выполнением всех записанных ранее ограничений целостности. Подобный двухуровневый контроль ограничений — на уровне приложений и СУБД — общепринятая практика [6].

Эффективные "триггерные" алгоритмы, ввиду того, что запускаются для всех транзакций (в случае контроля СУБД), должны обрабатываться за ограниченное время. Самой трудоемкой из всех операций окажется операция, выполняющая поиск циклов в графе. Временная сложность алгоритма обхода графа в глубину, посредством которого может быть выполнена эта проверка, составляет  $O(n + m)$ , где  $n$  и  $m$  — это соответственно число вершин и ребер в графе КТ.

Тем не менее далеко не во всех случаях требуется проводить подобный поиск. При вставке ребер, одна из вершин которых лист, корень или и то, и другое одновременно, циклы в графе не возникают.

Далее излагается алгоритм проверки ограничений, которые необходимо контролировать при добавлении каждого очередного кортежа в  $S$ -отношение.

*Вход: кортеж  $t_i = \langle t_i(A_1), t_i(A_2) \rangle$ , подлежащий вставке.*

```
// -----
// Проверка ограничения № 1
// -----
Если  $t_n(A_1) = t_n(A_2)$  // Нарушение ограничения
Выдать сообщение и выйти из программы Return(1)
// -----
// Проверка ограничения № 2
// -----
Получить кортеж  $t_f$  в  $S$ -отношении, удовлетворяющий
условию:  $t_f(A_1) = t_i(A_2) \wedge t_f(A_2) = t_i(A_1)$ .
Если кортеж существует, выдать сообщение о нарушении
условия № 2 и выйти из программы Return(2)
// -----
// Проверка ограничения № 3
// -----
Выполнить проверку на возможность появления в графе
КТ цикла из  $t_i(A_1)$  в  $t_i(A_1)$  после вставки  $t_i$ . Если цикл
отсутствует, добавить  $t_i$  в  $S$ -отношение и выйти из
программы Return(0), если нет — выдать сообщение об
ошибке и выйти из программы Return(3).
// -----
```

Рассмотренные ограничения расширяют известные границы стандартизации [7] и могут в принудительном порядке поддерживаться реляционными СУБД после включения в состав баз данных соответствующих триггеров.

## 6. Направления дальнейших исследований

Можно обозначить, по меньшей мере, две задачи, которые возникают по мере расширения границ исследования. Первая задача — это разработка оригинального алгоритма поиска циклов в графе КТ, который хранится в базе данных как реляционное  $S$ -отношение. Желательно иметь алгоритм с временной сложностью  $O(\lg n)$ , поскольку для алгоритма с  $O(n)$  время ожидания результата проверки, например для  $n \approx 10^5$ , составит 0,1 мс [8] (что не так уж и много). Соответственно, в случае, например 10 транзакций, запрашиваемых одновременно, выполнение данной проверочной функции, среднее время обработки каждой транзакции (непосредственное выполнение плюс ожидание) составит  $\approx 0,56$  мс. Такая продолжительность, вероятнее всего, для многих информационных систем окажется неприемлемой.

Алгоритм должен принимать в расчет отличительные признаки графа КТ, в частности, наличие у него корневых вершин и вершин—листьев, а также его частичную упорядоченность, и использовать наиболее предпочтительную структуру хранения в оперативной памяти: матрицу смежности, списки смежности, связанное представление или представление в виде  $S$ -отношения.

Другая задача связана с выбором одной из альтернативных схем хранения, изображенных ранее на рис. 2. Сложность задачи окажется тем выше, чем больше факторов будет принято в расчет и чем больше, и каких выходных показателей будет рассматриваться. Из факторов назовем только два: общее число "компонент" и "composition"-структур в IPO-кластере, а среди критериев оптимальности — среднее время ожидания выполнения "корректирующих" транзакций и среднее квадратическое отклонение этого времени.

### Заключение

Исследование и моделирование отношения IPO, возникающего на множестве, включающего простые и композитные объекты, сначала с помощью деревьев и графа, а далее посредством реляционных структур позволило установить набор ограничений, которые, как и внутренние ограничения реляционных моделей, могут и должны подвергаться обязательной проверке для обеспечения согласованности данных. Эти ограничения являются одним из следствий взаимодействия "состава" [9] между сущностями и могут рассматриваться в качестве отдельной, самодостаточной группы ограничений, наряду с явными и внутренними ограничениями.

Предложено специальным образом различать "слабые сущности" —  $S$ -отношения, в которых хранится информация о составе композитных объектов. С этой целью вводится отношение порядка на множестве ключевых атрибутов  $S$ -отношений и за каждым семантическим атрибутом закрепляется определенная семантическая роль.

### Список литературы

1. Bohannon P., Fan W., Geerts F., Jia A., Kementsietsidis A. Conditional functional dependencies for data clearing // Proc. of ICDE. 2007. P. 746—755.
2. Fan W. Dependencies revisited for improving data quality // Proc. of PODS. 2008. P. 159—170.
3. Мартин Дж. Организация баз данных в вычислительных системах. М.: Мир, 1980.
4. Родионов А. Н. Организация баз данных. Семантическая идентификация и концептуальное моделирование сущностей. Германия, Саарбрюкен: LAP LAMBERT Academic Publishing, 2013.
5. Сеппа Д. Программирование на Microsoft® ADO.NET 2.0. Мастер-класс. М.: Русская Редакция; СПб.: Питер, 2007.
6. Zhang H., Tan H. B. K., Zhang L., Lin H., Wang X., Zhang C., Mei H. Checking enforcement of integrity constraints in database applications based on code patterns // The Journal of Systems and Software. 2011. V. 84. P. 2253—2264.
7. Zviran M., Glezer C. Towards generating a data integrity standards // Data and knowledge engineering. 2000. V. 32 (3). P. 291—313.
8. Скиена С. Алгоритмы. Руководство по разработке. СПб.: БХВ-Петербург, 2013.
9. Родионов А. Н. Критерии качества даталогических схем. Полнота моделей данных // Вестник ХГАЭП. 2010. № 2. С. 28—51.

A. N. Rodionov, Leading Researcher,  
Computer Centre Of Far-Eastern Branch of RAS, e-mail: ran@newmail.ru

## Modeling and Implementation of "Is Part of" Relation at a Set of Databases Composite Entities

*Any domain of interest incorporates entities which consist of other entities. The letter may be both simple, indivisible entities and composite entities. The "is part of (IPO)" relation is always arisen on the set which integrates such entities. The implementation of given relation is become compulsory if it needs for the application domain under consideration.*

*In this article we research IPO relation to find out the set of the functional dependencies  $FD_c$ , which exist between the members of the set above-mentioned. For this purpose it utilizes a graph as a most suitable formal construction. In data models the structure of composite entities are often represented by means of the objects referred as week entities. The relations, which correspond to a week entities in a relational model, comprises the owned functional dependencies  $FD_w$ . Comparison of  $FD_c$  with  $FD_w$  shows that  $FD_c \subset FD_w$ . In this case the difference of  $FD_w$  and  $FD_c$  sets may consider as a new integrity constraint class for relational models. These constraints must be fulfilled for all relations where accommodate data about the structure of composite types. Finally, we propose the algorithm for checking of derived constraints and formulate the new problems in this avenue of research.*

**Keywords:** entity, composite type, relation, integrity constraints, conceptual and logical models

### References

1. Bohannon P., Fan W., Geerts F., Jia A., Kementsietsidis A. Conditional functional dependencies for data clearing. ICDE, 2007. P. 746—755.
2. Fan W. Dependencies revisited for improving data quality. PODS, 2008. P. 159—170.
3. Martin D. Organizatsiya baz dannykh v vychislitel'nykh sistemakh. M.: Mir, 1980.
4. Rodionov A. N. Organizatsiya baz dannykh. Semanticheskaya identifi-katsiya i kontseptual'noe modelirovanie sushchnostey. Germaniya, Saarbryuken: LAP LAMBERT Academic Publishing, 2013.
5. Seppa D. Programmirovaniye na Microsoft® ADO.NET 2.0. Master-klass. M.: Russkaya Redaktsiya; SPb.: Piter, 2007.
6. Zhang H., Tan H. B. K., Zhang L., Lin H., Wang X., Zhang C., Mei H. Checking enforcement of integrity constraints in database applications based on code patterns. The Journal of Systems and Software. 2011. Vol. 84. P. 2253—2264.
7. Zviran M., Glezer C. Towards generating a data integrity standards. Data and knowledge engineering. 2000. Vol. 32 (3). P. 291—313.
8. Skiena S. Algoritmy. Rukovodstvo po razrabotke. SPb.: BKhV-Peterburg. 2013.
9. Rodionov A. N. Kriterii kachestva datalogicheskikh skhem. Polnota modeley dannykh. Vestnik KhGAEP. 2010. N. 2. P. 28—51.

УДК 004.652.4(045)

Ю. В. Полищук, канд. техн. наук, доц., e-mail: youra\_polishuk@bk.ru,  
Т. А. Черных, канд. техн. наук, ст. преподаватель, e-mail: chatty84@mail.ru,  
ФГБОУ ВПО "Оренбургский государственный университет"

## О способах реализации концепции единого источника

Рассмотрены наиболее распространенные способы реализации концепции единого источника. Перечислены преимущества и недостатки данных способов. Предложен способ реализации концепции единого источника с применением хранилища данных, ориентированного на работу с квазиструктурированным информационным наполнением документов. Рассмотрено математическое описание и приведено графическое представление квазиструктурированной модели информационного наполнения документа. В качестве примера приведено описание модели "Руководство пользователей автоматизированного рабочего места" из пакета документов рабочей документации. Структура данного документа разработана в соответствии с ГОСТ 19.505—79 "Руководство оператора. Требования к содержанию и оформлению". Предложенная в работе реализация концепции единого источника совмещает в себе преимущества традиционных способов построения систем данного вида и обеспечивает дополнительный контроль корректности фактографического контента документов за счет ограничений, накладываемых моделями документов.

**Ключевые слова:** концепция единого источника, квазиструктурированный контент

### Введение

Разработка и поддержание технической документации в актуальном состоянии вызывают множество проблем, связанных, как правило, с необходимостью многократного внесения корректировок в документацию. В практике разработчиков технической документации распространено понятие "концепция единого источника" (*single source publishing*). Это концепция публикации документов, согласно которой один и тот же контент может быть использован в разных выпускаемых документах или в разных форматах [1]. Фактографическая информация из документации формируется в виде единого источника, который может быть представлен коллекцией документов или единым хранилищем данных, а формирование нового документа заключается в выборке фрагментов из единого источника.

Таким образом, трудоемкая работа по редактированию должна быть проведена только однажды в одном месте (внутри единого источника), дальнейшие преобразования в выходных документах различных форматов выполняются автоматически соответствующим программным средством [2].

### Постановка задачи

Рассмотрим подробнее описанные выше подходы в организации единого источника.

Схематичное представление реализации концепции единого источника в случае, когда он представлен коллекцией документов, приведено на рис. 1.

На рис. 1 штриховыми линиями показаны связи между фрагментами документов. Данный способ реализации характеризует низкая защищенность, обусловленная отсутствием в системе единого централизованного хранилища данных и сложностью в организации многопользовательской работы.

Способ реализации концепции единого источника с применением хранилища данных схематично изображен на рис. 2.

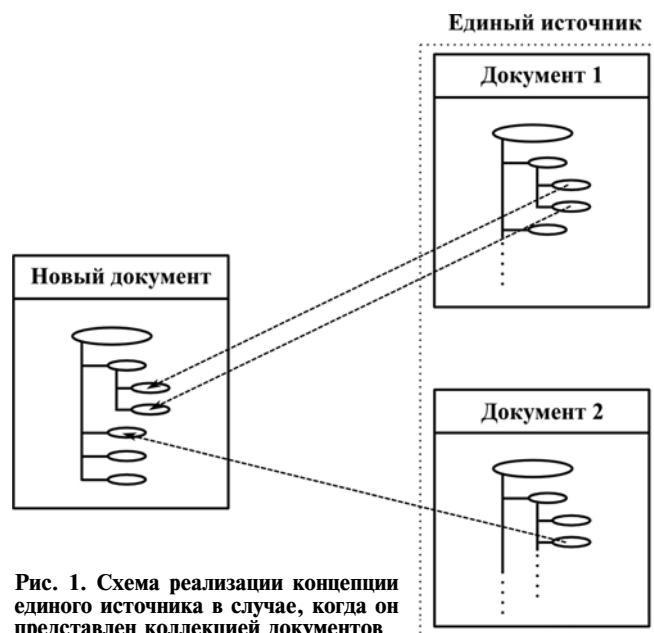


Рис. 1. Схема реализации концепции единого источника в случае, когда он представлен коллекцией документов

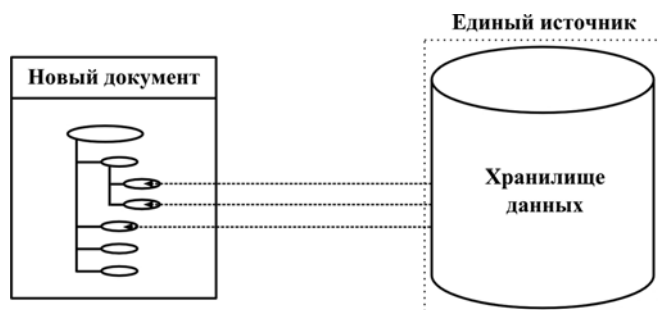


Рис. 2. Схема реализации концепции единого источника, когда он представлен в виде хранилища данных

Применение хранилища данных в качестве основы для системы единого источника характеризуется рядом преимуществ. Последнее объясняется тем, что хранилище данных реализуется посредством применения современной системы управления базами данных (СУБД), которая обеспечивает высокий уровень защищенности данных, возможность эффективной многопользовательской работы, встроенные средства резервирования данных и т. д.

Существенным недостатком данного подхода является необходимость извлечения фактографических данных из электронных документов для дальнейшего размещения в хранилище данных. При извлечении фактографических данных происходит их отрыв от первоисточника (документа), что может привести к частичной потере этих данных или появлению ошибок.

Перспективным направлением исследования является реализация концепции единого источника, совмещающая преимущества обоих рассмотренных способов.

### Практическая реализация

В состав технической документации входят документы, содержание которых определено согласно требованиям ГОСТ или нормативно-технических документов. Таким образом, для описания фактографического контента документов единого вида можно использовать квазиструктурированную модель информационного наполнения [3]:

$$S = \langle root, sObj, LObj, minOccurs, maxOccurs, sMet, Obj\_smet \rangle, \quad (1)$$

где  $root$  — корневой объект,  $root \in sObj$ ;  $sObj$  — конечное множество объектов, каждый из которых содержит фрагмент информационного наполнения документа (текст, рисунок и т. д.) или выполняет роль контейнера для одного или нескольких объектов.

Для объектов-контейнеров доступны следующие метасвойства:  $smet_c$  — определяет объект в качестве контейнера;  $mixed$  — разрешает использование объектов-потомков в произвольном порядке.  $LObj$  — отображение, определенное на множестве  $sObj$ , такое что  $sObj \xrightarrow{LObj} \{obj_1, \dots, obj_n\}$ , где  $obj_i \in sObj$  — дочерний объект;  $n$  — число дочерних объектов;  $Obj\_smet$  —

отображение, определенное на множестве  $sObj$ , такое что  $sObj \xrightarrow{Obj\_smet} \{smet_c | smet_c, mixed | smet_1, \dots, smet_k\}$ , где  $smet_i \in sMet$  — метасвойство ограничения на содержимое объекта;  $minOccurs$  — функция, определяющая минимально возможное число раз использования объекта в модели;  $maxOccurs$  — функция, определяющая максимально возможное число раз использования объекта в модели.

Рассмотрим графическое представление электронного документа (рис. 3). Документ, представленный на рис. 3, состоит из пяти объектов. Объект  $A$  выполняет роль контейнера для объектов  $B$  и  $C$ ; объект  $B$  выполняет роль контейнера для объектов  $D$  и  $E$ . Объекты  $A, B, D$  обязательно должны быть использованы при разработке документа, объект  $C$  является необязательным к использованию, объект  $E$  в рассматриваемом примере должен быть использован от 3 до 5 раз. Объекту-контейнеру  $A$  соответствует метасвойство ограничения  $smet_c$ , а для объекта-контейнера  $B$  определено дополнительно метасвойство  $mixed$ . Объект  $C$  представлен числовым наполнением, т. е. ему соответствует метасвойство ограничения  $smet_2$ . Объекты  $D, E$  имеют символьное информационное наполнение, которому соответствует метасвойство ограничения  $smet_1$ .

Документ может быть представлен с помощью модели следующим образом:

$$\begin{aligned} root &= \{A\}; sObj = \{A, B, C, D, E\}; \\ LObj(A) &= \{B, C\}, LObj(B) = \{D, E\}, \\ LObj(C) &= \{\}, LObj(D) = \{\}, LObj(E) = \{\}; \\ Obj\_smet(A) &= \{smet_c\}, Obj\_smet(B) = \{smet_c, mixed\}, \\ Obj\_smet(C) &= \{smet_2\}, \\ Obj\_smet(D) &= \{smet_1\}, Obj\_smet(E) = \{smet_1\}; \\ minOccurs(A) &= 1, maxOccurs(A) = 1; \\ minOccurs(B) &= 1, maxOccurs(B) = 1; \\ minOccurs(C) &= 0, maxOccurs(C) = 1; \\ minOccurs(D) &= 1, maxOccurs(D) = 1; \\ minOccurs(E) &= 3, maxOccurs(E) = 5. \end{aligned}$$

В качестве примера рассмотрим модель для документов "Руководство пользователей АРМ" (АРМ — автоматизированное рабочее место) из пакета документов рабочей документации [3]. Структура данного документа разрабатывается в соответствии с ГОСТ 19.505—79 "Руководство оператора. Требования к содержанию и оформлению".

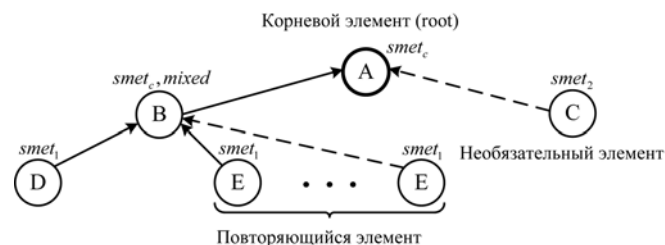


Рис. 3. Граф информационного наполнения документа

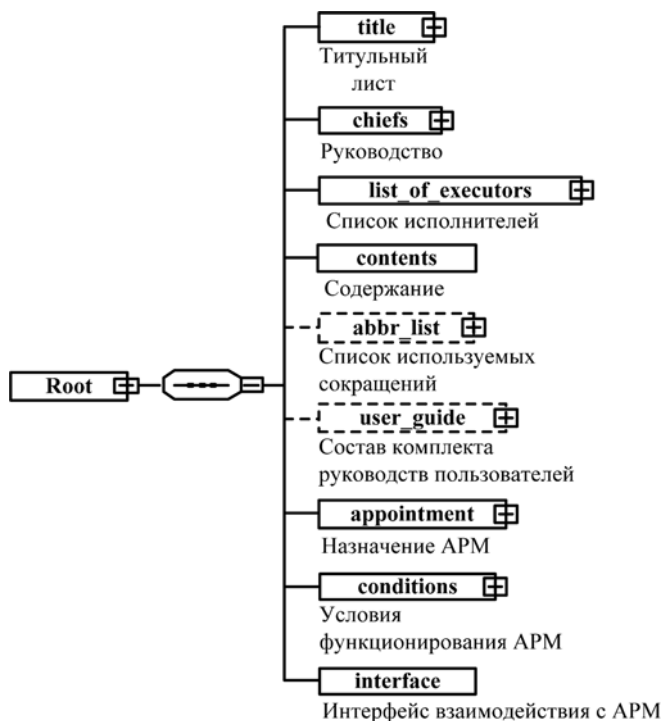


Рис. 4. Базовый сегмент схемы

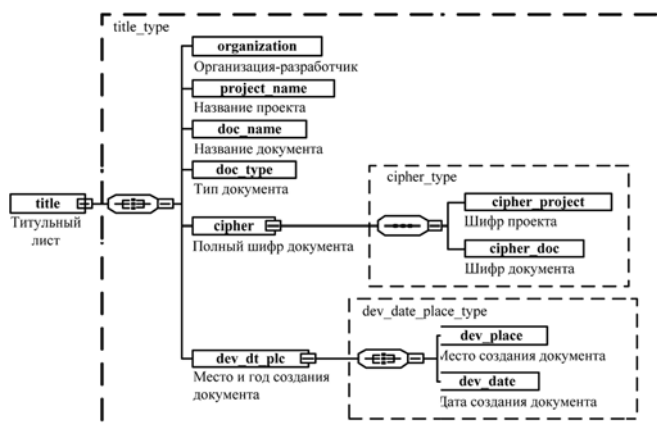


Рис. 5. Сегмент "Титульный лист"

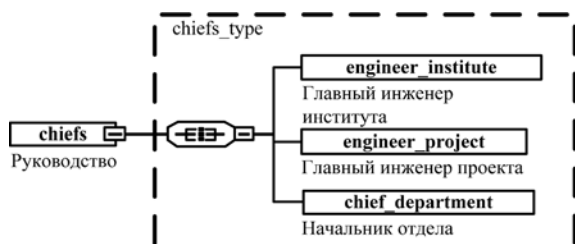


Рис. 6. Сегмент "Второй титульный лист"

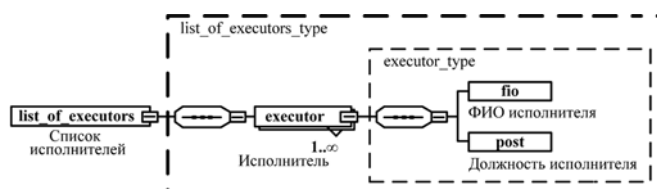


Рис. 7. Сегмент "Список исполнителей документа"

Схема документа "Руководство пользователей АРМ" состоит из сегментов (рис. 4).

Каждый сегмент отвечает за описание раздела документа, а он, в свою очередь, подразделяется на более мелкие сегменты и атрибуты. Сегменты "abbr\_list" и "user\_guide" показаны на схеме штриховой линией. При описании документа эти элементы являются необязательными и могут не присутствовать в документе.

Рассмотрим подробнее структуру модели. Сегмент "title" описывает информационное наполнение титульного листа документа (рис. 5), которое состоит из атрибутов: "organization" — организация-разработчик; "project\_name" — название проекта; "doc\_name" — название документа; "doc\_type" — тип документа; и двух сегментов: сегмента "cipher" — полный шифр документа, состоящего из двух атрибутов: "cipher\_project" — шифр проекта и "cipher\_doc" — шифр документа, и сегмента "dev\_dt\_plc" — место и год создания документа, в котором определены два атрибута: "dev\_place" и "dev\_date" — место создания документа и дата создания документа соответственно.

На втором листе документа размещена информация о руководителях проектной организации, которая описана с помощью сегмента "chiefs" (рис. 6). Этот сегмент включает в себя следующие атрибуты: "engineer\_institute" — главный инженер института, "engineer\_project" — главный инженер проекта, "chief\_department" — начальник отдела. Каждый из перечисленных атрибутов будет содержать ФИО соответствующего руководителя.

Далее использован сегмент "list\_of\_executors" — список исполнителей документа (рис. 7). Этот сегмент состоит из списка авторов документа, для каждого из которых определены его ФИО и занимаемая им должность. Число исполнителей документа неограниченно.

Содержание документа описано с помощью атрибута "contents", который представляет собой блок неограниченного размера.

Сегмент "abbr\_list" позволяет описать сокращения, использованные в документе. Структура этого сегмента представлена на рис. 8. Число использованных аббревиатур в документе неограниченно. Каждое сокращение определено с помощью двух атрибутов: "abbr\_name" — аббревиатура и "abbr\_decryption" — расшифровка аббревиатуры.

Состав комплекта руководств пользователей описан с помощью сегмента "user\_guide" (рис. 9). Число руководств комплекта неограниченно. Для каждого

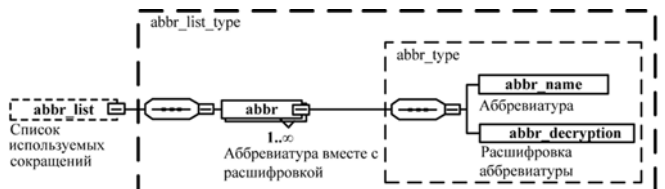


Рис. 8. Сегмент "Список использованных сокращений"

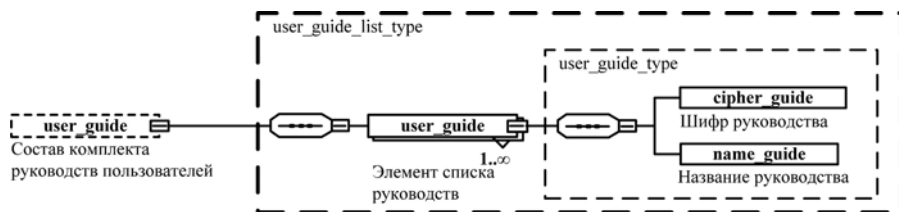


Рис. 9. Сегмент "Состав комплекта руководств пользователей"

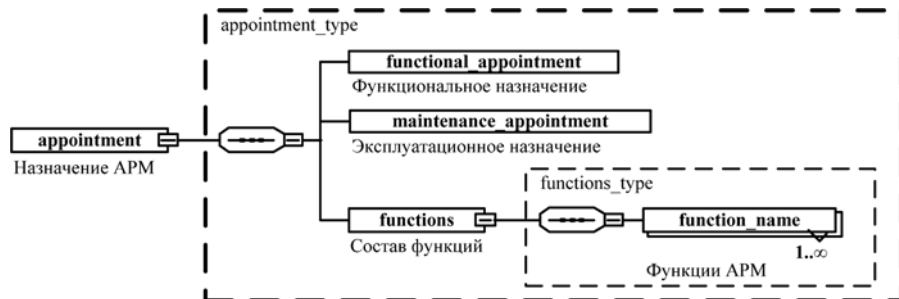


Рис. 10. Сегмент "Назначение АРМ"

руководства определены следующие атрибуты: "cipher\_guide" — шифр руководства; "name\_guide" — название руководства.

Для описания функционального и эксплуатационного назначения АРМ используется сегмент "appointment" (рис. 10).

Для него определены следующие атрибуты: "functional\_appointment" — функциональное назначение; "maintenance\_appointment" — эксплуатационное назначение. А также с помощью атрибута "function\_name" — функции АРМ может быть описано неограниченное число функций, реализуемых АРМом.

Далее в документе определены условия функционирования АРМ. Их описание реализует сегмент "conditions" (рис. 11), который, в свою очередь, состоит из сегментов "min\_system\_requirements" — минимальные системные требования, "adv\_system\_requirements" — рекомендуемые системные требо-

вания и атрибутов "conditions\_operator" — условия эксплуатации АРМ, "software" — список программного обеспечения (ПО).

Атрибут "conditions\_operator" определяет требования к рабочему месту оператора АРМ. Сегменты "min\_system\_requirements" и "adv\_system\_requirements" состоят из неограниченного числа атрибутов "hardware\_name" — наименование и "hardware\_metrics" — характеристика. Сегмент, формирующий рекомендуемые системные требования, является необязательным и может отсутствовать в документе. Эти сегменты позволяют описать минимальные и рекомендованные требования к компьютеру АРМ. Неограниченная последовательность атрибутов "software" позво-

ляет задать перечень программных средств, требуемых для функционирования АРМ на компьютере.

Атрибут "interface" использован в модели для маркирования разделов, содержащих описание интерфейса взаимодействия с АРМ.

Рассмотренная модель документов "Руководство пользователей АРМ" позволяет описать информационное наполнение документов этого вида и упростить процесс создания документа, так как в этом случае документ разрабатывается в соответствии со структурой и информационным наполнением, определенным моделью.

Процесс синтеза квазиструктурированных моделей фактографического контента документов подробно описан в работе [4].

