

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 23
2017
№ 12

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Разоренов А. А.** Математические основы применения теории К-представлений к разработке алгоритмов исполнения прикладными системами естественно-языковых предписаний 843
- Зак Ю. А.** Об одном алгоритме Fuzzy-регрессионного анализа 850
- Стемпковский А. Л., Левченко Н. Н., Окунев А. С., Климов А. В., Змеев Д. Н.** Программирование задачи "молекулярная динамика" в потоковой модели вычислений 859

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Казаков П. В.** Модель параллельных вычислений для генетических алгоритмов многокритериальной оптимизации, ориентированная на графические процессорные устройства с архитектурой CUDA 868
- Курейчик В. В., Бова В. В., Лещанов Д. В.** Модель семантического поиска в системах управления знаниями на основе генетических процедур 876

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Решетова Е. Н.** Опыт преподавания учебной дисциплины "Групповая динамика и коммуникация в профессиональной практике программной инженерии" . 884

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Коротченко Р. А.** Применение сингулярного спектрального анализа для обнаружения акустических сигналов от серых китов 893

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Филиппов Б. И.** Разработка и исследование устройства защиты от ошибок для системы передачи изображений по гидроакустическому каналу связи 897

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Андреева Е. А., Храмов И. С.** Алгоритм построения приближенного оптимального решения задач на основе искусственных нейронных сетей с учетом запаздывания 904
- Указатель статей, опубликованных в журнале "Информационные технологии" в 2017 г.** 910

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

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

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

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

Редакция:
БЕЗМЕНОВА М. Ю.
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 и базу данных RSCI на платформе Web of Science.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

CONTENTS

INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Razorenov A. A.** Mathematical Foundations of Applying the Theory of K-Representations to Developing Algorithms of Executing Natural Language Instructions by Computer Systems 843
- Zack Yu. A.** About One Algorithm of Fuzzy-Regression Analysis 850
- Stempkovskiy A. L., Levchenko N. N., Okunev A. S., Klimov A. V., Zmejey D. N.** Programming of the Molecular Dynamics Task in the Dataflow Computing Model 859

MODELING AND OPTIMIZATION

- Kazakov P. V.** The Model of Parallel Computing Based on CUDA Technology for Multi-Objective Genetic Algorithms 868
- Kureichik V. V., Bova V. V., Leshchanov D. V.** Semantic Search Model for Knowledge Management Systems Based on Genetic Procedures 876

INFORMATION TECHNOLOGIES IN EDUCATION

- Reshetova E. N.** Teaching Experience the Academic Discipline "Group Dynamics and Communication in Professional Software Engineering Practice" 884

APPLIED INFORMATION TECHNOLOGIES

- Korotchenko R. A.** Application of Singular Spectral Analysis to Detection of Acoustic Signals of Gray Whales 893

CRYPTOSAFETY INFORMATION

- Filippov B. I.** Research and Development of the Device of Protection against Mistakes for System of Transfer of Images on the Hydroacoustic Communication Channel 897

NEUROTECHNOLOGIES

- Andreeva E. A., Khramov I. S.** Optimization of the Artificial Neural Network with Allowing for Delay 904

- Index** of articles published in the journal "Information Technology" in 2017 910

Editor-in-Chief:

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

Deputy Editor-in-Chief:

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

Chairman:

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

Editorial Board Members:

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

Editors:

Bezmenova M. Yu.
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"

УДК 004.512; 004.89

А. А. Разоренов, аспирант, e-mail: arazorenov@semsynt.ru,
Национальный исследовательский университет "Высшая школа экономики", г. Москва

Математические основы применения теории К-представлений к разработке алгоритмов исполнения прикладными системами естественно-языковых предписаний

Статья продолжает публикацию автора в журнале Информационные технологии 2017, № 10, посвященную изложению принципов применения теории К-представлений к разработке новых методологических основ проектирования алгоритмов исполнения естественно-языковых предписаний компьютерными системами. Цель данной статьи — формальное описание бинарного отношения нового вида на семантических структурах, послужившее основой нового метода исполнения прикладными системами естественно-языковых предписаний. Построена математическая модель преобразований первичных семантических представлений (СП) в другие СП, а также математическая модель формирования сценариев/программ на языке управляемой системы для выполнения предписаний, выраженных в виде СП. Кроме того, исследованы свойства полученных отношений и описан опыт практического применения разработанного метода к проектированию естественно-языкового интерфейса файлового менеджера.

Ключевые слова: обработка естественного языка, семантический парсинг, теория К-представлений, СК-языки, К-цепочка, управление программным обеспечением, естественно-языковые интерфейсы, отношение соответствия образу, семантико-командный словарь, Natural Language Commander

Введение

В последние годы значительно возрос интерес к теоретическим и практическим вопросам семантической обработки (семантического парсинга) естественно-языковых предписаний (ЕЯП), или команд [1—10]. Это нашло отражение как в публикациях, посвященных управлению роботами, виртуальными персонажами видеоигр, поиску в базах данных и знаний, а также управлению прикладными программными системами, так и в разработке естественно-языковых интерфейсов (ЕЯИ) для различных программных и аппаратных систем, в частности систем *Siri* [11], *Sobesednik HD* [12], *Braina Project* [13], *NLUI Server* [14], *NLC-1* [15—19], *NLC-2* [20—22].

В рамках предыдущих публикаций цикла [19—22] были описаны математические основы семантического парсинга предписаний прикладными компьютерными системами *NLC-1* и *NLC-2*. Эти системы, являющиеся файловыми менеджерами с ЕЯИ, позволили значительно упростить операции пользователя с файлами, хранящимися в его компьютере. Научные результаты, опубликованные в работах [19—22], существенно обогащают теорию К-представлений (ТКП) [23—37], в рамках которой они были разработаны. Кроме того, были изложены основные идеи нового метода обработки и испол-

нения семантических представлений (СП) предписаний [38]. Основная цель данной статьи — формальное определение нового отношения на СК-языке (стандартном концептуальном языке), названного отношением соответствия образцу, и исследование его свойств, имеющих теоретическое и практическое значение.

Данная статья имеет следующую структуру. Сначала дается формальное определение отношения соответствия образцу и исследуются его свойства. Затем описываются применение введенного отношения к трансформации первичного СП во вторичное СП и преобразование в язык целевой системы. Завершается статья описанием принципов функционирования ЕЯИ системы *NLC-2* (Natural Language Commander — Version 2).

Формальное определение отношения соответствия образцу

В статье [38] были изложены основания для включения пар К-цепочек в отношение соответствия образцу *Match* с точностью до отношения синонимии *Syn*. При этом под К-цепочками понимаются выражения СК-языка в произвольном рассматриваемом концептуальном базисе (к.б.).

Рассматриваемое ниже определение интегрирует идеи, изложенные в работе [38]. Это определение включает обозначение функции tpl и обозначение $Tconc(B)$ для множества типов понятий, где B — рассматриваемый к.б. Функция tpl определена на СК-языке $Ls(B)$. Каждому элементу d множества $Ls(B)$ функция tpl ставит в соответствие тип $tpl(d)$ из множества типов $Tr(S(B))$, где $S(B)$ — компонент к.б. B , называемый сортовой системой [20]. Например, К-цепочке *Больше(Расстояние(Москва, Париж), Расстояние(Москва, Прага))* отображение tpl может поставить в соответствие тип *сообщ*, интерпретируемый как тип СП высказываний (тип "смысл сообщения").

Множество $Tconc(B)$, являющееся подмножеством множества типов $Tr(S(B))$, включает, в частности, три специальных элемента $[\uparrow\text{сущн}]$, $[\uparrow\text{пон}]$, $[\uparrow\text{об}]$. Эти элементы интерпретируются как типы семантических единиц, соответствующих словам "сущность", "понятие" и "объект". Остальные элементы множества $Tconc(B)$ интерпретируются как типы понятий, характеризующих различные реальные и абстрактные объекты, а также ситуации. Например, тип $\uparrow\text{интс} * \text{дин.физ.об}$ (после стрелки следует склейка сортов "интеллектуальная система" и "динамический физический объект") может интерпретироваться как тип понятия "человек".

Определение 1. Пусть B — произвольный к.б. и $y, z \in Ls(B)$. Пусть Syn — некоторое бинарное рефлексивное отношение на первичном информационном универсуме $X(B)$. Тогда отношением соответствия образцу с точностью до отношения синонимии Syn (для выражения факта "у соответствует образцу z " будем использовать обозначение $y > z$) будем называть наименьшее множество упорядоченных пар вида (y, z) , для которого выполнены следующие условия.

1. Если y и z — произвольные элементы первичного информационного универсума $X(B)$ к.б. B и $(y, z) \in Syn$, то y соответствует образцу z .

2. Если z — произвольная переменная из множества $V(B)$ и $tpl(z)$ является более общим типом, чем $tpl(y)$ (обозначается $tpl(z) \vdash tpl(y)$), то y соответствует образцу z .

3. Если z является К-цепочкой вида $\text{intq } a$, где a — обозначение понятия (т.е. $a \in X(B)$ и $tpl(a) \in Tconc(B)$, где $Tconc(B) = \{t \in Tr(S(B)) \mid t \text{ начинается с символа } \uparrow\} \cup \{\uparrow\text{сущн}, \uparrow\text{пон}, \uparrow\text{об}\}$ — множество типов, соответствующих понятиям), intq — интенсиональный квантор, и y является К-цепочкой вида $\text{intq}_1 a_1$, причем $(\text{intq}, \text{intq}_1) \in Syn$ и $a_1 > a$, то y соответствует образцу z .

4. Если z является К-цепочкой вида $f(a_1, \dots, a_n)$, y является К-цепочкой вида $g(b_1, \dots, b_n)$, где f, g являются n -местными функциональными символами первичного информационного универсума $X(B)$ и $g > f, b_1 > a_1, \dots, b_n > a_n$, то y соответствует образцу z .

5. Если z является К-цепочкой ($a \equiv b$), y является К-цепочкой ($a_1 \equiv b_1$) и $a_1 > a, b_1 > b$, то y соответствует образцу z .

6. Если z является К-цепочкой вида $r(a_1, \dots, a_n)$, y является К-цепочкой вида $q(b_1, \dots, b_n)$, где r, q являются n -местными реляционными символами первичного информационного универсума $X(B)$, $n \geq 1$ и $q > r, b_1 > a_1, \dots, b_n > a_n$, то y соответствует образцу z .

7. Если $y > z$ и v является переменной из множества $V(B)$, то $y:v > z$.

8. Если $a > b$, то $\neg a > \neg b$.

9. Если z является К-цепочкой, y является К-цепочкой вида $(t_1 \vee \dots \vee t_m)$ и существует j от 1 до m , для которого выполнено условие $t_j > z$, то y соответствует образцу z .

10. Если z является К-цепочкой вида $(s_1 \vee \dots \vee s_n)$, y является такой К-цепочкой, что существует i от 1 до n , для которого выполнено условие $y > s_i$, то y соответствует образцу z .

11. Если $z \in Ls(B)$, y является выражением вида $(t_1 \wedge \dots \wedge t_m)$, таким что существует j от 1 до m , для которого выполнено условие $t_j > z$, то y соответствует образцу z .

12. Если z является К-цепочкой вида $(s_1 \wedge \dots \wedge s_n)$, y является произвольной К-цепочкой, такой что для любого i от 1 до n выполнено условие $y > s_i$, то y соответствует образцу z .

13. Если z является обозначением понятия (т.е. $z \in X(B)$ и $tpl(z) \in Tconc(B)$), y является выражением вида $a^*(b_1, c_1) \dots (b_n, c_n)$ и $a > z$, то y соответствует образцу z .

14. Если z является К-цепочкой вида $z^*(u_1, v_1) \dots (u_m, v_m)$, y является К-цепочкой вида $a^*(b_1, c_1) \dots (b_n, c_n)$ и $a > z$, а также для любого i от 1 до m существует j такое, что $b_j > u_i$ и $c_j > v_i$, то y соответствует образцу z .

15. Если z является К-цепочкой вида $Qv(a)S$ и y является К-цепочкой вида $Qu(b)S_1$, где Q — один из кванторов \forall или \exists , и $S_1 > S, b > a$, то y соответствует образцу z .

16. Если z является К-цепочкой вида $\langle s_1, \dots, s_n \rangle$, y является К-цепочкой вида $\langle t_1, \dots, t_n \rangle$, где $n > 1$, и для любого i от 1 до n выполнено условие $t_i > s_i$, то y соответствует образцу z .

Номера пунктов определения 1 совпадают с номерами соответствующих оснований из предыдущей статьи [38]. Обратим внимание на пункт 2. Данный пункт позволяет создавать некоторое подобие масок для подстановки сложных подцепочек. Например, К-цепочка *Команда (удаление1*(Объект1, нек файл1*(Наименование, filename)))* будет соответствовать образцу *Команда (удаление1*(Объект1, y))*.

Проиллюстрируем данное определение на примерах.

Пусть отношение синонимии Syn включает в себя пары (*файл1, документ1*), (*напка1, каталог1*),

(папка1, директория1), (диск1, носитель1), (диск1, флешка1), а также симметричные им.

Тогда, в силу данного выше определения, будут выполнены следующие соотношения:

- $каталог1 > папка1$;
- $диск1 > z$, если $z \in V(B)$, $tpl(папка1) = инф.объект *дин.физ.об.$ и $tpl(z) = инф.объект$, так как $инф.объект \vdash инф.объект *дин.физ.об.$;
- $нек каталог1 > нек папка1$;
- $Размер(нек каталог1) > Размер(нек папка1)$;
- $Размер(нек каталог1*(Наименование, "Docs")) > Размер(нек каталог1)$;
- $(нек файл1 \wedge нек каталог1) > нек файл1$;
- $нек каталог1 > (нек файл1 \vee нек каталог1)$.

Свойства отношения соответствия образцу

Обратим внимание на следующие свойства отношения соответствия образцу.

Утверждение 1. Для любого к.б. B и любого отношения семантической синонимии Syn на первичном информационном универсуме $X(B)$ отношение соответствия образцу на СК-языке $Ls(B)$ с точностью до Syn существует и единственно.

Доказательство. Существование следует из определения. Докажем единственность от противного. Пусть есть два отношения соответствия образцу A и B с точностью до Syn , не совпадающие друг с другом и не входящие одно в другое (иначе "большее" отношение автоматически не будет минимальным). Тогда, без ограничения общности, некоторая пара К-цепочек $(y, z) \in A$ и $(y, z) \notin B$. Тогда либо пара (y, z) не получается с помощью рекурсивного применения пунктов определения 2, и тогда A не является отношением соответствия образцу, либо B не является отношением соответствия образцу. Что и требовалось доказать.

Утверждение 2. Для любых к.б. B , СК-языка $Ls(B)$ и отношения семантической синонимии Syn на первичном информационном универсуме $X(B)$ отношение соответствия образцу на СК-языке $Ls(B)$ с точностью до Syn является рефлексивным.

Доказательство. Пусть y — К-цепочка. Тогда, последовательно применяя необходимые пункты определения отношения соответствия образцу, получаем, что $y > y$.

Гипотеза 1. Для любых к.б. B , СК-языка $Ls(B)$ и транзитивного отношения семантической синонимии Syn на первичном информационном универсуме $X(B)$ отношение соответствия образцу на СК-языке $Ls(B)$ с точностью до Syn является транзитивным.

Предполагается, что данное утверждение может быть доказано индукцией по построению К-цепочек. Более того, скорее всего, верным является и более сильное утверждение, что отношение соответствия образцу с точностью до отношения синонимии Syn является транзитивным тогда и только тогда, когда отношение синонимии транзитивно.

Однако для дальнейших рассуждений данное утверждение не является на данный момент необходимым, поэтому оставим его без рассмотрения.

Замечание. Симметричность отношения синонимии не приводит к симметричности отношения соответствия образцу в силу пунктов 2, 12.

Утверждение 3. Пусть B — к.б., $Ls(B)$ — СК-язык, тогда отношение соответствия образцу на СК-языке $Ls(B)$ с точностью до отношения семантической синонимии Syn на первичном информационном универсуме $X(B)$ разрешимо тогда и только тогда, когда Syn — разрешимо.

Доказательство. Неразрешимость Syn по определению влечет за собой неразрешимость отношения синонимии. Пусть Syn разрешимо, тогда в силу разрешимости СК-языков и конечности их К-цепочек алгоритм проверки соответствия образцу существует (он следует из определения) и его выполнение всегда конечно, так как он рекурсивен, и на каждом шаге рекурсия переходит к меньшим подцепочкам вплоть до элементов первичного информационного универсума. Что и требовалось доказать.

В дальнейших рассуждениях нам также пригодится следующая теорема.

Теорема о соответствии образцу. Пусть B — произвольный к.б. и К-цепочки $y, z \in Ls(B)$. Тогда если $y > z$ и z не является переменной $V(B)$, то выполнено одно из следующих условий:

1) либо y и z образованы с применением одного и того же правила построения на последнем шаге построения и $tpl(y) = tpl(z)$;

2) либо если y является семантическим представлением составного понятия (построен по правилу $P[8]^*$) и z является понятием, т.е. $tpl(z) \in Tconc(B)$, то $tpl(y) = tpl(z)$;

3) либо если y является К-цепочкой вида $x_j \cdot v$, то для x_j и z выполнено одно из условий 1, 2 или 4;

4) либо z является К-цепочкой вида $(s_1 \lambda \dots \lambda s_n)$, где $\lambda \in \{\vee, \wedge\}$, существует такое i , что для s_i и y выполнено одно из условий, приведенных выше (1, 2 или 3).

Доказательство индукцией по построению К-цепочек y и z .

Следствие 1. Пусть B — произвольный к.б. и К-цепочки $y, z \in Ls(B)$. Если $y > z$, то $tpl(z) \vdash tpl(y)$ и, если z не является переменной $V(B)$, то $tpl(y) = tpl(z)$.

Трансформация КП из первичной в унифицированную форму

Дадим более формализованное определение идеи, изложенной ранее в статье [38].

Назовем **свободным вхождением переменной** y в К-цепочку z произвольное вхождение y в z , если оно не образовано непосредственно с помощью

* Правила построения описаны в работах [23–37].

правила построения P[5] или P[8], т.е. вхождение не является определением переменной и переменная не связана квантором существования или всеобщности. Согласно определению соответствия образцу свободные вхождения могут соответствовать подцепочкам К-цепочек, проверяемых на соответствие. Иными словами, это означает, что переменные выполняют роль символов-джокеров (wildcard) в поисковом шаблоне.

Псевдоформулой К-языка будем называть такую цепочку символов, которая может быть преобразована в выражение некоторого СК-языка $Ls(B)$, где B — рассматриваемый к.б., путем замены свободных вхождений некоторых (возможно, не всех) переменных базиса B на элементы первичного информационного универсума $X(B)$, при условии, что тип заменяющего элемента из множества $X(B)$ является конкретизацией типа заменяемой переменной.