Используя разработанную модель, можно маркировать фактографическое информационное наполнение непосредственно в самом электронном документе, что позволяет получить доступ к содержи-

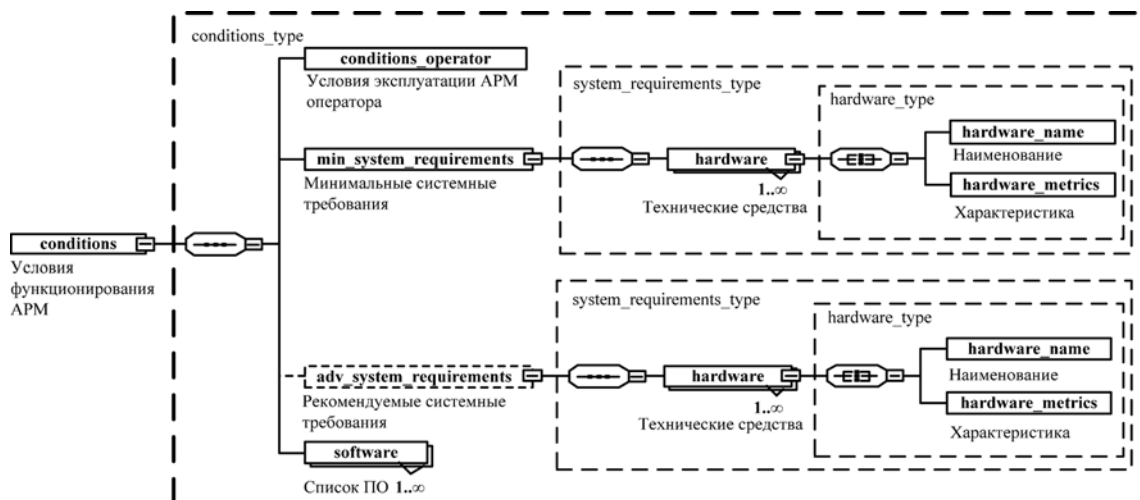


Рис. 11. Сегмент "Условия функционирования АРМ"



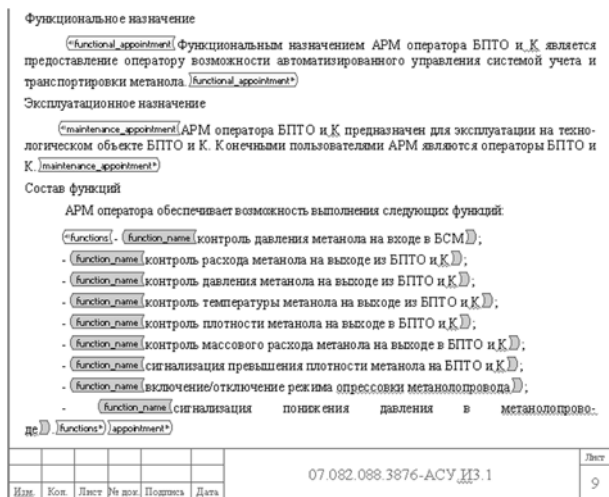


Рис. 12. Фрагмент документа с маркированным информационным наполнением

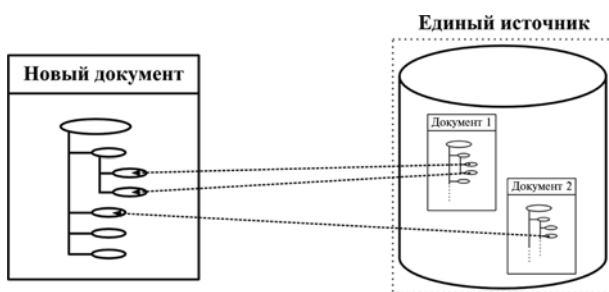


Рис. 13. Схема реализации концепции единого источника в случае применения хранилища данных, ориентированного на работу с квазиструктурированным информационным наполнением документов

тому фрагментов модели при хранении документа непосредственно в хранилище данных (рис. 12).

Таким образом, решение поставленной задачи может быть достигнуто за счет применения хранилища данных, ориентированного на работу с квазиструктурированным информационным наполнением документов. Описанный способ реализации схематично изображен на рис. 13.

## Заключение

Предлагаемая реализация концепции единого источника совмещает в себе преимущества традиционных способов построения систем данного вида и обеспечивает дополнительный контроль корректности фактографического контента документов за счет ограничений, накладываемых моделями документов.

## Список литературы

1. Гуринович В. Д., Просвирнов А. А., Нафталъ М. М., Северин А. Ю., Янченко Ю. А. Принцип единого источника применительно к разработке интерактивных электронных технических руководств. URL: <http://seplm.ru>. (дата обращения: 15.02.2014).
2. Острогорский М. Разработка технической документации на основе единого источника. URL: <http://www.philosoft.ru/ssth-01.zhtml>. (дата обращения: 15.02.2014).
3. Полищук Ю. В., Черных Т. А. Построение хранилищ данных, ориентированных на работу с квазиструктурированным информационным наполнением документов // Информационные технологии. 2010. № 4 (164). С. 9—14.
4. Полищук Ю. В., Черных Т. А. Синтез квазиструктурированных моделей информационного наполнения электронных документов // Вестник компьютерных и информационных технологий. 2012. № 6. С. 20—27.

Yu. V. Polishuk, PhD, Associate Professor of Computer Security mathematical software and information systems, e-mail: [youra\\_polishuk@bk.ru](mailto:youra_polishuk@bk.ru),

T. A. Chernyh, PhD, Associate Professor of Computer Science, Orenburg State University

## On the Methods of Implementing the Concept of a Single Source

The most widespread methods of implementing the concept of a single source are considered. The advantages and disadvantages of these methods are listed. Provides a method of implementing the concept of a single source using a data warehouse based on work with semistructured information content of documents. The mathematical description and graphical presentation of a semistructured content model document are considered. As an example, the model describes the "Manual workstation" from the document package documentation. The structure of this document was developed in accordance with GOST 19.505—79 "Operator's Manual. Requirements for content and design". Proposed in the implementation of the concept of a single source combines the advantages of traditional methods for constructing systems of this type and provides additional control of correctness of factual content of documents due to restrictions imposed by the document models

**Keywords:** single source publishing, semistructured content

### References

1. Gurinovich V. D., Prosvirnov A. A., Naftal' M. M., Severin A. Ju., Janchenko Ju. A. Princip edinogo istochnika primenitel'no k razrabotke interaktivnyh jelektronnyh tehniceskikh rukovodstv. URL: <http://seplm.ru>. (date of treatment: 15.02.2014).
2. Ostrogorskij M. Razrabotka tehniceskoy dokumentacii na osnove edinogo istochnika. URL: <http://www.philosoft.ru/ssth-01.zhtml>. (date of treatment: 15.02.2014).

3. Polishhuk Ju. V., Chernyh T. A. Postroenie hranilishh dannyh, orientirovannyh na rabotu s kvazistrukturovannym informacionnym napolneniem dokumentov // Informacionnye tehnologii, 2010. N. 4 (164). P. 9—14.
4. Polishhuk Ju. V., Chernyh T. A. Sintez kvazistrukturovannyh modelej informacionnogo napolnenija jelektronnyh dokumentov // Vestnik komp'juternyh i informacionnyh tehnologij. 2012. N. 6. P. 20—27.

Л. А. Козлова, ст. преподаватель, e-mail: lkozlova@hse.ru,  
 Н. К. Трубочкина, д-р техн. наук, проф., e-mail: ntrubochkina@hse.ru,  
 НИУ Высшая школа экономики, Москва

## Информационные технологии в английской лингвистике — визуализация правил грамматики

*Описана методика представления и изучения английской грамматики с использованием информационных технологий. Последовательное запоминание больших текстовых блоков, описывающих правила формирования времен, заменяется одномоментным запоминанием простого изображения — визуализации сущностно-временных процессов. Предложен простой язык символов, облегчающий и ускоряющий запоминание правил грамматики английского языка. Сведенная в таблицу систематизированная грамматика английского языка, где каждое время получает свой законченный графический образ, становится легкой для понимания и быстрого запоминания как человеком, так и компьютером.*

**Ключевые слова:** междисциплинарное исследование, информационные технологии, грамматика английского языка, визуализация правил грамматики, сущностно-временные процессы, словарь и язык символов, сжатие объема информации

### Введение

При изучении английского языка (как иностранного) очень важно правильно подобрать методику. Их существует несколько [1—6]. Наиболее популярными являются следующие:

- традиционная, она опирается на запоминание грамматических правил, исключений, ключевых слов, предполагает обязательное составление историй и диалогов. Эту методику можно считать фундаментальной, так как она дает качественное знание, но изучение английского языка с помощью такого подхода процесс длительный;
- альтернативная, она опирается на ролевые игры с элементарным пониманием происходящего. Недостатком этой методики является то, что полученные знания практически не используются на практике, так как они поверхностны. Эта методика допустима в том случае, если мотивация изучения английского языка — это простое общение, и времени на изучение языка мало. О качественном знании английской грамматики и языка здесь говорить не приходится.

Поэтому актуальной стала задача создания методики ускоренного и облегченного изучения грамматики английского языка без снижения качества знаний. Как ни странно, решение нашлось на пересечении английской лингвистики и информационных технологий, а точнее, более узкого их раздела — мультимедийных (графика + текст) и компьютерных (анализ объема информации) технологий.

### Последовательное и параллельное восприятие информации

Человек по-разному воспринимает информацию. При первом прочтении правил формирования какого-либо времени английской грамматики, страницу текста человек читает последовательно слово

за словом, чтобы лучше разобраться в смысле. На это тратится несколько минут (последовательный ввод информации). При необходимости воспроизведения этого знания на практике человек каждый раз как бы считывает эту информационную форму заново, так же последовательно. Чем больше практика, тем быстрее воспроизведение, но все равно процесс "вывода" информации остается последовательным, т. е. медленным. Если человек смотрит на картинку, он воспринимает ее сразу (параллельный ввод информации, все пиксели сразу). И точно так же, при воспроизведении знания, мозг одномоментно будет "выдавать" эту картинку. Из чего следует, что традиционная методика изучения английского языка может быть оптимизирована по времени с сохранением качества, если заменить легко формализуемые текстовые правила формирования времен картинками. Если картинки будут еще и бинарными (в двоичной, компьютерной системе исчисления, по сути, текст в символьном языке), то их не только мозг человека будет воспринимать быстрее, но и возможной станет предельная формализация языка на компьютере.

### Обзор символьных языков

Промежуточная форма между текстом и графикой — это символы и символьные языки. В истории известно большое число символьных языков: древне-шумерские [7, 8], древне-египетские [9, 10], китайские. В них каждому символу предполагается соотнесение объекта или действия (рис. 1).

Так как объекты и действия конкретны и их огромное количество, то для формального кодирования (например, для представления в компьютерной памяти) потребуются большие объемы битовой информации. В современной науке также используется визуализация объектов и процессов. Так,

| Древне-шумерские  | Древне-египетские   | Китайские   |
|---|---|---|
|  Глаз    |  Видеть  |  Глаз    |
|  Лес     |  Вода    |  Вода    |
|  Горы    |  Города  |  Гора    |
|  Факел   |  Огонь   |  Огонь   |
|  Человек |  Мужчины |  Человек |
|   |  Женщины |  Женщина |

Рис. 1. Символы древне-шумерского, древне-египетского и китайского языков

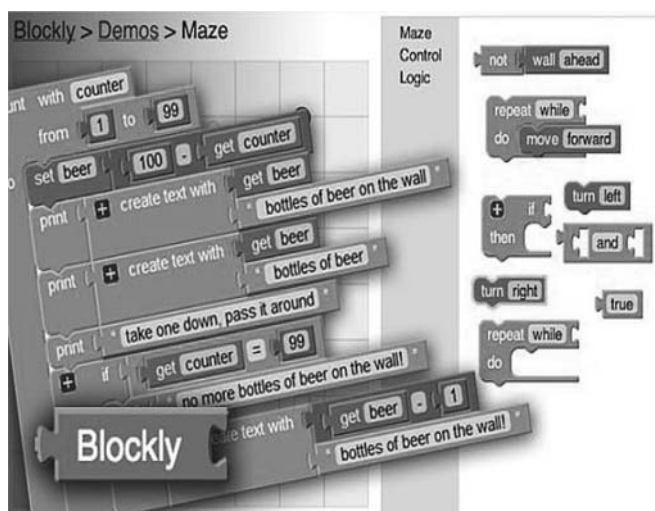


Рис. 2. Визуальный язык программирования Blockly

например, в новом визуальном языке программирования с веб-интерфейсом Blockly [11] программа создается путем соединения блоков (рис. 2). Один из главных разработчиков языка Нил Фрейзер (*Neil Fraser*) считает, что главная цель создания такого языка: "...Это попытка подняться на еще более высокий уровень абстракции, чтобы лишить программирование кастовости и закрытости, сделать его более естественным и доступным для как можно более широкого круга лиц". У этой легкости есть свое скрытое, но вполне понятное объяснение: количество информации для запоминания и воспроизведения таких блочных последовательностей значительно меньше, чем у текстовой информации, где блоком является буква, цифра или текстовый символ, поэтому время на понимание и запоминание "текстов" на формальном визуальном языке уменьшается значительно. В математической логике и информатике формальный язык — это набор конечных слов в конечном алфавите. И если удастся снизить число знаков в алфавите без потери понимания передаваемой информации, то можно получить ускорение восприятия информации компьютером, ускорение и легкость понимания и запоминания информации человеком.

Данная работа возникла на занятиях английского языка при изучении грамматики, где среди слушателей был специалист по информационным технологиям. Эта работа выполнена на стыке информационных технологий, английской лингвистики и методологии образования.

### Словарь графического языка символов для эффективного представления грамматики английского языка

Для облегчения изучения грамматики английского языка предлагается графический язык символов, элементы которого представлены на рис. 3.

На рис. 4—19 представлены графические алгоритмы (визуализация) формирования английских времен и пояснения к ним.

### Графический алгоритм Present Simple (PS)

**Информационное прочтение (определение).** С помощью времени Present Simple (PS) (простое настоящее) описывается событие, происходящее в настоящее время (рис. 4).

**Определение лингвиста.** Это время глагола используется в речи, когда действие обычное, привычное, последовательное, повторяющееся или просто факт. **Способы образования.** Во множественном числе глагол-сказуемое стоит в форме инфинитива без

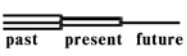
| Элементы графического языка  | Название элемента (символа) | Пояснения  |
|--|-----------------------------|--|
|  past present future | Шкала времени               | Символ «Шкала времени» состоит из трех частей, соответствующих трем временным интервалам: прошлому (past), настоящему (present), и будущему (future) |
|                      | Событие                     | Символ используется для описания события   |
|                      | Процесс                     | Символ используется для описания некоторого процесса (события незавершенного, которое еще продолжается)  |
| [ ]  | Период                      | Символы используются для фиксации периодов времени, в которых происходят события и процессы  |
| 1 2  | Номера событий, процессов   | Нумерация событий, определяющая их последовательность во времени или моменты времени   |
|                     | Связь, ссылка               | Символ используется для связности событий и процессов, например при описании прямой речи   |
| I, II, III   | Формы глаголов              | Символы форм глаголов  |

Рис. 3. Словарь графического языка символов для облегчения изучения грамматики английского языка


| Графический алгоритм   | Характерные слова  | Примеры  |  |
|--|--|--|--|
| <b>Present Simple (PS)</b><br><br>verb<br>do (does) | Every day (month, year, Sunday), from time to time, in the morning (evening, afternoon), usually, always, often, seldom, sometimes, ever | They like cats.<br>Do they like dogs?<br>They do not like animals. | He likes dogs.<br>Does he like cats?<br>He does not like horses. |

Рис. 4. Графический алгоритм Present Simple (PS)

частицы "to" (bare infinitive, т. е. "голый" инфинитив). С местоимением "I" глаголы используются в форме множественного числа. В *единственном числе* к глаголу-сказуемому присоединяется суффикс единственного числа "s" (для глаголов, оканчивающихся на "o" — "es"). В вопросительных и отрицательных предложениях появляется вспомогательный глагол "do" (для множественного числа) и "does" (для единственного числа). В отрицательных предложениях используется как полная форма "do not", "does not", так и краткая форма "don't" и "doesn't".

### Графический алгоритм Past Simple (PsS)

**Информационное прочтение (определение).** С помощью времени Past Simple (PsS) (простое прошедшее) описывается событие, происшедшее в прошлом (рис. 5).

**Определение лингвиста.** Это время глагола используется в речи, когда действие последовательное, единичное или регулярно повторяющееся, происходившее некоторое время и закончившееся в прошлом. **Способы образования.** Образуется прибавлением суффикса прошедшего времени "ed" к

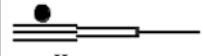
| Графический алгоритм   | Характерные слова   | Примеры   |   |
|--|---|---|---|
| <b>Past Simple (PsS)</b><br><br>II<br>(-ed) | last Sunday (week, month, year), yesterday, the day before yesterday, 2 days ago, in 2014 | They went to work.<br>Did they go to school?<br>They did not go home. | He went home.<br>Did he go to work?<br>He did not go there. |

Рис. 5. Графический алгоритм Past Simple (PsS)

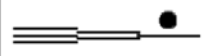
| Графический алгоритм  | Характерные слова  | Примеры  |  |
|---|--|--|--|
| <b>Future Simple (FS)</b><br><br>will (shall)<br>to do | Tomorrow, the day after tomorrow, next week (month, year, Sunday), soon, as soon as possible, in an hour | They will play tennis well. Will we join them?<br>They will not do the work. | He will come soon.<br>Will she have a test?<br>He will not be right. |

Рис. 6. Графический алгоритм Future Simple (FS)


| Графический алгоритм  | Характерные слова  | Примеры  |   |
|---|--|--|---|
| <b>Present Continuous (PC)</b><br><br>am/are/is+<br>(-ing) | Now, at the moment, at the present moment, right now, still. Look! Listen! | They are planting trees.<br>Are they planting bushes? Why are we quarreling?<br>We are not quarreling. | He is singing.<br>He is not dancing.<br>Is he singing well? |

Рис. 7. Графический алгоритм Present Continuous (PC)

правильным глаголам или использованием индивидуальной формы прошедшего времени неправильного глагола. В вопросительных и отрицательных предложениях появляется вспомогательный глагол прошедшего времени "did" ("did not"), за которым следует глагол-сказуемое в форме "bare infinitive".

### Графический алгоритм Future Simple (FS)

**Информационное прочтение (определение).** С помощью времени Future Simple (FS) (простое будущее) описывается событие, которое будет происходить в будущем (рис. 6).

**Определение лингвиста.** Это время используется в речи, когда действие единичное, повторяющееся или последовательное, произойдет в будущем. **Способы образования.** Образуется с помощью вспомогательного глагола будущего времени "will" (очень редко в настоящее время "shall"), за которым следует "bare infinitive". В отрицательном предложении используется "will not" ("shall not").

### Графический алгоритм Present Continuous (PC)

**Информационное прочтение (определение).** С помощью времени Present Continuous (PC) (настоящее продолженное) описывается процесс (незавершенное событие), которое происходит в настоящем (рис. 7).

**Определение лингвиста.** Это время глагола используется в речи, когда действие — процесс, происходящий в данный момент, в момент речи. **Способы образования.** Образуется с помощью глагола "to be" в соответствующей форме "am, is, are" и Причастия I (смыслового глагола с суффиксом "ing"). В отрицательном предложении отрицательная частица "not" ставится после глагола "to be" (am not, isn't, aren't).

### Графический алгоритм Past Continuous (PsC)

**Информационное прочтение (определение).** С помощью времени Past Continuous (PsC) (прошедшее продолженное) описывается процесс (незавершенное событие), которое происходило в прошлом (рис. 8).

**Определение лингвиста.** Это время глагола используется в речи, если действие происходило в определенный момент или период в прошлом. **Способы образования.** Образуется с помощью формы прошедшего времени глагола "to be" ("was, were") и Причастия I

(смыслового глагола с суффиксом "ing"). В отрицательном предложении отрицательная частица ставится после глагола "to be" ("was not, were not").

### Графический алгоритм Future Continuous (FC)

**Информационное прочтение (определение).** С помощью времени Future Continuous (FC) (будущее продолженное) описывается процесс (незавершенное событие), которое будет происходить в будущем (рис. 9).

**Определение лингвиста.** Это время глагола используется в речи, если действие будет происходить в определенный момент или период времени в будущем. **Способы образования.** Образуется с помощью вспомогательного глагола будущего времени "will", следующего за ним глагола "be" и Причастия I (смыслового глагола с суффиксом "ing"). В отрицательном предложении отрицательная частица "not" ставится после глагола "will".

### Графический алгоритм Present Perfect (PPf)

**Информационное прочтение (определение).** С помощью времени Present Perfect (PPf) (настоящее совершенное) описывается событие, которое, начавшись в прошлом, завершилось к некоторому моменту в настоящем (рис. 10).

**Определение лингвиста.** Это время глагола используется в речи, когда: а) действие совершено (закончено) в неопределенном прошлом, и результат которого мы сейчас обсуждаем; б) действие закончено, а период времени еще нет (сегодня, на этой неделе, в этом году, всю жизнь). **Способы образования.** Образуется с помощью глагола have (has) плюс правильный глагол с окончанием "— ed" или III форма неправильного глагола. В отрицательных предложениях отрицательная частица "not" стоит сразу после глагола "have" (have not, has not).

### Графический алгоритм Past Perfect (PsPf)

**Информационное прочтение (определение).** С помощью времени Past Perfect (PsPf) (прошедшее совершенное) описывается событие, которое, начавшись в прошлом (1), завершилось к некоторому моменту в прошлом (2) (рис. 11).

**Определение лингвиста.** Это время глагола используется в речи, когда: а) действие произошло раньше другого действия или момента в прошлом;

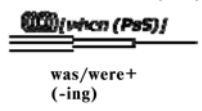
| Графический алгоритм  | Характерные слова  | Примеры  |   |
|---|--|--|---|
|  <p>was/were+ (-ing)</p> | at 5 o'clock yesterday, from 5 till 7 yesterday, while, ...when he came, this time last summer, all day (night) long | They were working.<br>Were they playing?<br>Why were they working?<br>They were not playing. | He was driving.<br>Was he playing?<br>He was not playing. |
|   |  | I was sleeping when someone knocked at the door].  |   |

Рис. 8. Графический алгоритм Past Continuous (PsC)

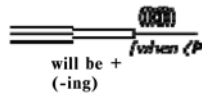
| Графический алгоритм  | Характерные слова   | Примеры  |  |
|---|---|--|--|
|  <p>will be + (-ing)</p> | at 5 o'clock tomorrow, from 5 till 7 next Sunday, all Sunday long, ...when he comes | They will be playing.<br>Will they be singing?<br>Why will they be playing tennis?<br>They will not be playing football. | He will be sleeping.<br>Will he be working?<br>He will not be working.<br>Where will he be sleeping? |
|   |   | I will be sleeping when my mother comes.   |  |

Рис. 9. Графический алгоритм Future Continuous (FC)

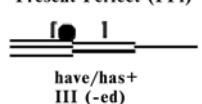
| Графический алгоритм   | Характерные слова   | Примеры   |   |
|--|---|---|---|
|  <p>have/has+ III (-ed)</p> | Already, yet, ever, never, just, recently, lately, before, so far, since, for, three times, It is the first time... | They have done it.<br>Have they come yet?<br>They have not heard from him recently. | He has just left.<br>Has he told you?<br>He has not informed us about it yet. |

Рис. 10. Графический алгоритм Present Perfect (PPf)

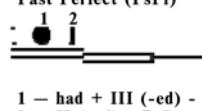
| Графический алгоритм   | Характерные слова | Примеры  |  |
|--|-------------------|--|--|
|  <p>1 — had + III (-ed) - Ps<br/>2 — II (-ed) - PsS</p> | by, after, before | I had finished writing the novel by my birthday. We learned about that after he rang us up.<br>He hadn't heard from him for a long time until he saw him at the door. Had he come before she left?<br>Had you done the housework by the time he arrived? |  |

Рис. 11. Графический алгоритм Past Perfect (PsPf)

б) когда действие завершилось к ("by") какому-то моменту в прошлом. **Способы образования.** Образуется с помощью глагола "had" и следующего за ним правильного глагола в прошедшем времени (с суффиксом "ed") или III формы неправильного глагола. В отрицательных предложениях отрицательная частица "not" стоит сразу за глаголом "had".

### Графический алгоритм Future Perfect (FPf)

**Информационное прочтение (определение).** С помощью времени Future Perfect (FPf) (будущее совершенное) описывается событие, которое, начавшись в будущем (1), завершится к некоторому моменту в будущем (2) (рис. 12).

**Определение лингвиста.** Это время глагола используется в речи, когда действие закончится к определенному моменту в будущем. **Способы образо-**

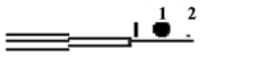
| Графический алгоритм  | Характерные слова  | Примеры  |
|---|--|--|
| <b>Future Perfect (FPf)</b><br><br>1 – shall/will+have+III (-ed)<br>2 – PS | By 5 o'clock tomorrow, by Monday, by that time, by the time he comes | They will have finished cleaning by 5.<br>Will they have discussed it?<br>They will not have stopped when he asked them to.<br>Will you have read all these books by the exam time?<br>I guess, I shall not have received you're my order before Christmas.<br>You will have spent much effort before you can get tickets to this theatre. |

Рис. 12. Графический алгоритм Future Perfect (FPf)

**вания.** Образуется с помощью вспомогательных глаголов будущего времени "will, shall" плюс "have" и следующего за ним смыслового глагола с суффиксом прошедшего времени "ed" или неправильного глагола в III форме. В отрицательных предложениях отрицательная частица "not" стоит сразу после глагола "will" ("shall").

### Графический алгоритм Present Perfect Continuous (PPfC)

**Информационное прочтение (определение).** С помощью времени Present Perfect Continuous (PPfC) (настоящее совершенное длящееся) описывается процесс, который, начавшись в прошлом, продолжается и в настоящем (рис. 13).

**Определение лингвиста.** Действие началось в прошлом, длилось какое-то время и продолжается сейчас или только что закончилось с очевидным результатом. **Способы образования.** Образуется с помощью глагола "have" ("has") плюс глагол "been"

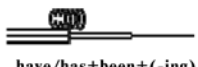
| Графический алгоритм  | Характерные слова    | Примеры   |
|---|----------------------|---|
| <b>Present Perfect Continuous (PPfC)</b><br><br>have/has+been+(-ing) | for, since, how long | They have been watching TV.<br>Have they been sleeping?<br>They have not been sleeping.<br>You smell tobacco. Have you been smoking?<br>Has she been using my computer again?<br>It has not been snowing here since 2000. |

Рис. 13. Графический алгоритм Present Perfect Continuous (PPfC)

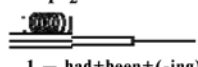
| Графический алгоритм  | Характерные слова                     | Примеры  |
|---|---------------------------------------|--|
| <b>Past Perfect Continuous (PsPfC)</b><br><br>1 – had+been+(-ing)<br>2 – PsS | for, since, all the in time, how long | We had been working.<br>He had not been sleeping!<br>Has she been writing a composition? |

Рис. 14. Графический алгоритм Past Perfect Continuous (PsPfC)

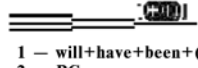
| Графический алгоритм  | Характерные слова        | Примеры  |
|---|--------------------------|--|
| <b>Future Perfect Continuous (FPfC)</b><br><br>1 – will+have+been+(-ing)<br>2 – PS | by, for, how long, since | I will have been living here for a week by next Sunday.<br>I won't have been waiting for you long when you come.<br>Will you have been living here for twenty years next autumn? |

Рис. 15. Графический алгоритм Future Perfect Continuous (FPfC)

и следующего за ним смыслового глагола с окончанием "ing". В отрицательных предложениях отрицательная частица "not" ставится после глагола "have" ("have not, has not")

### Графический алгоритм Past Perfect Continuous (PsPfC)

**Информационное прочтение (определение).** С помощью времени Past Perfect Continuous (PsPfC) (прошлое совершенное длящееся) описывается процесс (2), который, начавшись в прошлом, закончился к некоторому моменту (2) (рис. 14).

**Определение лингвиста.** Это время глагола используется в речи, когда действие началось до определенного момента или действия в прошлом и все еще продолжалось в тот момент или только что закончилось с очевидным результатом. **Способы образования.** Образуется с помощью глагола "had" плюс глагол "been" и следующий за ним смысловой глагол с суффиксом "ing". В отрицательных предложениях частица "not" ставится после глагола "had" (had not).

### Графический алгоритм Future Perfect Continuous (FPfC)

**Информационное прочтение (определение).** С помощью времени Future Perfect Continuous (FPfC) (будущее совершенное длящееся) описывается процесс (1), который начнется в будущем и закончится к некоторому моменту в будущем (2) (рис. 15).

**Определение лингвиста.** Это время глагола используется в речи, когда действие началось в прошлом, длилось какое-то время и будет длиться еще до определенного момента в будущем, включая сам момент. **Способы образования.** Образуется с помощью вспомогательного глагола будущего времени "will" (редко "shall") плюс глагол "have", затем глагол "been" и следующего за ним смыслового глагола с суффиксом "ing". В отрицательных предложениях отрицательная частица "not" ставится после глаголов "will, shall" (will not, shall not).

### Графический алгоритм Simple Future in the Past (FPs)

**Информационное прочтение (определение).** С помощью времени Simple Future in the Past (FPs) (простое будущее в прошедшем) описывается событие в прошлом (1), которое ссылается на событие (2) в будущем (прямая речь) (рис. 16).

**Определение лингвиста.** Это время глагола обозначает будущее действие относительно определенного момента в прошлом и используется в речи для выражения простых, обычных действий или последовательности действий в будущем, обозреваемом из прошлого.

**Способы образования.** Образуется с помощью прошедшего времени вспомогательных глаголов "will" (would) и "shall" (should) и следующего за ними инфинитива смыслового глагола без частицы "to". В отрицательных предложениях отрицательная частица "not" стоит сразу после глагола "would" (would not)

### Графический алгоритм Continuous Future in the Past (CFPs)

**Информационное прочтение (определение).** С помощью времени Continuous Future in the Past (CFPs) (будущее продолженное в прошедшем) описывается процесс в прошлом (1), который ссылается на процесс (2) в будущем (рис. 17).

**Определение лингвиста.** Это время глагола обозначает будущее действие относительно определенного момента в прошлом и используется в речи для выражения длительного действия, которое произойдет в определенный момент или период времени в будущем. Оба действия соотносятся с прошлым. **Способы образования.** Образуется с помощью прошедшего времени вспомогательных глаголов "will" (would) и "shall" (should), следующего за ними глагола "be" без частицы "to" и смыслового глагола с суффиксом "ing". В отрицательных предложениях отрицательная частица "not" стоит сразу после глагола "would" (would not).

### Графический алгоритм Perfect Future in the Past (PrFPs)

**Информационное прочтение (определение).** С помощью времени Perfect Future in the Past (PrFPs) (совершенное будущее в прошедшем) описывается событие в прошлом (1), которое ссылается на событие (2) в будущем, которое закончится к некоторому моменту времени в будущем (by) (рис. 18).

**Определение лингвиста.** Это время глагола обозначает будущее действие относительно определенного момента в прошлом и используется в речи для выражения действия, которое завершится к определенному моменту в будущем относительно про-

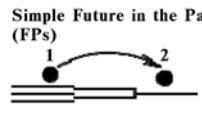
| Графический алгоритм  | Характерные слова  | Примеры   |
|---|--|---|
| <b>Simple Future in the Past (FPs)</b><br><br>1 – PsS<br>2 – would+be verb | Любой глагол в главном предложении, стоящий в прошедшем времени. | He said he would go to the theatre.<br>1                    2<br>She hoped he would help her.<br>1                    2 |

Рис. 16. Графический алгоритм Simple Future in the Past (FPs)

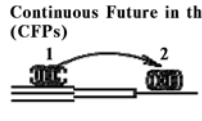
| Графический алгоритм   | Характерные слова  | Примеры  |
|--|--|--|
| <b>Continuous Future in the Past (CFPs)</b><br><br>1 – PsC<br>2 – would+be+(-ing) | Любой глагол в главном предложении, стоящий в прошедшем времени и в придаточном предложении наличие одного из следующих слов: at 5 o'clock that day, when ..., from 5 till 7, all day long last Sunday | He told us how he was driving in the desert when his wife interrupted him. |

Рис. 17. Графический алгоритм Continuous Future in the Past (CFPs)

| Графический алгоритм   | Характерные слова   | Примеры   |
|--|---|---|
| <b>Perfect Future in the Past (PrFPs)</b><br><br>1 – PsS<br>2 – should/would+have+III | Любой глагол в главном предложении, стоящий в прошедшем времени и в придаточном предложении наличие одного из следующих слов: by, when... | He realized that he would have finished his work long before midnight.<br>We thought she wouldn't have come by 5.<br>She said they wouldn't have done the work by the time the boss came. |

Рис. 18. Графический алгоритм Perfect Future in the Past (PrFPs)

шлого. **Способы образования.** Образуется с помощью прошедшего времени вспомогательных глаголов "will" (would) или "shall" (should), следующего за ним глагола "have" плюс глагол "been" и III форма смыслового глагола.

### Графический алгоритм Perfect Continuous Future in the Past (PrCFPs)

**Информационное прочтение (определение).** С помощью времени Perfect Continuous Future in the Past (PrCFPs) (совершенное продолженное будущее в прошедшем) описывается событие в прошлом (1), которое ссылается на процесс (2) в будущем, который закончится к некоторому моменту времени в будущем (by) (рис. 19).

**Определение лингвиста.** Это время глагола обозначает будущее действие относительно определенного момента в прошлом и используется в речи для выражения длительного действия, которое начнется до определенного момента в будущем от-

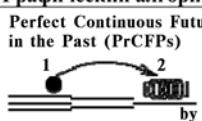
| Графический алгоритм   | Характерные слова  | Примеры  |
|--|--|--|
| <b>Perfect Continuous Future in the Past (PrCFPs)</b><br><br>1 – PsS<br>2 – should/would+have+been+(-ing) | Любой глагол в главном предложении, стоящий в прошедшем времени и в придаточном предложении наличие одного из следующих слов: for. | He said that by the time he picked us up he would have been driving for five hours.<br>He said he would have been living here for ten years next year. |

Рис. 19. Графический алгоритм Perfect Continuous Future in the Past (PrCFPs)

## Графический язык символов в грамматике английского языка

|  |   |   |
|--|---|---|
| <p><b>Present Simple (PS)</b><br/> <br/>         every day, morning, always, often, never, sometime<br/>         do they play? Does he play?<br/>         They do not play. He does not play.</p>  | <p><b>Past Simple (PsS)</b><br/> <br/>         five days ago, last year, yesterday, in 2012<br/>         Did they play? Did he play?<br/>         They did not play. He did not play.</p>   | <p><b>Future Simple (FS)</b><br/> <br/>         tomorrow, next year, in five years, in 2035<br/>         Will they play? Will he play?<br/>         They will not play. He will not play.</p>   |
| <p><b>Present Continuous (PC)</b><br/> <br/>         now, at the moment, Look!<br/>         They are playing. He is playing.<br/>         Are they playing? Is he playing?<br/>         They are not playing. He is not playing.</p>   | <p><b>Past Continuous (PsC)</b><br/> <br/>         yesterday at 5 o'clock, when you called, when rain started, all day long<br/>         They were playing. He was playing.<br/>         Were they playing? Was he playing?<br/>         Why were they playing?<br/>         They were not playing. He was not playing.</p>   | <p><b>Future Continuous (FC)</b><br/> <br/>         tomorrow at 4 o'clock, when we meet, from 6 till 7, next Sunday, when my mother come<br/>         They will be playing. He will be playing.<br/>         Will they be playing? Will he be playing?<br/>         Why will they be playing?<br/>         They will not be playing. He will not be playing.</p>  |
| <p><b>Present Perfect (PP)</b><br/> <br/>         3 times (past), this week, since childhood, for 45 years, already, just, yet, ever, never, all my life<br/>         They have played. He has played.<br/>         Have they played? Has he played?<br/>         They have not played. He has not played.</p>                 | <p><b>Past Perfect (PsP)</b><br/> <br/>         after the Sun had set, you brushed your teeth before you went to bed?<br/>         I finally caught Lucky and looked around. The nasty dog had scratched the furniture, had torn the wallpapers and had eaten my lunch on the table.<br/>         Had you brushed your teeth before you went to bed?<br/>         I finally caught Lucky and looked around. The nasty dog had scratched the furniture, had torn the wallpapers and had eaten my lunch on the table.</p> | <p><b>Future Perfect (FP)</b><br/> <br/>         by before<br/>         They will have played. Will they have played?<br/>         They will not have played.<br/>         Will you read all these books by the exam time? I guess, I shall not have received your next letter before Christmas.<br/>         You will have spent much effort before you can run a marathon.</p>  |
| <p><b>Present Perfect Continuous (PPC)</b><br/> <br/>         for, since<br/>         They have been playing. Have they been playing?<br/>         They have not been playing.<br/>         I smell tobacco. Have you been smoking? Has she been using my car again?<br/>         It has not been snowing here since 1993.</p> | <p><b>Past Perfect Continuous (PsPC)</b><br/> <br/>         for, since, all the week, in time<br/>         He had not been washing like a horse all the week. We had been working hard and managed to finish the project in time.<br/>         Her voice started to tremble. I did not realize what a hard time she had been having.</p>  | <p><b>Future Perfect Continuous (FPC)</b><br/> <br/>         by<br/>         They will have been playing. Will they have been playing?<br/>         They will not have been playing.<br/>         I hear, you are going to retire next month? For how many years will you have been working here by that time?<br/>         By 2015 you will not have been living in this country long enough to participate in the election.</p> |

|  |   |   |
|--|---|---|
| <p><b>Simple Future in the Past (FPs)</b><br/> <br/>         1 - PsS, 2 - would+do verb<br/>         said (that), told (that), thought (that)<br/>         He said he would go to the dentist.</p>   | <p><b>Perfect Future in the Past (PrFPs)</b><br/> <br/>         1 - PsS, 2 - should/would+have+III<br/>         said (that), told (that), thought (that), how, by<br/>         We hoped we should have done our homework by midday.</p>   | <p><b>Perfect Future in the Past (PrFPs)</b><br/> <br/>         1 - PsS, 2 - should/would+have+III<br/>         said (that), told (that), thought (that), how, by<br/>         We hoped we should have done our homework by midday.</p>   |
| <p><b>Continuous Future in the Past (CFPs)</b><br/> <br/>         1 - PsC, 2 - would+be+(-ing)<br/>         said (that), told (that), thought (that), how<br/>         He was planning how he would be sipping cocktail on his vacation.</p> | <p><b>Perfect Continuous Future in the Past (PrCFPs)</b><br/> <br/>         1 - PsC, 2 - should/would+have+been+(-ing)<br/>         said (that), told (that), thought (that), how, by<br/>         He told us he would have been working on the plant for 30 years next December.</p> | <p><b>Perfect Continuous Future in the Past (PrCFPs)</b><br/> <br/>         1 - PsC, 2 - should/would+have+been+(-ing)<br/>         said (that), told (that), thought (that), how, by<br/>         He told us he would have been working on the plant for 30 years next December.</p> |



Рис. 20. Система символического представления синтеза грамматики английского языка



носителем прошлого и продлится до этого момента или все еще будет продолжаться в момент. Данная форма употребляется редко даже в книжном стиле. **Способы образования.** Образуется с помощью прошедшего времени вспомогательных глаголов "will" (would) или "shall" (should), следующего за ним глагола "have" плюс глагол "been" и смыслового глагола с суффиксом "ing".

Все графические алгоритмы формирования английских времен сведены в одну таблицу, изображенную на рис. 20.

### Заключение

Разработана методика изучения английской грамматики для неанглоязычных слушателей с использованием информационных технологий.

Разработаны словарь языка символов для изучения английской грамматики и простой язык символов, облегчающий и ускоряющий запоминание правил грамматики английского языка.

Систематизирована визуализация формирования времен английского языка, представленная в виде таблицы.

Разработанная методика изучения английской грамматики для неанглоязычных слушателей была апробирована в группе Intermediate профессорско-

преподавательского состава и показала высокую эффективность запоминания больших объемов информации.

Систематизированная таблица (рис. 20) визуализированных правил может быть рекомендована в качестве дополнительных методических материалов при изучении английской грамматики для неанглоязычных слушателей.

### Список литературы

1. **Соловова Е. Н.** Методика обучения иностранным языкам. Базовый курс: Пособие для студентов и преподавателей вузов. М.: Просвещение, 2006.
2. **Щукин А. Н.** Теория обучения иностранным языкам (лингвистические основы). М.: ВК, 2012. 335 с.
3. <http://www.study.ru/support/lib/note234.html>
4. <http://bigben.su/articles/methods/metodiki-prepodavaniya-angliyskogo-yazika>
5. <http://bigben.su/articles/methods/net-i-bit-ne-mozhet>
6. <http://bigben.su/articles/methods/metodi-izucheniya-inostrannogo-yazika.-ekskurs-v-metodologiyu>
7. **Канева И. Т., Козлова Н. В., Визирова Е. Ю.** Шумерский язык // Языки мира. Древние реликтовые языки Передней Азии. М.: Academia, 2010. С. 20–21.
8. **Thomsen M.-L.** The Sumerian Language. Copenhagen: Akademisk Forlag, 1984. P. 59–63.
9. **Gardiner A. H.** Egyptian Grammar. Being an Introduction to the Study of Hieroglyphs. Oxford, 3rd ed., 1957.
10. **Allen J. P.** Middle Egyptian: An Introduction to the Language and Culture of Hieroglyphs. Cambridge, 1999.
11. <https://code.google.com/p/blockly/>

**L. A. Kozlova**, Senior Lecturer, **N. K. Trubochkina**, Professor, e-mail: ntrubochkina@hse.ru, Higher School of Economics, Moscow

## Information Technology in English Linguistics — Visualization of Grammar Rules

*The article describes the method of presenting and studying English grammar using information technology. Sequential memorizing of large blocks of text describing the rules of verb tense formation is replaced by remembering a simple image, ie visualization of grammar rules.*

*The authors presented a dictionary, as well as the language of symbols, that facilitate and accelerate the memorization of English grammar rules. The "graphic" information is in the table, where each tense gets its finished graphic image and becomes easy to understand and memorize quickly. The method of studying English grammar for non-English speaking students was developed and tested in an Intermediate level group. The results showed high efficiency of storing large amounts of information. The table of integrated visualized rules can be recommended as a supplementary teaching material in the course of English grammar for non-English speaking students. Further development of the approach of optimal visualization of text blocks in English will improve the computer translation system by reducing time in addition to the educational effect.*

**Keywords:** interdisciplinary study, information technology, English grammar, grammar rules, visualization, essentially-temporal processes, vocabulary and language of symbols, compression of information

### References

1. **Solovova E. N.** Metodika obucheniya inostrannym yazykam. Bazovyy kurs. Posobie dlya studentov i prepodavatelej vuzov. Moskva: Prosveshchenie, 2006.
2. **Schukin A. N.** Teoriya obucheniya inostrannym yazykam (lingvodidakticheskie osnovy). Moskva: VK, 2012. 335 p.
3. <http://www.study.ru/support/lib/note234.html>
4. <http://bigben.su/articles/methods/metodiki-prepodavaniya-angliyskogo-yazika>
5. <http://bigben.su/articles/methods/net-i-bit-ne-mozhet>
6. <http://bigben.su/articles/methods/metodi-izucheniya-inostrannogo-yazika.-ekskurs-v-metodologiyu>
7. **Kaneva I. T., Kozlova N. V., Vizirova E. Yu.** Shumerskiy yazyk // Yazyki mira. Drevnie reliktovye yazyki Perednej Azii. M.: Academia, 2010. P. 20–21.
8. **Thomsen M.-L.** The Sumerian Language. Copenhagen: Akademisk Forlag, 1984. P. 59–63.
9. **Gardiner A. H.** Egyptian Grammar. Being an Introduction to the Study of Hieroglyphs. Oxford, 3rd ed., 1957.
10. **Allen J. P.** Middle Egyptian: An Introduction to the Language and Culture of Hieroglyphs. Cambridge, 1999.
11. <https://code.google.com/p/blockly/>

УДК 378.1 + 311

**Б. Д. Залещанский**, д-р экон. наук, проф., Московский государственный технический университет радиотехники, электроники и автоматики (МГТУ МИРЭА),  
**А. П. Свиридов**, д-р техн. наук, проф., вице-президент, e-mail: prof\_sviridov@mail.ru, Международная академия информатизации,  
**О. А. Павлова**, аспирант, **Е. А. Шалобина**, аспирант,  
Московский государственный технический университет радиотехники, электроники и автоматики (МГТУ МИРЭА)

## Вероятностно-статистические стратегии обеспечения качества подготовки персонала социо-технических систем путем оптимизации полных и частичных проверок

*Рассматриваются периодические стратегии управления качеством профессиональной подготовки обучаемых (персонала) с учетом интенсивности забывания и экономических показателей (затрат на обучение, контроль и др.), основанные на оптимальном планировании во времени полных и частичных проверок и последующих восстановлений выявленных забытых знаний. Исходные данные для них: интенсивности забывания, затраты на частичные и полные проверки, минимальная вероятность правильного выполнения задачи, случайно выбранной для проверки. Показана применимость предлагаемых стратегий для оптимизации полных и частичных проверок предприятий, организаций и фирм.*

**Ключевые слова:** стратегии, переподготовка персонала, интенсивности забывания, затраты на полные и частичные проверки

### Введение

Современные предприятия, организации и фирмы являются социо-техническими системами. Эффективность их функционирования определяется не только техническими и программными средствами, но и деятельностью людей в их составе. Для обеспечения надежности технических и программных средств широко используются методы технической диагностики и стратегии проведения профилактических работ с учетом характеристик надежности. Важнейшей проблемой обеспечения эффективности социо-технических систем в настоящее время является управление профессиональной надежностью персонала предприятий, организаций и фирм на основе количественных характеристик, динамики знаний, умений, навыков и компетенций, экономических показателей и др. [1—4]. Особо актуальна эта проблема для процессов и производств с высокой "ценой" ошибки персонала [5]. Следует отметить, что интенсивность отказа персонала значительно превышает интенсивность отказа технических и программных средств. В американских рекомендациях WASH-1400 предполагается, что интенсивность отказа человека по крайней мере в 100 раз больше интенсивности отказа технических средств атомной станции. На немецких атомных

станциях ошибки персонала присутствуют в более чем 60 % всех несчастных случаев [6].

В работе [7] были рассмотрены периодические стратегии обеспечения качества профессиональной подготовки без учета и с учетом рисков недооценки и переоценки знаний при известной функции распределения времени забывания. В продолжение ее в данной статье предлагаются стратегии обеспечения качества профессиональной подготовки на основе оптимизации двух видов проверок (полных и частичных).

Приведем содержание понятий "знание" и "обучаемые". Термин "знание" применяется для обозначения знаний, умений, навыков и компетенций, усвоение которых необходимо для выполнения соответствующих задач и функций [8, 9]. Требования к качеству подготовки могут быть сформулированы в виде знаний, умений, навыков, компетенций и/или задач и функций, подлежащих выполнению. Под "обучаемыми" в данной работе понимаются:

- учащиеся школ, студенты вузов и университетов, слушатели курсов подготовки, переподготовки и повышения квалификации;
- обслуживающий персонал технических и программных средств информационно-телекоммуникационных систем;
- персонал атомных, тепловых и гидроэлектростанций, транспорта;

— операторы химико-технологических и иных процессов и производств (в первую очередь, операторы процессов и производств с "высокой ценной" ошибки);

— личный состав спасательных служб (например, по оказанию первой медицинской и психологической помощи населению и организации взаимодействия);

— военнослужащие;

— отдельные индивиды (население) для подготовки к поведению в случае неблагоприятных событий;

— предприятие или организация, функционирование которых совершенствуется в результате контроля и обучения. При этом предприятия и организации рассматриваются в качестве обучаемых и обучавшихся.

## 1. Проблема оптимизации

На основе количественных характеристик динамики знаний можно оптимизировать управление учебным процессом. При этом проблема оптимизации в общем случае ставится следующим образом. Усвоения, забывания, повторения и восстановления знаний образуют некоторый случайный процесс. Имеется некоторое множество управлений этим процессом. Для каждой комбинации из реализации случайного процесса и соответствующего управления можно определить функционал цели (степень полезности). Этот функционал содержит некоторые известные и неизвестные параметры.

**Примеры известных параметров:** 1) интенсивности усвоения и забывания; 2) различные затраты; 3) риски недооценки и переоценки знаний при (компьютерном) контроле знаний и др.

**Примеры неизвестных параметров:** 1) интервал восстановления знаний, под которым понимается промежуток времени  $a$  до следующего восстановления знаний; 2) объем и глубина повторения или восстановления знаний по учебному материалу и др.

Проблема оптимизации состоит при этом в синтезе стратегии управления случайным процессом, экстремизирующей (максимизирующей или минимизирующей) функционал цели или качества.

## 2. Оптимизация объема и периодичности двух видов контроля (на примере контроля знаний по языкам программирования TurboPascal и С)

Путем проведения регулярного (компьютерного) контроля знаний или проверок выполнения требуемых задач или функций возможно выявление забытых знаний и ошибок при их выполнении. В результате этого с достаточно большой вероятностью в значительной степени можно предотвратить опасные последствия ошибок людей.

Пусть  $N = 1, 2, \dots, n$  — множество знаний (задач, функций), которые должны быть усвоены (выполнены) обучаемым (оператором). Для обеспечения профессиональной готовности целесообразно проведение полных и частичных проверок знаний и выполнение требуемых задач и функций с после-

дующим восстановлением выявленных забытых знаний. В случае полной проверки контролируется усвоение всех знаний и выполнение всех требуемых задач и функций множества  $N$ , а при частичной проверке — усвоение лишь подмножества  $A$  ( $A \subset N$ ). Частичные проверки знаний по сравнению с полными проводят чаще.

Подчеркнем применимость рассматриваемых ниже стратегий не только к отдельным индивидам, но и к целым организациям, предприятиям и фирмам.

Рассмотрим два случая: 1) интенсивности забывания знаний (задач, функций) различны; 2) множество знаний (задач, функций) разделяется на два подмножества  $A$  и  $N \setminus A$  с интенсивностями забывания  $\lambda_1$  и  $\lambda_2$ .

Рассмотрим сначала первый случай. Пусть  $[0, T_0]$  — интервал времени, в котором  $n$  знаний (задач, функций) множества  $N$  должны быть усвоены обучаемым (оператором);  $\lambda_i$  — интенсивность забывания;  $c_i$  — затраты на проверку усвоения  $i$ -го знания ( $i$ -й задачи, функции). Частичные и полные проверки знаний проводят периодически через промежутки времени  $a_1$  и  $a_2$  ( $a_2 > a_1$ ). Ограничимся случаем кратных величин:  $k_1 = a_2/a_1 = 1, 2, \dots$  и  $k_2 = T_0/a_2 = 1, 2, \dots$ .

Необходимо определить значения  $A$ ,  $a_1$  и  $a_2$  по минимуму суммарных затрат на контроль при следующем условии: вероятность усвоения всех знаний (задач, функций) в заданном интервале  $[0, T_0]$  не меньше некоторого установленного значения  $Q_{\min}$ . Сделаем одно замечание относительно определения этой вероятности. В общем случае положения некоторой учебной дисциплины, естественно, зависимы или связаны. Соответственно времена сохранения знаний по этим положениям  $\tau_1, \dots, \tau_n$  — положительно коррелированные случайные величины, для которых выполняется условие

$$P(\tau_1 \geq t_1; \dots; \tau_n \geq t_n) \geq \prod_{i=1}^n P(\tau_i \geq t_i). \quad (1)$$

Таким образом, если оценивается вероятность усвоения знаний (задач, функций) учебной дисциплины в предположении их взаимной независимости, хотя на самом деле они статистически связаны, то полученная оценка будет заниженной по сравнению с действительным значением (оценка снизу).

Задачу оптимизации решим при следующих допущениях:

1) время забывания  $i$ -го знания ( $i$ -й задачи, функции) имеет экспоненциальное распределение  $\text{Exp}(\lambda_i)$ ;

2) время проверки и последующего восстановления забытых знаний пренебрежимо мало по сравнению с  $T$  (мгновенное восстановление знаний);

3) в момент времени  $t = 0$  все знания (задачи, функции) множества  $N$  усвоены обучаемым (оператором);

4) вероятность усвоения  $i$ -го знания ( $i$ -й задачи, функции) после его (ее) восстановления равна единице:  $P(x_i = 1) = 1$ .

В соответствии с принятыми допущениями после каждой полной проверки вероятность  $Q$  усвоения всех  $n$  знаний (задач, функций) оказывается равной единице, т. е. эта вероятность имеет повторяющийся

характер с периодом  $a_2$ . Ее можно определить в виде вероятности усвоения всех  $n$  знаний (задач, функций) перед очередным полным контролем:

$$Q = Q(a_2 - 0) = Q_1 Q_2 \geq Q_{\min}, \quad (2)$$

где  $Q_1 = \prod_{i \in A} \exp(-\lambda_i a_1)$  — вероятность усвоения знаний (задач, функций) множества  $A$ , неповторяемых в течение времени  $a_1$ , и  $Q_2 = \prod_{j \in N \setminus A} \exp(-\lambda_j a_2)$  — вероятность усвоения знаний (задач, функций) подмножества  $N \setminus A$ , неповторяемых в течение времени  $a_2$ . Из соотношения (2) получим

$$\ln Q = -T_0(k_1 \Lambda + (k_1 - 1) \sum_{i \in A} \lambda_i) / k_1 k_2,$$

где  $\Lambda = \sum_{i \in N} \lambda_i$  — интенсивность забывания всех  $n$  знаний (задач, функций).

Полный контроль на интервале от 0 до  $T_0$  проводится  $(k_2 - 1)$  раз, частичный контроль на интервале  $[0, a_2]$  —  $(k_1 - 1)$  раз. Соответственно суммарные затраты определяются выражением

$$C(C_N, k_1, k_2) = (k_2 - 1)C_N + k_2(k_1 - 1) \sum_{i \in A} C_i,$$

где  $C_N = \sum_{i \in N} c_i$  — затраты на полную проверку.

Введем бинарную переменную

$$r_i = \begin{cases} 0, & \text{если } i \in A \\ 1, & \text{если } i \notin A. \end{cases}$$

Тогда можно записать:

$$C(C_N, k_1, k_2) = (k_2 - 1)C_N + k_2(k_1 - 1)(C_N - \sum_{i \in N} c_i r_i);$$

$$\sum_{i \in A} \lambda_i = \sum_{i \in N} (1 - r_i) \lambda_i;$$

$$\sum_{i \in A} c_i = \sum_{i \in N} (1 - r_i) c_i.$$

Задача оптимизации сводится к вычислению набора  $R_0 = (r_{0i}, i \in N) \subset (r_i, i \in N)$ , а также значений  $a_{10} \in (1, 2, \dots)$  и  $a_{20} \in (1, 2, \dots)$ , обеспечивающих минимум суммарных затрат  $C(C_N, k_1, k_2)$ :

$$C(R_0, k_{10}, k_{20}) = \min_{r_i, k_1, k_2} C(C_N, k_1, k_2),$$

или

$$C(R_0, k_{10}, k_{20}) = \min_{k_1, k_2} [(k_2 - 1)C_N + k_2(k_1 - 1)(C_N - \max_{r_i} \sum_{i \in N} c_i r_i)],$$

при выполнении ограничения

$$T_0[k_1 \Lambda - (k_1 - 1) \sum_{i \in N} \lambda_i r_i] / k_1 k_2 \leq -\ln Q_{\min}.$$

Это задача целочисленного программирования. При ее решении целесообразно выделить и предварительно решить частную задачу о "рюкзаке":

$$C(\lambda) = \max_{r_i} \sum_{i \in N} c_i r_i; \quad \sum_{i \in N} \lambda_i r_i \leq \lambda \quad (0 \leq \lambda < \Lambda),$$

решаемую, например, методом динамического программирования.

Рассмотрим второй случай: множество знаний (задач, функций) разделено на два подмножества  $A$  и  $N \setminus A$ . Введем обозначения:  $\lambda_1$  и  $\lambda_2$  — интенсивности забывания знаний (задач, функций) первого и второго подмножества;  $c_1$  и  $c_2$  ( $c_2 > c_1$ ) — затраты на частичный и полный контроль и последующее восстановление знаний (задач, функций) для первого подмножества  $A$  и всего множества  $N$ ;  $a_1$  и  $a_2$  — интервалы между проверками знаний (задач, функций) первого и обоих подмножеств, причем  $a_2 = r a_1$  ( $r > 1$ ).