Определение отношения соответствия образцу может быть естественным образом расширено на множество псевдоформул простой заменой первого предложения определения 1 на следующее: "Пусть B — произвольный к.б. и y, z — псевдоформулы СК-языка $Ls(B)$ в базисе B ." Остальной текст определения остается неизменным. Применение псевдоформул в образцах позволяет в ряде случаев существенно сократить объем базы преобразований за счет обобщения правил.

Определение 2 (преобразования К-цепочек). Пусть есть некоторая пара псевдоформул (A, B) и пусть некоторая подцепочка c К-цепочки C соответствует образцу A с точностью до некоторого отношения синонимии Sup . Кроме того, свободным вхождениям переменных x_1, \dots, x_n в псевдоформулу A соответствуют подцепочки c_1, \dots, c_n К-цепочки C . Тогда трансформацией К-цепочки C с помощью правила трансформации $A \rightarrow B$ будем называть К-цепочку D , полученную из К-цепочки C заменой подцепочки c на К-цепочку B таким образом, что каждое свободное вхождение переменной x_i в псевдоформуле B заменяется на подцепочку c_i , где $i = 1, \dots, n$.

Правила преобразования могут применяться для получения новых К-цепочек не только при решении задачи управления программным обеспечением, но и для решения широкого спектра других задач. Полученные таким образом К-цепочки в зависимости от решаемой задачи могут либо замещать преобразуемые К-цепочки в некотором исходном множестве, либо дополнять это множество.

Для решения задачи управления программным обеспечением представляется более целесообразным замещение преобразуемых К-цепочек.

Таким образом, процесс преобразования СП требуемого действия в СП, которое может быть выполнено (выполнено) программой, заключается в последовательном применении правил преобразования, пока не будут исчерпаны все возможные правила преобразования. Если полученное в итоге

СП может быть выполнено, то процесс завершается выполнением требуемого действия. Если нет, то возможно либо завершение с ошибкой, либо взаимодействие с пользователем в целях уточнения неясных частей команды.

Преобразование КП предписаний в сценарий командной оболочки

Изложенный ниже метод может применяться как для преобразования СП в одиночные команды, так и в сценарии командной оболочки операционной системы. В дальнейшем, без ограничения общности, будем говорить о сценариях, так как одиночные команды являются их частным случаем.

Для преобразования СП простой команды в сценарий командной оболочки воспользуемся словом, состоящим из пар вида

$$(m, c),$$

где m — цепочка-образец, а c — шаблон сценария. Пример такой пары:

(Команда (удаление1*(Объект1, нек файл1* (Наименование, fileName))),
"rm -f {fileName}")

В данном примере *fileName* — это переменная типа *Literal*. Данный тип будем использовать для всех конструкций: имен файлов, папок дисков, других подцепочек, заключаемых в двойные кавычки.

Договоримся называть словарь, состоящий из пар указанного вида, *семантико-командным словом* (СКС).

Тогда преобразование СП в сценарий командной оболочки будем осуществлять по следующему алгоритму. Если в СКС существует такая пара (m, c) , что СП предписания пользователя f соответствует образцу m , то в качестве основы сценария командной оболочки принимается шаблон c . Для того чтобы из шаблона сценария получить законченный сценарий, требуется заменить в сценарии подстроки вида $\{varName\}$, где *varName* — свободная переменная шаблона m , (а) либо на результат преобразования подцепочки f , соответствующей переменной *varName*, с помощью этого же алгоритма преобразования с этим же СКС, (б) либо на текст в кавычках, если свободной переменной *varName* соответствует конструкт. То есть мы выполняем рекурсивно этот алгоритм для всех свободных переменных, входящих в шаблон m .

В приведенном выше примере для предписания пользователя "Удалить файл a.txt" с СП

Команда(удаление1*(Наименование, "a.txt"))

в качестве основы сценария будет использована строка

$$rm -f \{fileName\}$$

Так как переменной *fileName* будет соответствовать конструктор "a.txt", то вместо подстроки *{fileName}* шаблона будет подставлена строка "a.txt". Итоговая команда примет вид

$$rm -f "a.txt"$$

ЕЯ-интерфейс файлового менеджера Natural Language Commander

Предложенные ранее математическая модель лингвистической базы данных [20, 21] и алгоритм SemSyntRA [20, 22] лежали в основе разработанной программы управления файловой системой Natural Language Commander — Version 2 (NLC-2). Данная программа принимает на вход команды пользователя из некоторого ограниченного подмножества естественного языка (русского, английского или другого). Далее программа с помощью алгоритма SemSyntRA преобразует введенную пользователем команду в ее первичное КП и выполняет необходимые трансформации из базы преобразований, получая в результате вторичное КП. Данное КП принадлежит тому же СК-языку, что и исходное представление. Однако в нем сложные понятия, такие как "документ", "музыка", "видео", заменены более простыми понятиями с дополнительными квалификаторами. Например, "документ" интерпретируется как "файл с расширением doc, docx или odt".

Далее вторичное КП преобразуется в сценарий на языке командной оболочки операционной системы. NLC-2 использует для этих целей Bourne Again Shell (Bash) как для POSIX-совместимых операционных систем, так и для Windows.

Пример. Покажем, как NLC-2 обрабатывает предписание "Переместить файл "a.txt" в папку "Мои документы", если размер файла "a.txt" меньше 2 Мбайт". Это предписание алгоритм SemSyntRA преобразует в первичное К-представление

```
If(Less(Size(certain file1*(Name, "a.txt")), "2000000"),  
movement1*(Destination, certain folder1*(Name,  
"Desktop"))(Source, certain file1*(Name, "a.txt"))).
```

Вторичное представление является таким же. Затем программа формирует и исполняет итоговый сценарий Bourne-Again Shell (BASH):

```
if [S(du -s "a.txt" | awk '{print S1}') -lt "2000000"];  
then mv "a.txt" "Desktop"/; fi
```

Заключение

В данной статье предложен новый подход к формализации механизма исполнения СП предписаний прикладными компьютерными системами. Рассматриваемые СП являются К-представлениями, т.е. выражениями СК-языка в некотором концептуальном базисе.

Предложенный метод исполнения СП предписаний базируется на введенном определении соот-

ветствия К-цепочки образцу. Это соответствие используется как при выполнении преобразований на множестве К-цепочек, так и при трансляции К-представлений в язык целевой системы. Исследование свойств предложенного отношения соответствия образцу показало существование признаков, которые могут использоваться при индексации терминологической базы данных (ТБД) и семантико-командного словаря (СКС). Это открывает возможность построения объемных баз данных без существенной потери производительности, так как благодаря индексации нет необходимости проводить полный перебор правил преобразования, содержащихся в ТБД и СКС.

Следует отметить, что полученные научные результаты могут найти применение не только в разработке методов и алгоритмов исполнения предписаний, но и при проектировании иных алгоритмов, требующих осуществлять выбор СП из множества заданных семантических структур, например при поиске информации в базе знаний.

Предложенный метод позволяет проводить обработку сколь угодно сложных СП, описывающих последовательности команд, команды с условиями, а также команды, предусматривающие многократное выполнение действий до достижения определенных условий.

Наряду с ранее предложенными моделью лингвистической базы данных и алгоритмом SemSyntRA [20—22] описанные отношения и метод обработки СП стали основой разработки файлового менеджера с ЕЯИ Natural Language Commander — Version 2 (NLC-2) и его информационно-справочной Интернет-версии. Созданное программное обеспечение было успешно применено при проведении лабораторных работ со студентами Школы бизнес-информатики НИУ ВШЭ (бакалавриат и магистратура) и студентами НИУ МАИ (магистратура).

Список литературы

1. Artzi Y., Zettlemoyer L. Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions // Transactions of the Association for Computational Linguistics, 2013, N. 1. P. 49—62. URL: <https://aclweb.org/anthology/Q/Q13/Q13-1005.pdf> (дата обращения: 11.03.2016).
2. Babes-Vroman M., MacGlashan J., Gao R., Winner K., Adjogah R.; desJardins M., Littman M., Muresan S. Learning to Interpret Natural Language Instructions // Proceedings of the Second Workshop on Semantic Interpretation in an Actionable Context (Montreal, Canada, June 3—8, 2012), 2012. P. 1—6. URL: <https://aclweb.org/anthology/W/W12/W12-2801.pdf> (дата обращения: 11.03.2016).
3. Bisk Y., Yuret D., Marcu D. Natural Language Communication with Robots. // Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (San Diego, California, June, 2016), 2016. P. 751—761. URL: <http://aclweb.org/anthology/N16-1089> (дата обращения: 14.03.2017).
4. Bastianelli E., Castellucci C., Croce D., Basili R. Textual Inference and Meaning Representation in Human Robot Interaction // Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora, 2013. P. 65—69. URL: <https://aclweb.org/anthology/W/W13/W13-3820v2.pdf> (дата обращения: 12.03.2016).

5. **Benotti L., Villalba M., Lau T., Cerruti J.** Corpus-based Interpretation of Instructions in Virtual Environments. // Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Jeju, Republic of Korea, 8–14 July 2012), 2012. Vol. 2: Short Papers. P. 181–186. URL: <https://aclweb.org/anthology/P/P12/P12-2036.pdf> (дата обращения: 8.03.2016).
6. **Carlos C. S.** Natural Language Programming Using Class Sequential Rules // Proceedings of the 5th International Joint Conference on Natural Language Processing, (Chiang Mai, Thailand, November 8–13, 2011), 2011. P. 237–245. URL: <https://aclweb.org/anthology/I/I11/I11-1027.pdf> (дата обращения: 13.03.2016).
7. **Marge M., Rudnicky A.** Comparing Spoken Language Route Instructions for Robots across Environment Representations // Proceedings of the SIGDIAL 2010 Conference (The University of Tokyo, September 24–25, 2010). 2010. P. 157–164. URL: <https://aclweb.org/anthology/W/W10/W10-4328.pdf> (дата обращения: 10.03.2016).
8. **Misra D. K., Tao K., Liang P., Saxena A.** Environment-Driven Lexicon Induction for High-Level Instructions. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Beijing, China, July 26–31 2015). 2015. Vol. 1: Long Papers. P. 992–1002. URL: <https://aclweb.org/anthology/P/P15/P15-1096.pdf> (дата обращения: 11.03.2016).
9. **She L., Yang S., Cheng Y., Jia Y., Cha J., Xi N.** Back to the Blocks World: Learning New Actions through Situated Human-Robot Dialogue. // Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL) (Philadelphia, U. S.A., 18–20 June 2014). 2014. P. 89–97. URL: <https://aclweb.org/anthology/W/W14/W14-4313.pdf> (дата обращения: 11.03.2016).
10. **Lei T., Long F., Barzilay R., Rinard M.** From Natural Language Specifications to Program Input Parsers // Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Sofia, Bulgaria, August 4–9 2013). 2013. Vol. 1: Long Papers, pp. 1294–1303. URL: <https://aclweb.org/anthology/P/P13/P13-1127.pdf> (дата обращения: 10.03.2016).
11. **Siri.** URL: <http://www.apple.com/ios/siri/> (дата обращения: 3.03.2017).
12. **Sobesednik HD.** URL: <http://sobesednik.idevelops.com/> (дата обращения: 3.03.2017).
13. **Briana Project.** URL: <http://www.brainasoft.com/braina/#overview> (дата обращения: 13.03.2017).
14. **NLUI Server.** URL: http://www.linguasys.net/web_production/server-item/NLUI%20Server (дата обращения: 9.02.2014).
15. **Fomichov V. A., Razorenov A. A.** A New Method of Extracting Structured Meanings from Natural Language Texts and Its Application // Proceedings of the 19th International Conference on Applications of Natural Language to Information Systems, NLDB 2014 (Montpellier, France, June 18–20, 2014). Cham: Springer International Publishing. 2014. P. 81–84.
16. **Razorenov A. A., Fomichov V. A.** The Design of a Natural Language Interface for File System Operations on the Basis of a Structured Meanings Model. *Procedia Computer Science*. 2014. N. 31. P. 1005–1011.
17. **Разоренов А. А.** Построение естественно-языкового интерфейса управления файловой системой на основе теории К-представлений // Информационные средства и технологии. Труды XXII Международной научно-технической конференции. В 3 т. Т. 3. М.: Издательский дом МЭИ, 2014. С. 59–65.
18. **Разоренов А. А.** Разработка метода управления программами на естественном языке // Научно-техническая конференция студентов, аспирантов и молодых специалистов НИУ ВШЭ им. Е. В. Арменского. Материалы конференции / Под общ. ред.: А. Н. Тихонов, В. Н. Азаров, У. В. Аристова, М. В. Карасев, В. П. Кулагин, Ю. Л. Леохин, Б. Г. Львов, Н. С. Титкова, С. У. Увайсов. М.: МИЭМ НИУ ВШЭ, 2015. С. 59–60.
19. **Фомичев В. А., Разоренов А. А.** Значение теории К-представлений для исследований по автоматическому выявлению семантических ролей // Информационные технологии. 2015. Т. 21. № 6. С. 403–411.
20. **Razorenov A. A., Fomichov V. A.** A new formal approach to semantic parsing of instructions and to file manager design // Database and Expert Systems Applications. Proceedings of the 27th International Conference, DEXA 2016, Porto, Portugal, September 5–8. 2016. Vol. 9827. Part. I. Cham: Springer, 2016. P. 416–430.
21. **Разоренов А. А., Фомичев В. А.** Компактная формализация входных и промежуточных данных алгоритмов семантического анализа предписаний // Информационные технологии. 2016. Т. 22, № 12. С. 883–891.
22. **Разоренов А. А., Фомичев В. А.** Новый подход к формализации семантической обработки предписаний на основе теории К-представлений // Информационные технологии. 2017. Т. 23, № 1. С. 3–14.
23. **Fomichov V. A.** Mathematical models of natural-language-processing systems as cybernetic models of a new kind // *Cybernetica. Quarterly Review of the International Association for Cybernetics* (Belgium, Namur). 1992. Vol. 35, N. 1. P. 63–91.
24. **Fomichov V. A.** K-Calculuses and K-Languages as Powerful Formal Means to Design Intelligent Systems Processing Medical Texts // *Cybernetica. Quarterly Review of the International Association for Cybernetics* (Belgium, Namur). 1993. Vol. 36, N. 2. P. 161–182.
25. **Fomichov V. A.** Integral Formal Semantics and the Design of Legal Full-Text Databases // *Cybernetica. Quarterly Review of the International Association for Cybernetics* (Belgium, Namur). 1994. Vol. 37, N. 2. P. 145–177.
26. **Fomichov V. A.** A Mathematical Model for Describing Structured Items of Conceptual Level // *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia). 1996. Vol. 20, N. 1. P. 5–32.
27. **Fomichov V. A.** Theory of Restricted K-calculuses as a Comprehensive Framework for Constructing Agent Communication Languages // Fomichov V. A., Zeleznikar A. P. (eds.). Special Issue on NLP and Multi-Agent Systems. *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia). 1998. Vol. 22, N. 4. P. 451–463.
28. **Fomichov V. A.** An ontological mathematical framework for electronic commerce and semantically-structured Web // Zhang Y., Fomichov V. A., Zeleznikar A. P. (Eds.) Special Issue on Database, Web, and Cooperative Systems. *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia). 2000. Vol. 24, N. 1. P. 39–49.
29. **Fomichov V. A.** Theory of K-calculuses as a powerful and flexible mathematical framework for building ontologies and designing natural language-processing systems // Andreassen T., Motro A., Christiansen H., Larsen H. L. (Eds.), *Flexible Query Answering Systems, 5th Intern. Conference, FQAS 2002, Proceedings, Lecture Notes in Artificial Intelligence*. Berlin, Heidelberg, New York: Springer. 2002. Vol. 2522. P. 183–196.
30. **Фомичев В. А.** Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть I // Информационные технологии. 2002. № 10. С. 16–25.
31. **Фомичев В. А.** Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть II // Информационные технологии. 2002. № 11. С. 34–45.
32. **Фомичев В. А.** Формализация проектирования лингвистических процессоров. М.: МАКС Пресс, 2005. 368 с.
33. **Фомичев В. А.** Математические основы представления содержания посланий компьютерных интеллектуальных агентов. М.: Издательство ТЕИС, 2007. 176 с.
34. **Fomichov V. A.** A Comprehensive Mathematical Framework for Bridging a Gap between Two Approaches to Creating a Meaning-Understanding Web // *Intern. Journal of Intelligent Computing and Cybernetics*. 2008. Vol. 1, N. 1. P. 143–163.
35. **Fomichov V. A.** Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms. IFSR International Series on Systems Science and Engineering. New York, Dordrecht, Heidelberg, London: Springer, 2010. Vol. 27. 352 p.
36. **Fomichov V. A.** Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web // *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia). 2010. Vol. 34, N. 3. P. 387–396.
37. **Fomichov V. A.** SK-languages as a Powerful and Flexible Semantic Formalism for the Systems of Cross-Lingual Intelligent Information Access // *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia). 2017. Vol. 41, N. 2. P. 221–232.
38. **Разоренов А. А.** Принципы применения теории К-представлений к разработке математических основ исполнения прикладными системами естественно-языковых предписаний // Информационные технологии. 2017. Т. 23, № 10.

Mathematical Foundations of Applying the Theory of K-Representations to Developing Algorithms of Executing Natural Language Instructions by Computer Systems

The paper continues the author's publication in *Informational Technologies 2017*, No. 10 describing the principles of applying the theory of K-representations to developing methodological foundations of designing computer systems processing natural language (NL) instructions. The goal of this paper is to formally define a binary relation of a new kind on semantic structures used as the ground of a new method of executing NL instructions by applied systems. The paper introduces a mathematical model describing the transformations of semantic representations (SRs) into another SRs. Besides, the paper proposes a mathematical model of forming the scripts/programs in the language of the controlled system for executing instructions expressed in the form of SRs. The properties of the introduced relations are investigated, and an experience of applying the developed method to designing a NL interface of a file manager is set forth.