В момент времени  $t = 0$  обучаемый обладает всеми необходимыми знаниями для выполнения требуемых задач и функций множества  $N$ . Через промежуток времени  $a_1$  непосредственно перед частичной проверкой и восстановлением выявленных забытых знаний вероятность усвоения всех знаний составит  $Q(a_1 - 0) = \exp[-(\lambda_1 + \lambda_2)a_1]$ .

В результате контроля и восстановления выявленных забытых знаний первого подмножества эта вероятность увеличится до  $Q(a_1 + 0) = \exp(-\lambda_2 a_1)$ .

Допустим, что контроль проводится безошибочно. После  $(r - 1)$ -й частичной проверки и последующего восстановления знаний подмножества  $A$  вероятность усвоения знания этого подмножества будет определяться выражением

$$Q((r - 1)a_1 + 0) = \exp(-\lambda_2(r - 1)a_1).$$

Через промежуток времени  $a_2$  непосредственно перед полной проверкой и восстановлением знаний эта вероятность уменьшится до

$$Q(a_2 - 0) = \exp[-\lambda_2(r - 1)a_1] \exp(-\lambda a_1),$$

где  $\lambda = \lambda_1 + \lambda_2$  — интенсивность забывания всех знаний (задач, функций) множества  $N$ .

После проверки и восстановления забытых знаний всего множества  $N$  через интервал времени  $a_2$  весь процесс повторяется. Задача оптимизации состоит в определении таких значений  $a_{10}$  и  $a_{20}$ , при которых вероятность  $Q(a_2 - 0)$  не меньше заданного значения  $Q_{\min}$ :  $\exp[-\lambda_2(r - 1)a_1] \exp(-\lambda a_1) = Q_{\min}$ .

Прологарифмировав это равенство, получим

$$\lambda_1 a_1 + \lambda_2 a_2 = -\ln Q_{\min}, \quad a_1, a_2 > 0. \quad (3)$$

Вычислим суммарные затраты на контроль и восстановление знаний в интервале  $[0, a_2]$ :  $C = r c_1 + c_2$ . Тогда средние удельные затраты в единицу времени (интенсивность затрат) можно записать в виде

$$c = \frac{C}{a_2} = \frac{c_1}{a_1} + \frac{c_2}{a_2}. \quad (4)$$

Контроль, обеспечивающий заданную вероятность усвоения всех  $n$  знаний (задач, функций) множества  $N$  при минимуме средних удельных затрат, назовем оптимальным. Задача оптимизации состоит в определении значений  $a_{10}$  и  $a_{20}$  по минимуму средних удельных затрат (4) при выполнении условия (3).

В случае  $c_1\lambda_1 - c_2\lambda_2 \neq 0$  решение имеет следующий вид:

$$a_{10} = (-\ln Q_{\min})(c_1\lambda_1 - \sqrt{c_1c_2\lambda_1\lambda_2})/[\lambda_1(c_1\lambda_1 - c_2\lambda_2)]; \quad (5)$$

$$a_{20} = -(\ln Q_{\min} + \lambda_1 a_{10})/\lambda_2. \quad (6)$$

В случае  $c_1\lambda_1 - c_2\lambda_2 = 0$  получаем

$$a_{10} = -\ln Q_{\min}/2\lambda_1,$$

а оптимальное значение  $a_{20}$  рассчитывается на основе соотношения (4).

Рассмотрим пример использования второй стратегии.

**Пример.** Оптимизация периодичности полного и частичного контроля по языкам программирования *Turbo Pascal* (*TP*) и *C*.

Приведем основные вопросы (задачи) подмножества **A** для *TP* и *C*.

Для языка *TP*:

N1. Какие типы данных можно объявить в *TP*? Приведите пример программы.

N3. Составьте программу сортировки одномерного массива. Приведите пример программы.

N5. Приведите пример программы (функции) нахождения минимального элемента в массиве.

N7. Каковы различия между процедурой и функцией? Приведите пример программы.

N11. Составьте процедуру графического представления прямоугольника с его заполнением в графическом режиме.

N12. Приведите основные процедуры ввода-вывода, их различия и пример программы их применения.

N16. Что может быть типом компоненты данных? Приведите пример программы объявления данных.

N18. Укажите основные операции с символами в *TP* и любой пример программы их применения.

N19. Укажите не арифметические операции в *TP* по их уменьшающимся приоритетам и любой пример программы их применения.

N31. Приведите любой пример программы декларирования типов данных.

N33. Структурированные типы данных в *TP*.

Для языка *C*:

N1. Дайте определение структурированных данных языка программирования *C*.

N2. Приведите пример программы определения данных в *C*.

N6. Приведите пример функции нахождения наибольшего элемента в массиве определенной длины.

N10. Покажите различие между операторами инкрементов.

N14. Приведите пример программы объявления структур.

N23. Приведите пример программы определения нового типа.

N28. Назовите арифметические операции в *C*.

| Параметры    | <i>TP</i>  | <i>C</i>  |
|--------------|--|---|
| <b>N</b>     | {1,2,...,44}, 44 задачи                            | {1,2,...,51}, 51 задача                           |
| <b>A</b>     | {1,3,5,7,11,12,16,18,19,31,33}, 11 задач           | {1,2,6,10,14,23,28,42,46}, 9 задач                |
| $Q_{\min}$   | 0,4  | 0,4   |
| $\lambda_1$  | 8,031 год <sup>-1</sup> = 0,00093 ч <sup>-1</sup>  | 9,245 год <sup>-1</sup> = 0,00107 ч <sup>-1</sup> |
| $\lambda_2$  | 29,23 год <sup>-1</sup> = 0,003383 ч <sup>-1</sup> | 52,6 год <sup>-1</sup> = 0,006088 ч <sup>-1</sup> |
| $c_1$        | 11   | 9   |
| $c_2$        | 33   | 51  |
| $a_{10}$ , ч | 114,07   | 73,26   |
| $a_{20}$ , ч | 456,26   | 439,54  |

N42. Каковы логические операции в *C*.

N46. Разработайте процедуру графического построения круга со штриховым заполнением.

В таблице приведены исходные данные и результаты расчета оптимальной периодичности частичных и полных проверок по языкам программирования *TP* и *C* с помощью диалоговой системы "Планирование подготовки кадров":

**N** — множество задач по *TP* (44) и *C* (51);

**A** — множество основных задач по *TP* (11) и *C* (9) по одной из каждого основного раздела: в *TP* — 11 разделов, в *C* — 9 разделов;

$Q_{\min}$  — минимальная вероятность правильного выполнения случайно выбранной задачи по *TP* и *C*;

$\lambda_1$  — суммарная интенсивность забывания основных задач множества **A** по *TP* и *C*;

$\lambda_2$  — суммарная интенсивность забывания остальных задач множества **N**\A по *TP* и *C*: 22 (33-11) и 42 (51-9) по *TP* и *C*;

$c_1$  — затраты на частичную проверку и восстановление забытых знаний множества **A**;

$c_2$  — затраты на полную проверку и восстановление забытых знаний множества **N**;

$a_{10}$  и  $a_{20}$  — оптимальные интервалы между частичными и полными проверками и восстановлением знаний.

Таким образом, частичные проверки и восстановления знаний по языкам *TP* и *C* следует про-

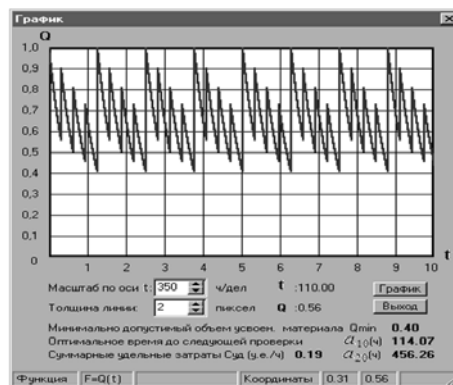


Рис. 1. Зависимость вероятности  $Q(t)$  правильного выполнения случайно выбранной задачи по *TP* обучаемым/пользователем или сотрудником с учетом частичных и полных проверок и восстановления забытых знаний

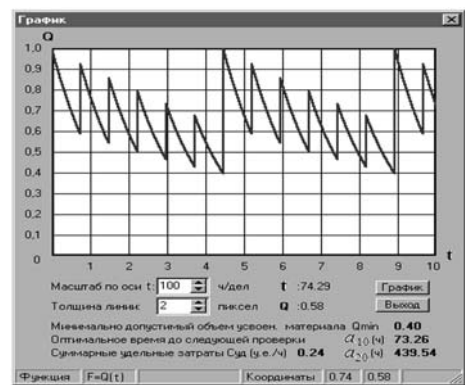


Рис. 2. Зависимость вероятности  $Q(t)$  правильного выполнения случайно выбранной задачи по *C* обучаемым/пользователем или сотрудником с учетом частичных и полных проверок и восстановления забытых знаний

водить через 114,07 и 73,26 ч, а полные — через 456,26 и 439,54 ч. На рис. 1 и 2 приведены зависимости вероятности  $Q(t)$  правильного выполнения случайно выбранной задачи по  $TP$  и  $C$  с учетом частичных и полных проверок и восстановления забытых знаний.

Число способов разделения  $n$  знаний (задач, функций) множества  $\mathbf{N}$  на два подмножества  $\mathbf{A}$  и  $\mathbf{N} \setminus \mathbf{A}$  равно  $\mathbf{K} = 2^n - 1$ . Для каждого из них вычислим интенсивности забывания  $\lambda_1$  и  $\lambda_2$  и затраты  $c_1$  и  $c_2$  на контроль и последующее восстановление знаний, а по ним —  $a_{10}(i)$ ,  $a_{20}(i)$  и  $c_{\min}(i)$ . Из полученных значений  $c_{\min}(i)$  следует взять минимальное:

$$c_{\min} = \min_i c_{\min}^{(i)}.$$

По нему и определяется периодичность контроля.

### Заключение

В работе предложены две периодические стратегии управления качеством профессиональной подготовки обучаемых (персонала) с учетом интенсивности забывания и экономических показателей (затрат на обучение, контроль и др.). Они основаны на оптимальном планировании во времени полных и частичных проверок и последующих восстанов-

лений выявленных забытых знаний. Показана применимость предлагаемых стратегий для оптимизации полных и частичных проверок предприятий, организаций и фирм.

### Список литературы

1. **Birolini A.** Qualitat und Zuverlässigkeit technischer Systeme. Theorie, Praxis, Management. Berlin; Heidelberg; New York, u.a.: Springer—Verlag, 1991. 507 S.
2. **E-Learning** deutscher Universitäten. Fallbeispiele aus der Hochschulpraxis // Kurzinformatio Bau und Technik, HIS Hochschulinformationssystem. 2005. B 4. 97 S.
3. **Свиридов А. П.** Введение в статистическую теорию обучения и контроля знаний. Ч. 2. Элементы статистической динамики знаний. М.: Изд-во МЭИ, 1974. 152 с.
4. **Свиридов А. П.** Основы статистической теории обучения и контроля знаний. М.: Высшая школа, 1981. 262 с.
5. **Проблемы** подготовки кадров в сфере инфокоммуникационных технологий, Санкт-Петербургская научно-практическая конференция, Санкт-Петербург, 1—3 марта 2005 г.: Материалы конференции. СПб.: СПОИСУ, 2005. 156 с.
6. **Человеческий** фактор (Handboock of Human Factors) / Под ред. Сальвенди Г. М.: Мир, 1991. 599 с.
7. **Залещанский Б. Д., Свиридов А. П., Шалобина О. А., Шалобина Е. А.** Вероятностно-статистические стратегии обеспечения качества подготовки персонала социотехнических систем // Информационные технологии. 2013. № 8. С. 67—70.
8. **Ушинский К. Д.** Человек как предмет воспитания. Опыт педагогической антропологии. М.: Учпедгиз, 1950.
9. **Талызина Н. Ф.** Управление процессом усвоения знаний (психологические основы), 2-е изд. М.: Изд-во МГУ, 1984. 344 с.

**B. D. Zaleshchanskiy, Prof., A. P. Sviridov, Prof.,  
O. A. Pavlova, Graduate Student, E. A. Shalobina, Graduate Student**

## Probabilistic and Statistical Strategy of Quality Ensurance of Personnel Training in Sotsio-Tehnickal Systems by Means of Full and Partial Testing Optimisation

*Periodical strategies of vocational training quality management are examined with consideration of forgetting infectivity and economical parameters (training costs, material losses due to personal errors). Strategies are based optimal planning of repetitions and preventive knowledge renewal. There initial data: forgetting infectivity, costs of full and partial testing, minimal probability of correct answer to the random chosen test. Aplicability of this strategies for full and partial testing optimization is shown.*

*Examples of strategies application are given for periodicity optimization of full and partial students knowledge checking in of Pascal (TP) and C programming course. With this purpose forgetting intensity is obtained for each of 44 task in TP and 51 tasks in C. On this base forgetting intensity for partial checking is calculated and summary forgetting intensity of all tasks for full checking.*

**Keywords:** strategies, personnel training, forgetting infectivity, test costs

### References

1. **Birolini A.** Qualitat und Zuverlässigkeit technischer Systeme. Theorie, Praxis, Management. Berlin, Heidelberg, New York u. a.: Springer—Verlag, 1991. 507 S.
2. **E-Learning** deutscher Universitäten. Fallbeispiele aus der Hochschulpraxis. Kurzinformatio Bau und Technik, HIS Hochschulinformationssystem. 2005. B 4. 97 S.
3. **Sviridov A. P.** Vvedenie v statisticheskuyu teoriyu obucheniya i kontrolya znaniy. P. 2. Jelementy statisticheskoy dinamiki znaniy. M.: Moskovskiy jenergeticheskij institut, 1974. 152 p.
4. **Sviridov A. P.** Osnovy statisticheskoy teorii obucheniya i kontrolya znaniy. M.: Vysshaja shkola, 1981. 262 p.
5. **Problemy** podgotovki kadrov v sfere infokommunikacionnyh tehnologiy. Sankt-Peterburgskaya nauchno-prakticheskaya konferenciya. Materialy konferencii: Sankt-Peterburg, 1—3 marta 2005 g.: SPb.: SPOISU, 2005. 156 p.
6. **Sal'vendi G.** (Ed.) Chelovecheskij faktor (Handboock of Human Factors). M.: Mir, 1991. 599 p.
7. **Zaleshhanskiy B. D., Sviridov A. P., Shalobina O. A., Shalobina E. A.** Veroyatnostno-statisticheskie strategii obespecheniya kachestva podgotovki personala sociotekhnicheskikh sistem. *Informacionnye tehnologii*. 2013. N. 8. P. 67—70.
8. **Ushinskiy K. D.** Chelovek kak predmet vospitaniya. Opyt pedagogicheskoy antropologii. M.: Uchpedgiz, 1950.
9. **Talyzina N. F.** Upravlenie processom usvoeniya znaniy (psihologicheskie osnovy), 2-oe izd. M.: MGU, 1984. 344 p.

# ЖУРНАЛ В ЖУРНАЛЕ



## НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

№ 7

ИЮЛЬ

2014

### Главный редактор:

ГАЛУШКИН А.И.

### Редакционная коллегия:

АВЕДЬЯН Э.Д.  
БАЗИЯН Б.Х.  
БЕНЕВОЛЕНСКИЙ С.Б.  
БОРИСОВ В.В.  
ГОРБАЧЕНКО В.И.  
ЖДАНОВ А.А.  
ЗЕФИРОВ Н.С.  
ЗОЗУЛЯ Ю.И.  
КРИЖИЖАНОВСКИЙ Б.В.  
КУДРЯВЦЕВ В.Б.  
КУЛИК С.Д.  
КУРАВСКИЙ Л.С.  
РЕДЬКО В.Г.  
РУДИНСКИЙ А.В.  
СИМОРОВ С.Н.  
ФЕДУЛОВ А.С.  
ЧЕРВЯКОВ Н.И.

### Иностранные члены редколлегии:

БОЯНОВ К.  
ВЕЛИЧКОВСКИЙ Б.М.  
ГРАБАРЧУК В.  
РУТКОВСКИЙ Л.

### Редакция:

БЕЗМЕНОВА М.Ю.  
ГРИГОРИН-РЯБОВА Е.В.  
ЛЫСЕНКО А.В.  
ЧУГУНОВА А.В.

### Осипов В. Ю.

Рекуррентная нейронная сеть со структурой слоев в виде двойной спирали . . . . . 56

### Мандрикова О. В., Полозов Ю. А.

Аппроксимация и анализ ионосферных параметров на основе совмещения вейвлет-преобразования с коллективами нейронных сетей . . . . . 61

### Галушкин А. И.

Метод обратного распространения ошибки и российские работы по теории нейронных сетей . . . . . 66

### Боронников Д. А., Пантюхин Д. В., Данько С. В.

Нейросетевой алгоритм организации пространственных данных о рельефе местности . . . . . 77

**В. Ю. Осипов**, д-р техн. наук, проф., вед. науч. сотр., e-mail: osipov\_vasily@mail.ru,  
Федеральное государственное бюджетное учреждение науки  
Санкт-Петербургский институт информатики и автоматизации Российской академии наук

## Рекуррентная нейронная сеть со структурой слоев в виде двойной спирали

*Предложен усовершенствованный метод обработки информации в двухслойной рекуррентной нейронной сети (РНС) с управляемыми синапсами. Рекомендовано за счет изменения функций ослабления синапсов наделять эту РНС структурой слоев в виде двойной спирали. Раскрыты особенности реализации РНС с такой структурой. Путем математического моделирования сопоставлены ее возможности с известными решениями. Показано, что РНС со структурой слоев в виде двойной спирали обладает рядом преимуществ по интеллектуальной обработке динамических сигналов.*

**Ключевые слова:** рекуррентная нейронная сеть, структура, двойная спираль, обработка информации

### Введение

На искусственные рекуррентные нейронные сети (РНС) возлагают большие надежды в части возможного прорыва по интеллектуальной обработке различной информации. Пока РНС во многом еще не совершенны [1–4].

Одним из направлений развития РНС выступает расширение их возможностей по распознаванию и ассоциативному запоминанию динамических сигналов.

Анализ известных решений [1, 2, 5, 6] по обеспечению таких возможностей РНС показывает, что они обладают рядом недостатков. При использовании этих решений возможна быстрая перегрузка памяти РНС входным потоком сигналов. Велика избыточность запоминания сигналов. Одни и те же пространственно-временные структуры сигналов запоминаются на элементах РНС многократно. Не высока выраженность ассоциативного взаимодействия сигналов в РНС. Результаты распознавания динамических сигналов слишком "размазаны" в памяти РНС, что затрудняет их ассоциативный вызов.

Некоторыми преимуществами по сравнению с известными решениями [1, 2, 5, 6] обладают методы, рассмотренные в работах [7–11]. Согласно [9] удается исключить перегрузку памяти сети входным потоком сигналов за счет стирания устаревшей информации. Как показано в [10, 11], путем продвижения сигналов вдоль слоев в двухслойной РНС по спирали достижимо также расширение возможностей ее памяти и улучшение распознавания входных воздействий. Однако уровень неоправданного избыточного запоминания информации в РНС остается еще высоким. Также не достигнуты пределы по повышению выраженности в пространстве и во времени ассоциативного взаимодействия сигналов в РНС, изменения ее параметров.

Необходим поиск новых решений, позволяющих устранить имеющиеся недостатки, расширить

возможности РНС по распознаванию и ассоциативному запоминанию динамических сигналов.

### Постановка задачи

Известен метод [10, 11] обработки информации в двухслойной нейронной сети с обратными связями, замыкающими двухслойные контуры со временем задержки единичных образов (ЕО) меньше времени невосприимчивости нейронов сети после их возбуждения. Согласно этому методу сигнал подают в РНС после разложения на составляющие в базисе, согласованном с входным слоем сети. При этом каждую составляющую перед подачей в РНС преобразуют в последовательность единичных образов (импульсов) с частотой повторения как функции от амплитуды составляющей. Сигнал представляют в РНС в виде последовательных совокупностей ЕО в соответствии с предварительно заданными правилами его распознавания с учетом обратных результатов. При передаче совокупностей ЕО от слоя к слою их сдвигают вдоль слоев. Сдвиги осуществляют с изменяемыми параметрами с учетом текущих состояний слоев. Продвигают совокупности ЕО вдоль слоев по спирали с изменяемым диаметром. Результаты распознавания запоминают на элементах сети. В качестве результатов обработки используют последовательные совокупности ЕО на выходном слое сети после обратного преобразования в соответствующие им исходные сигналы.

На рис. 1 приведена структурная схема двухслойной рекуррентной нейронной сети с управляемыми синапсами [7], реализующей известный метод [10, 11]. При подаче на вход первого слоя соответствующим образом преобразованного сигнала на выходе этого слоя присутствуют совокупности ЕО, несущие всю информацию о входном воздействии. Каждый ЕО из текущих совокупностей подается одновременно в первом блоке динамических синапсов на совокупность своих синапсов. Обеспе-





Рис. 1. Структурная схема двухслойной рекуррентной нейронной сети с управляемыми синапсами

чивается связь каждого нейрона, породившего ЕО, в общем случае со всеми нейронами второго слоя. Связи между нейронами одного и того же слоя отсутствуют. Амплитуда импульсного тока на выходе каждого синапса равна амплитуде напряжения входного импульса (единичного образа), умноженной на проводимость (вес) синапса. Веса синапсов определяются через произведение их весовых коэффициентов и функций ослабления. Весовые коэффициенты изменяются в зависимости от воздействий на синапсы единичных образов и выступают в качестве элементов долговременной памяти сети. При прохождении ЕО через синапсы они снимают с них информацию о предыдущих воздействиях и оставляют информацию о своем появлении через изменения весовых коэффициентов. Значения весовых коэффициентов определяют по известным правилам [8, 11]. Каждая из связей (синапсов) имеет свое значение функции ослабления ЕО, зависящее от удаленности связываемых через синапсы нейронов (расстояний между ними). Изменяя функции ослабления синапсов, можно менять направления продвижения совокупностей ЕО вдоль слоев сети.

В общем случае смещенные совокупности ЕО с выхода первого блока динамических синапсов поступают на вход второго слоя нейронов. Все ЕО, поступившие на один и тот же нейрон по различным синапсам, суммируются. При превышении этой суммой заданного порога нейрон возбуждается и на его выходе формируется ЕО. Затем имеемая сумма обнуляется, а сам нейрон затем переходит в состояние невосприимчивости входных сигналов, в котором находится заданное время.

Совокупности ЕО с выхода второго слоя после задержки поступают на второй блок динамических синапсов. В этом блоке они, как и в первом блоке динамических синапсов, обрабатываются. Сдвинутые вдоль первого слоя в зависимости от состояний первого и второго слоев совокупности ЕО поступают на первый слой.

С учетом этого поступающие на первый слой нейронов прямые и обратные совокупности ЕО в

нем корректно связываются, распознаются и порождают на его выходе новые совокупности ЕО, несущие в себе информацию как о текущих, так и ранее запомненных сетью сигналах, связанных с первыми.

За счет пространственных сдвигов совокупностей ЕО вдоль слоев и цикличности обмена информацией между слоями они разбиваются на логические поля. Согласно известному методу при этом совокупности ЕО продвигают вдоль слоев по спирали.

Пример продольной спиральной структуры РНС приведен на рис. 2, где  $d, q$  — единичные сдвиги совокупностей ЕО вдоль слоев соответственно по координатам  $X, Y$ . Стрелки отражают направления продвижения совокупностей ЕО вдоль слоев. Согласно рис. 2 входной сигнал в виде последовательных совокупностей ЕО подается на первое поле первого слоя РНС, а снимается с последнего поля. Второй слой сети по структуре идентичен первому.

За счет встречного продвижения совокупностей ЕО вдоль слоев удастся снизить избыточность запоминания результатов распознавания сигналов на элементах сети. При этом частично повышается выраженность ассоциативной обработки сигналов. Однако это достигается, когда совокупности ЕО продвигаются вдоль слоев навстречу друг другу по смежным полувиткам спирали. При продвижении совокупностей ЕО вдоль слоев РНС в пределах одного полувитка все недостатки, связанные с избыточностью запоминания сигналов и невысокой выраженностью ассоциативной обработки, остаются.

Необходимо усовершенствовать известный метод обработки информации и предложить новую логическую структуру РНС, расширяющую возможности по распознаванию и ассоциативному запоминанию динамических сигналов.

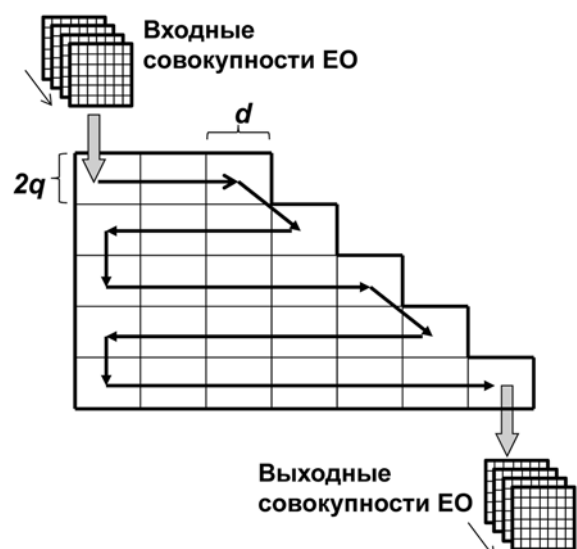


Рис. 2. Пример спиральной структуры РНС

## Метод обработки информации в РНС

В интересах расширения функциональных возможностей РНС предлагается усовершенствованный метод обработки в ней информации. Согласно ему сигнал представляют в виде последовательных совокупностей ЕО, разделенных на пары частных совокупностей этих образов. Нечетные частные совокупности ЕО из этих пар подаются на один групповой вход первого слоя сети, а четные — на другой групповой вход, пространственно удаленный от первого. Частные совокупности ЕО из каждой такой пары предлагается продвигать вдоль слоев РНС по двойной спирали с изменяемым диаметром и скручивать попарно последовательности частных совокупностей ЕО.

Это достижимо путем осуществления разных пространственных сдвигов частных совокупностей ЕО в сформированных парах. Напомним, что с физической точки зрения такие сдвиги реализуемы за счет изменения функций ослабления синапсов.

За счет такого решения обеспечиваются не только выраженные ассоциации между совокупностями ЕО, продвигающимися вдоль слоев по смежным полувиткам спирали. Реализуется сосредоточенное по пространству и времени взаимодействие между совокупностями ЕО, продвигающимися по разным спиральям. Это позволяет сосредоточенно запоминать на элементах РНС результаты распознавания сигналов, а не "размазывать" их по всей сети. В итоге при одних и тех же размерах РНС способна лучше распознавать сигналы и больше запоминать различной информации.

Соответствие между входами и выходами сети обеспечивается за счет приоритетности коротких связей между нейронами взаимодействующих слоев. Это позволяет, так же как и в известном подходе [8, 10], использовать в качестве результатов обработки последовательные совокупности ЕО на выходном слое сети после обратного преобразования в соответствующие им исходные сигналы.

### Структура РНС в виде двойной спирали

Для пояснения предлагаемого метода рассмотрим структуру РНС в виде двойной спирали. На рис. 3 приведен пример ее продольной структуры, где стрелками отражены направления продвижения частных совокупностей ЕО вдоль слоев. Штриховые стрелки показывают продвижение совокупностей ЕО под пересекаемыми полями. Параметры  $d$ ,  $q/2$  — единичные сдвиги совокупностей ЕО вдоль слоев. На рис. 4, а—г показаны разрезы этой РНС (см. рис. 3) по двум слоям по первой (рис. 4, а), второй (рис. 4, б), третьей (рис. 4, в), четвертой (рис. 4, г) строкам и вид сбоку (рис. 4, д). Стрелками на рис. 4 обозначены направления продвижения совокупностей ЕО между слоями.