Keywords: natural language processing, semantic parsing, theory of K-representations, SK-language, K-string, natural language interface, pattern matching relation, semantic-command dictionary, Natural Language Commander

References

1. Artzi Y., Zettlemoyer L. Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions. *Transactions of the Association for Computational Linguistics*, 2013, no. 1, pp. 49–62. Available at: <https://aclweb.org/anthology/Q/Q13/Q13-1005.pdf> (accessed 11.03.2016).
2. Babes-Vroman M., MacGlashan J., Gao R., Winner K., Adjogah R., desJardins M., Littman M., Muresan S. Learning to Interpret Natural Language Instructions. *Proceedings of the Second Workshop on Semantic Interpretation in an Actionable Context* (Montreal, Canada, June 3–8, 2012), 2012, pp. 1–6. Available at: <https://aclweb.org/anthology/W/W12/W12-2801.pdf> (accessed 11.03.2016).
3. Bisk Y., Yuret D., Marcu D. Natural Language Communication with Robots. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (San Diego, California, June, 2016), 2016, pp. 751–761. URL: <http://aclweb.org/anthology/N16-1089> (accessed 14.03.2017).
4. Bastianelli E., Castellucci C., Croce D., Basili R. Textual Inference and Meaning Representation in Human Robot Interaction. *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, 2013, pp. 65–69. Available at: <https://aclweb.org/anthology/W/W13/W13-3820v2.pdf> (accessed 12.03.2016).
5. Benotti L., Villalba M., Lau T., Cerruti J. Corpus-based Interpretation of Instructions in Virtual Environments. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* (Jeju, Republic of Korea, 8–14 July 2012), 2012, vol. 2: Short Papers, pp. 181–186. Available at: <https://aclweb.org/anthology/P/P12/P12-2036.pdf> (accessed 8.03.2016).
6. Carlos C. S. Natural Language Programming Using Class Sequential Rules. *Proceedings of the 5th International Joint Conference on Natural Language Processing*, (Chiang Mai, Thailand, November 8–13, 2011), 2011, pp. 237–245. Available at: <https://aclweb.org/anthology/I/I11/I11-1027.pdf> (accessed 13.03.2016).
7. Marge M., Rudnicky A. Comparing Spoken Language Route Instructions for Robots across Environment Representations. *Proceedings of the SIGDIAL 2010 Conference* (The University of Tokyo, September 24–25, 2010), 2010, pp. 157–164. Available at: <https://aclweb.org/anthology/W/W10/W10-4328.pdf> (accessed 10.03.2016).
8. Misra D. K., Tao K., Liang P., Saxena A. Environment-Driven Lexicon Induction for High-Level Instructions. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (Beijing, China, July 26–31 2015), 2015, vol. 1: Long Papers, pp. 992–1002. Available at: <https://aclweb.org/anthology/P/P15/P15-1096.pdf> (accessed 11.03.2016).
9. She L., Yang S., Cheng Y., Jia Y., Cha J., Xi N. Back to the Blocks World: Learning New Actions through Situated Human-Robot Dialogue. *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue* (SIGDIAL) (Philadelphia, U.S.A., 18–20 June 2014), 2014, pp. 89–97. Available at: <https://aclweb.org/anthology/W/W14/W14-4313.pdf> (accessed 11.03.2016).
10. Lei T., Long F., Barzilay R., Rinard M. From Natural Language Specifications to Program Input Parsers. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (Sofia, Bulgaria, August 4–9 2013), 2013, vol. 1: Long Papers, pp. 1294–1303. Available at: <https://aclweb.org/anthology/P/P13/P13-1127.pdf> (accessed 10.03.2016).
11. Web-resource: Siri. Available at: <http://www.apple.com/ios/siri/> (accessed 3.03.2017).
12. Web-resource: Sobesednik HD. Available at: <http://sobesednik.idevelops.com/> (accessed 3.03.2017).
13. Web-resource: Braina Project. Available at: <http://www.brainasoft.com/braina/#overview> (accessed 13.03.2017).
14. Web-resource: NLUI Server. Available at: http://www.linguasys.net/web_production/server-item/NLUI%20Server (accessed 9.02.2014).
15. Fomichov V. A., Razorenov A. A. A New Method of Extracting Structured Meanings from Natural Language Texts and Its Application. *Proceedings of the 19th International Conference on Applications of Natural Language to Information Systems, NLDB 2014* (Montpellier, France, June 18–20, 2014). Cham: Springer International Publishing, 2014, pp. 81–84.
16. Razorenov A. A., Fomichov V. A. The Design of a Natural Language Interface for File System Operations on the Basis of a Structured Meanings Model. *Procedia Computer Science*, 2014, no. 31, pp. 1005–1011.
17. Razorenov A. A. Postroenie estestvenno-yazykovogo interfejsa upravleniya fajlovoj sistemoy na osnove teorii K-predstavlenij (Building Natural Language Interface for File System Management Based on the Theory of K-representations) *Informatsionnye sredstva i tekhnologii. Trudy XXII Mezhdunarodnoj nauchno-tekhnicheskoy konferentsii. V 3 tomakh. T. 3. (Information Tools and Technologies. Proceedings of 22nd International Scientific and Technical Conference)* Moscow: Izdatel'skij dom MEI, 2014, vol. 3, pp. 59–65 (in Russian).
18. Razorenov A. A. Razrabotka metoda upravleniya programmami na estestvennom yazyke (Developing Method of Software Management with Natural Language Interface). *Nauchno-tekhnicheskaya konferenciya studentov, aspirantov i molodyh specialistov HSE im. E. V. Armenskogo. Materialy konferencii. Pod obshch. red.: A. N. Tihonov, V. N. Azarov, U. V. Aristova, M. V. Karasev, V. P. Kulagin, Yu. L. Leonhin, B. G. L'vov, N. S. Titkova, S. U. Uvajsov. (Proceedings of Scientific and Technical Conference for Students, PhD Students and Young Specialists of HSE named after E. V. Armenskiy)* Moscow: MIEM NRU HSE, 2015, pp. 59–60 (in Russian).
19. Fomichov V. A., Razorenov A. A. Znachenie teorii K-predstavlenij dlya issledovaniy po avtomaticheskomu vyavleniyu semanticheskikh roley (Significance of the Theory of K-Representations for the Studies on Automatic Semantic Role Labeling). *Informatsionnye Tekhnologii*, 2015, no. 6, pp. 403–411 (in Russian).
20. Razorenov A. A., Fomichov V. A. A new formal approach to semantic parsing of instructions and to file manager design *Database*

and Expert Systems Applications. *Proceedings of the 27th International Conference, DEXA 2016*, Porto, Portugal, September 5–8, 2016. Cham: Springer, 2016, vol. 9827. Part. I, pp. 416–430.

21. **Razorenov A. A., Fomichov V. A.** Kompaktnaya formalizatsiya vhodnykh i promezhutochnykh dannykh algoritmov semanticheskogo analiza predpisaniy (Compact Formalization of Instructions Semantic Parsing Algorithms' and Intermediary Data Structure). *Informatsionnye tekhnologii*, 2016, vol. 22, no. 12, pp. 883–891 (in Russian).

22. **Razorenov A. A., Fomichov V. A.** Novii podkhod k formalizatsii semanticheskoi obrabotki predpisaniy na osnove teorii K-predstavlenii (A New approach to Formalization of Instructions' Semantic Processing Based on the Theory of K-representations). *Informatsionnye tekhnologii*, 2017, vol. 23, no. 1, pp. 3–14 (in Russian).

23. **Fomichov V. A.** Mathematical models of natural-language-processing systems as cybernetic models of a new kind *Cybernetica. Quarterly Review of the International Association for Cybernetics* (Belgium, Namur), 1992, vol. 35, no. 1, pp. 63–91.

24. **Fomichov V. A.** K-Calculuses and K-Languages as Powerful Formal Means to Design Intelligent Systems Processing Medical Texts *Cybernetica. Quarterly Review of the International Association for Cybernetics* (Belgium, Namur), 1993, vol. 36, no. 2, pp. 161–182.

25. **Fomichov V. A.** Integral Formal Semantics and the Design of Legal Full-Text Databases *Cybernetica. Quarterly Review of the International Association for Cybernetics* (Belgium, Namur), 1994, vol. 37, no. 2, pp. 145–177.

26. **Fomichov V. A.** A Mathematical Model for Describing Structured Items of Conceptual Level *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia), 1996, vol. 20, no. 1, pp. 5–32.

27. **Fomichov V. A.** Theory of Restricted K-calculuses as a Comprehensive Framework for Constructing Agent Communication Languages *Fomichov V. A., Zeleznikar A. P. (eds.). Special Issue on NLP and Multi-Agent Systems. Informatica. An Intern. Journal of Computing and Informatics* (Slovenia), 1998, vol. 22, no. 4, pp. 451–463.

28. **Fomichov V. A.** An ontological mathematical framework for electronic commerce and semantically-structured Web. *Zhang Y., Fomichov V. A., Zeleznikar A. P. (Eds.) Special Issue on Database, Web, and Cooperative Systems. Informatica. An Intern. Journal of Computing and Informatics* (Slovenia), 2000, vol. 24, no. 1, pp. 39–49.

29. **Fomichov V. A.** Theory of K-calculuses as a powerful and flexible mathematical framework for building ontologies and designing natural language-processing systems *Andreasen T., Motro A., Chris-*

tiansen H., Larsen H. L. (Eds.), Flexible Query Answering Systems, 5th Intern. Conference, FQAS 2002, Proceedings, Lecture Notes in Artificial Intelligence. Berlin, Heidelberg, New York: Springer. 2002, vol. 2522, pp. 183–196.

30. **Fomichov V. A.** Matematicheskie osnovy predstavleniya smysla tekstov dlya razrabotki lingvisticheskikh informatsionnykh tekhnologii. Part I. *Informatsionnye Tekhnologii*, 2002, no. 10, pp. 16–25 (in Russian).

31. **Fomichov V. A.** Matematicheskie osnovy predstavleniya smysla tekstov dlya razrabotki lingvisticheskikh informatsionnykh tekhnologii. Part II. *Informatsionnye Tekhnologii*, 2002, no. 11, pp. 34–45 (in Russian).

32. **Fomichov V. A.** *Formalizatsiya proektirovaniya lingvisticheskikh protessorov*. Moscow: MAKS Press, 2005. 368 p. (in Russian).

33. **Fomichov V. A.** *Matematicheskie osnovy predstavleniya sodержaniya poslanii kompyuternykh intellektualnykh agentov*. Moscow: State University — Higher School of Economics, Publishing House "TEIS", 2007. 176 p. (in Russian).

34. **Fomichov V. A.** A Comprehensive Mathematical Framework for Bridging a Gap between Two Approaches to Creating a Meaning-Understanding Web. *Intern. Journal of Intelligent Computing and Cybernetics*, 2008, vol. 1, no. 1, pp. 143–163.

35. **Fomichov V. A.** *Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms*. IFSR International Series on Systems Science and Engineering, vol. 27. New York, Dordrecht, Heidelberg, London: Springer, 2010. 352 p.

36. **Fomichov V. A.** Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web. *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia), 2010, vol. 34, no. 3, pp. 387–396.

37. **Fomichov V. A.** SK-languages as a Powerful and Flexible Semantic Formalism for the Systems of Cross-Lingual Intelligent Information Access. *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia), 2017, vol. 41, no. 2, pp. 221–232.

38. **Razorenov A. A.** Printsipy primeneniya teorii K-predstavlenij k razrabotke matematicheskikh osnov ispolneniya prikladnymi sistemami estestvenno-jazykovykh predpisaniy (The principles of applying the theory of K-representations to developing mathematical foundations of executing natural language instructions by computer systems). *Informatsionnye tekhnologii*, 2017, vol. 23, no. 10 (in Russian).

УДК 51-74/330.4

Ю. А. Зак, д-р техн. наук, эксперт и консультант, e-mail: Yuriy_Zack@hotmail.com;
Аахен, Германия

Об одном алгоритме Fuzzy-регрессионного анализа

Предложены методы решения задачи Fuzzy-регрессионного анализа в условиях, когда входные, выходная переменная и свободный член уравнения регрессии представлены нечеткими множествами самого общего вида, а коэффициенты регрессии — действительные числа. В качестве критериев аппроксимации использованы сумма квадратов среднезвешенных координат минимальных и максимальных значений абсцисс различных сечений функций принадлежности нечетких множеств выходной переменной и их оценок по Fuzzy-регрессионной модели, а также координат центров тяжести функций принадлежности этих Fuzzy-множеств. Получены детерминированные эквиваленты данного класса нечетких регрессионных моделей, оценки адекватности построенных уравнений регрессии и приведена вычислительная схема алгоритма решения задачи.

Ключевые слова: Fuzzy-регрессионный анализ, нечеткие множества, нечисловая статистика, лингвистические и булевы переменные, метод наименьших квадратов

Введение

В эконометрике, маркетинговых и социологических исследованиях при построении математических моделей широко используется аппарат математической статистики [1]. Полученные математи-

ческие модели применяют преимущественно для прогноза экономических и социально-экономических показателей, характеризующих состояние и развитие анализируемых процессов, а также для имитации различных возможных сценариев социально-экономического развития анализируемых сис-

тем. Так как в экономике невозможно проведение управляемого активного эксперимента, широкое распространение получили здесь регрессионные модели, построенные на основе статистических данных предыдущих периодов или аналогичных объектов [1]. Характерная особенность построения регрессионных моделей в этих областях исследования заключается в том, что широкое распространение здесь получили методы нечисловой статистики (см., например, [2]), характерными представителями которой являются:

— бинарные отношения, значения качественных признаков, т.е. результаты кодировки объектов (например, ответов на вопросы экспертной, маркетинговой социологической анкеты) с помощью заданного перечня категорий (градаций);

— результаты парных сравнений;

— упорядочения (ранжировки), классификации, толерантности;

— множества (обычные или нечеткие).

Большинство отмеченных входных факторов — это показатели нечисловой природы, которые зачастую могут быть определены субъективно и представлены либо в виде лингвистических переменных с определенным количеством лингвистических термов, либо в виде некоторого числа градаций числовой шкалы или булевых переменных.

Отметим, что в экономике доля нечисловых данных существенно выше, чем в технике и технологии, и неопределенность зачастую приходится описывать не в терминах вероятностно-статистических моделей, а иными способами, например в терминах теории нечеткости или математики и статистики интервальных данных. Даже обычные данные, рассматриваемые в этих моделях как детерминированные числовые величины (например, рыночная цена товара, прибыль предприятия, индексы инфляции и дисконтирования, валовой внутренний продукт, дисконтированный поток платежей и т. д.), как и многие другие макроэкономические показатели, могут быть определены лишь с точностью 5...10 %. Фактически подобные величины не имеют определенного численного значения, они размыты, имеют нечисловой характер, и описывать их представляется целесообразным с помощью нечисловых математических понятий, относящихся к тем или иным классам объектов нечисловой природы, таким как нечеткие множества, интервалы распределения вероятностей и др. Аналогичные проблемы возникают также и в социологических, маркетинговых исследованиях и в логистике.

Алгоритмы построения регрессионных моделей предусматривают представление всей входной информации только в форме действительных чисел. В ситуациях, когда многие входные факторы модели не являются числовыми величинами, в качестве альтернативных подходов можно использовать методы *Fuzzy*-регрессионного анализа [3, 4]. Эти

методы позволяют в качестве исходной информации использовать как нечеткую информацию, так и информацию, представленную действительными числами. Результатом расчета на основе математических моделей *Fuzzy*-регрессионного анализа является некоторое нечеткое множество с функцией принадлежности непрерывного вида, которое определяет диапазон возможных значений выходной переменной и оценку (некоторый аналог вероятности) получения этого значения в пределах данного диапазона. В этих условиях применение *Fuzzy*-регрессионных моделей прогнозирования, которые отражают возможный фактический диапазон реальных значений выходной величины, является экономически оправданным и может в дальнейшем быть более эффективно использован в процессах анализа и принятия решений (см., например, [5—9, 14, 25]). Методы преобразования термов лингвистических переменных, интервальных данных, показателей числовых шкал и булевых переменных в нечеткие множества с функцией принадлежности определенного вида рассматривались в работах автора [4, 5, 8, 26], и поэтому в данной работе на них останавливаться не будем.

1. Состояние разработок в области *Fuzzy*-регрессионного анализа

Известны три метода *Fuzzy*-регрессионного анализа [22—24]:

а) нечеткая регрессия, основанная на критерии минимизации нечеткости;

б) метод, получивший название FLSRA (*Fuzzy least-square regression analysis*), который, в свою очередь, имеет две разновидности, в одной из которых используется критерий максимальной совместности, а в другой — критерий минимизации нечеткости;

в) регрессия интервала [19, 22].

Во всех трех методах в качестве исходной информации могут быть использованы как нечеткие множества, представленные функциями принадлежности, так и детерминированные данные.

В расчетах нечетких коэффициентов модели используются два критерия:

- для всех рассматриваемых данных принадлежность фактического значения выходной переменной к его нечеткой оценке должна быть не ниже некоторой величины, определяемой как уровень доверия;
- общая нечеткость предсказываемого значения зависимой переменной должна быть минимизирована; это может быть достигнуто минимизацией суммы разбросов нечетких чисел для всех наборов данных.

При этом во многих работах используется дефазифицированное значение выходной переменной или ее оценки по *Fuzzy*-регрессионной модели либо критерии вида $\max\{0; \mu_{\bar{Y}_i}[F(Y_i)] - \mu_{\bar{Y}_i}(Y_i)\}$ —

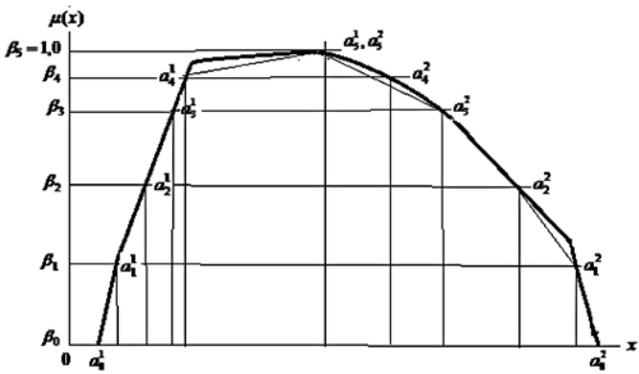
ограниченная разность нечетких чисел. Здесь $\mu_{\bar{Y}}(Y_i)$ и $\mu_{\bar{F}}[F(Y_i)]$ — соответственно функции принадлежности нечеткого множества выходной переменной и ее расчетного значения на основе *Fuzzy*-регрессионной модели. В большинстве публикаций по данной тематике либо рассматривали некоторые частные случаи одной из этих общих постановок задачи, либо давали интересные новые приложения ее применения, либо описывали алгоритмы решения известных постановок этой задачи. П. Даймонд [16, 17] ввел новое понятие — расстояние на множестве нечетких чисел между прогнозируемыми и экспериментальными данными. В работах Х. Танака (1982 г.) [13, 14], как и в работах [15–17, 24] и многих других публикациях, рассмотрена модель линейной регрессии с нечеткими коэффициентами в виде треугольных *Fuzzy*-чисел. В работах [13, 14] для определения значений этих коэффициентов, минимизирующих суммарную средневзвешенную размытость параметров функции принадлежности, рассматриваемую в различных метриках, предложены методы линейного программирования. В 1987 г. А. Келминс [15] и П. Даймонд [16, 17], а также Р. Rousseeuw [21], Янг и Лиу в 2003 г. [22], также предложили методику построения моделей нечеткой регрессии методом наименьших квадратов. Для построения критериев аппроксимации (целевая функция нечеткой идентификации) использовали различные метрики, среди которых наибольшее распространение получили показатели α -сечений нечетких множеств. В ряде случаев сформулированная оптимизационная задача становится нелинейной и многоэкстремальной. Для ее решения применяли градиентные, поисковые методы и генетические алгоритмы (см., например, Алиев в [7]). В работах С. Д. Штовбы [10] при построении регрессионных моделей с нечеткими коэффициентами каждой точке факторного пространства ставится в соответствие нечеткое число с параметрической функцией принадлежности, определенной с точностью до неизвестных параметров. Зависимость параметров этой функции принадлежности от влияющих факторов описывается четкими моделями, коэффициенты которых определяются методами регрессионного анализа на основе экспериментальной выборки данных. В. А. Домрачев и О. М. Полещук [12] рассмотрели построение нечеткой комбинированной линейной регрессионной модели, в которой исходные данные и коэффициенты модели — *Fuzzy*-множества в виде унимодальных (L - R) чисел, и предложили метод решения задачи, основанный на определении взвешенных отрезков для толерантных чисел с последующим применением метода наименьших квадратов. Робастные регрессионные модели представлены в работах [11, 20, 21].

В данной работе представлено решение задач нечеткого регрессионного анализа в условиях, когда входные и выходная переменные представлены

Fuzzy-множествами самого общего вида, а коэффициенты регрессии — положительные действительные числа. Свободный член уравнения регрессии — нечеткое множество. Такая постановка задачи актуальна для построения многих моделей макроэкономики. Рассмотрены некоторые новые критерии аппроксимации, основанные на сравнении средневзвешенных координат минимальных и максимальных значений абсцисс α -сечений функций принадлежности нечетких множеств выходной переменной и ее оценки по *Fuzzy*-регрессионной модели, а также координат центров тяжести функций принадлежности *Fuzzy*-множеств. Число сечений, значения функции принадлежности каждого из выбираемых сечений и весовые коэффициенты соответствующих отклонений, учитывающихся в критерии аппроксимации, определяются экспертами и лицом, принимающим решение. Результат расчета *Fuzzy*-регрессионной модели — нечеткое множество, функция принадлежности которого — многоугольник, соединяющий координаты расчетных точек. Разработаны детерминированные эквиваленты сформулированных задач и алгоритмы расчета параметров критериев аппроксимации, детерминированных значений коэффициентов уравнения регрессии, а также свободного члена, который представлен *Fuzzy*-множеством, определенным с точностью до неизвестных параметров. Эти алгоритмы основаны на методе наименьших квадратов. Полученные результаты позволят разработать эффективные алгоритмы построения моделей для многих частных видов функций принадлежности исходных данных задачи, а также решить многие прикладные проблемы в экономике, логистике, социологии и маркетинге.

2. Критерии аппроксимации, математическая модель и алгоритм решения задачи

Нечеткое множество с функцией принадлежности произвольного вида аппроксимируем некоторой ломаной линией, соединяющей точки минимальных $a_0^1, a_1^1, \dots, a_k^1, \dots, a_K^1$ и максимальных значений координат абсцисс $a_0^2, a_1^2, \dots, a_k^2, \dots, a_K^2$ некоторого множества сечений $\beta_0, \beta_1, \dots, \beta_k, \dots, \beta_K$ и представляем его многоугольником $(a_0^1, a_1^1, \dots, a_k^1, \dots, a_K^1, a_K^2, a_{K-1}^2, \dots, a_k^2, \dots, a_0^2)$. Такая аппроксимация поясняется на рисунке, на котором выбрано пять сечений: $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$ и исходное множество представлено многоугольником $(a_0^1, a_1^1, a_2^1, a_3^1, a_4^1, a_5^1, a_5^2, a_4^2, a_3^2, a_2^2, a_1^2, a_0^2)$.



Аппроксимация нечетких множеств

В данной работе рассматриваются в основном нечеткие множества с LR -представлением функции принадлежности (см., например, [6–9]).

В задаче *Fuzzy*-регрессионного анализа в каждом из N комплектов информации, $i = 1, \dots, N$, нечеткие множества входной X_{ik} и выходной переменной Y_i представляются $2(K + 1)$ детерминированными параметрами, соответственно $a_0^1(X_i^j), \dots, a_k^1(X_i^j), \dots, a_K^1(X_i^j), a_0^2(X_i^j), \dots, a_k^2(X_i^j), \dots, a_K^2(X_i^j)$ и $a_0^1(Y_i), \dots, a_k^1(Y_i), \dots, a_K^1(Y_i), a_0^2(Y_i), \dots, a_k^2(Y_i), \dots, a_K^2(Y_i)$.

В качестве неизвестных детерминированных параметров (коэффициентов модели) определены детерминированные значения $b^0, b^1, \dots, b^j, \dots, b^n$, и уравнение регрессии будем искать в виде

$$a_k^r(\bar{Y}) = \sum_{j=0}^n b^j \cdot a_k^{jr}(X^j) + B_k, \quad r = 1, 2; \quad k = 0, 1, \dots, K. \quad (1)$$

Здесь $B(B_0, B_1, \dots, B_k, \dots, B_K)$ — нечеткое множество с функцией принадлежности, определенной с точностью до неизвестных значений детерминированных параметров.

Задача выбора коэффициентов уравнения регрессии может быть представлена в виде задачи многокритериальной оптимизации вида

$$F_k = \min_{(b^0, b^1, \dots, b^j, \dots, b^n, B_k)} \sum_{i=1}^N \left\{ \eta_{k,1} \left[a_k^1(Y_i) - \sum_{j=0}^n b^j \cdot a_k^1(X_i^j) - B_{k,1} \right]^2 + \eta_{k,2} \left[a_k^2(Y_i) - \sum_{j=1}^n b^j \cdot a_k^2(X_i^j) - B_{k,2} \right]^2 \right\}, \quad k = 0, 1, \dots, K, \quad (2)$$

где $0 \leq \eta_{k,1} \leq 1, 0 \leq \eta_{k,2} \leq 1$ — весовые коэффициенты, удовлетворяющие соотношению

$\sum_{k=0}^K (\eta_{k,1} + \eta_{k,2}) = 1; a_k^1(Y_i), a_k^2(Y_i)F_k$ и $a_k^1(X_i^j), a_k^2(X_i^j)$ — соответственно минимальные и максимальные значения координаты абсцисс *Fuzzy*-множеств выходной переменной и j -й входной переменной в k -м сечении i -го комплекта информации (исходных данных); $B_{k,1}$ и $B_{k,2}$ — соответственно значения левой и правой координат абсциссы *Fuzzy*-множества свободного члена уравнения регрессии в k -м сечении.

В качестве одного из методов решения сформулированной задачи рассмотрим аддитивную свертку локальных критериев:

$$\Phi_1 = \min_{(b^0, \dots, b^j, \dots, b^n, B_{k,1}, B_{k,2})} \sum_{k=0}^K \left\{ \eta_{k,1} \sum_{i=1}^N \left[a_k^1(Y_i) - \sum_{j=1}^n b^j a_k^1(X_i^j) - B_{k,1} \right]^2 + \eta_{k,2} \sum_{i=1}^N \left[a_k^2(Y_i) - \sum_{j=1}^n b^j a_k^2(X_i^j) - B_{k,2} \right]^2 \right\}. \quad (3)$$

Здесь $0 \leq \eta_{k,1} = \eta_k \eta_1 \leq 1, 0 \leq \eta_{k,2} = \eta_k \eta_2 \leq 1$ — весовые коэффициенты, удовлетворяющие условию $\sum_{k=0}^K (\eta_{k,1} + \eta_{k,2}) = 1$.

Необходимые и достаточные условия минимума критерия оптимальности (3) — равенство нулю первой производной этого критерия:

$$\frac{d\Phi_1}{db^j} = 2 \sum_{k=0}^K \left\{ \eta_{k,1} \sum_{i=1}^N \left[a_k^1(Y_i) - \sum_{j=1}^n b^j a_k^1(X_i^j) - B_{k,1} \right] a_k^1(X_i^j) + \eta_{k,2} \sum_{i=1}^N \left[a_k^2(Y_i) - \sum_{j=1}^n b^j a_k^2(X_i^j) - B_{k,2} \right] a_k^2(X_i^j) \right\} = 0, \quad j = 1, \dots, n. \quad (4)$$

Следовательно, система линейных алгебраических уравнений относительно переменных $b^j, j = 1, \dots, n$, и $B_{k,1}, B_{k,2}, k = 0, 1, \dots, K$, имеет вид

$$b^1 \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(X_i^1) a_k^j(X_i^j) + \eta_{k,2} a_k^2(X_i^1) a_k^j(X_i^j)] + \dots + b^j \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} (a_k^1(X_i^j))^2 + \eta_{k,2} (a_k^2(X_i^j))^2] + \dots + b^n \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(X_i^n) a_k^j(X_i^j) + \eta_{k,2} a_k^2(X_i^n) a_k^j(X_i^j)] + \sum_{k=0}^K B_{k,1} N \eta_{k,1} + \sum_{k=0}^K B_{k,2} N \eta_{k,2} = 0$$

$$= \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(Y_i) a_k^j(X_i^j) + \eta_{k,2} a_k^2(Y_i) a_k^j(X_i^j)],$$

$$j = 1, \dots, n. \quad (5)$$

$$W_j^1 b^1 + \dots + W_j^j b^j + \dots + W_j^n b^n + \sum_{k=0}^K V_{j,k,1} B_{k,1} +$$

$$+ \sum_{k=0}^K V_{j,k,2} B_{k,2} = T_j, j = 1, \dots, n, \quad (6)$$

где

$$W_j^l = \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(X_i^l) a_k^j(X_i^j) + \eta_{k,2} a_k^2(X_i^l) a_k^j(X_i^j)];$$

$$W_j^j = \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} (a_k^1(X_i^j))^2 + \eta_{k,2} (a_k^2(X_i^j))^2];$$

$$V_{j,k,1} = N\eta_{k,1}; V_{j,k,2} = N\eta_{k,2}, k = 0, 1, \dots, K;$$

$$T_j = \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(Y_i) a_k^j(X_i^j) + \eta_{k,2} a_k^2(Y_i) a_k^j(X_i^j)];$$

$$l, j = 1, \dots, n. \quad (7)$$

$$\frac{d\Phi_1}{dB_{k,l}} = 2 \sum_{k=0}^K \left\{ \eta_{k,1} \sum_{i=1}^N \left[a_k^1(Y_i) - \sum_{j=1}^n b^j a_k^1(X_i^j) - B_{k,1} \right] + \eta_{k,2} \sum_{i=1}^N \left[a_k^2(Y_i) - \sum_{j=1}^n b^j a_k^2(X_i^j) - B_{k,2} \right] \right\} = 0, l = 1, 2; k = 0, 1, \dots, K.$$

Следовательно, система линейных алгебраических уравнений относительно переменных $b^j, j = 1, \dots, n$, и $B_{k,1}, B_{k,2}, k = 0, 1, \dots, K$, имеет вид

$$b^1 \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(X_i^1) + \eta_{k,2} a_k^2(X_i^1)] + \dots +$$

$$+ b^j \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(X_i^j) + \eta_{k,2} a_k^2(X_i^j)] + \dots +$$

$$+ b^n \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(X_i^n) + \eta_{k,2} a_k^2(X_i^n)] +$$

$$+ \sum_{k=0}^K B_{k,1} N\eta_{k,1} + \sum_{k=0}^K B_{k,2} N\eta_{k,2} =$$

$$= \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(Y_i) + \eta_{k,2} a_k^2(Y_i)],$$

$$k = 0, 1, \dots, 2(K+1). \quad (8)$$

То есть

$$U_k^1 b^1 + \dots + U_k^j b^j + \dots + U_k^n b^n + R_{k,1} B_{k,1} +$$

$$+ R_{k,2} B_{k,2} = G_k, k = 0, 1, \dots, 2(K+1), \quad (9)$$

где

$$U_k^j = \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(X_i^j) + \eta_{k,2} a_k^2(X_i^j)],$$

$$R_{k,1} = N\eta_{k,1}, R_{k,2} = N\eta_{k,2};$$

$$G_k = \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(Y_i) + \eta_{k,2} a_k^2(Y_i)],$$

$$k = 0, 1, \dots, K. \quad (10)$$

Здесь $0 \leq \eta_{k,1} \leq 1$ и $0 \leq \eta_{k,2} \leq 1, k = 0, 1, \dots, K$, — весовые коэффициенты, удовлетворяющие соотношению $\sum_{k=0}^K (\eta_{k,1} + \eta_{k,2}) = 1$.

Таким образом для расчета детерминированных коэффициентов уравнения регрессии $b^j, j = 1, \dots, n$ и *Fuzzy*-множества (параметров свободного члена $B_{k,1}, B_{k,2}, k = 0, 1, \dots, K$) необходимо решить систему $n + 2(K+1)$ линейных алгебраических уравнений (5), (8) или (6), (9), коэффициенты и правые части которых вычисляются соответственно по формулам (7) и (10), относительно $n + 2(K+1)$ неизвестных.

Так как система линейных алгебраических уравнений (6), (9) может быть достаточно большого размера, рассмотрим другой алгоритм решения задачи, предусматривающий существенно меньший объем вычислений. На первом этапе решения задачи уравнение регрессии (1) и критерий оптимальности (3) представим без *Fuzzy*-множества свободного члена, т. е. без переменных $B_{k,1}$ и $B_{k,2}$. Тогда на первом этапе решается система n линейных алгебраических уравнений относительно n неизвестных коэффициентов уравнения регрессии $b^j, j = 1, \dots, n$, вида

$$b^1 g_j^1 + \dots + b^j g_j^j + \dots + b^n g_j^n = Q_j; j = 1, \dots, n, \quad (11)$$

где

$$g_j^l = \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(X_i^l) a_k^j(X_i^j) + \eta_{k,2} a_k^2(X_i^l) a_k^j(X_i^j)],$$

$$l = 1, \dots, n, l \neq j;$$

$$g_j^j = \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} (a_k^1(X_i^j))^2 + \eta_{k,2} (a_k^2(X_i^j))^2],$$

$$Q_j = \sum_{i=1}^N \sum_{k=0}^K [\eta_{k,1} a_k^1(Y_i) a_k^j(X_i^j) + \eta_{k,2} a_k^2(Y_i) a_k^j(X_i^j)];$$

$$j = 1, \dots, n. \quad (12)$$

В результате выполненных вычислений для каждого комплекта информации $i = 1, \dots, N$ могут быть вычислены расчетные значения левой и правой частей координат всех $k = 0, 1, \dots, K$ сечений нечеткого множества выходной переменной Z_i , т.е. значения $Z_{i,k,1}$ и $Z_{i,k,2}, i = 1, \dots, N, k = 0, 1, \dots, K$. Оп-

ределим параметры свободного члена в результате минимизации следующего функционала:

$$\bar{\Phi} = \min_{B_{k,1}, B_{k,2}, k=0, 1, \dots, K} \sum_{i=1}^N [\eta_{k,1} (a_k^1(Y_i) - Z_{i,k,1} - B_{k,1})^2 + \eta_{k,2} (a_k^2(Y_i) - Z_{i,k,2} - B_{k,2})^2]. \quad (13)$$

Определим необходимые и достаточные условия минимума:

$$\frac{d\bar{\Phi}}{dB_{k,1}} = \eta_{k,1} \sum_{i=1}^N (a_k^1(Y_i) - Z_{i,k,1} - B_{k,1}) = 0;$$

$$\frac{d\bar{\Phi}}{dB_{k,2}} = \eta_{k,2} \sum_{i=1}^N (a_k^2(Y_i) - Z_{i,k,2} - B_{k,2}) = 0,$$

$$k = 0, 1, \dots, K. \quad (14)$$

Следовательно, значения $B_{k,1}$ и $B_{k,2}$, $k = 0, 1, \dots, K$, можно вычислить по простым формулам

$$B_{k,1} = \frac{1}{N} \sum_{i=1}^N (a_k^1(Y_i) - Z_{i,k,1}) = 0,$$

$$B_{k,2} = \frac{1}{N} \sum_{i=1}^N (a_k^2(Y_i) - Z_{i,k,2}), \quad k = 0, 1, \dots, K. \quad (15)$$

В результате расчетов значений выходной переменной по уравнению регрессии (1) для каждого i -го комплекта информации будут вычислены параметры $a_k^1(Y_i)$ и $a_k^2(Y_i)$, $i = 1, \dots, N$, нечеткого множества выходной переменной Y_i , т.е. значения $a_k^1(Y_i)$ и $a_k^2(Y_i)$, $k = 0, 1, \dots, K$, соединив которые ломаной линией можно будет построить функцию принадлежности этого нечеткого множества. Заметим, что если бы в результате расчетов значение некоторого коэффициента b^j , $j = 1, \dots, n$, получилось бы отрицательным, то в левой части уравнения регрессии значение соответствующих коэффициентов $a_k^1(X_{ij})$ и $a_k^2(X_{ij})$, $i = 1, \dots, N$, необходимо было бы подставлять из условий представления соответствующего обратного *Fuzzy*-множества.

В качестве другой математической модели *Fuzzy*-регрессионного анализа рассмотрим более простой детерминированный эквивалент задачи, когда в качестве характеристики нечеткого множества представлена координата абсциссы его функции принадлежности

$$G(X_{ij}) = \frac{\int_{-\infty}^{\infty} \bar{X}_{ij} \mu_i(X_{ij}) dX_{ij}}{\int_{-\infty}^{\infty} \mu_i(X_{ij}) dX_{ij}}; \quad G(Y_i) = \frac{\int_{-\infty}^{\infty} Y_i \mu_i(Y) dY_i}{\int_{-\infty}^{\infty} \mu_i(Y) dY_i},$$

$$j = 1, \dots, n; \quad i = 1, \dots, N. \quad (16)$$

Расчет детерминированных значений коэффициентов регрессии b^j , $j = 1, \dots, n$, в котором отсут-

ствует свободный член B , выполняется из условий минимизации функционала

$$\Phi_1 = \min_{b^1, b^2, \dots, b^j, \dots, b^n} \sum_{i=1}^N (G(Y_i) - [b^1 G(X_i^1) + b^2 G(X_i^2) + \dots + b^j G(X_i^j) + \dots + b^n G(X_i^n)])^2 \quad (17)$$

и сводится к решению следующей системы n линейных алгебраических уравнений относительно переменных b^j , $j = 1, \dots, n$:

$$b^1 \sum_{i=1}^N G(X_i^1) G(X_i^j) + b^2 \sum_{i=1}^N G(X_{i2}) G(X_i^j) + \dots + b^j \sum_{i=1}^N [G(X_i^j)]^2 + \dots + b^n \sum_{i=1}^N G(X_{in}) G(X_i^j) =$$

$$= \sum_{i=1}^N G(Y_i) G(X_i^j), \quad j = 1, \dots, n.$$

Определив коэффициенты переменных уравнения регрессии, параметры сечений нечеткого множества свободного члена B , т.е. значения левых $B_{k,1}$ и правых их частей $B_{k,2}$, $k = 1, \dots, K$, этого нечеткого множества, определяются, как и в выше описанном алгоритме, по формулам (15).