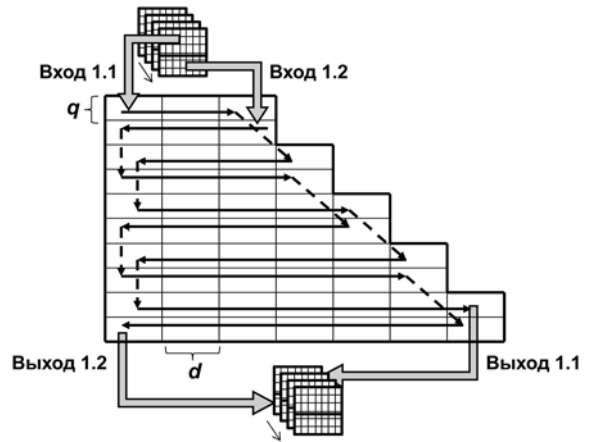


Рис. 3. Пример продольной структуры РНС в виде двойной спирали

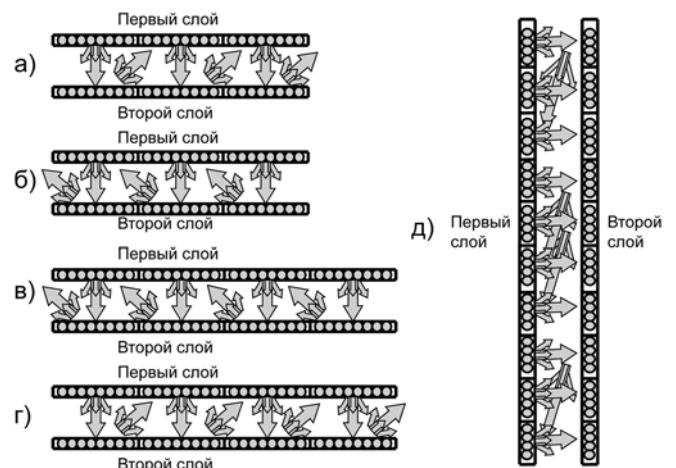


Рис. 4. Поперечная структура РНС в виде двойной спирали

Согласно рис. 3 входной сигнал в виде последовательных совокупностей ЕО, разделенных на пары частных совокупностей этих образов, подают в сеть. Причем нечетные частные совокупности ЕО подают на вход 1.1 и продвигают по первой строке слева направо, а четные — на вход 1.2 и продвигают по второй строке справа налево. В целом эти совокупности ЕО продвигаются по двойной спирали в виде скрутки двух последовательностей частных совокупностей ЕО.

Наибольшее ассоциативное взаимодействие частных совокупностей ЕО друг с другом наблюдается, когда расстояние между этими совокупностями становится минимальным. Из анализа рис. 3, 4 видно, что за счет предлагаемого решения возможно сосредоточенное по пространству и времени взаимодействие между совокупностями ЕО, продвигающимися как по одной, так и по разным спиральям. Это позволяет устранить недостатки, свойственные известным подходам.

## Результаты моделирования

Для подтверждения преимуществ предлагаемого решения по сравнению с известным подходом [10, 11] проводилось математическое моделирование. Были разработаны программные модели нейронных сетей, реализующих подход [10, 11] и предлагаемый метод обработки информации в нейронной сети. В модели известной РНС с обычной спиральной структурой формировался один полный виток продвижения совокупностей единичных образов вдоль слоев. Каждый слой этой сети за счет пространственных сдвигов совокупностей ЕО, передаваемых от слоя к слою, разбивался на две строки по 25 полей. Размер полей  $6 \times 8$  нейронов. В модели РНС, реализующей предлагаемый метод, число нейронов было таким же, как и в первой модели. Однако каждый слой этой РНС разбивался на четыре строки по 25 полей с размерами  $6 \times 4$  нейрона. Две строки приходились на виток одной, а другие две — на виток другой спирали. За счет этого обеспечивалась скрутка двух частных последовательностей ЕО.

Сопоставлялись результаты ассоциативного взаимодействия сигналов, полученные с помощью этих моделей. Результаты моделирования приведены на рис. 5, 6. На рис. 5 показаны приросты  $\Delta W$  суммарного веса синапсов на каждом шаге  $t$  функционирования сети относительно предыдущего шага на одном полувитке спирали при использовании известного решения (кривая 1) и предлагаемого метода (кривая 2) для случая обработки одной совокупности. Приросты  $\Delta W$  при обработке двух последовательных совокупностей ЕО на одном витке спирали известным подходом и предлагаемым методом отражены на рис. 6, кривые 1 и 2 соответственно.

Из анализа рис. 5 видно, что согласно известному методу при вводе в сеть неразделенной совокупности ЕО наблюдается скачок прироста  $\Delta W$  суммарного веса синапсов до значения 0,019 (кривая 1). Затем значение прироста суммарного веса синапсов, при продвижении совокупности ЕО в пределах полувитка спирали, не изменяется. Это свидетельствует о том, что на различных элементах сети запоминается одна и та же схема пространственно-временных связей между ЕО. Присутствует высокая неоправданная избыточность запоминания сигналов.

При использовании предлагаемого метода в начале обработки также присутствует скачок прироста  $\Delta W$  суммарного веса синапсов, но лишь до значения 0,015 (рис. 5, кривая 2), а не до 0,019. Максимум этого прироста приходится на момент встречи двух частных совокупностей ЕО, продвигающихся навстречу друг другу по смежным более узким строкам.

Анализ рис. 6 показывает, что при обработке двух последовательных совокупностей ЕО ситуация аналогична предыдущей. Только в начале обработки присутствует не один, а два скачка при-

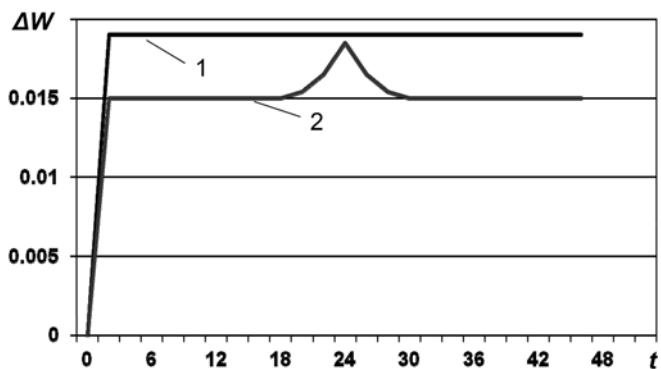


Рис. 5. Результаты обработки одной совокупности единичных образов известным подходом (кривая 1) и предлагаемым методом (кривая 2)

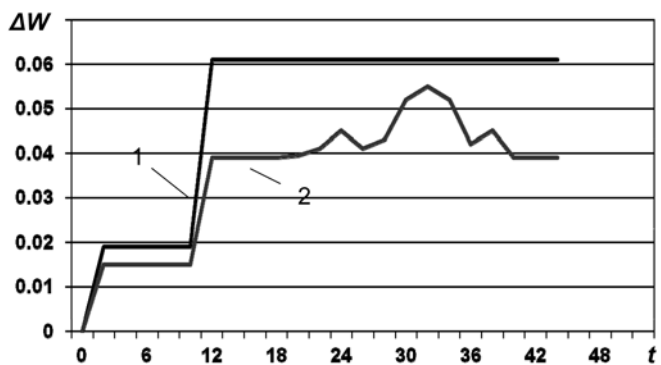


Рис. 6. Результаты обработки двух последовательных совокупностей единичных образов известным подходом (кривая 1) и предлагаемым методом (кривая 2)

роста  $\Delta W$ , обусловленные вводом в РНС первой и второй последовательностей ЕО. В этой ситуации, судя по результатам использования известного метода (рис. 6, кривая 1), недостатки его проявляются еще сильнее, чем в предыдущем случае (рис. 5, кривая 1). Максимумы ассоциативного взаимодействия сигналов при предлагаемом методе приходятся на моменты минимизации расстояний между частными совокупностями (рис. 6, кривая 2). В другие моменты времени уровни такого взаимодействия существенно ниже. При этом наблюдаемые на рис. 5, 6 (кривые 2) всплески  $\Delta W$  свидетельствуют о ярко выраженном по пространству и времени ассоциативном взаимодействии обрабатываемых сигналов в РНС со структурой слоев в виде двойной спирали.

Из этих результатов моделирования следует вывод, что предлагаемый метод обладает меньшей избыточностью запоминания сигналов. Результаты распознавания сигналов лучше выражены по пространству и времени через изменение параметров сети. В целом это расширяет возможности по интеллектуальной обработке информации в нейронной сети, по распознаванию и запоминанию динамических сигналов.

Разработан новый метод обработки информации и предложена реализующая его РНС со структурой слоев в виде двойной спирали. При обработке информации в двухслойной РНС предлагается сигналы представлять в виде пар последовательных частных совокупностей ЕО. Частные совокупности ЕО из каждой такой пары рекомендуется продвигать вдоль слоев РНС по двойной спирали с изменяемым диаметром и скручивать попарно последовательности этих совокупностей. Предложенные решения позволяют расширить функциональные возможности РНС по интеллектуальной обработке информации, улучшить распознавание и ассоциативное запоминание динамических сигналов. Эти новые решения могут быть использованы при создании перспективных ассоциативных интеллектуальных машин и систем.

Реализация РНС в виде двойной спирали осуществима как программно, так и аппаратно с применением известных цифровых и аналоговых технологий.

1. **Amari S.** Dreaming of mathematical neuroscience for half a century // *Neural Networks*. 2013. N. 37. P. 48–51.
2. **Palm G.** Neural associative memories and sparse coding // *Neural Networks*. 2013. N. 37. P. 165–171.
3. **Pentti O. A. Haikonen.** The Role of Associative Processing in Cognitive Computing // *Cognitive Computing*. 2009. N. 1. P. 42–49.
4. **Hawkins J., Blakeslee S.** On Intelligence. NY; Owl Books, 2005. 272 p.
5. **Galushkin A. I.** *Neural Networks Theory*. Berlin Heidelberg: Springer-Verlag, 2007. 396 p.
6. **Haykin S.** *Neural Networks: A Comprehensive Foundation*. Second Edition. Prentice Hall, 1988. 842 p.
7. **Осипов В. Ю.** Рекуррентная нейронная сеть с управляемыми синапсами // *Информационные технологии*. 2010. № 7. С. 43–47.
8. **Осипов В. Ю.** Ассоциативная интеллектуальная машина // *Информационные технологии и вычислительные системы*. 2010. № 2.
9. **Осипов В. Ю.** Стирание устаревшей информации в ассоциативных интеллектуальных системах // *Мехатроника, автоматизация, управление*. 2012. № 3. С. 16–20.
10. **Осипов В. Ю.** Оптимизация ассоциативных интеллектуальных систем // *Мехатроника, автоматизация, управление*. 2011. № 3. С. 35–39.
11. **Осипов В. Ю.** Рекуррентная нейронная сеть со спиральной структурой слоев // *Информационно-управляющие системы*. 2012. № 6. С. 28–32.

V. Yu. Osipov, Leading Researcher, e-mail: osipov\_vasily@mail.ru,  
Saint-Petersburg Institute for Informatics and Automation of RAS

## Recurrent Neural Network with Structure of Layers in the Form of the Double Spiral

**Purpose.** Search approaches to eliminate excess storage and to empower recognition of dynamic signals in recurrent neural networks. **Methods:** The proposed approach is based on well-known models and methods of information processing in recurrent neural networks (RNN) with operated synapses. To justify the proposed approach used method of mathematical modeling. **Results:** improved method of processing information in a bilayer RNN operated synapses. Recommended by changing the attenuation functions of synapses endow the network layer structure in the form of a double spiral. Features of the implementation of a neural network with the structure disclosed. Its capabilities compared with known solutions. It is shown that the recurrent neural network with the structure of the layers in the form of a double spiral has the benefits of intellectual processing dynamic signals. **Practical relevance:** empowering recurrent neural network proposed structure can significantly reduce the redundancy of storing information. Recognition results of signals are better expressed in space and time through a change in the network settings. The developed method is useful if you create associative perspective of intelligent machines and systems.

**Keywords:** recurrent neural network, structure, double spiral, information processing

### References

1. **Amari S.** Dreaming of mathematical neuroscience for half a century. *Neural Networks*. 2013. N. 37. P. 48–51.
2. **Palm G.** Neural associative memories and sparse coding. *Neural Networks*. 2013. N. 37. P. 165–171.
3. **Pentti O. A. Haikonen.** The Role of Associative Processing in Cognitive Computing. *Cognitive Computing*. 2009. N. 1. P. 42–49.
4. **Hawkins J., Blakeslee S.** On Intelligence. NY: Owl Books, 2005. 272 p.
5. **Galushkin A. I.** *Neural Networks Theory*. Berlin Heidelberg: Springer-Verlag, 2007. 396 p.
6. **Haykin S.** *Neural Networks: A Comprehensive Foundation*. Second Edition. Prentice Hall, 1988. 842 p.
7. **Osipov V. Yu.** Rekurrentnaja nejronnaja set' s upravljaemyimi sinapsami. *Informacionnye tehnologii*. 2010. N. 7. P. 43–47.
8. **Osipov V. Yu.** Associativnaja intellektual'naja mashina. *Informacionnye tehnologii i vychislitel'nye sistemy*. 2010. N. 2. P. 59–67.
9. **Osipov V. Yu.** Stiranje ustarevshej informacii v associativnyh intellektual'nyh sistemah. *Mehatronika, avtomatizacija, upravlenie*. 2012. N. 3. P. 16–20.
10. **Osipov V. Yu.** Optimizacija associativnyh intellektual'nyh sistem. *Mehatronika, avtomatizacija, upravlenie*. 2011. N. 3. P. 35–39.
11. **Osipov V. Yu.** Rekurrentnaja nejronnaja set' so spiral'noj strukturoj sloev. *Informacionno-upravliajushchie sistemy*. 2012. N. 6. P. 28–32.

О. В. Мандрикова<sup>1, 2</sup>, д-р техн. наук, проф., зав. лаб., e-mail: up\_agent@mail.ru,

Ю. А. Полозов<sup>1, 2</sup>, канд. техн. наук, науч. сотр.,

<sup>1</sup> Институт космофизических исследований и распространения радиоволн ДВО РАН

<sup>2</sup> Камчатский государственный технический университет

## Аппроксимация и анализ ионосферных параметров на основе совмещения вейвлет-преобразования с коллективами нейронных сетей

*Предложен способ аппроксимации и анализа временного хода ионосферных параметров, основанный на совмещении кратномасштабных вейвлет-разложений и коллективов нейронных сетей. Описаны вычислительные решения по формированию нейронных сетей и их объединению в коллективы. Аprobация способа выполнена на данных критической частоты ионосферы  $f_{OF2}$ , полученных на обсерватории "Паратунка" (ИКИР ДВО РАН, с. Паратунка, Камчатский край). На основе оценки ошибок аппроксимации в регистрируемом временном ряде критической частоты ионосферы выделены аномалии в периоды повышенной солнечной активности, а также накануне и в моменты сильных землетрясений на Камчатке.*

**Ключевые слова:** вейвлет-преобразование, нейронные сети, критическая частота ионосферы, аномалии, землетрясения

### Введение

Регистрируемые ионосферные параметры имеют сложную структуру, подвержены влиянию внешних факторов различной физической природы, что значительно усложняет процесс их изучения [1, 2]. Предметом данных исследований являются регистрируемые временные ряды критической частоты ионосферы  $f_{OF2}$ . Для временного ряда  $f_{OF2}$  характерны суточные и сезонные изменения, а также аномальные возмущения, возникающие вследствие различных факторов природного и техногенного характера [2—8]. В периоды возмущений длительность аномального поведения данных составляет от нескольких десятков минут до нескольких часов [3—8]. Данные аномалии могут быть обусловлены повышенной активностью Солнца, в сейсмоактивных областях они могут возникать в периоды сильных землетрясений [3—8].

Ввиду сложной структуры временного ряда  $f_{OF2}$  его изучение с помощью традиционных методов анализа временных рядов неэффективно [3]. Предложенный в данной статье способ его аппроксимации и анализа основан на применении кратномасштабных вейвлет-разложений [9] и коллективов нейронных сетей [10]. Применение вейвлет-преобразования показало свою эффективность при обработке и анализе ионосферных параметров [4, 11, 12]. Для аппроксимации получаемых в результате вейвлет-разложений компонент в работе предлагается использовать аппарат нейронных сетей. Нейронные сети хорошо зарекомендовали себя при воспроизведении сложных нелинейных зависимостей данных [5, 13, 14]. В более ранних работах авторов [5, 15, 16] показано, что данный аппарат является эффективным средством аппроксимации компонент ряда критической частоты ионосферы [5, 14—16]. Для построения аппроксимации детализирующих

компонент временного хода данных  $f_{OF2}$  предложено объединять нейронные сети в коллективы. В статье описаны вычислительные решения по формированию нейронных сетей и их объединению в коллективы.

На основе предлагаемых средств выполнено моделирование и анализ данных  $f_{OF2}$ , изучена их внутренняя структура. В различных компонентах вейвлет-разложений выделены аномалии, возникающие в ионосфере в периоды возмущений. Анализ полученных результатов показал, что аномалии можно наблюдать накануне и в моменты сильных землетрясений на Камчатке, а также в периоды повышенной солнечной и геомагнитной активностей.

### 1. Описание способа

#### 1.1. Представление данных на основе кратномасштабных вейвлет-разложений

На основе кратномасштабных вейвлет-разложений до 3-го масштабного уровня получаем следующее представление ионосферных данных [3]:

$$f_0(t) = \sum_{j=-1}^{-3} g(2^j t) + f(2^{-3} t), \quad (1)$$

где  $g(2^j t) \in W_j$ ,  $f(2^{-3} t) \in V_{-3}$ ,  $W_j$  — пространство с разрешением  $j$ , порожденное вейвлетом  $\Psi_{j, n}(t) = 2^{j/2} \Psi(2^j t - n)$ ; компоненты  $g(2^j t) = \sum_n d_{j, n} \Psi_{j, n}(t)$ ,

где  $d_{j, n} = \langle f, \Psi_{j, n} \rangle$  — детализирующие компоненты; сглаженная компонента  $f(2^{-3} t) = \sum_k c_{-3, k} \varphi_{-3, k}(t)$ ,

где  $c_{-3, k} = \langle f, \varphi_{-3, k} \rangle$ ,  $\varphi_{-3, k}(t) = 2^{-3/2} \varphi(2^{-3} t - k)$  — скейлинг-функция, порождающая пространство  $V_{-3}$ .

В соотношении (1) предполагается, что исходное разрешение данных  $j = 0$ . Масштабный уровень разложения определялся на основе критерия наименьших потерь (путем оценки апостериорного риска [17]) и основывался на результатах работ [4, 5].

### 1.2. Аппроксимация сглаженной компоненты на основе нейронной сети

#### Алгоритм формирования нейронной сети

1. Выполняем вейвлет-восстановление компоненты  $f(2^{-3}t)$  до исходного разрешения и получаем ее представление в виде [9]:

$$f(t) = \sum_k c_{0,k} \varphi_{0,k}(t).$$

2. Для исключения влияния краевого эффекта, возникающего при вейвлет-обработке, у восстановленного набора данных удаляем первые и последние  $2^{-j}q$  отсчетов (размер окрестности, содержащей краевой эффект, на масштабном уровне  $(-j)$  определяется по формуле  $h_{-j} = 2^{-j}q$ , где  $q$  — размер носителя используемого вейвлета).

3. Полученный массив данных  $\{c_{0,k}\}_{k=1}^K$ , где  $K$  — число элементов массива, делим на блоки:  $\{c_{0,k}\}_{k=1}^Q$ ,  $\{c_{0,k}\}_{k=2}^{Q+1}$ , ...,  $\{c_{0,k}\}_{k=K-Q}^K$ . Длину блока  $Q = 24$  определяли в соответствии с суточным ходом часовых данных  $f_0 F2$ .

4. Используя полученные блоки данных, формируем *сеть переменной структуры* (алгоритм обучения и формирования *сети переменной структуры* описан в работе [16]). В качестве критерия качества обучения сети используем следующий критерий:

$$E_A = \left( \frac{1}{I} \sqrt{\sum_{i=1}^I e_{A,i}^2(l)} \right) < \varepsilon_A,$$

где  $\varepsilon_A > 0$  — наперед заданное маленькое значение;  $e_{A,i}(l) = \hat{c}_{0,l}^i - c_{0,l}^i$  — ошибка решения в дискретный момент времени  $l$  с шагом упреждения  $i$ ;  $c_{0,l}^i$  — желаемое,  $\hat{c}_{0,l}^i$  — действительное выходное значение сети;  $I$  — длина выходного вектора сети.

### 1.3. Аппроксимация детализирующих компонент на основе коллектива нейронных сетей

#### Алгоритм формирования коллектива нейронных сетей

1. Выполняем вейвлет-восстановление компоненты  $g(2^j t)$  до исходного разрешения и получаем ее представление в виде [9]:

$$g_0^j(t) = \sum_n d_{0,n}^j \Psi_{0,n}(t),$$

где верхний индекс  $j$  обозначает разрешение компоненты до ее восстановления.

2. Для исключения влияния краевого эффекта у восстановленного набора данных удаляем первые и последние  $2^{-j}q$  отсчетов.

3. Полученный массив данных  $\{d_{0,n}^j\}_{n=1}^N$ , где  $N$  — число элементов массива, делим на блоки:  $D^m = \{\{d_{0,n}^j\}_{n=m}^{23+m}, \{d_{0,n}^j\}_{n=24+(23+m)}^{24+(23+m)}, \dots, \{d_{0,n}^j\}_{n=S+(23+m)}^{S+(23+m)}\}_{m=1,24}$ ,  $m$  — номер нейронной сети,  $S = N - 48$ .

4. Используя полученные блоки  $D^m$ ,  $m = \overline{1, 24}$ , формируем *24 сети переменной структуры* [16]. В качестве критерия качества обучения каждой сети используем следующий критерий:

$$E_D = \left( \frac{1}{I} \sqrt{\sum_{i=1}^I e_{D,i}^2(l)} \right) < \varepsilon_D,$$

где  $\varepsilon_D > 0$  — наперед заданное маленькое значение;  $e_{D,i}(l) = \hat{d}_{0,l}^{j,i} - d_{0,l}^{j,i}$  — ошибка сети в дискретный момент времени  $l$  с шагом упреждения  $i$ ;  $d_{0,l}^{j,i}$  — желаемое выходное значение сети;  $\hat{d}_{0,l}^{j,i}$  — действительное выходное значение сети;  $I$  — длина выходного вектора сети. Сформированные сети объединяем в коллектив.

5. Из выходных векторов полученного коллектива нейронных сетей формируем матрицу размерности  $(I \times M)$ , содержащую  $I$  — строк (номер строки  $i$  — номер выходного значения нейронной сети) и  $M$  — столбцов (номер столбца  $m$  — номер нейронной сети). В нашем случае число столбцов  $M = 24$ . Каждый столбец полученной матрицы определяет вектор выходных значений  $m$ -й нейронной сети коллектива. Каждая строка полученной матрицы определяет вектор упрежденных с шагом  $i$  часовых значений критической частоты ионосферы за сутки.

## 2. Результаты экспериментов

В экспериментах использовали часовые данные критической частоты ионосферного слоя  $F2$  за период с 1968 по 2010 г., полученные автоматической ионосферной станцией "Паратунка" (п-ов Камчатка, с. Паратунка, регистрацию выполняет ИКИР ДВО РАН). Для анализа геомагнитной активности, характеризующей степень активности Солнца, использовали данные Н-компоненты геомагнитного поля, регистрируемые на Камчатке (станция "Паратунка", регистрацию выполняет ИКИР ДВО РАН). Для анализа сейсмической активности использовали данные сейсмического каталога п-ова Камчатка (регистрацию данных выполняет Камчатский филиал геофизической службы РАН). С учетом сезонного хода ионосферного процесса, временные ряды  $f_0 F2$  разбивали на сезоны (весна, зима, лето, осень). В статье представлены результаты анализа данных зимнего и летнего сезонов. В случае наличия пропусков в данных их заполняли медианными значениями. Следуя результатам работы [4], в качестве базисных вейвлетов брали класс ортого-

нальных вейвлетов Добеши 3-го порядка. При обучении нейронных сетей использовали данные за сейсмически спокойные периоды времени.

В соответствии с алгоритмом, описанным в п. 1.2, была сформирована нейронная сеть по аппроксимации сглаженной компоненты ряда  $f_0F2$ . Архитектура сети представлена на рис. 1, в соответствии с ней шаг упреждения данных критической частоты равен одному часу. Результаты работы данной сети показаны на рис. 2. Анализ рис. 2 показывает, что в своем большинстве абсолютные значения ошибок сети не превышают значения 0,1, что говорит о хорошем качестве аппроксимирующих свойств сети. Можно отметить, что в период низкой геомагнитной активности наблюдается локальное увеличение ошибки сети (13, 14 декабря 2004 г.), которое, возможно, обусловлено литосферными процессами на Камчатке (18 декабря 2004 г. на Камчатке произошли два сильных землетрясения энергетического класса  $K = 13,3$  и  $13,4$  соответственно). В период времени 17—23 января 2005 г. произошли сильные магнитные бури, которые в связи с наличием существенных пропусков в данных не отразились на результатах обработки (общая длительность пропусков в этот период составляет 30 ч).

В соответствии с алгоритмом, описанным в п. 1.3, был сформирован коллектив нейронных сетей по аппроксимации детализирующей компоненты  $g(2^{-3}t)$  ряда  $f_0F2$ . Сформированный коллектив выполняет упреждение данных с шагом, равным одному часу. Результаты работы коллектива сетей представлены на рис. 3. Анализ рис. 3 показывает, что абсолютные значения ошибок аппроксимации данных не превышают значения 0,03, за исключением периода времени 13, 14 декабря 2004 г. Как отмечено выше, данная аномальная особенность, возможно, обусловлена литосферными процессами, связанными с подготовкой землетрясений на Камчатке.

На рис. 4 показаны результаты моделирования ионосферных данных за период 04.12.2007—25.02.2008. Как показывает анализ графиков на рис. 4, *в*, *е* абсолютные значения ошибок аппроксимации сглаженной компоненты ряда  $f_0F2$  не превышают значения 0,2, детализирующей компоненты — значения 0,02.

Возрастание ошибок в аппроксимирующей и детализирующей компонентах наблюдается накануне и в моменты землетрясений, произошедших 24 декабря 2007 г. и

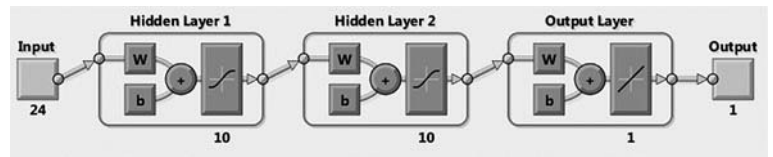


Рис. 1. Архитектура нейронной сети по аппроксимации сглаженной компоненты ряда  $f_0F2$

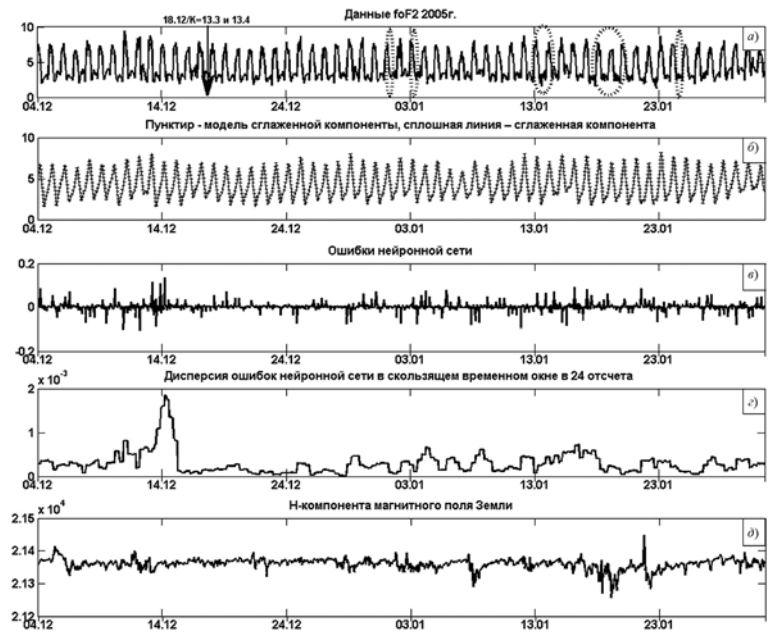


Рис. 2. Результаты работы нейронной сети по аппроксимации сглаженной компоненты  $f_0F2$  за период 04.12.2004—29.01.2005. Моменты возникновения сильных землетрясений на Камчатке отмечены стрелками. Пунктиром отмечены периоды, содержащие пропуски в данных

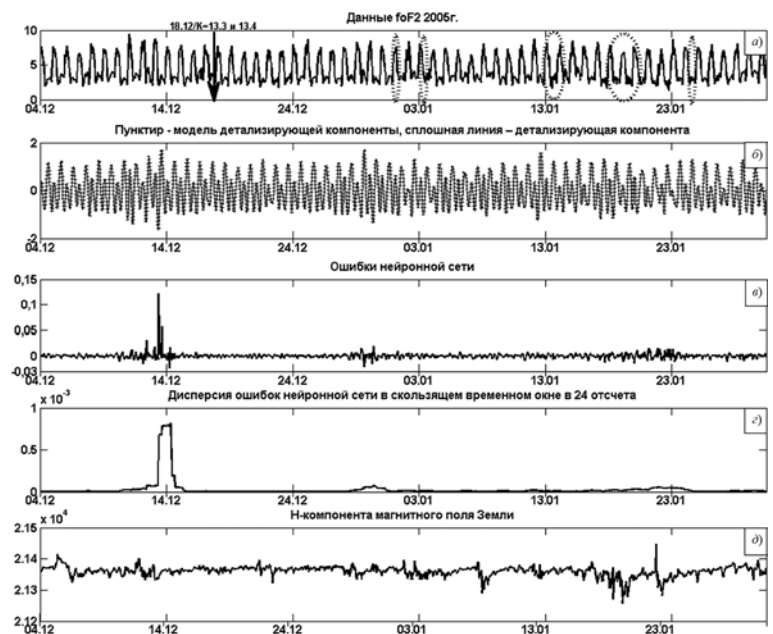


Рис. 3. Результаты работы коллектива нейронных сетей по аппроксимации детализирующей компоненты  $f_0F2$  за период 04.12.2004—28.01.2005. Моменты возникновения сильных землетрясений на Камчатке отмечены стрелками. Пунктиром отмечены периоды, содержащие пропуски в данных

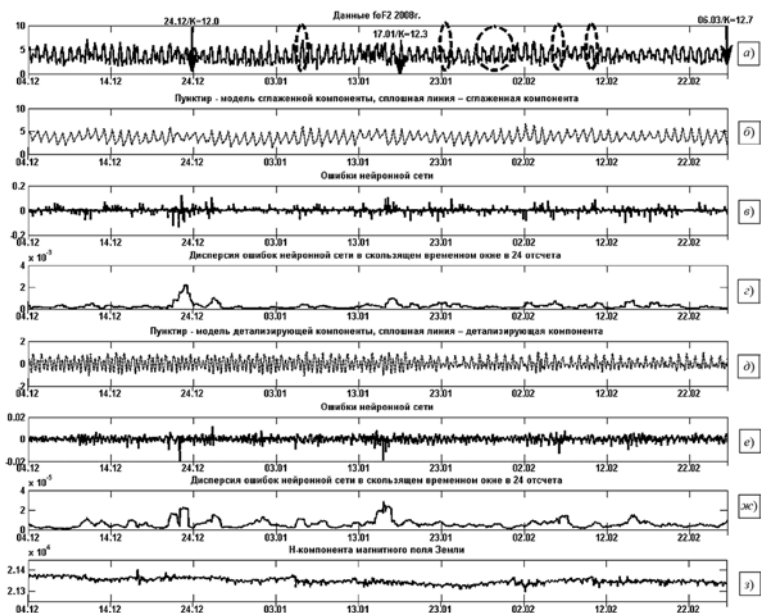


Рис. 4. Результаты обработки данных  $f_0F_2$  за период 04.12.2007—25.02.2008:

*a* — исходный временной ряд; *b*, *d* — модельные данные и исходная компонента ряда; *e*, *g* — ошибки аппроксимации сглаженной и детализирующей компонент, соответственно; *z* — Н-компонента магнитного поля Земли. Стрелками отмечены моменты возникновения землетрясений на Камчатке. Пунктиром отмечены периоды, содержащие пропуски в данных

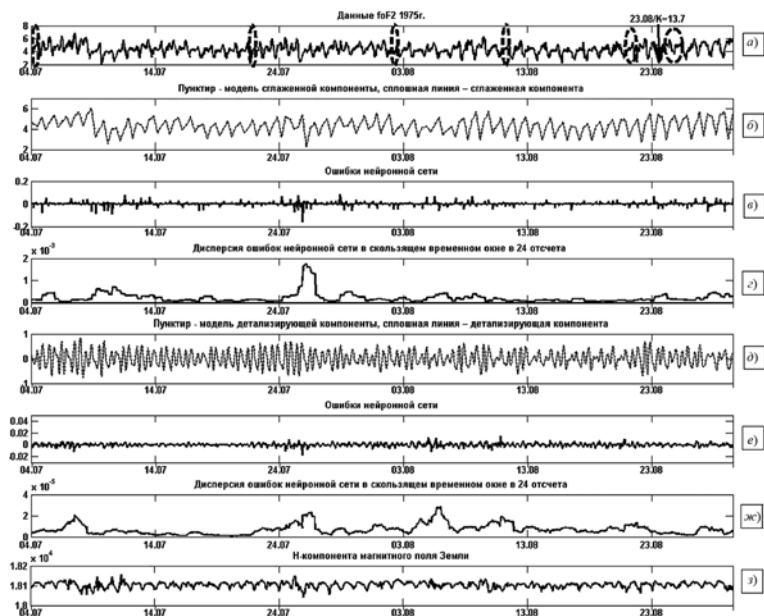


Рис. 5. Результаты обработки данных  $f_0F_2$  за период 04.07.1975—29.08.1975:

*a* — исходный временной ряд; *b*, *d* — модельные данные и исходная компонента ряда; *e*, *g* — ошибки аппроксимации сглаженной и детализирующей компонент, соответственно; *z* — Н-компонента магнитного поля Земли. Стрелками отмечены моменты возникновения землетрясений на Камчатке. Пунктиром отмечены периоды, содержащие пропуски в данных

17 января 2008 г. Представленные на рис. 5 результаты моделирования ионосферных данных за период 04.07.1975—29.08.1975 показывают, что увеличение ошибок аппроксимации анализируемых компонент наблюдается в периоды повышенной геомагнитной активности. Накануне и после землетрясения, произошедшего 23 августа 1975 г., наблюдаются пропуски в данных, что могло повлиять на возможность выявления аномалий в данный период.

## Заключение

На основе предложенного в работе способа аппроксимации временного хода ионосферных параметров выполнено моделирование и анализ данных критической частоты ионосферы и выделены разномасштабные аномалии, возникающие в периоды ионосферных возмущений. Сопоставление полученных результатов с геомагнитными данными показало зависимость интенсивности и частоты появления аномалий в ионосфере от уровня геомагнитной активности. Сопоставление результатов с данными каталога землетрясений также показало, что их возникновение может быть обусловлено повышением сейсмической активности на Камчатке.

*Работа поддержана грантом стипендии Президента Российской Федерации СП-2976.2013.5, грантом РФФИ — ДВО РАН № 11-07-98514-р\_восток\_a.*

*Данные сейсмического каталога любезно предоставлены Камчатским филиалом геофизической службы РАН (г. Петропавловск-Камчатский).*

## Список литературы

1. Афраймович Э. Л., Перевалова Н. П. GPS-мониторинг верхней атмосферы Земли. Иркутск: ГУ НУ РВХ ВСНЦ СО РАН, 2006. 480 с.
2. Демин М. Г. Ионосфера Земли. Плазменная гелиогеофизика. М.: Физматлит, 2008. Т. II. С. 92—163.
3. Мандрикова О. В., Глушкова Н. В., Полозов Ю. А. Алгоритмы выделения и анализа аномалий в параметрах критической частоты ионосферы  $f_0F_2$  на основе совмещения вейвлет-преобразования и авторегрессионных моделей // Цифровая обработка сигналов. 2013. № 1. С. 47—53.
4. Мандрикова О. В., Полозов Ю. А. Критерии выбора вейвлет-функции в задачах аппроксимации природных временных рядов сложной структуры // Информационные технологии. 2012. № 1. С. 31—36.
5. Полозов Ю. А. Метод формирования обучающего множества для нейронной сети на основе вейвлет-фильтрации // Известия вузов, Северо-Кавказский регион. Серия естественные науки. 2010. № 3. С. 12—16.
6. Краткосрочный прогноз катастрофических землетрясений с помощью радиофизических наземно-космических методов: докл. конф. М.: ОИФЗ РАН, 1998. 164 с.
7. Липеровская Е. В., Липеровский В. А., Похотелов О. А., Родкин М. В. О возмущениях в F-области ионосферы перед землетрясениями // Геофизические исследования: ред. Глико А. О. М.: Изд. ИФЗ РАН. 2006. Вып. 6. С. 51—58.



8. **Намгалдзе А. А.** Физический механизм и математическое моделирование ионосферных предвестников землетрясений, регистрируемых в полном электронном содержании // Геомагнетизм и аэронавигация. 2009. Т. 49. № 2. С. 267—277.

9. **Малла С.** Вейвлеты в обработке сигналов. Пер. с англ. М.: Мир, 2005. 672 с.

10. **Хайкин С.** Нейронные сети: полный курс. М.: Вильямс, 2006. 1104 с.

11. **Liming H. E., Lixin W. U., Shanjun L. I. U., Baodong M. A.** Seismo-ionospheric anomalies detection based on integrated wavelet // Proc. of Geoscience and Remote Sensing Symposium (IGARSS), 2011. IEEE International. 24—29 July 2011. P. 1834—1837. doi 10.1109/IGARSS.2011.6049479.

12. **Hamoudi M., Zaourar N., Mebarki R., Briquieu L., Parrot M.** Wavelet analysis of ionospheric disturbances // EGU General Assembly 2009. Held 19—24 April, Austria, 2009. P. 8523. URL: <http://meetings.copernicus.org/egu2009>.

13. **Нейроматематика:** учеб. пособие для вузов / общая ред. А. И. Галушкина. М.: ИПРЖР, 2002. 448 с.

14. **Martin J. D., Morton Yu. T., Zhou Q.** Neural network development for the forecasting of upper atmosphere parameter distributions // Advances in Space Research. 2005. Vol. 36. P. 2480—2485.

15. **Mandrikova O. V., Polozov Yu. A., Bogdanov V. V., Zhizhikina E. A.** Method of detection of abnormal features in ionosphere critical frequency data on the basis of wavelet transformation and neural networks combination // Journal of Software Engineering and Applications. 2012. Vol. 5, N. 12B. P. 181—187. doi:10.4236/jsea.2012.512b035. Published Online December 2012.

16. **Мандрикова О. В., Полозов Ю. А.** Метод выделения аномальных особенностей в данных критической частоты ионосферы на основе совмещения вейвлет-преобразования и нейронных сетей // Цифровая обработка сигналов. 2012. № 2. С. 29—35.

17. **Левин Б. Р.** Теоретические основы статистической радиотехники. Изд. 2-е. М.: Сов. радио, 1975. 392 с.

**O. V. Mandrikova**<sup>1, 2</sup>, Prof., Head of Laboratory, **Yu. A. Polozov**<sup>1, 2</sup>, Researcher, e-mail: [up\\_agent@mail.ru](mailto:up_agent@mail.ru),

<sup>1</sup> Institute of Cosmophysical Researches and Radio Wave Propagation FEB RAS

<sup>2</sup> Kamchatka State Technical University

## Approximation and Analysis of Ionospheric Parameters Based on a Combination of Wavelet Transformation and Neural Networks Groups

*The paper suggests a method of approximation and analysis of ionospheric parameter time variation based on the combination of multi-scale wavelet decompositions and neural network groups. Calculation solutions to determine ionospheric parameter time series components, to form neural networks and to combine them into groups are described. In order to approximate a series smoothed component, a three-layer neural network of signal direct transmission was developed; approximation of detailing components is carried out on the basis of neural network groups. Analyzing approximation errors in ionospheric parameters, anomalies are detected.*

*Testing of the method was carried out on ionospheric critical frequency  $f_oF_2$  data for 1969—2009, obtained at "Paratunka" observatory (IKIR FEB RAS, Paratunka, Kamchatskiy krai). On the basis of estimation of approximation errors in the recorded ionosphere critical frequency time series, anomalies were detected during increased solar activity as well as before and at the time of strong earthquakes in Kamchatka. The suggested method of ionospheric data analysis combined with other methods will allow us to increase the quality of evaluation and control of ionosphere state and detection of anomalies occurring during ionospheric disturbances.*

**Keywords:** wavelet transformation, neural networks, the critical frequency of the ionosphere, anomalies, earthquakes

### References

1. **Afraimovich Je. L., Perevalova N. P.** GPS-monitoring verhnjej atmosfery Zemli. Irkutsk: GU NU RVH VSNC SO RAMN. 2006. 480 p.

2. **Djomim M. G.** Ionosfera Zemli. Plazmennaja geliogeofizika. M.: Fizmatlit, 2008. Vol. II. P. 92—163.

3. **Mandrikova O. V., Glushkova N. V., Polozov Ju. A.** Algoritmy vydelenija i analiza anomalij v parametrah kriticheskoj chastoty ionosfery foF2 na osnove sovmeshhenija vejvlet-preobrazovanija i avtoregressivnyh modelej. *Cifrovaja obrabotka signalov*. M.: RNTORJeS. 2013. N. 1. P. 47—53.

4. **Mandrikova O. V., Polozov Ju. A.** Kriterii vybora vejvlet-funkcij v zadachah approksimacii prirodnyh vpmennyh pjadov slozhnoj struktury. *Informacionnye tehnologii*. 2012. N. 1. P. 31—36.

5. **Polozov Ju. A.** Metod formirovanija obuchajushhego mnozhestva dlja nejronnoj seti na osnove vejvlet-fil'tracii. *Izvestija vuzov, Severo-Kavkazskij region. Serija estestvennye nauki*. Rostov-na-Donu. 2010. N. 3. P. 12—16.

6. **Kratkosrochnyj prognoz katastroficheskikh zemletrjasenij s pomosh'ju radiofizicheskikh nazemno-kosmicheskikh metodov:** dokl. konf. M.: OIFZ RAN, 1998. 164 p.

7. **Lipervovskaja E. V., Liperovskij V. A., Pohotelov O. A., Rodkin M. V.** O vozmushhenijah v F-oblasti ionosfery pered zemletrjasenijami. *Geofizicheskie issledovanija*: red. Gliko A. O. Institut fiziki Zemli RAN. M.: IFZ RAN. 2006. Vol. 6. P. 51—58.

8. **Namgaladze A. A.** Fizicheskij mehanizm i matematicheskoe modelirovanie ionosfernyh predvestnikov zemletrjasenij, registriruemyh v polnom jelektronnom sodержanii. *Geomagnetizm i ajeronomija*. 2009. Vol. 49, N. 2. P. 267—277.

9. **Malla S.** Vjevlyety v obrabotke signalov. Per. s angl. M.: Mir, 2005. 671 p.

10. **Hajkin S.** Nejronnye seti: polnyj kurs. M.: Vil'jams, 2006. 1104 p.

11. **Liming H. E., Lixin W. U., Shanjun L., Baodong M. A.** Seismo-ionospheric anomalies detection based on integrated wavelet. *Geoscience and Remote Sensing Symposium (IGARSS)*, 2011. IEEE International. 24—29 July 2011. P. 1834—1837. Doi 10.1109/IGARSS.2011.6049479.

12. **Hamoudi M., Zaourar N., Mebarki R., Briquieu L., Parrot M.** Wavelet analysis of ionospheric disturbances. *EGU General Assembly 2009*, held 19—24 April, 2009, Austria. P. 8523. URL: <http://meetings.copernicus.org/egu2009>.

13. **Нейроматематика:** учеб. пособие длja vuzov / obshhaja red. А. И. Galushkin. M.: IPRZhR, 2002. 448 p.

14. **Martin J. D., Morton Yu. T., Zhou Q.** Neural network development for the forecasting of upper atmosphere parameter distributions. *Advances in Space Research*. 2005 Vol. 36. P. 2480—2485.

15. **Mandrikova O. V., Polozov Yu. A., Bogdanov V. V., Zhizhikina E. A.** Method of detection of abnormal features in ionosphere critical frequency data on the basis of wavelet transformation and neural networks combination. *Journal of Software Engineering and Applications*. 2012. Vol. 5, N. 12B. P. 181—187. Doi:10.4236/jsea.2012.512b035. Published Online December 2012.

16. **Mandrikova O. V., Polozov Ju. A.** Metod vydelenija anomal'nyh osobennostej v dannyh kriticheskoj chastoty ionosfery na osnove sovmeshhenija vejvlet-preobrazovanija i nejronnyh setej. *Cifrovaja obrabotka signalov*. 2012. N. 2. P. 29—35.

17. **Levin B. R.** Teoreticheskie osnovy statisticheskoj radiotekhniki. Izd. 2. M.: Sov. Radio, 1975. 392 p.

**А. И. Галушкин**, д-р техн. наук, проф., нач. лаборатории, e-mail: neurocomputer@yandex.ru,  
 Центр информационных технологий и систем органов исполнительной власти

## Метод обратного распространения ошибки и российские работы по теории нейронных сетей

*Отмечается значительная роль работ Пауля Вербоса [1, 2] и авторов [3–5] в развитии и применении нейросетевых технологий, место российских работ в области теории нейронных сетей, а также перспективы обоих направлений в разработке методов и алгоритмов настройки многослойных нейронных сетей и их реализацией в виде нейрокомпьютеров с применением мемристоров.*

**Ключевые слова:** многослойные нейронные сети, теория нейронных сетей, алгоритмы настройки многослойных нейронных сетей, нейрокомпьютеры с применением мемристоров, метод обратного распространения

### Введение

Особый интерес к нейронным сетям у автора появился после изучения основополагающей монографии Ф. Розенблатта [6] и посещения семинаров академика П. К. Анохина в начале 60-х годов прошлого века. Еще студентом кафедры "Системы автоматического управления" МВТУ имени Н. Э. Баумана кроме общих курсов по следящим системам, теории управления и других автор слушал курсы по принципу максимума Понтрягина, динамическому программированию Беллмана, статистической теории линейных систем и другим дисциплинам. Изучая результаты исследовательских работ своих учителей Л. Т. Кузина в области дискретных систем управления [7], В. В. Солодовникова в области поисковых и аналитических самонастраивающихся систем [8], Я. З. Цыпкина в области адаптивных систем [9], автор в середине 60-х годов прошлого столетия сформировал у себя представление о нейронной сети как частном виде объекта управления (многомерного, многосвязного, нелинейного, с переменной структурой), к которому можно было бы применить методы управления, разработанные его учителями.

Недаром монографии [10–12] автор посвятил своим учителям в области теории управления, аналитических самонастраивающихся систем, дискретных систем управления Солодовникову Владимиру Викторовичу, Кузину Льву Тимофеевичу и Цыпкину Якову Залмановичу.

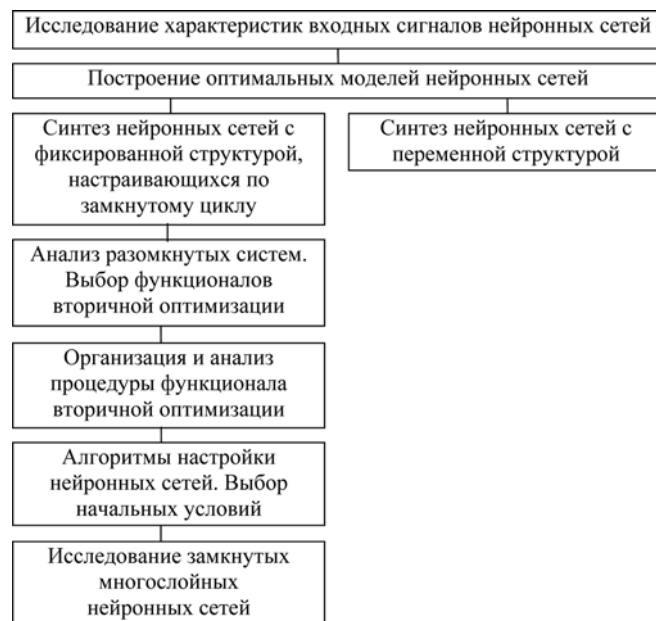
### 1. О системном подходе к синтезу алгоритмов настройки коэффициентов в многослойных нейронных сетях

Во второй половине 60-х годов прошлого века в России родилась методика синтеза нейронных сетей, которая развивалась и успешно применялась в течение последующих почти 50 лет. Общая структура этой методики представлена на рис. 1. Основной целью данной методики является представление тем, кто решает какую-либо задачу в нейросетевом логическом базисе, способов синтеза нейро-

сетевого алгоритма, адекватного именно решаемой задаче. При этом процесс решения задачи представляется в виде динамического процесса настройки коэффициентов нейронной сети определенной структуры с определением топологии данной сети (числа слоев, числа элементов в слое, характере связей и структуры).

Необходимо отметить, что на формирование, развитие и применение нейросетевых технологий в России в 70-е и последующие годы не повлияла работа американских авторов [13], которая практически на 15 лет затормозила развитие в США нейросетевых технологий, так как авторы не нашли в то время эффективного математического аппарата решения задач в нейросетевом логическом базисе.

Основа системного подхода к синтезу алгоритмов настройки коэффициентов в многослойных нейронных сетях была заложена в России в работах [14–19].



**Рис. 1.** Системный подход к синтезу многослойных нейронных сетей

## 2. Нейронные сети — частный вид объекта управления в адаптивной системе с аналитическими методами настройки коэффициентов (адаптации)

В начале 60-х годов прошлого столетия адаптивные системы строились на основе двух основополагающих принципов:

- поисковые, когда градиент функционала оптимизации вычислялся в системе путем введения искусственных поисковых колебаний настраиваемых параметров;
- аналитические, когда градиент функционала оптимизации и, следовательно, направление перестройки коэффициентов вычислялись аналитически через входные, выходные и промежуточные сигналы в системе.

В работах [14, 19] были предложены и исследованы поисковые алгоритмы настройки коэффициентов в нейронных сетях. Однако они не получили должного развития по следующим причинам:

- поисковые колебания вносят в систему определенные дополнительные шумы;
- при усложнении структуры нейронных сетей резко увеличивается число настраиваемых коэффициентов;
- аналитические методы настройки показали свою эффективность.

Нейронные сети явились в теории управления одним из первых примеров перехода от управления простейшими линейными стационарными системами к управлению сложными нелинейными, нестационарными, многомерными, многосвязными системами.

На рис. 2 показано то каноническое представление структуры нейронной сети, которое было принято в середине 60-х годов прошлого века для решения задач и для разработки алгоритмов настройки коэффициентов. Это были нейронные сети с полными (или неполными по какой-либо причине) последовательными, перекрестными и обратными связями.

Отметим, что введение обратных связей и, как следствие, разработка алгоритмов настройки их коэффициентов в 60-е, 70-е, 80-е годы прошлого столетия имели чисто теоретический смысл, так как не

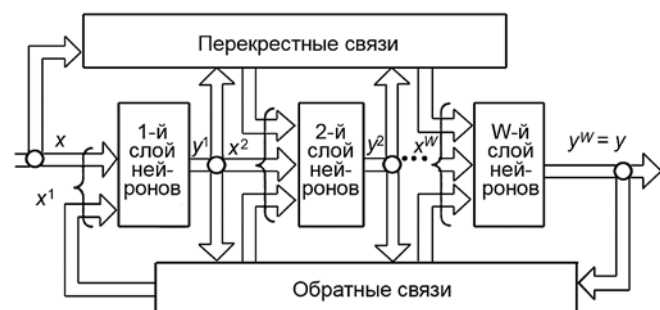


Рис. 2. Многослойная нейронная сеть (каноническое представление 60-х годов XX в.)

было практических задач, адекватных таким структурам. Лишь в конце 80-х — начале 90-х годов стали появляться такие задачи и простейшие структуры с настраиваемыми обратными связями для их решения (так называемые рекуррентные нейронные сети).

Иначе в 60-е и 70-е годы прошлого столетия рассматривались перекрестные связи. В течение всей своей 50-летней деятельности в области нейросетевых технологий автор занимался не только разработкой алгоритмов настройки многослойных нейронных сетей и нейросетевыми алгоритмами решения различных задач, но и наиболее эффективными (на текущий момент развития технологии электроники) аппаратными эмуляторами нейросетевых алгоритмов.

В 60-е годы прошлого столетия до появления микропроцессора наиболее эффективными эмуляторами нейронных сетей были аналоговые реализации разомкнутых нейронных сетей с реализацией алгоритмов настройки на универсальных ЭВМ [20], иногда системы на адаптивных элементах с аналоговой памятью [21, 22]. Такой уровень развития электроники делал актуальным введение перекрестных связей в структуры нейронных сетей, поскольку, как показано в [10—12, 19], это приводило к значительному уменьшению числа нейронов в нейронной сети при сохранении качества решения задачи (например, дискриминантной способности при решении задач распознавания образов). Исследования 60-х и 70-х годов прошлого столетия в области оптимизации структур нейронных сетей с перекрестными связями наверняка найдут развитие при реализации мемристорных систем с учетом их специфики в части аналогово-цифровой обработки информации.

Специфические требования прикладных задач определяли некоторые особенности структур нейронных сетей, являющихся объектом управления с помощью специфических алгоритмов настройки, представленных в работах [10—12, 19]. К таким особенностям следует отнести:

- *континуум числа классов*, когда указание учителя системы формируется в виде непрерывного значения функции в некотором диапазоне изменения;
- *континуум решений многослойной нейронной сети*, формируемый выбором континуальной функции активации нейрона последнего слоя;
- *континуум числа признаков*, формируемый переходом в пространстве признаков от представления выходного сигнала в виде  $N$ -мерного вектора вещественных чисел к вещественной функции в некотором диапазоне изменения аргумента. Континуум числа признаков, как следствие, требует специфической программной и аппаратной реализации нейронной сети;
- *континуум числа нейронов в слое*.

Реализация многослойных нейронных сетей с континуумами классов и решений проводится выбором соответствующих видов функций активации

Таблица 1

## Классификация нейронных сетей

| Пространство (число) решений | Входной сигнал       |      |                   |
|------------------------------|----------------------|------|-------------------|
|                              | Два класса           |      | Континуум классов |
| Два                          | 1                    |      | 8                 |
| $K_p$                        | $K_p = 3$            | $3a$ | $K < K_p$ 9       |
|                              |                      |      | $K = K_p$ 2       |
|                              | $K_p = \text{const}$ | $3b$ | $K > K_p$ 4       |
| Континуум                    | 5                    |      | 11                |

нейронов последнего слоя. Вариант континуума признаков входного пространства был реализован в задаче распознавания периодических сигналов без преобразования их с помощью аналого-цифрового преобразователя (АЦП) на входе системы, а также реализацией аналого-цифровой многослойной нейронной сети [20].

В табл. 1 представлен систематизированный набор вариантов алгоритмов настройки (обозначенных арабскими цифрами) многослойных нейронных сетей в пространстве "Входной сигнал — пространство решений", разработанных и представленных в работе [15].

Основными преимуществами нейронных сетей как логического базиса алгоритмов решения сложных задач являются следующие:

- инвариантность методов синтеза нейронных сетей от размерности пространства признаков;
- возможность выбора структуры нейронных сетей в значительном диапазоне параметров в зависимости от сложности и специфики решаемой задачи в целях достижения требуемого качества решения;
- адекватность текущим и перспективным технологиям микроэлектроники;
- отказоустойчивость в смысле его монотонного, а не катастрофического изменения качества решения задачи в зависимости от числа вышедших из строя элементов.

### 3. О входном сигнале нейронных сетей

Вероятностная модель окружающего мира является основой нейросетевых технологий. Подобная модель является основой математической статистики. Нейронные сети возникли как раз в то время, когда экспериментаторы, использующие методы математической статистики, задали себе вопрос: А почему мы обязаны описывать функции распределения входных случайных сигналов в виде конкретных аналитических выражений (нормальное распределение, распределение Пуассона и т. д.). Если это правильно и на это есть какая-то физическая причина, то задача обработки случайных сигналов становится достаточно простой.

Специалисты по нейросетевым технологиям сказали: "Мы ничего не знаем о функции распределения входных сигналов, мы отказываемся от необходимости формального описания функции распределения входных сигналов, даже если мы сузим класс решаемых задач. Мы считаем функции распределения входных сигналов сложными, неизвестными и будем решать частные конкретные задачи в условиях подобной априорной неопределенности".

Именно поэтому нейронные сети в начале 60-х годов прошлого столетия эффективно применяются при решении задач распознавания образов. Причем задача распознавания образов трактуется как задача аппроксимации многомерной случайной функции, принимающей  $K$  значений, где  $K$  — число классов образов.

Ниже отмечены некоторые режимы работы многослойных нейронных сетей, определяемые характеристиками случайных входных сигналов, для которых еще в конце 60-х годов прошлого столетия были разработаны алгоритмы настройки коэффициентов.