В самом общем случае значения коэффициентов регрессии уравнения *Fuzzy*-регрессии, полученные в результате решения системы линейных алгебраических уравнений, могут быть как положительными, так и отрицательными величинами. Заметим, что если в результате выполненных расчетов какой-либо из коэффициентов уравнений линейной регрессии b^j является отрицательным, то соответствующие параметры нечеткого множества этого члена уравнения определяются по формулам

$$\alpha_k^{1j} (Z_{k,i}^{1j} = b^j \alpha_k^{1j} (X_i^j)) = b^j \alpha_k^{2j} (X_i^j);$$

$$\alpha_k^{2j} (Z_{k,i}^{2j} = b^j \alpha_k^{2j} (X_i^j)) = b^j \alpha_k^{1j} (X_i^j), \quad k = 1, \dots, K,$$

что должно быть предусмотрено при построении математической модели.

Эта ситуация требует существенно большего объема вычислений и не является темой этой статьи.

3. Оценка адекватности *Fuzzy*-регрессионной модели

Результат расчета выходных показателей на основе нечеткой регрессионной модели — *Fuzzy*-множество $F(Y_i)$, функция принадлежности которого $\mu_{Y_i}[F(Y_i)]$ представлена многоугольником заданного вида (см. рис. 1). В качестве детерминированного аналога прогнозируемой величины могут быть приняты средние квадратические значения суммы квадратов расчетного и фактического отклонений:

— координат абсцисс центра тяжести нечеткого множества, полученного в результате расчета по

Fuzzy-регрессионной модели $(G[Y_i] - G[Z_i = F(Y_i)])^2$, где $G[Z_i = F(Y_i)]$ вычисляется по формулам аналогично (16);

— некоторая средневзвешенная величина значений разности координат фактического и расчетного значений функций принадлежности нечеткого множества $(Q[Y_i] - Q[Z_i = F(Y_i)])^2$:

$$Q[Y_i] = \sum_{k=0}^K \sum_{r=1}^2 \eta_{k,r} a_r^k(Y_i),$$

$$Q[Z_i = F(Y_i)] = \sum_{k=0}^K \sum_{r=1}^2 \eta_{k,r} a_r^k[F(Y_i)], \quad i = 1, \dots, N,$$

где $a_1^k[Z_i = F(Y_i)]$ и $a_2^k[Z_i = F(Y_i)]$ — соответственно левые и правые крайние точки координат оси абсцисс функции принадлежности нечеткого множества, построенного по уравнениям регрессионной модели для i -го комплекта информации; $\eta_{k,1}, \eta_{k,2}$ — весовые коэффициенты, значения которых в частном случае могут быть приняты такими же, как при расчете параметров регрессионной модели;

— среднее квадратическое значение суммы отклонений расчетного и фактического значений функций принадлежности одних и тех же контрольных точек выходной переменной

$$(\Lambda[\mu_{\bar{Y}}(Y_i)] - \Lambda[\mu_{F(\bar{Z})}(Z_i)])^2,$$

где

$$\Lambda[\mu_{\bar{Y}}(Y_i)] = \sum_{k=0}^K \sum_{r=1}^2 \eta_{k,r} \mu_{\bar{Y}}[a_r^k(Y_i)],$$

$$\Lambda[\mu_{\bar{Y}}(Z_i)] = \sum_{k=0}^K \sum_{r=1}^2 \eta_{k,r} \mu_{\bar{Z}}[a_r^k(Z_i)], \quad i = 1, \dots, N.$$

В качестве оценки качества прогнозирования на основе нечеткой регрессионной модели может рассматриваться следующий показатель:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N [W(Y_i^{\text{расч}} = Y_i) - W(Y_i^{\Phi} = Z_i)]^2}, \quad (18)$$

где в качестве значения $W(Y_i^{\Phi})$ могут быть использованы показатели $G[Y_i]$, $Q[Y_i]$ или $\Lambda[\mu_{\bar{Y}}(Y_i)]$, а в качестве значения $W(Y_i^{\text{расч}})$ — показатели $G(Z_i)$, $Q(Z_i)$ или $\Lambda[\mu_{F(\bar{Z})}(Z_i)]$.

С достаточной для практических приложений точностью в большинстве случаев можно использовать построенные Fuzzy-регрессионные модели для прогнозирования значения переменной состав-

ляющей затрат, если справедливы следующие показатели их адекватности:

$$\sigma \leq \vartheta \sqrt{\frac{1}{N-1} \sum_{i=1}^N [W(Y_i^{\Phi}) - M\{W(Y_i^{\Phi})\}]^2},$$

где

$$M\{W(Y_i^{\Phi})\} = \frac{1}{N} \sum_{i=1}^N W(Y_i^{\Phi}), \quad \text{а } \vartheta \leq 0,1.$$

Заключение

В условиях, когда в статистической выборке некоторые или все входные и выходная переменные представлены нечисловой информацией (булевы, лингвистические переменные, данные числовых шкал, Fuzzy-множества и т.п.), использование Fuzzy-регрессионных моделей является эффективной альтернативой получения количественных зависимостей установленных экспертами качественных закономерностей изучаемых явлений.

В работе предложены методы решения задачи в условиях, когда входные и выходная переменные и свободный член уравнения регрессии представлены Fuzzy-множествами самого общего вида, а коэффициенты регрессии — действительные числа. В качестве критериев аппроксимации использованы сумма квадратов средневзвешенных координат минимальных и максимальных значений абсцисс сечений функций принадлежности нечетких множеств выходной переменной и их оценок по Fuzzy-регрессионной модели, а также координат центров тяжести функций принадлежности этих Fuzzy-множеств. Построены детерминированный эквивалент и приведена вычислительная схема алгоритма решения задачи.

Для расчета значения выходной переменной по уравнению Fuzzy-модели необходимо приведенными в работе формулами преобразовать все входные переменные, представленные булевыми, лингвистическими переменными, данными числовых шкал или Fuzzy-множествами, в числовые величины, подставить их в уравнение регрессии и определить расчетные параметры функции принадлежности нечеткого множества выходной переменной. Полученные в работе результаты расширяют область приложения Fuzzy-регрессионных моделей в экономике, социологии, маркетинге и других приложениях.

Приложение

Иллюстративный пример

Рассмотрим простейший пример построения Fuzzy-регрессионного уравнения $Y = bX + B$ в случае наличия только одной входной переменной на основе восьми комплектов информации, представленных семью параметрами абсцисс их сечений.

Значения этих параметров для входной и выходной переменных представлены соответственно в табл. 1 и 2. Весовые коэффициенты параметров этих сечений заданы вектором $\eta = (0,05; 0,1; 0,2; 0,3; 0,2; 0,1; 0,005)$. Значения функции принадлежности этих сечений и значения соответствующих весовых коэффициентов сведены в табл. 3.

Коэффициент уравнения нечеткой регрессии b определим из условий минимизации следующего функционала:

$$\Phi = \min_b \sum_{i=1}^8 \sum_{k=0}^6 \eta_k [a_k^i(Y_i) - b a_k^i(X_i)]^2.$$

Таблица 1

№ п/п	Параметры нечетких множеств входной переменной $a_k^i(X_i)$						
	0	1	2	3	4	5	6
1	1,0	2,0	2,5	3,0	3,7	4,0	5,0
2	0,5	0,7	0,9	1,2	1,5	1,7	2,0
3	3,0	3,3	3,5	4,0	4,2	4,5	5,0
4	2,0	2,5	2,8	3,3	3,5	4,0	4,4
5	1,5	1,8	2,0	2,3	2,6	2,8	3,2
6	3,5	4,0	4,2	4,5	5,0	5,5	6,0
7	4,2	4,5	4,6	5,0	5,5	6,0	7,0
8	2,5	3,0	3,2	3,6	3,9	4,2	4,5

Таблица 2

№ п/п	Параметры нечетких множеств выходной переменной $a_k^i(Y_i)$						
	0	1	2	3	4	5	6
1	5,0	5,6	6,0	6,2	6,5	7,0	7,3
2	2,5	3,0	3,3	3,5	4,0	4,2	4,4
3	7,5	8,0	8,5	9,0	9,2	9,5	10,0
4	2,0	2,5	2,8	3,3	3,5	4,0	4,4
5	5,3	5,5	6,0	6,1	6,3	6,7	6,9
6	8,5	9,0	9,3	9,5	10,0	10,5	11,5
7	10,0	10,2	10,3	10,8	11,5	12,0	12,6
8	6,5	7,0	7,3	7,8	8,0	8,3	8,5

Таблица 3

Параметры	Значения функции принадлежности сечений $k =$						
	0	1	2	3	4	5	6
$\mu_k(A)$	0	0,4	0,8	1,0	0,8	0,4	0
η_k	0,05	0,1	0,2	0,3	0,2	0,1	0,05

Таблица 4

Параметры	Параметры сечений Fuzzy-множества $B - k =$						
	0	1	2	3	4	5	6
$a_k(B)$	-1,349	-0,575	-0,209	0,272	0,649	0,937	1,230
$\mu_k[a_k(B)]$	0	0,4	0,8	1,0	0,8	0,4	0

$$\text{Следовательно, } b = \frac{\sum_{i=1}^8 \sum_{k=0}^6 \eta_k a_k^i(Y_i) a_k^i(X_i)}{\sum_{i=1}^8 \sum_{k=0}^6 \eta_k [a_k^i(X_i)]^2}.$$

$$\sum_{i=1}^8 \sum_{k=0}^6 \eta_k a_k^i(Y_i) a_k^i(X_i) = 225,078,$$

$$\sum_{i=1}^8 \sum_{k=0}^6 \eta_k [a_k^i(X_i)]^2 = 104,309, \quad b = 2,156.$$

Параметры нечеткого множества свободного члена B , т. е. значения $a_k(B)$, $k = 0, 1, \dots, 6$, определяются из условий минимизации следующих функционалов:

$$\varphi_k = \min_{B_k} \sum_{i=1}^8 \eta_k [a_k^i(Y_i) - b a_k^i(X_i) - B_k]^2, \quad k = 0, 1, \dots, 6;$$

и определяются по формулам

$$B_k = \frac{1}{8} \sum_{i=1}^8 [a_k^i(Y_i) - b a_k^i(X_i)], \quad k = 0, 1, \dots, 6.$$

Выполнив необходимые арифметические вычисления, получаем следующие параметры функции принадлежности нечеткого множества свободного члена B (табл. 4).

Так, в результате вычисленных параметров свободного члена $a_k(B)$ было определено $a_6(B) < a_5(B) < a_4(B) < 0$, а $a_0(B) > a_1(B) > a_2(B) > 0$, в результате-рующем представлении этого Fuzzy-множества (см. табл. 4) представлено $a_0(B) = a_6(B)$, $a_1(B) = a_5(B)$, $a_2(B) = a_4(B)$ и соответственно $a_4(B) = a_2(B)$, $a_5(B) = a_1(B)$, $a_6(B) = a_0(B)$.

Список литературы

1. **Расин Дж.** Непараметрическая эконометрика: вводный курс // Квантиль. 2008. № 4. С. 7–56.
2. **Орлов А. И.** Прикладная статистика: учебник. М.: Экзамен, 2006. 672 с.
3. **Дрейпер Н., Смит Г.** Прикладной регрессионный анализ. М.: Издательский дом "Вильямс". 2007, 912 с.
4. **Зак Ю. А.** Методы обработки нечисловой информации в маркетинговых исследованиях. // Маркетинг и маркетинговые исследования, 2013, № 1, С. 20–33.
5. **Зак Ю. А.** Математические модели прогнозирования затрат времени и стоимости перевозки грузов // Логистика сегодня. 2015. № 1. С. 162–172.
6. **Ярушкина Н. Г.** Нечеткие системы: обзор итогов и тенденций развития // Искусственный интеллект и принятие решений, 2008, № 4, С. 26–38.
7. **Алиев Р. А., Церковный А. Э., Мамедова Г. А.** Управление производством при нечеткой исходной информации. М.: Энергоатомиздат, 1991. 240 с.
8. **Зак Ю. А.** Принятие решений в условиях размытых и нечетких данных. М.: URSS, 2013. 352 с.
9. **Ярушкина Н. Г.** Основы теории нечетких и гибридных систем. М.: Финансы и статистика, 2004. 320 с.
10. **Штовба С. Д.** Нечеткая идентификация на основе регрессионных моделей параметрической функции принадлежности // Проблемы управления и информатики. 2006. № 6. С. 1–8.

11. Грицюк В. И. Нечеткий робастный регрессионный анализ для нечетких входных и выходных данных. // Технологический аудит и резервы производства. 2015. № 6. С. 4—8.

12. Домрачев В. Г., Полячук О. М. О построении регрессионной модели при нечетких исходных данных // Автоматика и телемеханика. 2003. № 11. С. 74—83.

13. Tanaka H., Uejima S., Asai K. Linear regression analysis with fuzzy model // IEEE Trans. Systems Man Cybernet. 1982. Vol. 12. N. 6. P. 903—907.

14. Tanaka H., Warada J. Possibilistic linear system and their application to the linear regression model // Fuzzy Sets and Systems. 1988. Vol. 27. P. 275—289.

15. Celmins A. Least Squares Model Fitting to Fuzzy Vector Data // Fuzzy Sets and System. 1987. Vol. 22. P. 260—269.

16. Diamond P. Fuzzy least squares // Information Sciences. 1988. 46. P. 141—157.

17. Diamond P. Least Squares Fitting of Several Fuzzy Variables // Proc. of Second IFSA Congress, Tokyo, 1987. P. 20—25.

18. Aliev R., Fazlollahi B., Vahidov R. Genetic algorithms-based fuzzy regression analysis // Soft Computing. 2002. N. 6. P. 470—475.

19. Papadopoulos B., Sirpi M. Similarities and distances in fuzzy regression modeling // Soft Computing. 2004. N. 8. P. 556—561.

20. D-Urso P., Massan R., Santoro A. Robust fuzzy regression analysis. // Informations Sciences. 2011. Vol. 181. P. 1154—1174.

21. Rousseeuw P. Applying robust regression to insurance. // Insurance: Mathematics and Economics. 1984. Vol. 3. N. 1. P. 67—72.

22. Yang M.-S., Liu H. H. Fuzzy Least Squares Algorithms for Interactive Fuzzy Linear Regression Models // Fuzzy Sets and Systems. 2003. Vol. 135. N. 2. Pp. 305—316.

23. Chang Yun-Hsi O. Fuzzy regression methods — a comparative assessment. // Fuzzy Sets and Systems. 2001. Vol. 119 (2). P. 187—203.

24. Chang Yun-Hsi O. Hybrid fuzzy least-squares regression analysis and its reliability measures // Fuzzy Sets and Systems. 2001. Vol. 119 (2). P. 225—246.

25. Zaychenko Yu. The Fuzzy Group Method of Data Handling and Its Application for Economical Processes forecasting // Scientific Inquiry, June, 2006, Vol. 7, N. 1. P. 83—98.

26. Зак Ю. А. Fuzzy-регрессионные модели в условиях наличия в статистической выборке нечисловой информации // Системні дослідження та інформаційні технології. 2017. № 1. С. 88—96.

Yu. A. Zack, Dokt.-Ing., E-mail: yuriy_zack@hotmail.com, http: optimorum.de
Aachen, Deutschland

About One Algorithm of Fuzzy-Regression Analysis

Methods for solving the Fuzzy regression analysis problem are proposed under conditions when the input, output variable and free term of the regression equation are represented by fuzzy sets of the most general kind, and the regression coefficients are real numbers. The sum of the squares of the average weighted coordinates of the minimum and maximum abscissas of different sections of the fuzzy-set membership functions of the output variable and their estimations by the Fuzzy regression model, and also the coordinates of the centers of gravity of the membership functions of these Fuzzy sets are used as the approximation criteria. Deterministic equivalents of this class of fuzzy regression models, estimates of the adequacy of the constructed regression equations, and a computational scheme of the algorithm for solving the problem are obtained.

Keywords: Fuzzy-regression analysis, fuzzy sets, non-numeric statistics, linguistic and Boolean variables, least-squares method

References

- Rasin Dzh. Neparаметricheskaja ekonomika: vvodnij kurs, *Kvantil*. 2008, no. 4, pp. 7—56.
- Orlov A. I. *Prikladnaja statistika*. Uchebnik, Moscow: Eksamen, 2006, 672 p.
- Drejper N., Smit G. *Prikladnoj regressionnij analiz*. Moscow, Viljams, 2007, 912 P.
- Zack Yu. A. Mitodi obrabotki nechislovoj informazii v marketingovih issledovanijach, *Marketing i marketingovije issledovanja*, 2013, no. 1, pp. 20—33.
- Zack Yu. A. Matematicheskiye modeli prognosirovanija zatrat vremeni i stoimosti perevoski grusov, *Logistika segodnya*, 2015, no. 1, pp. 162—172.
- Yarushkina N. G. Nечetkije sistemi: obsor itogov i tendenzij razvitiya, *Iskusstvennij intellekt i prinyatije reshenij*, 2008, no. 4, pp. 26—38.
- Alijev R. A., Zerkovnij A. E., Mamedova G. A. *Upravlenije proizvod-stvom pri nechetkoj ischodnoj informazii*, Moscow, Energoatomizdat, 1991, 240 p.
- Zack Yu. A. *Prinyatije reshenij v uslovojach razmitich i nechetkikh dannich*, Moscow, URSS, 2013, 352 p.
- Yarushkina N. G. *Osnovi teorii nechetkikh i gibridnich sistem*, Moscow, Finans i statika, 2004, 320 p.
- Shtovba S. D. Nечetkaja identifikacija na osnove regressionnih modelej parametricheskoi funkzii prinadlezhnosti, *Problemi upravlenija i informatiki*, 2006, no. 6, pp. 1—8.
- Grizyuk V. I. Nечetkij robastij regressionnij analiz dlya nechetkikh vvodnich i vychodnich dannich, *Technologicheskij audit i rezervy proizvodstva*, 2015, no. 6, pp. 4—8.
- Domrachev V. G., Poleschuk O. M. O postroenii regressionnoj modeli pri nechetkikh ischodnich dannich, *Avtomatika i telemechanika*, 2003, no. 11, pp. 74—83.
- Tanaka H., Uejima S., Asai K. Linear regression analysis with fuzzy model, *IEEE Trans. Systems Man Cybernet.*, 1982, vol. 12, no. 6, pp. 903—907.
- Tanaka H., Warada J. Possibilistic linear system and their application to the linear regression model, *Fuzzy Sets and Systems*, 1988, vol. 27, pp. 275—289.
- Celmins A. Least Squares Model Fitting to Fuzzy Vector Data, *Fuzzy Sets and System*, 1987, vol. 22, pp. 260—269.
- Diamond P. Fuzzy least squares, *Information Sciences*, 1988, vol. 46, pp. 141—157.
- Diamond P. Least Squares Fitting of Several Fuzzy Variables // *Proc. of Second IFSA Congress, Tokyo*, 1987, pp. 20—25.
- Aliev R., Fazlollahi B., Vahidov R. Genetic algorithms-based fuzzy regression analysis, *Soft Computing*, 2002, no. 6, pp. 470—475.
- Papadopoulos B., Sirpi M. Similarities and distances in fuzzy regression modeling, *Soft Computing*, 2004, no. 8, pp. 556—561.
- D-Urso P., Massan R., Santoro A. Robust fuzzy regression analysis, *Informations Sciences*, 2011, vol. 181, pp. 1154—1174.
- Rousseeuw P. Applying robust regression to insurance, *Insurance: Mathematics and Economics*, 1984, vol. 3, no. 1, pp. 67—72.
- Yang M.-S., Liu H. H. Fuzzy Least Squares Algorithms for Interactive Fuzzy Linear Regression Models, *Fuzzy Sets and Systems*, 2003, vol. 135, no. 2, pp. 305—316.
- Chang Yun-Hsi O. Fuzzy regression methods — a comparative assessment, *Fuzzy Sets and Systems*, 2001, vol. 119 (2), pp. 187—203.
- Chang Yun-Hsi O. Hybrid fuzzy least-squares regression analysis and its reliability measures, *Fuzzy Sets and Systems*, 2001, vol. 119 (2), pp. 225—246.
- Zaychenko Yu. The Fuzzy Group Method of Data Handling and Its Application for Economical Processes forecasting, *Scientific Inquiry*, June, 2006, vol. 7, no. 1, pp. 83—98.
- Zack Yu. A. Fuzzy-regressionije modeli v uslovijach nalichija v statisticheskoj viberke nechislovoj informazii, *Sistemni doslidzhennija ta informazijni tehnologii*, 2017, no. 1, pp. 88—96.

А. Л. Стемповский, академик РАН, директор, e-mail: ippm@ippm.ru,

Н. Н. Левченко, канд. техн. наук, зав. отделом, e-mail: nick@ippm.ru,

А. С. Окунев, канд. техн. наук, вед. науч. сотр., e-mail: oku@ippm.ru,

А. В. Климов, ст. науч. сотр., e-mail: klimov@ippm.ru,

Д. Н. Змеев, науч. сотр., e-mail: zmejevdn@ippm.ru,

Федеральное государственное бюджетное учреждение науки

Институт проблем проектирования в микроэлектронике Российской академии наук, Москва, 124365

Программирование задачи "молекулярная динамика" в потоковой модели вычислений

Во многих НРС-задачах масштабирование сдерживается излишним применением барьеров, которые в условиях неравномерной и изменяющейся загрузки приводят к дополнительным простоям. Предлагаемый в работе подход к решению этой проблемы опирается на парадигму вычислений с управлением потоком данных. Этот подход описывается на примере алгоритма решения задачи молекулярной динамики. В этой задаче вместо барьеров вводится подсчет находящейся поблизости частиц, который происходит асинхронно и параллельно с основной работой. Синхронизация опирается только на локальные взаимодействия между вычислительными ядрами, обрабатывающими близкие области.

Ключевые слова: молекулярная динамика, масштабирование, барьерная синхронизация, параллельные вычисления, потоковая модель вычисления

Введение

Масштабирование многих актуальных задач, выполняемых на кластерных суперкомпьютерах, сдерживается излишним применением барьеров, которые в условиях неравномерной и изменяющейся загрузки приводят к дополнительным простоям, не говоря о затратах на сам барьер. Выходом из создавшегося положения является переход к алгоритмам, опирающимся исключительно на локальные попарные взаимодействия между соседями. При традиционном подходе к программированию (SPMD, MPI, CSP, CUDA) проще использовать барьерную синхронизацию. Предлагаемый в работе подход опирается на потоковую модель вычислений с динамически формируемым контекстом, в которой синхронизация выполняется по готовности данных.

Масштабированию задач класса молекулярной динамики (МД) на суперкомпьютерах кластерного типа обычно мешают глобальные барьеры, используемые при пересчетах списков соседних частиц. Поэтому была поставлена задача написания полностью асинхронного и эффективного алгоритма, опирающегося исключительно на локальные взаимодействия между вычислительными ядрами, ответственными за соседние области. Аналогичные цели ставились и другими авторами, при том что они использовали традиционные подходы к программированию на кластерах: MPI, специальная библиотека RDMA [1], Java RMI [2].

В этой работе демонстрируется использование для указанных целей потоковой модели вычислений, основанной на односторонних сообщениях — токенах. Программа может быть представлена в форме графа, где типы узлов изображаются блоками,

а пути токенов — направленными дугами в виде стрелок. Эта форма более удобна и для разработки, и для понимания, поэтому в ней и будет представлен рассматриваемый алгоритм. Причем это не просто иллюстрация, а сама программа, которая может быть исполнена в соответствии с семантикой.

Потоковая модель вычислений реализована в параллельной потоковой вычислительной системе (ППВС) "Буран" [3] и хорошо подходит для создания асинхронных алгоритмов. Операция отправки токена в ней играет ту же фундаментальную роль, что и операции чтения и записи в память в обычных архитектурах. Помимо того что потоковая модель эффективна для асинхронных алгоритмов, она может претендовать на то, чтобы служить языком первичной разработки алгоритмов. Эти алгоритмы могут быть неэффективными изначально, но они будут допускать последующую систематическую (возможно, и автоматическую) трансформацию в эффективные программы в различных существующих моделях параллельного программирования: MPI, OpenMP, CUDA и др.

1. Алгоритм моделирования молекулярной динамики

Моделирование молекулярной динамики заключается в расчете на основе законов Ньютона новых координат и скоростей всех частиц (атомов) на каждом временном шаге. Предполагается, что на каждой итерации надо вычислить, зная координаты всех частиц (N), суммарную силу, действующую на каждую частицу со стороны всех остальных частиц, и затем пересчитать все скорости и координаты. Однако потенциал взаимодействия пары очень быстро убывает с расстоянием. Поэтому прежде

всего надо воспользоваться классическим приемом — игнорировать все пары, расстояние между которыми превышает радиус отсечения R_c . Это также улучшает локальность и делает предпочтительным распараллеливание путем пространственной декомпозиции и геометрического распределения пространства по процессорам.

Для организации взаимодействия между частицами из соседних процессоров могут применяться различные приемы. Каноническим считается метод, в котором каждый кубоид для расчета своих частиц получает информацию о положении всех частиц из всех 26 соседних кубоидов, находящихся в пределах оболочки радиуса R_c от границы данного кубоида (предполагая, что размер кубоида превышает R_c по всем размерностям). Недостатки метода — большой объем передаваемой информации и необходимость дублировать вычисления для пар частиц, лежащих в разных кубоидах. Достоинством является то, что в каждой итерации нужна только одна фаза обмена.

Для реализации был выбран метод "средней точки", предложенный в работе [1]. Это один из методов так называемой нейтральной территории, с помощью которого вычисляется взаимодействие между двумя частицами в кубоиде, возможно, не содержащем ни одну из частиц данной пары. В методе "средней точки" вычисление взаимодействия происходит в вычислительном ядре, кубоид которого содержит середину отрезка между этими двумя частицами. Этот метод позволяет ограничиться импортом в каждый кубоид частиц лишь из его оболочки радиуса $R_c/2$ (а не R_c). Предполагается, что $1 < R_c/b < 2 - 2\varepsilon$, где b — длина стороны кубоида и $\varepsilon > 0$. Таким образом, область импорта ограничена 26 соседними кубоидами и изолирована от других кубоидов слоем, толщина которого не менее ε . Недостатком данного метода является необходимость второй фазы обмена для передачи обратно в "домашний" кубоид суммарных сил взаимодействия.

На рис. 1 показана область импорта для одного кубоида. Соответственно, в кубоиде-экспортере образуются зоны экспорта, свои для каждого соседа.

Каждый цикл решения задачи (одна итерация) состоит из следующих фаз.

А. В каждый кубоид передаются копии всех частиц из его области импорта, а оригинал частицы помещается в кубоид, определяемый ее координатами.

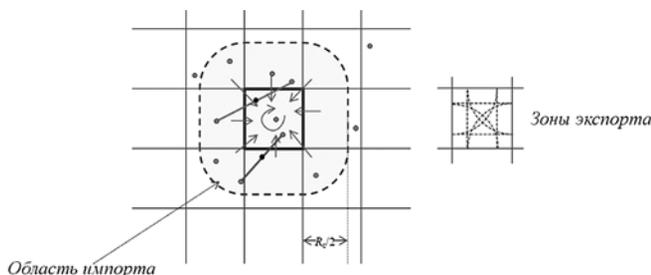


Рис. 1. Метод "средней точки": область импорта и зоны экспорта

В. Силы взаимодействия вычисляются для каждой пары импортированных или собственных частиц, которые затем суммируются по каждой частице отдельно внутри каждого кубоида до получения парциальных сил.

С. Возвращение парциальных сил в "домашний" кубоид для каждой (импортированной) частицы.

Д. Суммирование парциальных сил и интегрирование уравнения движения частиц.

При выходе частицы за границу кубоида происходит "миграция" — смена "домашнего" кубоида частицы. Эта фаза объединяется с фазой А: если в фазе D частица пересекает границу, тогда в следующей фазе А в ее экспортируемых копиях указывается ее новый кубоид (отличный от того, из которого эта копия фактически пришла), а ее оригинал переходит, соответственно, в этот новый кубоид в качестве его собственной частицы.

2. Модель вычисления ППВС "Буран"

В потоковой модели вычислений с динамически формируемым контекстом [4, 5] (рис. 2, см. вторую сторону обложки), реализуемой в ППВС "Буран", вычисления разделены на относительно небольшие фрагменты, называемые программными узлами (или просто узлами), каждый из которых активируется по готовности своих входных данных. Программа DFL, написанная на параллельном языке ППВС "Буран", состоит из набора таких программных узлов.

Описание программного узла состоит из заголовка и набора операторов. Заголовок включает в себя ключевое слово **node**, имя узла, список входов и контекста. Пример:

node M1(a, b) { i, j }.

Контекст представляет собой список имен полей (параметров), заключенных в фигурные скобки {}. Значения этих полей определяют положение экземпляра конкретного узла в виртуальном адресном пространстве. Экземпляр узла называется виртуальным, поскольку реально он создается только при приходе операнда на один из входов узла. Экземпляр узла активируется, когда на все входы этого экземпляра поступают требуемые операнды. В ходе выполнения программы узла могут быть созданы новые токены с данными, которые отправляются на другие экземпляры узлов. При формировании токена программа узла может использовать только те данные, которые пришли на входы узла. В состав токена обычно входит значение (элемент данных), имя целевого узла, имя входа, значение контекста целевого узла и ряд служебных атрибутов. Значение может быть составным (структурой, вектором). Имя узла и значение контекста образуют ключ, или адрес, который однозначно идентифицирует конкретный экземпляр узла. Некоторые поля контекста в токене могут быть заданы символом "*": такой токен посылается на многие экземпляры узлов. Однако активация может произойти только с

конкретным контекстом, определяемым всей совокупностью входных токенов.

Например, $x \rightarrow \mathbf{M1}.a \{i + 1, *\}$ или $y \rightarrow \mathbf{M1}.b \{*, j - 1\}$. Здесь x, y — элементы данных, $\mathbf{M1}$ — имя узла, a, b — имена входов, в фигурных скобках — список значений контекста.

Обычно токены, участвующие в активации программных узлов, удаляются после генерации экземпляра узла. Однако если указать кратность (между посылаемым данным и стрелкой через #), то токен будет сохраняться на входе до исчерпания кратности, а сама кратность при каждой активации будет уменьшаться на 1. Кратность может быть бесконечной (##).

Порядок прихода токенов на узел может отличаться от порядка их отправки. Каждый виртуальный узел привязан к вычислительному ядру в соответствии со своим контекстом. Функция распределения, задаваемая отдельно от основной DFL-программы, преобразует контекст в номер вычислительного ядра. По умолчанию программист может положиться на стандартную функцию распределения, но это может привести к неоправданно высоким коммуникационным издержкам. Для их минимизации важно найти хорошую функцию распределения для каждой конкретной программы.

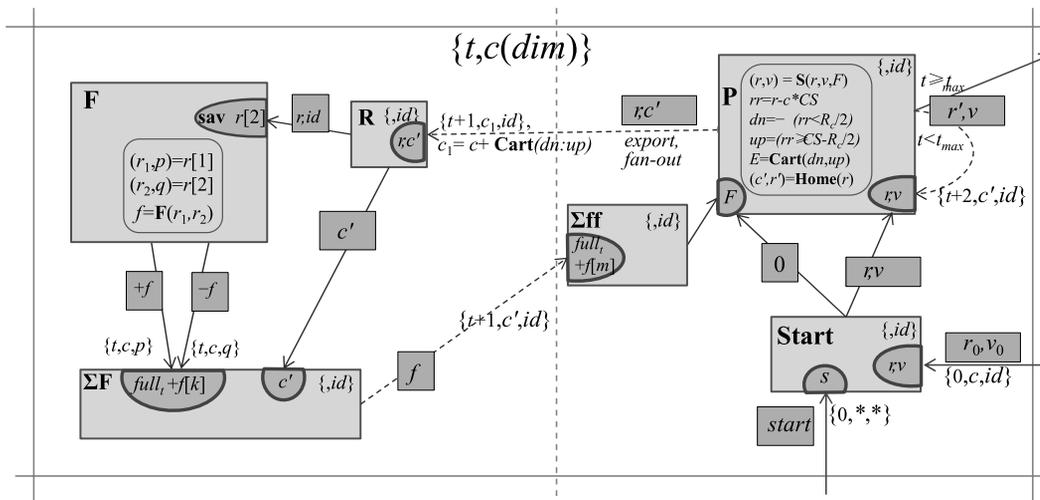
3. Реализация задачи молекулярной динамики на ППВС "Буря"

Модельное пространство разбито на кубоиды размера **CubSize** (типа **Vec**, с компонентами по из-

мерениям). Всякое вычислительное действие привязано к некоторому кубоиду и некоторому шагу итераций. Поэтому в контексте всегда присутствуют номер итерации it и координаты кубоида c .

Исходя из общего размера области моделирования **BoxSize** и размера кубоида перед началом работы определяется число кубоидов **NCubes**. На рис. 3 потоковый алгоритм представлен в графической форме. Каждый "большой" блок изображает программный узел. Атрибуты (поля) контекста показаны в фигурных скобках. Чтобы не загромождать схему, общие поля контекста разных узлов $\{t, c\}$ вынесены наверх, а в каждом узле указываются только возможные дополнительные поля, если они есть, например $\{, id\}$. Здесь t — номер цикла, $c = (i, j, k)$ — трехмерные координаты кубоида, id — уникальный код частицы.

"Малые" блоки на стрелках изображают передаваемые данные. На концах стрелок указан контекст целевого узла, в начале стрелок — условие отправки. Усеченный диск у границы блока обозначает вход (порт). В него входят стрелки — это операторы отправки токенов. Вход помечен именем и другими атрибутами. Квадратные скобки указывают, что это собирающий вход, т. е. на него должно прийти несколько токенов, и в скобках задано их число. Это либо константа, либо символическое имя, значение которого требует своего определения. Если перед именем стоит знак (+), то значения, приходящие на этот вход, будут просуммированы и полученная сумма станет значением входа.



Обозначения:

$+n[m]$ — суммирующий вход, $full$ — число слагаемых заранее не известно и определяется условием, что должны пройти все (например, по тишине). Здесь индекс t — номер раунда определения тишины.
 c' — координаты кубоида, в котором будет главная копия частицы (home)
 $dn(up)$ — вектор из компонент -1 или 0 (соответственно, 0 или 1) — нижние (верхние) границы 3D-интервала направлений экспорта
 c_1 — множество направлений экспорта $c + dn \leq c_1 \leq c + up$
 $Cart$ — операция взятия декартова произведения - вектора интервалов, например, $Cart((-1,0,-1), (1,1,0))$ имеет 12 элементов (параллелепипед $3 \times 2 \times 2$)

Узлы:

F — вычисляет силу взаимодействия пары
ΣF — суммирует силы в рамках кубоида
Σff — суммирует парциальные силы
P — вычисляет новые координаты и скорости (r, v), посылая ее соседям c_1 и на новое место c'
R — подача координат на **F** и обратного адреса на **ΣF**
Start — инициация процесса при $t=0$

Рис. 3. Схема базового алгоритма молекулярной динамики

Узел **F** обеспечивает взаимодействие частиц. Его единственный вход r является собирающим и описан как **sav** $r[2]$. Описатель **sav** указывает, что использованные для активации токены должны удерживаться в памяти и, таким образом, могут дальше участвовать в других активациях. Для активации требуется два токена. Существует общее правило, что каждая комбинация токенов может породить не более одной активации. Поэтому, когда на этот вход придет второй токен, он породит активацию с первым и оба останутся на входе. Когда придет третий, он породит активацию с первым и тоже останется. Это приведет к тому, что он породит активацию еще и со вторым. Четвертый породит активацию с первым, вторым и третьим и так далее. То есть будет порождаться по одной активации для каждой пары присланных токенов.

Каждый токен, поступающий на вход **F**, r , соответствует одной импортированной частице (включая собственные частицы кубоида) и содержит ее координаты и код: (r, id) . Каждая активация рассчитывает силу взаимодействия $f = \mathbf{F}(r_1, r_2)$ для одной пары $((r_1, p), (r_2, q))$ и посылает ее с разными знаками на узлы $\Sigma\mathbf{F}$, аккумулирующие силы для частиц p и q соответственно. Это парциальные (частичные) силы, накопленные внутри одного кубоида. Внутри функции **F** также проверяется условие средней точки — середина отрезка лежит в данном кубоиде, и если оно не выполнено, а также если расстояние между частицами больше R_c , то сила взаимодействия считается нулевой.

Узел **P** активируется для каждой частицы в каждой итерации. На вход **P**, F подается полная сумма сил, действующих на частицу, исходя из которой пересчитываются новые скорости и координаты.

Далее новое положение r используется для определения множества кубоидов c_1 (включая себя, c), в которые частицу надо экспортировать для взаимодействий, и кубоида c' , в котором частица должна будет теперь находиться.

Затем происходит рассылка информации о частице во все кубоиды из множества c_1 . узел **R** принимает токен с информацией об импортировании частицы и направляет ее положение r на узел **F** (вместе с ее id), а обратный адрес c' — на одноименный вход узла $\Sigma\mathbf{F}$.

Узел $\Sigma\mathbf{F}$ принимает силы от узла **F** на вход f (описанный как суммирующий: $+ f[k]$) и координаты владельца частицы на вход c' . Поскольку в контексте имеется id частицы, для каждой частицы будет порожден отдельный экземпляр узла, у которого на входе f будет накапливаться сумма всех сил, действующих на эту частицу от других частиц, импортированных этим кубоидом. Накопленная сумма должна быть послана на вход f узла $\Sigma\mathbf{ff}$ с контекстом $\{t + 1, c', id\}$. Сюда же придут с тем же контекстом токены из других кубоидов, куда была экспортирована частица id . Там они будут просуммированы, а сумма отправлена на вход F узла **P**.