**Априорные вероятности появления классов.** При всей недостаточности априорной информации о функциях распределения входных сигналов неучет некоторой полезной информации может привести к потере качества решения задачи. Это в первую очередь касается априорных вероятностей появления классов. В работах [10—12, 19] представлены алгоритмы настройки многослойных нейронных сетей с учетом имеющейся информации об априорных вероятностях появления классов. Это имеет место в таких задачах, как распознавание букв в тексте, когда для данного языка вероятность появления каждой буквы известна, и эту информацию необходимо использовать при построении алгоритма настройки коэффициентов многослойной нейронной сети.

**О квалификации учителя.** В задачах распознавания образов, как правило, по умолчанию квалификация учителя является полной, т. е. вероятность правильного отнесения учителем образов к тому или иному классу равна единице. На практике, при наличии косвенных измерений, это зачастую не соответствует действительности, например, в задачах медицинской диагностики, когда при верификации архива медицинских данных, предназначенных для обучения системы медицинской диагностики, квалифицированным врачом или консилиумом врачей вероятность отнесения медицинских данных к тому или иному заболеванию не равна единице.

Введение понятия конечной квалификации учителя видоизменяет алгоритм настройки коэффициентов многослойной нейронной сети [15]. В общем случае квалификация учителя может быть различной для различных классов образов.

Введение понятия квалификации учителя в [15] позволило разработать единые алгоритмы настройки коэффициентов многослойных нейронных сетей для режимов обучения, обучения с учителем, обладающим конечной квалификацией, и

самообучения (кластеризации), когда при наличии  $k$  или двух классов образов квалификация учителя (вероятность отнесения образов к тому или иному классу) равна  $1/k$  или  $1/2$ .

Введение понятия квалификации учителя в системах распознавания образов позволило чисто теоретически рассмотреть режимы "вредительства" системе, когда системе сообщается заведомо ложное (с различной степенью ложности) отнесение образов к тому или иному классу. Данный режим настройки коэффициентов многослойной нейронной сети пока не нашел практического применения.

**Кластеризация.** Кластеризация (самообучение) — это частный режим работы многослойных нейронных сетей, когда системе не сообщается информация о принадлежности образов к тому или иному классу. Кластеризации (самообучению) посвящено множество научных работ. Основная задача кластеризации заключается в обработке множества векторов в многомерном пространстве признаков с выделением компактных подмножеств (подмножеств, близко расположенных друг к другу), их количества и свойств.

Наиболее распространенным методом кластеризации является метод " $K$ -means", практически не связанный с методами обратного распространения и не обобщаемый на архитектуры типа многослойных нейронных сетей.

Введение понятия квалификации учителя и единого подхода к обучению и самообучению в работах [16—18] позволило фактически создать основу для реализации режима кластеризации в многослойных нейронных сетях широкого класса структур.

**Нестационарные образы.** Существующие разработки в области систем распознавания образов на базе многослойных нейронных сетей в основном относятся к стационарным образам, т. е. к случайным входным сигналам, имеющим сложные неизвестные, но стационарные во времени функции распределения. В работах [16, 17] впервые сделана попытка распространить предлагаемую методику настройки многослойных нейронных сетей на нестационарные образы, когда предполагаемая неизвестная функция распределения входного сигнала зависит от времени или входной случайный сигнал является суперпозицией регулярной составляющей и случайной составляющей с неизвестной сложной функцией распределения, не зависящей от времени.

#### 4. О критериях первичной оптимизации в многослойных нейронных сетях

Вероятностная модель мира, взятая за основу при построении алгоритмов адаптации в многослойных нейронных сетях, позволила формировать критерии первичной оптимизации в рассматриваемых системах в виде требований минимума средней функции риска и его модификаций [15, 18]:

- максимум апостериорной вероятности;

- минимум средней функции риска;
- минимум средней функции риска при условии равенства условных функций риска для различных классов;
- минимум средней функции риска при условии заданного значения условной функции риска для одного из классов;
- другие критерии первичной оптимизации, вытекающие из требований конкретной практической задачи.

В работах [10—12, 18, 19] были представлены модификации алгоритмов настройки многослойных нейронных сетей для указанных выше критериев первичной оптимизации. Отметим, что в подавляющем большинстве работ в области теории нейронных сетей и в алгоритмах обратного распространения рассматривается простейший критерий — минимума среднеквадратической ошибки без каких бы то ни было ограничений на условные функции риска.

В режиме самообучения (кластеризации) предпосылкой формирования критерия и функционала первичной оптимизации нейронных сетей служит представление функции распределения входного сигнала в виде многомодальной функции в многомерном пространстве признаков, где каждой моде с некоторой вероятностью соответствует класс. В качестве критериев первичной оптимизации в режиме самообучения (кластеризации) использовались модификации средней функции риска.

В работах [10—12, 19] представленные выше модификации критериев первичной оптимизации были обобщены (при произвольной квалификации учителя) на случаи:

- континуума классов и решений;
- континуума признаков входного пространства;
- континуума числа нейронов в слое.

Важным разделом формирования критерия и функционала первичной оптимизации в многослойных нейронных сетях при вероятностной модели мира является выбор матрицы потерь, которая в теории статистических решений определяет коэффициент потерь  $l_{1,2}$  при ошибочном отнесении образов 1-го класса ко 2-му, и коэффициент потерь  $l_{2,1}$  при отнесении образов 2-го класса к 1-му. Как правило, по умолчанию матрица  $L$  этих коэффициентов при синтезе алгоритмов настройки многослойных нейронных сетей, в том числе и при применении метода обратного распространения, принимается симметричной, т. е.  $l_{1,2} = l_{2,1}$ . На практике это не соответствует действительности. Характерным примером является система обнаружения мин с применением геолокатора. В этом случае потери при ошибочном отнесении камня к mine равнозначны некоторой небольшой потере времени пользователем геолокатора. Потери, обусловленные ошибочным отнесением мины к классу камней, связаны с жизнью или значительной потерей здоровья пользователями геолокатора.

В работах [10–12, 19] представлены алгоритмы настройки многослойных нейронных сетей для произвольных, в том числе недиагональных матриц потерь.

### 5. Анализ разомкнутых нейронных сетей

Данный этап синтеза ставит своей целью определение в общем виде статистических характеристик выходных и промежуточных сигналов нейронных сетей как многомерных, нелинейных объектов управления для дальнейшего формирования критерия и функционала *вторичной* оптимизации, т. е. функционала, реально оптимизируемого алгоритмом адаптации в конкретной нейронной сети. В подавляющем большинстве работ в качестве такого функционала принимается среднеквадратическая ошибка, что ухудшает качество решения или вообще не соответствует задаче оптимизации, поставленной критерием первичной оптимизации.

В работах [10–12, 19] представлена методика и алгоритмы формирования функционала вторичной оптимизации, соответствующего заданному критерию первичной оптимизации из числа приведенных выше в п. 4.

### 6. Алгоритмы поиска экстремума функционалов вторичной оптимизации

Алгоритм поиска экстремума применительно к конкретному функционалу вторичной оптимизации определяет алгоритм настройки коэффициентов многослойной нейронной сети.

В настоящее время наибольший практический интерес представляют подобные алгоритмы, реализованные в системе MATLAB (Neural Network Toolbox) [24].

Однако необходимо отметить частность алгоритмов адаптации в многослойных нейронных сетях, используемых в системах MATLAB (Neural Network Toolbox) версия 2013 г., и ориентацию этих алгоритмов не на специфику решаемых задач, а на воображаемую "геометрию" функционалов вторичной оптимизации [24].

Алгоритмы, приведенные в табл. 2, не учитывают многих деталей специфики применения многослойных нейронных сетей при решении конкретных задач, и, естественно, требуют коренной, если не принципиальной переработки при переходе к мемристорным нейронным системам.

В работе [26] проведен детальный сравнительный анализ метода обратного распространения [1–5] и российских методов 60-х–70-х годов прошлого века. Основная особенность данных алгоритмов заключается в необходимости поиска локальных и зачастую глобального экстремумов многоэкстремального функционала в многомерном пространстве настраиваемых коэффициентов нейронной сети. Рост размеров нейронной сети ведет

к значительному росту числа настраиваемых коэффициентов, т. е. к росту размерности пространства поиска. Еще в 60-х годах прошлого века в работах [14, 19] предлагались поисковые и аналитические процедуры расчета градиента функционала вторичной оптимизации, а в классе аналитических процедур предлагалось и исследовалось применение для организации поиска не только первой, но и второй производной функционала вторичной оптимизации. Специфика многоэкстремальности функционала вторичной оптимизации привела в течение последующих десятилетий к появлению различных модификаций методов поиска (генетические алгоритмы и т. п.).

Объединение в одном коллективе в 60-е годы прошлого столетия и в последующие десятилетия групп, занимающихся теорией нейронных сетей, нейроматематикой и аппаратной реализацией нейроматематических алгоритмов привело к созданию алгоритмов поиска экстремумов функционалов вторичной оптимизации с ограничениями на величину, скорости и другие параметры весовых коэффициентов нейронных сетей.

Именно эти методы должны быть основой работ по методам настройки нейронных сетей с применением мемристоров с учетом таких специфических характеристик мемристоров (весовых коэффициентов), как передаточные функции.

Необходимо отметить еще одно, на первый взгляд незначительное, отличие алгоритмов обратного распространения от алгоритмов, разработанных и представленных в работах [10–12, 18, 19]. Условно их различие показано на рис. 3.

Таблица 2  
Алгоритмы, реализованные в системе MATLAB (Neural Network Toolbox)

| Функция  | Алгоритм                                      |
|----------|---|
| trainlm  | Levenberg-Marquardt                           |
| trainbr  | Bayesian Regularization                       |
| trainbfg | BFGS Quasi-Newton                             |
| trainrp  | Resilient Backpropagation                     |
| trainscg | Scaled Conjugate Gradient                     |
| traincgb | Conjugate Gradient with Powell/Beale Restarts |
| traincgf | Fletcher-Powell Conjugate Gradient            |
| traincgp | Polak-Ribiere Conjugate Gradient              |
| trainoss | One Step Secant                               |
| traingdx | Variable Learning Rate Gradient Descent       |
| traingdm | Gradient Descent with Momentum                |
| traingd  | Gradient Descent                              |

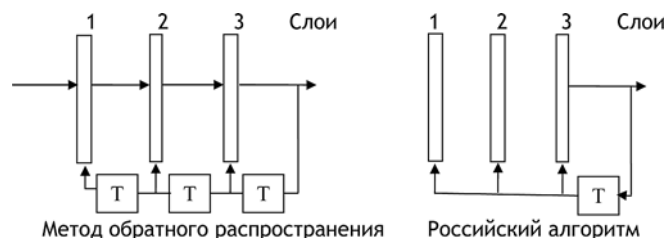


Рис. 3. Иллюстрация различия метода обратного распространения и методов, представленных в работах [10–12, 18, 19]

Метод обратного распространения реализуется рекуррентным уравнением, причем аргументом рекуррентного уравнения является номер слоя многослойной нейронной сети (рекурсии выполняются в направлении убывания номера слоя — от последнего к первому). Российская методика реализует целиком параллельный алгоритм изменений весовых коэффициентов всех слоев.

В будущем это в значительной степени будет влиять на реализацию методов распараллеливания алгоритмов настройки коэффициентов многослойной нейронной сети с помощью перспективных суперЭВМ.

### 7. Начальные условия при настройке коэффициентов

Выбор начальных условий итерационной процедуры поиска экстремумов функционалов вторичной оптимизации является важным этапом синтеза алгоритмов настройки многослойных нейронных сетей. Качественное решение этой задачи в значительной степени может сократить время настройки.

Априорная сложность функционала вторичной оптимизации сделала необходимой введение процедуры выбора начальных условий в виде случайных значений коэффициентов с повторением этой процедуры и процедуры настройки коэффициентов. Эта процедура еще в 60-е годы прошлого столетия казалась чрезвычайно избыточной с точки зрения времени, затрачиваемого на настройку коэффициентов. Однако несмотря на это она достаточно широко применяется и в настоящее время.

Для отдельных задач еще в 60-е годы прошлого века была принята идея выбора начальных условий, специфических для данной решаемой задачи. Такая процедура была отработана для трех задач [15, 17—19]:

- распознавание образов;
- кластеризация;
- нейроидентификация нелинейных динамических объектов.

В принципе выбор начальных условий должен проводиться специфически для каждой задачи, решаемой нейронной сетью, и быть неотъемлемой составляющей общей процедуры синтеза алгоритмов настройки многослойных нейронных сетей.

### 8. Память фильтра в контуре настройки коэффициентов

Системный подход к построению алгоритмов поиска экстремума функционала вторичной оптимизации предполагает в качестве одного из режимов настройки перенастройку коэффициентов в каждом такте поступления образов на входе по текущему значению градиента функционала вторичной оптимизации. В работах [10—12, 19] представлены алгоритмы настройки многослойных нейронных сетей с

*фильтрацией* последовательности значений градиентов функционала вторичной оптимизации:

- фильтром нулевого порядка с памятью  $m_n$  (для стационарных образов);
- фильтром 1, ...,  $k$ -го порядка с памятью  $m_n$  (для нестационарных образов) с различной гипотезой изменения во времени функций распределения для образов различных классов.

### 9. Исследование алгоритмов адаптации в нейронных сетях

Главный вопрос — как выбрать структуру многослойной нейронной сети для решения выбранной конкретной задачи — до сих пор в значительной степени не решен. Можно предложить лишь структуру разумного направленного перебора вариантов структур с оценкой их эффективности в процессе решения задачи.

Однако оценка качества работы алгоритма настройки на конкретной выбранной структуре, т. е. на конкретной задаче, может быть недостаточно корректной.

Так, для оценки качества работы линейных динамических систем управления применяются типовые входные сигналы (ступенчатый, квадратичный и т. д.), по реакции на которые оценивается установившаяся ошибка (астатизм системы) и ошибки в переходных процессах.

Подобно этому для многослойных нейронных сетей еще в 60-е годы прошлого столетия были разработаны [19] типовые входные сигналы для проверки и сравнения работоспособности различных алгоритмов настройки.

Естественно, что типовые входные сигналы для таких объектов, как многослойные нейронные сети, являются специфическими для каждой решаемой задачи. В работах [10—12, 19] представлены типовые входные сигналы для следующих задач:

- распознавание образов;
- кластеризация;
- нейрорегулирование динамическими объектами.

Основным аксиоматическим принципом применения нейросетевых технологий вместо методов классической математической статистики является отказ от формализованного описания функций распределения вероятностей для входных сигналов и принятие концепции неизвестных, сложных функций распределения. Именно по этой причине в работах [10—12, 19] были предложены следующие типовые входные сигналы.

Для задачи кластеризации была предложена выборка случайного сигнала с многомодальным распределением, реализуемая в  $N$ -мерном пространстве признаков с модами функции распределения, центры которых в количестве  $s$  размещаются на гипербиссектрисе  $N$ -мерного пространства признаков. Каждая мода реализует составляющую случайной выборки с нормальным распределением и

среднеквадратичным отклонением  $\sigma$ , равным для всех из мод  $z$ . Предметом сравнения различных методов кластеризации будет динамика настройки и качество решения задачи в зависимости от  $N$ ,  $z$  и  $\sigma$  при достаточно большой случайной выборке  $M$ . Этот подход можно считать одним из первых достаточно объективных подходов к сравнению алгоритмов кластеризации, в том числе основанных на многослойных нейронных сетях [10–12, 19] с соответствующим выбором структуры для достижения необходимого качества кластеризации.

Для задач классификации входные сигналы для испытаний аналогичны сигналам для кластеризации с тем изменением, что выборка с многомодальным распределением делится на 2 (в случае двух классов) или  $K$  (в случае  $K$  классов) частей с перемежающимися модами функции распределения для отдельных классов.

### 10. Нейронные сети с переменной структурой

Отказ в нейросетевых технологиях от априорной информации, от информации о функциях распределения входных сигналов приводит к необходимости реализации разумного перебора параметров структуры многослойных нейронных сетей для обеспечения необходимого качества решения задачи.

В 60-е годы прошлого столетия для весьма актуального в то время класса задач — распознавания образов — была предложена [14, 19] процедура настройки многослойных нейронных сетей, в которой структура априори не фиксируется, а является результатом настройки наряду со значениями настраиваемых коэффициентов. При этом в процессе настройки выбираются число слоев и число нейронов в слоях.

Процедура настройки коэффициентов многослойной нейронной сети с переменной структурой легко переносится с задачи распознавания двух классов образов на задачу распознавания  $K$  классов образов. Причем здесь результатом настройки являются  $K$  нейронных сетей, в каждой из которых первым классом является  $k$ -й класс ( $k = 1, \dots, K$ ), а вторым — все остальные.

Подобная идея настройки многослойных нейронных сетей с переменной структурой применима и к решению задачи кластеризации. При этом в качестве первого класса образов принимается исходная анализируемая выборка, а в качестве второго класса — выборка с равномерным распределением в диапазоне изменения признаков.

При этом реализуемая в процессе настройки многослойная нейронная сеть с переменной структурой качественно и количественно отражает сложность решения задачи. С этой точки зрения задача кластеризации, как задача рождения новых знаний об изучаемом объекте, заключается в выделении и анализе тех областей многомерного пространства признаков, в которых функция распределения вероятностей превышает уровень равно-

мерного распределения в диапазоне изменения величин признаков.

### 11. Нейроматематика и нейроуправление

Эти два важных раздела нейросетевых технологий обусловлены развитием теории нейронных сетей. *Нейроматематика* — раздел вычислительной математики, связанный с развитием сложных формализуемых и неформализуемых задач в нейросетевом логическом базисе. В работах [27–30] показано, что в последние 20–30 лет нейросетевые технологии применялись для решения практически всех формализуемых задач при превышении ими некоторого порога сложности, в частности размерности. В настоящее время нейросетевые технологии являются основными, применяемыми для решения неформализуемых задач.

*Нейроуправление* — одно из направлений нейросетевых технологий — становятся главным способом решения задач идентификации и управления сложными динамическими объектами (нелинейными, многомерными, с переменными параметрами) [31].

### 12. Нейрокомпьютеры с использованием мемристов и перспективы развития теории нейронных сетей

Главной целью российской научной школы в области нейросетевых технологий является не создание набора нейросетевых парадигм, как в пакете программ Neural Network Toolbox, а отработка методики, которая для каждой решаемой задачи позволяет сформировать нейросетевой алгоритм, адекватный именно решаемой задаче. Это в значительной степени касается распространения разработанных в [10–12, 19] методов настройки многослойных нейронных сетей на нейрокомпьютеры с применением мемристов и на задачи нейроматематики и нейроуправления, решаемые с помощью нейрокомпьютеров с применением мемристов [32–34].

При разработке методов настройки многослойных нейронных сетей для нейрокомпьютеров с применением мемристов должен быть учтен опыт создания алгоритмов настройки многослойных нейронных сетей с ограничениями:

- на весовые коэффициенты;
- на априорные вероятности появления классов;
- на квалификацию учителя;
- на матрицы потерь;
- на условные функции риска;
- на размерность (точность) представления весовых коэффициентов с возможностью адаптивного управления разрядностью (точностью);
- другие ограничения.

На протяжении всей истории развития вычислительной техники (однопроцессорные ЭВМ, транспьютерные системы, системы на базе графических процессоров) постоянно делались попытки



моделирования функций мозга и создавались специализированные нейрокомпьютеры.

Развитие теории нейронных сетей, нейроматематики и нейрорегуляции для нейрокомпьютеров с применением мемристоров позволит сделать эти вычислительные системы достаточно универсальными.

Причем одной из основных задач в теории нейронных сетей станет переход в многослойной нейронной сети от передаточной функции в простейшем случае:

$$y = f \sum a_i f \sum a_j f \sum a_k x_k$$

к передаточной функции

$$y = f \sum a_i(z) f \sum a_j(z) f \sum a_k(z) x_k,$$

где  $f$  — функция активации;  $a_i, a_j, a_k$  — матрицы весовых коэффициентов;  $x_k$  — входной сигнал нейронной сети;  $a(z)$  — передаточная функция мемристора.

В этом плане важными становятся задачи построения моделей мемристоров и мемристорных систем на FPGA и суперЭВМ на базе графических процессоров.

### Заключение

В своих монографиях [11, 12] автор изложил результаты работ коллектива за несколько десятилетий в области теории нейронных сетей и сердечно благодарен трем ведущим ученым: Лотфи Заде, Роберту Хехт Нильсену и Shun-ichi Amagi за высокую оценку наших работ.

Принципиальные отличия российских [10—12, 14—19] и зарубежных [1—5] методов настройки многослойных нейронных сетей на период их зарождения в 60-е, 70-е и 80-е годы прошлого столетия собраны в табл. 3. Конечно, за рубежом предложения этих лет были в значительной мере развиты. Однако сейчас на очередном этапе развития нейросетевых технологий, связанном с нанотехнологиями и разработками мемристорных систем, с нашей точки зрения, именно российский опыт может служить фундаментом для разработки методов настройки, нейроматематики и нейрорегуляции для нейрокомпьютеров с применением мемристоров.

Теория нейронных сетей является в настоящее время самостоятельным направлением науки. Основные перспективные направления теории нейронных сетей связаны, конечно, с решением наиболее сложных практических задач, среди которых можно отметить следующие:

- континуальные нейронные сети, в которых на формальном уровне рассматривается континуум, например, числа входных каналов или нейронов в слоях;
- исследование структурной и параметрической надежности нейронных сетей применительно к конкретной технологии реализации нейрокомпьютера;

- распараллеливание нейросетевых алгоритмов на различные типы коммутационных ядер в супернейрокомпьютерах;
- нейронные сети, обеспечивающие инвариантность к группе преобразований (например, к сдвигу, повороту, изменению масштаба изображения или сигнала);
- аналитическое описание нейронных сетей с контуром адаптации с помощью аппарата линейных последовательностных машин Гилла и многие другие.

К большому сожалению, видимо, из-за плохой информированности, возникает достаточно большое число "доморощенных" нейросетевых алгоритмов.

Получение с их помощью первых положительных результатов может создать впечатление о "законченности" теории нейронных сетей, хотя она находится на начальном этапе своего развития.

Число научных работ в области теории нейронных сетей возрастает. Именно поэтому требуют аналитического подхода сравнение и детальная классификация различных подходов к решению задач синтеза нейронных сетей. В первую очередь это нужно делать, сравнивая априорную информацию, необходимую для синтеза многослойных нейронных сетей в каждом конкретном случае, а именно:

- априорные характеристики пространства "указаний учителя" нейронной сети — число классов образов ( $2, K$ , континуум);
- априорные характеристики нестационарности входного сигнала нейронной сети;
- функция "квалификации учителя" нейронной сети — двух аргументов, являющихся индексами соответствующих классов;
- функция "собственного мнения учителя" нейронной сети о своих способностях (это также функция двух аргументов, являющихся индексами соответствующих классов);
- априорные вероятности появления классов;
- априорные характеристики пространства решений нейронной сети ( $2, K$ , континуум решений);
- класс критериев первичной оптимизации нейронной сети;
- функция потерь, возникающих при отнесении системой образов одного класса к другому;
- априорная информация об условных функциях распределения  $f'(x/\varepsilon)$ ;
- априорная информация о фиксированной структуре разомкнутой нейронной сети при построении нейронной сети с фиксированной структурой, настраиваемой по замкнутому циклу;
- априорная информация о классе структур при построении нейронной сети с переменной структурой;
- априорная информация об отличии функционалов первичной и вторичной оптимизации при построении нейронной сети с фиксированной

Сравнение методов настройки многослойных нейронных сетей

| № п.п. | Признак методики синтеза нейронных сетей   | Российские методы адаптации в многослойных нейронных сетях  | Метод обратного распространения   | Примечание   |
|--------|--|---|---|--|
| 1      | Срок разработки и опубликования  | 1965—1974 гт.   | 1976—1984 гт.   |  |
| 2      | Число классов образов (градаций по уровню сигнала указанному классу)   | 2, К, континуум   | 2   |  |
| 3      | Характеристика стационарности входных образов как случайных сигналов   | Стационарная, нестационарная  | Стационарные  |  |
| 4      | Характеристика "квалификации учителя"  | Произвольная  | Обучение, редко самообучение  |  |
| 5      | Собственное мнение учителя о своих способностях  | +   | —   |  |
| 6      | Априорные вероятности появления классов образов  | Произвольная  | Равные  |  |
| 7      | Число решений  | 2, К, континуум   | 2   | Для любого варианта числа классов  |
| 8      | Априорная информация об условной плотности распределения вероятностей относительно образов классов           | Может быть учтена   | Не учитывается  |  |
| 9      | Класс критериев первичной оптимизации  | Критерии<br>Средняя функция риска, без учета и при наличии ограничений на составляющие для различных классов, максимум апостериорной информации и другие критерии, соответствующие постановке задач   | Энергетическая функция, среднеквадратическая ошибка   | Российская методика:<br>• $\min R$ (средней функции риска);<br>• $\min R$ при $p_1 r_1 = \text{const}$ ( $p_1 r_1$ — составляющая средней функции риска);<br>• $\min R$ при $p_1 r_1 = p_2 r_2$ и других критериях |
| 10     | Матрица (функция потерь)   | Произвольная  | Диагональная, симметричная  |  |
| 11     | Типы структур многослойных нейронных сетей   | Структуры многослойных нейронных сетей<br>Многослойные нейронные сети с полными и неполными последовательными, перекрестными и обратными связями. Произвольные структуры, адекватные решаемым задачам | Многослойные нейронные сети с полными последовательными связями. Простейшие рекуррентные нейронные сети |  |
| 12     | Метод выбора функционала вторичной оптимизации, соответствующего функционалу первичной оптимизации           | Функционал вторичной оптимизации<br>+   | —   |  |
| 13     | Использование комбинированных (градиентных и случайных) методов поиска                                       | Методы поиска экстремума функционала вторичной оптимизации<br>+   | —   |  |
| 14     | Использование метода стохастической аппроксимации  | +   | —   |  |
| 15     | Учет информации об ограничениях на настраиваемые коэффициенты (например, по значению или скорости изменения) | +   | —   |  |
| 16     | Возможность использования поисковых колебаний  | +   | —   |  |
| 17     | Возможность фильтрации в контуре адаптации при оценке градиента функционала вторичной оптимизации            | +   | —   |  |
| 18     | Выбор начальных условий в контуре адаптации весовых коэффициентов  | +   | —   |  |
| 19     | Выбор типовых входных сигналов   | Типовые входные сигналы<br>+  | —   |  |

структурой, настраивающейся по замкнутому циклу;

- априорная информация о методе поиска экстремума-функционала вторичной оптимизации;
- априорная информация о наличии и виде ограничений на настраиваемые коэффициенты;
- априорная информация о методе выбора коэффициентов параметрической матрицы  $K^*$  системы поиска экстремума-функционала вторичной оптимизации;
- априорная информация о параметрах поисковых колебаний в том случае, если алгоритм адаптации нейронной сети не может быть построен в виде аналитической системы;
- априорная информация о начальных условиях для настройки;
- априорная информация о классе типовых входных сигналов нейронной сети;
- априорная информация о степени усложнения структуры разомкнутой нейронной сети на каждом шаге и способе проведения данного усложнения.

Объективное сравнение многослойных нейронных сетей типов необходимо проводить, сравнивая информацию, данную для их построения, а также качество работы многослойных нейронных сетей при типовых и реальных входных сигналах.

Автор в течение многих лет уделял большое внимание изучению научных работ других авторов в области теории нейронных сетей, но, возможно, упустил какие-то важные работы. В связи с этим напоминаем, что данная статья написана в порядке обсуждения, и автор будет сердечно благодарен тем, кто подтвердит или поправит его точку зрения в обсуждаемом вопросе.

Автор выражает глубокую благодарность своему другу и коллеге Аведьяну Эдуарду Дзероновичу за его преданность направлению нейросетевых технологий [25] и за то, что он первым в Советском Союзе и России понял общность [26] методов обратного распространения и российских методов настройки многослойных нейронных сетей и отметил приоритет российской научной школы в этой области знаний.

Автор рассматривает данную работу как некоторое "начальное условие" для разработчиков будущих алгоритмов настройки многослойных нейронных сетей с применением реальных моделей мемристоров с уверенностью в том, что эти структуры позволят более эффективно решать задачи в нейросетевом логическом базисе и строить более эффективные модели для исследователей-нейрофизиологов.

#### Список литературы

1. **Werbos P.** Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences // Thesis in applied mathematics, Harvard University. 1974.
2. **Werbos P.** The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting. NY: Wiley, 1994.