На схеме на рис. 3 ряд стрелок изображен штриховой линией — это сделано в случаях, когда посылаемый токен может уходить за пределы ядра в другое ядро. Это такие стрелки, на которых может измениться значение координат кубоида c (поскольку только они являются аргументами функции распределения). Таких стрелок на схеме три, они соответствуют экспорту-импорту, возврату парциальных сил и "миграции". Соответственно, сплошные стрелки — это те, на которых c не изменяется. Эти связи локальные, и они могут исполняться более эффективно. Например, левая группа связей объединяет узлы **R**, **F** и $\Sigma\mathbf{F}$. Узел **F** не имеет в контексте поля id , и на него приходят токены от многих узлов **R** с разными id . Сам узел **R** одноходовой, поэтому в нем нет поиска по id . В узле $\Sigma\mathbf{F}$ поиск по id формально нужен, но его можно устранить. Описанная схема будет рабочей, если узлы $\Sigma\mathbf{F}$ и $\Sigma\mathbf{ff}$ "будут знать", когда можно считать суммирование оконченным.

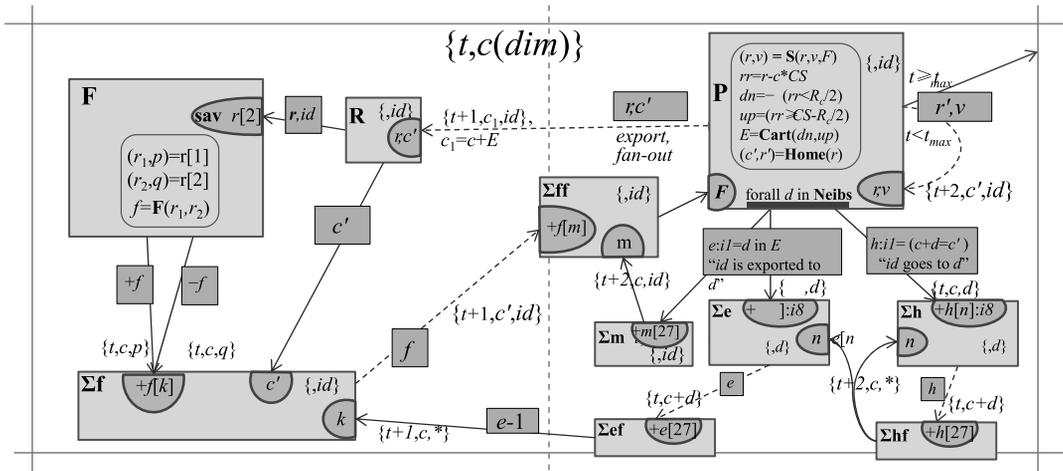
4. Извлечение алгоритма с локальной синхронизацией

Из рассмотренного варианта может быть выведен алгоритм, основанный на локальных критериях. Это можно делать как вручную, так и с помощью преобразователя, действующего автоматически или с подсказками человека. Ключевая проблема состоит в определении числа слагаемых на входах узлов $\Sigma\mathbf{F}$ и $\Sigma\mathbf{ff}$.

Рассмотрим узел $\Sigma\mathbf{F}$. Число токенов для каждого id — это фактически число сопоставлений, которое совершено в данном кубоиде c с участием частицы id . Это обеспечивается следующими условиями:

- все id на входе узла **F** различны;
- id со входа узла **F** без изменения переходит в контекст выходного токена;
- каждая пара входных токенов в узле **F** порождает ровно по одному токену со слагаемым для каждого из двух участвующих id .

На вход узла **F** подаются токены из узлов **R**, имеющих дополнительное поле id в контексте. Надо определить число активаций таких узлов (для данных t и c). А это число токенов, посланных из узлов **P** с полушага $t - 1$ со всех возможных кубоидов. Но: для кубоида c посылка из **P** в **R** возможна только в кубоиды c_1 , у которых каждая координата отличается от c максимум на 1 (в любую сторону). Соответственно, то же верно и для приемника: узлы $\mathbf{R}\{t, c_1, id\}$ активируются токенами из узлов $\mathbf{P}\{t - 1, c, id\}$, в которых c отличается от c_1 не более чем на 1 в каждой координате. Число таких различных c заранее известно, это $3^3 = 27$ (если бы не периодическое граничное условие, то число таких c могло быть разным для разных c_1 , но тоже заранее известно и не более 27). Поэтому можно организовать подсчет числа токенов, посланных на все $\mathbf{R}\{t, c, *id\}$ (для всевозможных $*id$) как сумму 27 слагаемых, если будем знать $\mathbf{N}_{P \rightarrow R}(c, c_1)$ — число токенов из **P** в **R** из каждого кубоида c в каждый



Обозначения:

- $+n[m]$ – суммирующий вход, m – число слагаемых
- t – номер (полу)итерации
- c' – координаты кубоида, в который идет главная копия частицы (home)
- $dn(up)$ – вектор из компонент -1 или 0 (соответственно, 0 или 1) – нижние (верхние) границы 3D-интервала направлений экспорта
- E – множество направлений экспорта, $dn \leq E \leq up$
- $c_1 = c + E$ – множество целевых кубоидов экспорта, $c + dn \leq c_1 \leq c + up$
- Cart** – операция взятия декартова произведения векторного интервала
- Neibs** = **Cart**((-1,-1,-1), (1,1,1)) имеет 27 элементов (параллелепипед $3 \times 3 \times 3$)

Узлы:

- F** – вычисляет силу взаимодействия пары
- Σf** – суммирует попарные силы в рамках кубоида
- Σff** – суммирует парциальные силы
- P** – вычисляет новое состояние частицы (r,v) , посылая ее на новое место c' и соседям c_1
- R** – подает координат на **F** и обратного адреса на **Σf**
- Σe, Σh** – подсчет экспорта и иммиграции (по соседям)
- Σef, Σhf** – суммирование импорта и иммиграции
- Σm** – подсчет направлений экспорта частицы

Рис. 4. Схема МД с подсчетом

кубоид c_1 . Эти числа надо будет послать из кубоидов c на суммирующий узел **ΣI** в кубоиде c_1 , который ожидает ровно 27 слагаемых.

Для вычисления чисел $N_{P \rightarrow R}(c, c_1)$ в кубоиде c применим стандартный прием – подсчет подмножества. Для начального момента $t = 0$ число n задано исходными данными как начальное число частиц в каждом кубоиде. Но возможна "миграция", поэтому в другие моменты оно будет отличаться.

Рассмотрим цепочку посылок: **P** -> **R** -> **ΣF** -> **Σff**. Нетрудно увидеть, что каждая следующая посылка в этой цепочке однозначно определяется предыдущей. Поэтому количество слагаемых на входе **Σff**, $f(t+1, c', id)$ будет в точности равно количеству токенов из **P** $\{t, c, id\}$ (для некоторого c) в **R** (t, c', id) . Его же подсчет сводится к определению числа элементов в множестве c_1 .

Эти рассуждения приводят к граф-программе, изображенной на рис. 4.

Узлы **Σe** $\{t, c, d\}$ и **Σh** $\{t, c, d\}$ принимают и суммируют (по частицам) логические условия (как 0 или 1), соответственно: "частица id экспортируется в направлении d " и "частица id идет в кубоид в направлении d ". Далее суммы передаются на окончательное суммирование в узлы **Σef** $\{t, c + d\}$ и **Σhf** $\{t, c + d\}$, куда эти суммы сходятся с 27 разных направлений. Результаты, соответственно, – общее число импортированных частиц e и общее число пришедших частиц h . Число $(e - 1)$ подается как число взаимодействий каждой частицы, а число h – как новое число частиц на следующем шаге.

Условие "частица id экспортируется в направлении d " также подается на узел **Σm** $\{t, c, id\}$, где для

каждой частицы формируются суммы по направлениям – количество экспортов данной частицы.

Эта схема описывает версию алгоритма, в котором нет глобальных синхронизаций. В каждом кубоиде следующая фаза может выполняться сразу после того, как закончена предыдущая фаза на этом и всех соседних кубоидах. Более того, фаза взаимодействия для каждой импортированной частицы начинает выполняться сразу при ее приходе, не дожидаясь прихода всех импортируемых частиц. Новая частица взаимодействует со всеми старыми и сохраняется, т. е. присоединяется к старым. Фаза взаимодействия в целом будет считаться законченной только после того, как придут не только все частицы, но и значение их количества, которое формируется по завершении фаз интегрирования во всех соседних кубоидах.

5. Пути повышения эффективности решения задачи МД

Все реализации МД так или иначе стремятся повысить свою эффективность, используя разные приемы оптимизаций. Реализация каждого приема может осуществляться на одном из нескольких уровней:

- на уровне исходного кода, когда требуется изменить сам алгоритм или вычислительный метод;
- на уровне исполняющей системы, когда библиотечные функции могут быть улучшены без участия пользователя;
- на смешанном уровне, когда программист, не меняя код, вставляет подсказки компилятору или исполнителю.

Многие из рассматриваемых приемов не являются специфичными для МД и могут применяться в широком классе приложений. МД — это достаточно сложная задача, охватывающая многие вычислительные аспекты.

Мелкозернистый параллелизм. В модели вычислений ППВС гранулой параллелизма является узел. В нашем решении МД узел выполняет расчет одной частицы или одного взаимодействия пары частиц. Взаимодействие пар частиц организуется через механизм взаимодействия токенов. Каждое взаимодействие порождает активацию узла, в результате которой и рассчитывается взаимодействие данной пары частиц. При этом семантикой модели вычислений предполагается, что формирование пар (активация узлов) проводится атомарно, и это очень быстрая операция, а дальнейшее вычисление программ активированных узлов может выполняться параллельно, если имеется достаточное число исполнителей. Поэтому в решении имеется мелкозернистый параллелизм, гранулами которого являются отдельные взаимодействия пар частиц.

Накладные расходы. Всякое распараллеливание в той или иной степени порождает накладные расходы. Использование метода средней точки тоже вносит накладные расходы. К ним относятся: затраты на определение направлений экспорта; затраты на определение направлений "миграции"; каждая пара обрабатывается не в одном, а в нескольких кубоидах, в каждом случае выполняется проверка вхождения средней точки в кубоид; затраты на организацию передачи информации (в другой кубоид) по экспорту, "миграции" и обратно по парциальным силам; затраты на подсчет и суммирование количеств передаваемых частиц и сил.

"Миграция" без дополнительной фазы передачи. В работе [1] описывается эффективная параллельная реализация МД Desmond, в которой "миграция" вынесена в отдельную фазу, тем самым вызывая дополнительную задержку. В этой работе сказано, что реализация без дополнительной фазы возможна, но ценой более сложного программирования. Эта дополнительная фаза была использована для других типов коммуникаций, которые в упрощенной версии отсутствуют. В реализации на DFL соответствующее усложнение является незначительным, и оно было с легкостью осуществлено.

Совмещение передач и вычислений. Важное достоинство предлагаемого решения в том, что обмены между процессорными ядрами происходят на фоне вычислений. Единственное место, где возникает "барьер", — начало фазы интегрирования, которое требует окончания фаз взаимодействия всех соседей. Это принципиально, поскольку пока не закончена фаза взаимодействий хотя бы одного соседа, нет информации о полной силе ни для одной частицы. В случае, когда радиус отсечения R_c меньше всех размеров кубоида, есть небольшая область внутри кубоида, частицы которой не экспортируются, по-

этому могут не дожидаться информации о силах от соседей для очередной фазы интегрирования.

Изменение размера узла. Это один из способов регулирования степени масштабирования программы МД на ППВС "Буран". Чем будет больше независимых "гранул", готовых к исполнению, тем больше вычислительных ядер можно "загрузить" работой и тем самым повысить эффективность решения задачи. Верно и обратное: при необходимости уменьшить уровень параллелизма решения задачи требуется увеличить размер программного узла, сократив общее число узлов. При уменьшении размера узла требуется учитывать "побочные эффекты", которые заключаются в следующем:

- увеличивается число токенов, связанное с необходимостью осуществлять взаимосвязь между вновь создаваемыми программными узлами;
- увеличивается число передач по коммуникационной сети;
- увеличивается количество сопоставлений в ассоциативной памяти.

Использование функций распределения. Итоговая производительность системы существенно зависит от выбора способа распределения виртуальных узлов по ядрам для каждой конкретной задачи. Распределение виртуальных узлов осуществляется посредством задания функции распределения (присоединяемой к DFL-программе), аргументом которой является ключ токена, а значением — номер вычислительного ядра. Эта функция должна вычисляться для каждого посылаемого токена при его создании. Ее значение определяет адрес (или номер) ядра, в который токен передается посредством коммуникационной сети. Эта функция должна быть "вычислительно дешевой", чтобы не снижалась производительность всей системы.

Изменение архитектуры ППВС. Архитектура создаваемой вычислительной системы является универсальной. Однако разработчику программного и аппаратного обеспечения (на данном этапе проектирования) достаточно легко удается "специализировать" вычисления, ориентируясь на ту или иную задачу. Так и при исследовании задачи МД было предложено несколько архитектурных решений, увеличивающих производительность системы. В частности, был реализован в системе команд новый тип токена (двойной групповой "индифферентный"), который позволяет резко сократить общее число токенов задачи и, соответственно, в несколько раз повысить эффективность использования ППВС при минимальных затратах оборудования.

6. Результаты экспериментов

Разработанный алгоритм был реализован на параллельном языке ППВС "Буран". Для подтверждения правильности результатов теоретических исследований были проведены эксперименты на поведенческой блочно-регистрационной модели [6], а также

на эмуляторе [7] вычислительной системы, работающем на суперкомпьютере "Ломоносов" [8].

Эксперименты на поведенческой блочно-регистрационной модели. На поведенческой блочно-регистрационной модели для прохождения задачи МД потребовалось написать программу с использованием расширения параллельного языка DFL — HPL. Программа МД была пропущена с помощью пакетного режима с использованием универсального клиента на основе общего шаблона. В шаблоне задается конфигурация вычислительной системы и проводится настройка различных параметров, в частности числа вычислительных ядер, типов хэш-функции и др.

Ниже приведены результаты (рис. 5, 6, см. вторую сторону обложки) прохождения задачи МД размерностью 256 частиц, 10 итераций на 64 вычислительных ядрах для различных функций распределения [9].

На рис. 5 (см. вторую сторону обложки) приведена загрузка ассоциативной памяти ключей (АПК) процессора сопоставления (ПС) при использовании функции локализации вычислений Fld. Максимальное значение токенов на одно вычислительное ядро составляет 3000 токенов (ось ординат), время выполнения задачи около 3 млн тактов (ось абсцисс). Из 64 ядер функционируют не все вычислительные ядра, что "говорит" о плохой функции распределения для данной задачи.

На рис. 6 (см. вторую сторону обложки) приведена загрузка ассоциативной памяти ключей (ось ординат) процессора сопоставления при использовании функции локализации вычислений — Zip. Максимальное число токенов на одну АПК вычислительного ядра составляет 1150 токенов, время выполнения задачи порядка 450 тысяч тактов (ось абсцисс). Из 64 ядер загружены работой практически все вычислительные ядра. Эти данные свидетельствуют о функции Zip как о наилучшей функции распределения вычислений для решения данной задачи.

На рис. 7 приведен сводный график прохождения задачи МД с функциями локализации вычислений Fld, Zip и Std на различном числе вычислительных ядер. Приведенный график подтверждает выводы о наибольшей эффективности применения функции локализации вычислений Zip. Виден и недостаточный уровень размерности задачи, что проявляется на 128 вычислительных ядрах. У функции локализации вычислений Std динамика немного отличается из-за худшего использования исполнительных устройств вычислительной системы.

Эксперименты на эмуляторе ППВС на кластере "Ломоносов". На графике, приведенном на рис. 8, отображена зависимость времени сопоставления токенов от числа вычислительных ядер, полученная на эмуляторе вычислительной системы (суперкомпьютер "Ломоносов"). Эксперименты проводились для задачи размерностью 64К частиц, 100 итераций, 128 кубоидов. На графике отображается практически линейное уменьшение времени сопоставления при увеличении числа вычислительных ядер.

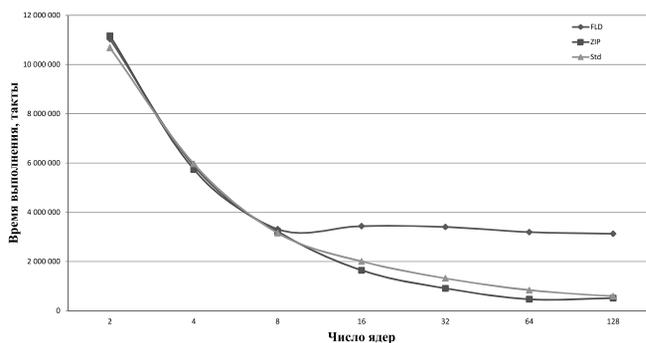


Рис. 7. Прохождение задачи МД (на различных конфигурациях) с использованием разных функций локализации

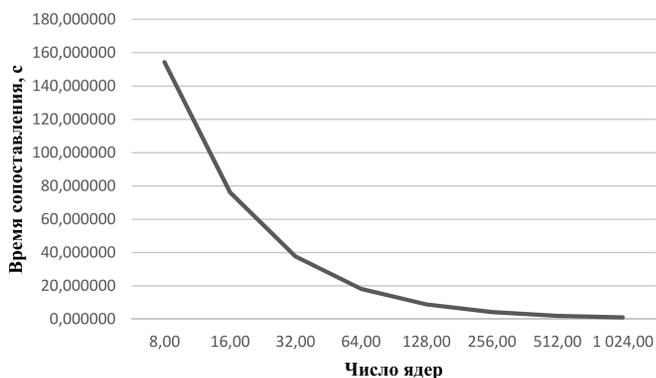


Рис. 8. Зависимость времени сопоставления токенов от числа ядер на суперкомпьютере "Ломоносов"

Практически линейное уменьшение времени сопоставления токенов при увеличении числа вычислительных ядер, на которых работает эмулятор, возможно только при равномерном распределении токенов по вычислительным ядрам. Это, в свою очередь, свидетельствует о том, что выбранная функция распределения Zip является наиболее эффективной для задач класса МД.

Зависимость времени прохождения задачи от числа ядер на суперкомпьютере "Ломоносов" на рис. 9 показана для размерностей: 1К, 100 итераций.