3. **Rumelhart D. E., Hinton G. E., Williams R. J.** Learning internal representations by error propagation // D. E. Rumelhart and J. L. McClelland, Eds. // Parallel Data Processing. Cambridge, VA: The M.I.T. Press, 1986. V. 1. P. 318–362.
4. **Rumelhart D. E., Hinton G. E., Williams R. J.** Learning representations by back-propagating errors // Nature. 1986. V. 323. P. 533–536.
5. **Rumelhart D. E.** / J. L. McClelland and the PDP Research Group, Eds. Parallel Distributed Processing. Cambridge, MA: The M.I.T. Press, 1986. V. 1 and 2.
6. **Rosenblatt F.** Principles of Neurodynamics. Washington, D. C.: Spartan Press, 1961.
7. **Кузин Л. Т.** Расчет и проектирование дискретных систем управления. М.: Машгиз, 1962.
8. **Самоастроивающиеся системы** / Под. ред. В. В. Солодовникова. Раздел VI. Техническая кибернетика. Теория автоматического регулирования. Кн. 3, ч. II. М.: Машиностроение, 1969.
9. **Цыпкин Я. З.** Адаптация и обучение в автоматических системах. М.: Наука, 1968.
10. **Галушкин А. И.** Теория нейронных сетей. М.: ИПРЖР, 2000.
11. **Galushkin A. I.** Neural Network Theory. Berlin: Springer, 2007.
12. **Галушкин А. И.** Нейронные сети: основы теории. М.: Горячая линия — Телеком, 2010.
13. **Минский М., Пейперт С.** Перцептроны. М.: Мир, 1971.
14. **Галушкин А. И.** Многослойные системы распознавания образов. М.: Изд-во МИЭМ, 1970.
15. **Галушкин А. И., Зак Л. С., Тюхов Б. П.** К сравнению критериев оптимизации адаптивных систем распознавания образов // Кибернетика. 1970. № 6. С. 122–130.
16. **Галушкин А. И., Василькова Т. Ф., Слободенюк В. Н., Тюхов Б. П.** Анализ динамики распознавания нестационарных образов // Тр. МИЭМ. 1971. Вып. 23. С. 210–227.
17. **Ванюшин В. А., Галушкин А. И.** Построение и исследование многослойных систем распознавания образов // Сб. "Некоторые проблемы биологической кибернетики" / Под ред. А. И. Берга. Л.: Наука, 1972.
18. **Галушкин А. И.** Об алгоритмах адаптации в многослойных системах распознавания образов // Докл. АН УССР, 1973. (Представлено акад. В. М. Глушковым).
19. **Галушкин А. И.** Синтез многослойных нейронных систем распознавания образов. М.: Энергия, 1974.
20. **Галушкин А. И.** Нейрокомпьютеры. М.: ИПРЖР, 2000.
21. **Аналоговые запоминающие и адаптивные элементы** / Под ред. Б. С. Сотскова. М.: Энергия, 1973.
22. **Трейер В. В., Елизаров А. Б.** Электрохимические интегрирующие и аналоговые запоминающие элементы. М.: Энергия, 1973.
23. **Нейронные сети: история развития теории** // Сер. Нейрокомпьютеры и их применение. / Под ред. Я. З. Цыпкина, А. И. Галушкина. М.: Радиотехника, 2001. Кн. 5.
24. **Beale M. H., Hagan M. T., Demuth H. B.** Neural Network Toolbox. User's Guide. R2013b. MathWorks. 2013.
25. **Aved'yan E.** Learning System. London: Springer. 1995.
26. **Аведьян Э. Д.** Исторические аспекты развития теории многослойных нейронных сетей // Информационные технологии. 2005. № 12.
27. **Галушкин А. И.** Нейроматематика: методы решения задач на нейрокомпьютерах // Математическое моделирование. 1991. Т. 3, № 8.
28. **Нейроматематика.** Сер. "Нейрокомпьютеры и их применение". М.: ИПРЖР, 2002. Кн. 6.
29. **Галушкин А. И.** Нейроматематика (проблемы развития) // Нейрокомпьютеры. 2003. № 1.
30. **Галушкин А. И.** О методике решения задач в нейросетевом логическом базисе // Информационные технологии. Приложение. 2006. № 9.
31. **Галушкин А. И.** Основы нейроуправления // Информационные технологии. Приложение. 2002. № 10.
32. **Herbert Ho-Ching Iu, Andrew L Fitch.** Development of Memristor Based Circuits // World Scientific — Imperial College Press, Nov. 2012.
33. **Adamatzky A.** (Author, Editor), **Chen G.** (Editor). Chaos, CNN, Memristors and Beyond: A Festschrift for Leon Chua (With DVD-ROM, composed by Eleonora Bilotta) // World Scientific Publishing Company, 2012.
34. **Cozma R.** et al. (eds.) Advances in Neuromorphic Memristor Science and Applications // Springer Series in Cognitive and Neural Systems, 2012.

## The Back Propagation Error Method and Russian Works on Neural Networks Theory

In this paper, the significant role of works by Paul Verbois and other American writers in the development of algorithms of readjustment of the multi-layer neural network weight coefficients is described. Gist and role of Russian works in this field are noted in this paper, as well as comparison of Russian methods and back propagation (back propagation error), and the prospects of both directions for promising neural computers using memristors. The structure of approaches to the synthesis of multilayer neural networks developed in the 60<sup>th</sup> of the last century is noted for multilayer neural networks with consistent cross and feedback with such features as:

- Continuum of the number of classes;
- Continuum of number of solutions;
- Continuum number of features.
- Continuum of the number of neurons in the layer.

Developed in Russia approaches' focus on specific hardware implementations of neural computers based on actual restrictions on the weights of neural networks is marked.

It is also noted that methods of multilayer neural networks readjustment developed in Russia in the 60<sup>th</sup> years of the last century correspond to various operating modes:

- In case of unequal probabilities for different classes of appearance;
- For classification and clustering modes;
- Training with the teacher who has the final qualification, that allows making algorithms for the classification and clustering unified;
- For non-stationary images;
- For controlled errors first and second types in decision-making;
- For non-diagonal matrices in allocating losses to the images of a particular class.

Formulated and partially solved the problem of the choice of initial conditions in the procedure of neural network coefficients' readjustment for some practical problems.

Outlined ways of development of readjustment methods of multilayer neural networks for advanced neural computers using memristors.

**Keywords:** multilayer neural networks, the theory of neural networks, readjustment of the neural network weight coefficients, neural computers, memristors, back propagation

### References

1. **Verbois P.** Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. *Thesis in applied mathematics*, Harvard University, 1974.
2. **Verbois P.** The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting. NY: Wiley, 1994.
3. **Rumelhart D. E., Hinton G. E., Williams R. J.** Learning internal representations by error propagation // D. E. Rumelhart and J. L. McClelland, Eds. *Parallel Data Processing*, Cambridge, VA: The M.I.T. Press, 1986. V. 1. P. 318—362.
4. **Rumelhart D. E., Hinton G. E., Williams R. J.** Learning representations by back-propagating errors. *Nature*. 1986. V. 323. P. 533—536.
5. **Rumelhart D. E. J. L. McClelland and the PDP Research Group**, Eds. *Parallel Distributed Processing*. Cambridge, MA: The M.I.T. Press, 1986. V. 1 and 2.
6. **Rosenblatt F.** Principles of Neurodynamics. Washington, D. C.: Spartan Press, 1961.
7. **Kuzin L. T.** Raschet i proektirovanie diskretnykh sistem upravleniya. M.: Mashgiz, 1962.
8. **Samonastraivajushhiesja** sistemy. Pod red. V. V. Solodovnikova. Razdel VI. *Tekhnicheskaja kibernetika. Teorija avtomaticheskogo regulirovanija*. Kn. 3, ch. II. M.: Mashinostroenie, 1969.
9. **Cypkin Ja. Z.** Adaptacija i obuchenie v avtomaticheskikh sistemah. M.: Nauka, 1968.
10. **Galushkin A. I.** Teorija nejronnykh setej. M.: IPRZhR, 2000.
11. **Galushkin A. I.** Neural Network Theory. Berlin: Springer, 2007.
12. **Galushkin A. I.** Nejronnye seti: osnovy teorii. M.: Gorjachaja linija — Telekom, 2010.
13. **Minskij M., Pejpert S.** Perseptrony. M.: Mir, 1971.
14. **Galushkin A. I.** Mnogoslojnyje sistemy raspoznavanija obrazov. M.: Izd-vo MIJeM, 1970.
15. **Galushkin A. I., Zak L. S., Tjuhov B. P.** K sravneniju kriteriev optimizacii adaptivnykh sistem raspoznavanija obrazov. *Kibernetika*. 1970. N. 6. P. 122—130.
16. **Galushkin A. I., Vasil'kova T. F., Slobodenjuk V. N., Tjuhov B. P.** Analiz dinamiki raspoznavanija nestacionarnykh obrazov. *Tr. MIJeM*. 1971. Vyp. 23. P. 210—227.
17. **Vanjushin V. A., Galushkin A. I.** Postroenie i issledovanie mnogoslojnykh sistem raspoznavanija obrazov. *Sb. "Nekotorye problemy biologicheskoi kibernetiki"*. Pod red. A. I. Berga. L.: Nauka, 1972.
18. **Galushkin A. I.** Ob algoritmah adaptacii v mnogoslojnykh sistemah raspoznavanija obrazov. *Dokl. AN USSR*, 1973. (Predstavleno akad. V. M. Glushkovym).
19. **Galushkin A. I.** Sintez mnogoslojnykh nejronnykh sistem raspoznavanija obrazov. M.: Jenergija, 1974.
20. **Galushkin A. I.** Nejrokomp'jutery. M.: IPRZhR, 2000.
21. **Analogovye** zapominajushhie i adaptivnye jelementy / Pod red. B. S. Sotskova. M.: Jenergija, 1973.
22. **Trejer V. V., Elizarov A. B.** Jelektrohimicheskie integrirujushhie i analogovye zapominajushhie jelementy. M.: Jenergija, 1973.
23. **Nejronnye seti: istorija razvitija teorii. Ser. Nejrokomp'jutery i ih primenenie.** Pod red. Cypkina Ja. Z., Galushkina A. I. Kn. 5. M.: Radiotekhnika, 2001.
24. **Beale M. H., Hagan M. T., Demuth H. B.** Neural Network Toolbox. User's Guide. R2013b. MathWorks, 2013.
25. **Aved'yan E.** Learning System. London: Springer, 1995.
26. **Aved'jan Je. D.** Istoricheskie aspekty razvitija teorii mnogoslojnykh nejronnykh setej. *Informacionnye tehnologii*. 2005. N. 12.
27. **Galushkin A. I.** Nejromatematika: metody reshenija zadach na nejrokomp'juterah. *Matematicheskoe modelirovanie*. 1991. V. 3, N. 8.
28. **Nejromatematika.** Serija "Nejrokomp'jutery i ih primenenie". M.: IPRZhR, 2002. Kn. 6.
29. **Galushkin A. I.** Nejromatematika (problemy razvitija). *Nejrokomp'jutery*. 2003. N. 1.
30. **Galushkin A. I.** O metodike reshenija zadach v nejrosetevom logicheskom baze. *Informacionnye tehnologii. Prilozhenie*. 2006. N. 9.
31. **Galushkin A. I.** Osnovy nejroupravljenija. *Informacionnye tehnologii. Prilozhenie*. 2002. N. 10.
32. **H. Ho-Ching Iu, A. L. Fitch.** Development of Memristor Based Circuits. *World Scientific — Imperial College Press*, 2012.
33. **Adamatzky A.** (Author, Editor), **Chen G.** (Editor) Chaos, CNN, Memristors and Beyond: A Festschrift for Leon Chua (With DVD-ROM, composed by Eleonora Bilotta). *World Scientific Publishing Company*, 2012.
34. **Cozma R.** et al. (eds.) Advances in Neuromorphic Memristor Science and Applications. *Springer Series in Cognitive and Neural Systems*. 2012.

Д. А. Боронников<sup>1</sup>, канд. экон. наук, нач. отд.,  
 Д. В. Пантюхин<sup>2</sup>, ассистент, e-mail: dim\_beavis@mail.ru, С. В. Данько<sup>2</sup>, студент,  
<sup>1</sup> Московский государственный машиностроительный университет  
<sup>2</sup> Московский физико-технический институт

## Нейросетевой алгоритм организации пространственных данных о рельефе местности

*Разработан нейросетевой алгоритм организации пространственных данных в области недропользования и его программная реализация на языке MATLAB. Экспериментальные исследования на примере данных о рельефе местности Озерного горно-обогатительного комбината показали, что нейронная сеть успешно запомнила и обобщила входную информацию о рельефе местности (110 149 пространственных точек), с ошибкой менее 0,5 м. При этом было достигнуто сжатие исходной информации в 12 раз.*

**Ключевые слова:** нейронные сети, пространственная организация данных, геоинформационные системы, недропользование

### Введение

Геоинформационные системы (ГИС) являются основой для оперирования большими объемами пространственно привязанных данных как в области недропользования, так и в других областях. Несмотря на различия в назначении геоинформационных систем принципы их построения в целом совпадают.

Нейронные сети хорошо зарекомендовали себя как универсальные аппроксиматоры сложных многомерных функций для больших объемов данных. Поэтому целесообразно провести работу по исследованию возможностей применения нейронных сетей в ГИС.

Поскольку число видов различных данных, которые могут быть использованы в ГИС, достаточно большое, то необходимо провести некоторую классификацию таких данных.

Пространственные объекты можно разделить на два типа:

- дискретные, когда объекты, измерением которых (например, фотографированием, лазерным сканированием и т. п.) получают данные, являются разнесенными в пространстве (например, отдельные деревья и т. п.);
- непрерывные — это различного рода площади, объемы и т. п. Примером может служить рельеф местности.

Вместе с тем данные об этих объектах практически всегда представляются в дискретном, цифровом виде. Так, например, рельеф местности, измеряемый лазерным сканером, представляется совокупностью точек — координат  $(x, y, z)$ , от которых отразился лазерный луч. Налицо противоречие между реальностью и представлением о ней. По этой причине представление данных о непрерывных объектах также должно быть непрерывно. И если для дискретных представлений разработано множество различных алгоритмов и программ, то для непрерывных представлений это число довольно мало.

Основным способом представления непрерывных данных является технология нейронных сетей.

Детальный анализ работ позволяет провести сравнение различных методов представления пространственных данных и сделать следующие выводы о преимуществах и недостатках нейросетевого подхода в сравнении с классическими подходами к организации пространственных данных:

1. Данные представляются в непрерывном виде, закодированном в значениях весовых коэффициентов нейронов. Возможно получение информации о данных, которые не были представлены в исходной выборке: автоматически выполняется обобщение (гладкая интерполяция) данных. Это свойство является одним из самых главных преимуществ применения нейросетевых технологий.

2. В нейронной сети одновременно происходят процессы запоминания и обобщения данных. С этим связана возможность дальнейшего извлечения знаний из данных. С одной стороны, это большое преимущество нейросетевого подхода, но, с другой стороны, возникает зависимость структуры нейронной сети от самих данных. Так, например, если данные полностью случайны, то нейронная сеть не сможет вывести общие закономерности (поскольку их нет) и тогда будет запоминать информацию. Однако простое запоминание информации требует огромного количества вычислительных ресурсов, гораздо больше того, что потребовалось бы для запоминания того же объема данных прямым методом. Если же в данных есть закономерность, то нейронная сеть пытается вывести такую закономерность и тем самым снизить объемы запоминаемой информации.

3. Нейронная сеть обучается на данных. Наличие процесса обучения делает нейросетевой подход, с одной стороны, предпочтительным в случаях, когда данные необходимо не просто хранить, но и использовать для получения знаний из них, выявления связей. С другой стороны, в зависимости от данных может случиться такой вариант, что нейронная сеть не обучится до заданного качества. И хотя теория говорит, что нейронная сеть может

аппроксимировать любую непрерывную функцию с заданной точностью, на практике для этого могут потребоваться нереальные вычислительные мощности. Поэтому нейросетевой подход следует использовать не сам по себе, а в сочетании с другими подходами, для тех данных, для которых преимущества превысят недостатки.

4. Нейросетевые алгоритмы обычно заключаются в двух этапах: обучение и непосредственное использование обученной нейронной сети (симуляция). Процесс симуляции — быстрый процесс. В силу изначальной параллельности нейронной сети этот процесс может быть относительно легко распределен на множество параллельных аппаратных ускорителей, например графических процессоров, что дает большой скачок в производительности системы, основанной на нейросетевой организации данных. Процесс же обучения гораздо более медленный. Часто он требует высоких вычислительных затрат, поэтому следует разносить эти процессы на разные вычислительные системы. Из этого следует, что нейросетевой подход в организации данных следует применять для данных, которые часто используются, но редко обновляются.

5. Поскольку нейронная сеть выполняет обобщение данных, то она способна нивелировать возможные ошибки в них. Так, наличие шумов или пропусков в данных учитывается автоматически, в процессе обучения. Повторим, что это свойство является как большим преимуществом нейросетевого подхода, в случаях, когда в исходных данных есть ошибки, так и недостатком в случае, когда требуется точное запоминание информации.

Таким образом, нейросетевой подход к организации данных имеет право на существование, но требует применения в ограниченных ситуациях. Поэтому в реальных ГИС нейросетевой алгоритм организации пространственных данных должен быть одним из многих, а выбор конкретного метода организации данных должен определяться свойствами данных и задачами их обработки.

Основными задачами в ГИС, для которых применяют нейросетевые методы решения, являются следующие:

- оценка параметров (загрязненность вод, оптимальности положения и др.);
- предсказание (мест залежей, траекторий движения и др.);
- кодирование информации (адресов и географических координат);
- обработка изображений (извлечение данных из карт);
- управление (например, трубопроводами).

Для решения этих задач данные переводят в нейросетевое представление.

Ниже проведен синтез нейросетевого алгоритма организации пространственных данных о рельефе местности для Озерного ГОК.

## 1. Нейросетевой алгоритм организации пространственных данных о рельефе местности

*Физическая постановка задачи.* Месторождение Озерного горно-обогатительного комбината (ГОК) расположено в Республике Бурятия в 160 км от Транссибирской железнодорожной магистрали.

Для реализации функций моделирования и проектирования необходимо разработать способ организации пространственных данных о рельефе местности.

Эти данные представляют собой координаты точек сканируемой лазерным методом поверхности с шагом 0,5 м. Данные с лазерного сканера поступают в среду Autodesk Revit, где проходят предварительную обработку. Затем данные выгружаются в текстовый файл, содержащий информацию о координатах точек поверхности рельефа местности. По выбору проектировщика данные могут быть представлены на регулярной или нерегулярной сетке с шагом до 0,5 м. В данных возможны пропуски.

*Математическая постановка задачи.* Математически задача состоит в выполнении преобразования координат  $(x, y)$  сетки пространственного разбиения в координату  $z$  — высоту текущего участка сканируемой поверхности. То есть необходимо преобразовать участок пространства размерности 2 в участок пространства размерности 1 на основе имеющихся измеримых точек. При этом должна выполняться гладкая интерполяция результатов для точек, не представленных в исходных данных, но находящихся в пределах изучаемого участка.

*Нейросетевая постановка задачи.* В соответствии с физической и математической постановкой исходными данными являются набор координат  $(x, y)$  и соответствующая им высота  $z$ . Данные представлены в текстовом файле, содержащем тройки чисел  $(x, y, z)$ .

Входным сигналом нейронной сети будут текущие координаты  $(x, y)$  точки вычислительной сетки (два входа).

Выходной сигнал нейронной сети будет интерпретироваться как координата  $z$  текущей точки рельефа (один выход).

Желаемый выходной сигнал нейронной сети — истинное значение координаты  $z$  текущей точки рельефа.

Сигнал ошибки нейронной сети — разность между действительным и желаемым выходными сигналами.

Функционал вторичной оптимизации нейронной сети — сумма квадратов ошибок на всех точках обучающего множества.

Метод поиска экстремума функционала вторичной оптимизации нейронной сети — метод шкалированных сопряженных градиентов *trainscg* [1].

Тип нейронной сети — многослойный перцептрон с двумя скрытыми слоями, число нейронов в слоях — варьируемый параметр.

Функции активации для скрытых слоев — гиперболический тангенс *tansig* [2], для выходного слоя — линейная.

Начальные условия при настройке нейронной сети — случайные числа из диапазона  $(-1,1)$ .

Начальные данные проходят предобработку — перенос и масштабирование к диапазону  $(-100, 100)$  для всех координат.

Выходные данные проходят постобработку и преобразуются в обратном порядке.

Тестирование проводится на примере реального рельефа местности Озерного ГОК из файла Relief.txt.

## 2. Программная реализация

Для выполнения программной реализации был выбран пакет MATLAB фирмы *Mathworks* версии 2013a под управлением операционной системы *Windows*, который включает в себя *Neural Network Toolbox* — специализированный инструментарий для создания, моделирования, обучения нейронных сетей различного типа.

Программная реализация состоит из одного модуля, который позволяет следующее.

1. Загружать пространственные данные в заданном выше формате с произвольным разрешением пространственной сетки. Загрузка данных происходит с помощью команд MATLAB открытия/закрытия и чтения текстовых файлов: *fopen()*, *fclose()*, *textscan()*, *fgetl()* и др.

2. Проводить обработку пространственных данных с произвольным пространственным разрешением. Применяемый способ вычислений позволяет узнать значение координаты  $z$  в произвольной точке внутри исследуемой области  $(x, y)$ , при этом точность вычислений зависит от шага пространственного разрешения исходных данных, а для промежуточных точек автоматически выполняется гладкая интерполяция результатов. Желаемая точность обучения нейронной сети подобрана таким образом, чтобы обеспечить максимальное отклонение обученной нейронной сети не больше 0,5 м.

Обработка данных выполняется в несколько этапов.

На первом этапе загруженные данные проходят предварительную обработку таким образом, чтобы привести диапазон данных к диапазону  $(-100, 100)$ . Это достигается за счет операции масштабирования и параллельного переноса, что необходимо для обеспечения работоспособности модуля с реальными данными, представленными в абсолютных координатах. Также это обеспечивает независимость модуля расчета от конкретных диапазонов данных.

На втором этапе создаются нейронные сети с заданными параметрами. Это выполняется командой *newff()*.

На третьем этапе созданная нейронная сеть обучается с помощью функции *train()* выбранным методом обучения. В процессе обучения выводится информация о текущем состоянии процесса.

На четвертом этапе нейронная сеть проверяется (симулируется) с помощью команды *sim()*. Полученная информация с выхода нейронной сети преобразуется к исходным диапазонам. Затем про-

исходит вычисление полученного максимального отклонения обученной нейронной сети от истинных значений. Если это значение меньше заданного (0,5 м), то нейронная сеть считается успешно обученной. Она запоминается в файл (команда *save()*). Сохраняется также информация о масштабирующих коэффициентах.

Именно эта информация (структура и весовые коэффициенты обученной нейронной сети) является нейросетевым представлением исходных пространственных данных. Таким образом, исходные пространственные данные организованы в структуре нейронной сети, а конкретные данные кодируются значениями весовых коэффициентов нейронной сети.

Для извлечения данных необходимо загрузить сохраненную обученную нейронную сеть (команда *load()*), подать на ее вход значения координат интересующей точки вычислительной сетки и запустить симуляцию нейронной сети. Результат вычислений будет искомым значением третьей координаты точки рельефа.

3. Выгружать и отображать данные о ходе вычислений и результатах. В процессе вычислений в командном окне MATLAB выводится информация о ходе процесса (командами *fprintf()*). Результаты вычислений — структура обученной нейронной сети и масштабирующие коэффициенты — сохраняются в файл. Рассчитывается и отображается в командном окне максимальная ошибка обучения.

4. Измерять достигнутые характеристики процесса обработки:

- а) время обработки в секундах и циклах;
- б) точность обработки в процентах и максимальное отклонение;
- в) размер получаемого представления в байтах и числе весовых коэффициентов.

Эти данные выводятся в командное окно MATLAB.

## 4. Экспериментальные результаты

Для тестового варианта, содержащегося в файле Relief.txt, были показаны следующие характеристики процесса вычислений:

Число обработанных точек: 110 149.

Ошибка обучения нейронной сети: 0,91719 при заданной максимальной ошибке 0,91735.

Среднее квадратичное отклонение результатов составило 0,0025.

Максимальное отклонение обученных данных: 0,451 м (2,109 %), что меньше заданного значения 0,5 м.

Время обработки составило 6757,6 с (1482 циклов). Из этого времени 0,46 с проводилась загрузка исходных данных; 0,005 с — предобработка данных (масштабирование); 6755,49 с — обучение нейронной сети; 1,719 с — симуляция и постобработка результатов.

Размер получаемого представления обученной нейронной сети (объект *net2*):  $b = 2\ 863\ 876$  байт, что соответствует 10 501 весовым коэффициентам. Из них  $a = 2\ 643\ 576$  байт соответствует исходным

данным (поля *exampleInput* и *exampleOutput*), хранение которых необязательно, они предназначены только для возможности однозначного повторения процесса обучения. Таким образом, на запоминание весовых коэффициентов, структуру нейронной сети и прочую служебную информацию приходится  $b - a = 2\,863\,876 - 2\,643\,576 = 220\,300$  байт. Коэффициент сжатия составил  $a/(b - a) = 12$  раз.

### Заключение

В процессе экспериментов было обнаружено, что используемые размерности нейронных сетей (два скрытых слоя по 100 нейронов в каждом) требуют достаточно заметных вычислительных затрат. Поэтому дальнейшее увеличение числа обрабатываемых пространственных точек делает необходимым переход на параллельные аппаратные вычислительные средства и разработку соответствующих программ для них.

На сегодняшний день наиболее эффективными аппаратными средствами являются ЭВМ с графическими процессорами. Этим объясняется наш выбор этого типа аппаратных ускорителей при разработке планов дальнейших исследований и разработок.

Дополнительно было изучено, что сжатие файла исходных данных с помощью упаковщика 7zip алгоритмом LZMA с максимальной степенью сжа-

тия приводит к представлению данных объемом 823 412 байт, т. е. в 3,74 раза хуже, чем достигнутое нами значение (220 300 байт). При этом нейросетевое представление сразу готово к вычислениям, без необходимости распаковки упакованных данных.

Процесс прямого расчета (симуляции) уже обученной нейронной сети является довольно быстрым (1,719 с).

Результаты экспериментальных исследований подтверждают, что разработанный нейросетевой алгоритм организации пространственных данных имеет большие перспективы для реализации в составе ГИС.

Реализация таких алгоритмов позволит:

- уменьшить объемы запоминаемой информации;
- увеличить скорость обработки информации;
- унифицировать различные операции по обработке пространственных данных в нейросетевом представлении, например, такие как аппроксимация, классификация, экстраполяция, кодирование и др.

*Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации по госконтракту №14.514.11.4096.*

### Список литературы

1. Moller M. F. A scaled conjugate gradient algorithm for fast supervised learning // Neural Networks. 1993. Vol. 6. P. 525–533.
2. Vogl T. P., Mangis J. K., Rigler A. K., Zink W. T., Alkon D. L. Accelerating the convergence of the backpropagation method // Biological Cybernetics. 1988. Vol. 59. P. 257–263.

D. A. Boronnikov, Head of Department, D. V. Pantiukhin, Assistant, S. V. Danko, Student

## Neural Networks Algorithm of Spatial Relief Data Organization

*Neural networks algorithm of spatial relief data organization and its implementation in MATLAB language are developed. Experimental studies on the data on terrain of the Ozernyi mining and processing plant showed that the neural network has successfully memorized and generalize input information about the terrain (110 149 spatial points), with an error less than 0,5 meters. Compressing ratio on the original data is about 12 times.*

**Keywords:** neural networks, spatial data organization, GIS, subsoil use

### References

1. Moller M. F. A scaled conjugate gradient algorithm for fast supervised learning // Neural Networks. 1993. Vol. 6. P. 525–533.

2. Vogl T. P., Mangis J. K., Rigler A. K., Zink W. T., Alkon D. L. Accelerating the convergence of the backpropagation method // Biological Cybernetics. 1988. Vol. 59. P. 257–263.

### Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер Т.Н. Погорелова. Технический редактор Е.В. Конова.

Корректор Е.В. Комиссарова.

Сдано в набор 08.05.2014. Подписано в печать 23.06.2014. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT714. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

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

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

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