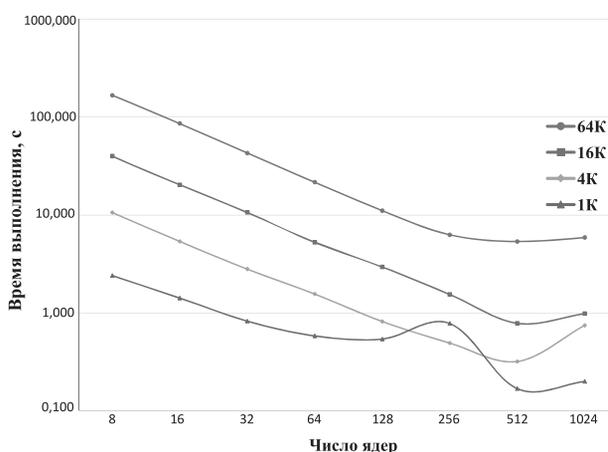


Рис. 9. Зависимость времени прохождения задачи МД от числа ядер на суперкомпьютере "Ломоносов" (логарифмическая шкала)

ций, 16 кубоидов; 4К, 100 итераций, 32 кубоида; 16К, 100 итераций, 64 кубоида; 64К, 100 итераций, 128 кубоида.

Эксперименты проводили на различных конфигурациях системы — от 8 до 1024 вычислительных ядер. На первых трех графиках видно, что с ростом числа ядер происходит практически линейное уменьшение времени выполнения задачи на эмуляторе. Это означает, что размерность самой задачи и параллелизм, заложенный в реализованном алгоритме, обеспечивают эффективное использование имеющихся вычислительных ресурсов. Время выполнения задачи начинает увеличиваться при большом числе ядер вследствие недозагруженности работы исполнительных устройств (недостаточная размерность задачи). Оптимальное число обрабатываемых частиц на одно ядро, при котором масштабирование не падает, составляет для ППВС порядка 10^3 частиц.

Следует отметить, что часть вычислительной нагрузки при работе эмулятора составляют накладные расходы на организацию работы самого эмулятора.

Заключение

Данная статья описывает возможности эффективного применения ППВС "Буран" для задач молекулярной динамики. На параллельном языке этой вычислительной системы алгоритм выражается естественно и адекватно: с одной стороны, в этой форме полностью определяется вычислительная схема алгоритма, а с другой стороны, не накладываются какие-либо ограничения на способ организации и порядок этих вычислений.

Исследования показали, что возможно написать простой базовый вариант алгоритма, из которого можно вывести более эффективный, полностью асинхронный алгоритм, основанный на подсчетах передаваемых частиц. Предлагается строить алгоритмы, опирающиеся исключительно на локальные попарные взаимодействия между соседями. При традиционном программировании для высокопроизводительных вычислительных систем это сделать трудно, в основном используют глобальные барьеры, которые в условиях неравномерной загрузки приводят к простоям.

В предлагаемом подходе к решению задачи МД используются только локальные коммуникации между соседними ядрами/кубоидами. Более того, разные частицы обрабатываются, в основном, независимо и асинхронно. Единственная синхронизация связана с подсчетом экспортированных и импортированных частиц. Поэтому можно ожидать, что данный алгоритм не имеет препятствий для неограниченного масштабирования при решении задачи МД на нашей системе.

Вообще задачи класса МД в настоящее время стали настолько востребованными, что создаются специально разработанные машины для этих задач, например Anton [10] компании D. E. Shaw Research. Будучи построенной на специализированных микросхемах, эта машина показала 100-кратное пре-

восходство по сравнению с традиционными кластерными суперкомпьютерами.

На ППВС оптимальное число обрабатываемых частиц на одно ядро, при котором масштабирование не падает, составляет порядка 10^3 частиц, что фактически на два порядка превосходит результаты, получаемые на традиционных кластерных системах [11]. Однако при этом ППВС остается универсальным вычислителем.

Итак, возможности и эффективность проектируемой ППВС "Буран" как универсального программируемого супервычислителя можно применить к актуальным задачам, таким как МД, а изменяя архитектуру вычислительных элементов, можно перейти и к построению спецвычислителей для конкретных задач.

Список литературы

1. Bowers K. J., Chow E., Xu X., Dror R. O., Eastwood M. P., Gregersen B. A., Klepeis J. L., Kolossvary I., Moraes M. A., Sacerdoti F. D., Salmon J. K., Shan Y., Shaw D. E. Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters // Proceedings of the ACM/IEEE Conference on Supercomputing (SC06). Article No. 84. New York, NY: IEEE, 2006.
2. Mederski J., Mikulski L., Bala P. Asynchronous Parallel Molecular Dynamics Simulations // Lecture Notes in Computer Science. 2008. Vol. 4967. P. 439–446.
3. Стемпковский А. Л., Левченко Н. Н., Окунев А. С., Цветков В. В. Параллельная потоковая вычислительная система — дальнейшее развитие архитектуры и структурной организации вычислительной системы с автоматическим распределением ресурсов // Информационные технологии. 2008. № 10. С. 2–7.
4. Климов А. В., Левченко Н. Н., Окунев А. С., Стемпковский А. Л. Вопросы применения и реализации потоковой модели вычислений // Проблемы разработки перспективных микро- и наноэлектронных систем — 2016: сб. трудов / под общ. ред. академика РАН А. Л. Стемпковского. М.: ИППМ РАН, 2016. Часть II. С. 100–106.
5. Климов А. В., Левченко Н. Н., Окунев А. С., Стемпковский А. Л. Суперкомпьютеры, иерархия памяти и потоковая модель вычислений // Программные системы: теория и приложения: электрон. научн. журн. 2014. Т. 5, № 1 (19). С. 15–36. URL: http://psta.psisras.ru/read/psta2014_1_15-36.pdf (дата обращения: 14.07.17).
6. Levchenko N. N., Okunev A. S., Zmejev D. N. Development Tools for High-Performance Computing Systems Using Associative Environment for Computing Process Organization // In bk.: Proceedings of IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS'2016). Yerevan, Armenia. October 14–17. 2016. P. 359–362.
7. Змеев Д. Н., Климов А. В., Левченко Н. Н., Окунев А. С., Стемпковский А. Л. Эмуляция аппаратно-программных средств параллельной потоковой вычислительной системы "Буран" // Информационные технологии. 2015. Т. 21, № 10. С. 757–762.
8. Воеводин В. В., Жуматий С. А., Соболев С. И., Антонов А. С., Брызгалов П. А., Никитенко Д. А., Стефанов К. С., Воеводин Вад. В. Практика суперкомпьютера "Ломоносов" // Открытые системы. 2012. № 7. С. 36–39.
9. Змеев Д. Н., Климов А. В., Левченко Н. Н. Средства распределения вычислений в ППВС "Буран" и варианты реализации блока выработки хэш-функций // Проблемы разработки перспективных микро- и наноэлектронных систем — 2016. Сборник трудов / под общ. ред. академика РАН А. Л. Стемпковского. М.: ИППМ РАН, 2016. Часть II. С. 107–113.
10. Dror R. O., Young C., Shaw D. E. Anton, a special-purpose molecular simulation machine // In: Encyclopedia of parallel computing. Springer. 2011. P. 60–71.
11. Куксин А. Ю., Ланкин А. В., Морозов И. В., Норман Г. Э., Орехов Н. Д., Писарев В. В., Смирнов Г. С., Стариков С. В., Стегайлов В. В., Тимофеев А. В. ЗАЧЕМ и КАКИЕ нужны суперкомпьютеры эксафлопсного класса? Предсказательное моделирование свойств и многомасштабных процессов в материаловедении // Программные системы: теория и приложения: электрон. научн. журн. 2014. Т. 5, № 1 (19). С. 191–244. URL: http://psta.psisras.ru/read/psta2014_1_191-244.pdf (дата обращения: 14.07.17).

A. L. Stempkovskiy, Academician, Director, e-mail: ippm@ippm.ru,
N. N. Levchenko, Ph. D., Head of the Department, e-mail: nick@ippm.ru,
A. S. Okunev, Ph. D., Chief Researcher, e-mail: oku@ippm.ru,
A. V. Klimov, Senior Researcher, e-mail: klimov@ippm.ru,
D. N. Zmejev, Researcher, e-mail: zmejevdn@ippm.ru,
Institute for Design Problems in Microelectronics (IPPM RAS), Moscow, 124365, Russia

Programming of the Molecular Dynamics Task in the Dataflow Computing Model

The scaling of HPC is often constrained due to excessive use of global barriers, especially when the load is non-uniform and varying, not to mention the costs to the barrier itself. The proposed in the article the approach for solving this problem relies on the dataflow computing paradigm. This approach is described by the example of an algorithm for the molecular dynamics task. In this task, instead of barriers, counting of nearby particles is introduced, which occurs asynchronously and in parallel with the main work. Synchronization relies only on local interactions between computational cores. The article also describes ways to increase the efficiency of molecular dynamics task execution on the parallel dataflow computing system (Buran) using various optimization techniques.

The approach to the solution of the molecular dynamics task considered in the article actually uses only local communications between neighbouring computational cores (cuboids). Moreover, different particles are processed, basically, independently and asynchronously. The only synchronization is related to the counting of the exported and imported particles. Therefore, it can be expected that this algorithm has no obstacles for unlimited scalability in solving the molecular dynamics task on the PDCS.

During the experiments, results were obtained that attest to the high degree of scalability of the molecular dynamics task on the cycle-accurate model of PDCS, and also on the emulator of the system on the "Lomonosov" supercomputer. For the PDCS, the optimal number of particles processed per computational core, at which scaling does not decrease, is about 10^3 particles, which is actually two orders of magnitude greater than the results obtained on traditional cluster systems. However, in this case, the PDCS remains universal.

On the basis of the universal parallel dataflow computing system "Buran", by changing the architecture of computing elements, it can be possible to proceed to the creation of special computers for the molecular dynamics task.

Keywords: molecular dynamics, scaling, asynchronous computing, barrier synchronization, parallel computing, dataflow computation model

References

1. Bowers K. J., Chow E., Xu X., Dror R. O., Eastwood M. P., Gregersen B. A., Klepeis J. L., Kolossvary I., Moraes M. A., Sacerdoti F. D., Salmon J. K., Shan Y., Shaw D. E. Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters, *Proceedings of the ACM/IEEE Conference on Supercomputing (SC06)*, Article No. 84, New York, NY: IEEE, 2006.
2. Mederski J., Mikulski L., Bala P. Asynchronous Parallel Molecular Dynamics Simulations, *Lecture Notes in Computer Science*. Vol. 4967, 2008, pp. 439–446.
3. Stempkovskij A. L., Levchenko N. N., Okunev A. S., Cvetkov V. V. Parallelnaja potokovaja vychislitel'naja sistema — dal'njshhee razvitiye arhitektury i strukturnoj organizacii vychislitel'noj sistemy s avtomaticheskim raspredeleniem resursov (Parallel dataflow computing system — the further development of architecture and the structural organization of the computing system with automatic distribution of resources), *Informacionnye tehnologii*, no. 10, 2008, pp. 2–7.
4. Klimov A. V., Levchenko N. N., Okunev A. S., Stempkovskij A. L. Voprosy pri-menenija i realizacii potokovoj modeli vychislenij (The application and implementation issues of dataflow computing system), *Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem — 2016. Sbornik trudov*, part II, ed. A. L. Stempkovskij, Moscow, IPPM RAS, 2016, pp. 100–106.
5. Klimov A. V., Levchenko N. N., Okunev A. S., Stempkovskij A. L. Superkomp'jutery, ierarhija pamjati i potokovaja model' vychislenij (Supercomputers, memory hierarchy and dataflow computation model), *Programmnye sistemy: teorija i prilozhenija: jelektron. nauchn. Zhurn*, 2014, vol. 5, no. 1 (19), pp. 15–36. URL: http://psta.psiras.ru/read/psta2014_1_15-36.pdf (date of access: 14.07.17).
6. Levchenko N. N., Okunev A. S., Zmejev D. N. Development Tools for High-Performance Computing Systems Using Associative Environment for Computing Process Organization, *Proceedings of*

IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS'2016), Yerevan, Armenia, October 14–17, 2016, pp. 359–362.

7. Zmeev D. N., Klimov A. V., Levchenko N. N., Okunev A. S., Stempkovskij A. L. Jemuljacija apparatno-programmnyh sredstv parallelnoj potokovoj vychislitel'noj sistemy "Buran" (Emulation on Hardware and Software of the Parallel Computing Streaming System "Buran"), *Informacionnye tehnologii*, 2015, vol. 21, no. 10, pp. 757–762.

8. Voevodin V. I., Zhumatij S. A., Sobolev S. I., Antonov A. S., Bryzgalov P. A., Nikitenko D. A., Stefanov K. S., Voevodin V. V. Praktika superkomp'jutera "Lomonosov" (Practice of the supercomputer "Lomonosov"), *Otkrytie sistemy*, 2012, № 7, pp. 36–39.

9. Zmeev D. N., Klimov A. V., Levchenko N. N. Sredstva raspredelenija vychislenij v PPVS "Buran" i varianty realizacii bloka vyrabotki hjeshe-funkcij (The Tools for Computation Distribution in the PDCS "Buran" and Hash-Functions Block Implementation Options), *Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem — 2016. Sbornik trudov*, part II, ed. A. L. Stempkovskogo, Moscow, IPPM RAS, 2016, pp. 107–113.

10. Dror R. O., Young C., Shaw D. E. Anton, a special-purpose molecular simulation machine, *Encyclopedia of parallel computing*, Springer, 2011, pp. 60–71.

11. Kuskina A. Ju., Lankin A. V., Morozov I. V., Norman G. Je., Orehov N. D., Pisarev V. V., Smirnov G. S., Starikov S. V., Stegajlov V. V., Timofeev A. V. ZACHEM i KAKIE nuzhny superkomp'jutery jeksaflopsnogo klassa? Predskazatel'noe modelirovanie svojstv i mnogomasshtabnyh processov v materialovedenii (Predictive modeling and simulation of properties and multi-scale processes in materials science. Tasks for Exaflops-era supercomputers), *Programmnye sistemy: teorija i prilozhenija: jelektron. nauchn. Zhurn*, 2014, vol. 5, № 1 (19), pp. 191–244. URL: http://psta.psiras.ru/read/psta2014_1_191-244.pdf (date of access: 14.07.17).

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 004.89 + 004.021

П. В. Казаков, канд. техн. наук, доц., e-mail: pvk_mail@list.ru,
Брянский государственный технический университет

Модель параллельных вычислений для генетических алгоритмов многокритериальной оптимизации, ориентированная на графические процессорные устройства с архитектурой CUDA

Предлагается подход к организации гибридной модели параллельных вычислений для генетических алгоритмов многокритериальной оптимизации. В нем используется технология CUDA. Этот подход позволяет не только увеличить скорость вычислений, но и реализовать различные способы распараллеливания вычислений в генетическом алгоритме. Приведен сравнительный анализ эффективности использования предлагаемой модели параллельных вычислений при решении задач многокритериальной оптимизации разной сложности.

Ключевые слова: многокритериальная оптимизация, множество и граница Парето, многокритериальные генетические алгоритмы, параллельные вычисления, технология CUDA

Введение

Задачи многокритериальной оптимизации (МКО) возникают при разработке и исследовании математических моделей различных систем. Возможность оценки качества их функционирования одновременно по набору критериев позволяет определить пределы совершенствования системы. В отличие от однокритериальной оптимизации, использование множества критериев оптимальности требует привлечения специальных методов оптимизации, позволяющих выполнять поиск решений с учетом конфликта критериев [1]. Он проявляется в невозможности улучшения решений одновременно по всем критериям. Подход, обеспечивающий в таких условиях наиболее удобное для исследователя решение задачи МКО, связан с непосредственным определением дискретной аппроксимации множества Парето-оптимальных решений, среди которых лицо, принимающее решение (ЛПР), руководствуясь своими субъективными предпочтениями, выбирает окончательное решение. Известно, что дискретная аппроксимация множества Парето может содержать любое число решений, при этом информация об их числе, возможных координатах в пространстве поиска, как правило, отсутствует. Все это усложняет нахождение дискретной аппроксимации множества Парето, делает эту процедуру вычислительно трудоемкой и, следовательно, плохо автоматизированной.

В настоящее время одним из высокоэффективных среди существующих подходов к построению множества Парето в виде его дискретной аппрок-

симации является применение генетических алгоритмов, их специальных версий для задач МКО (ГА МКО) [2—5]. Генетические алгоритмы относятся к многоточечным итерационным методам оптимизации, позволяющим одновременно исследовать множество решений. В терминах ГА МКО они называются индивидами, вместе образующими популяцию. После каждой итерации, включающей последовательность применения специальных генетических операторов, множество исследованных решений изменяется, создавая эффект эволюции индивидов популяции. Такое свойство коллективного исследования пространства поиска позволяет ГА МКО за один запуск находить множество Парето-оптимальных решений в виде его дискретной аппроксимации.

Выполнение различных операций обработки информации одновременно о целой популяции индивидов, включающих вычисление значений критериев для оценки качества решений, управление траекториями поиска в пространствах переменных и критериев, приводит к неудовлетворительному для пользователя росту времени выполнения ГА МКО. В то же время эти алгоритмы имеют возможности распараллеливания реализуемого ими процесса вычислений, позволяющие не только сократить время расчетов, но и повысить качество решений.

Подходы к разработке параллельных ГА МКО основываются на моделях параллельных вычислений для обычных генетических алгоритмов и классифицируются [6, 7] как: модель "главный—подчиненный", модель островов с подпопуляциями и

модель с массовым параллелизмом. Каждая из них позволяет по-разному организовать параллельную обработку данных в ГА МКО при определении множества Парето [8]. Так, первая модель предназначена для сокращения времени вычислений при решении задач МКО, требовательных к ресурсам компьютера при расчете значений критериев [9, 10]. В ней используется единая популяция, для которой все генетические операторы выполняются главным компьютером, а вычисление значений критериев распределяется по подчиненным компьютерам.

В модели с островами популяция разделяется на подпопуляции, каждая из которых обрабатывается на отдельном компьютере. Между подпопуляциями организуется кооперация, направленная на поддержание разнообразия индивидов, наиболее полное исследование пространства поиска [11].

В островной модели могут одновременно применяться различные ГА МКО, а также модификации одного генетического алгоритма с различными правилами обмена информацией об индивидах внутри популяции.

В модели с массовым параллелизмом популяция обрабатывается в многопроцессорной системе на единой аппаратной платформе. Считается, что число процессоров совпадает с размером популяции и они связаны между собой по топологии "решетка". Поэтому в генетических операторах участвуют только индивиды из непосредственно связанных процессоров. Во времени это может порождать движение (диффузию) индивидов с наилучшей пригодностью через все процессоры. Эта модель позволяет обеспечить высокопараллельный режим работы ГА МКО, а особенности обмена данными — повысить качество решений [12].

Наибольший эффект от применения параллельного ГА МКО может быть достигнут при объединении в нем основных преимуществ всех трех описанных моделей, что в общем случае должно проявляться в снижении времени вычислений и повышении качества найденных решений. Достижение этого возможно при использовании гибридных моделей параллельных вычислений, в различной степени интегрирующих в себе принципы функционирования и возможности основных моделей параллельных генетических алгоритмов. Однако попытки реализации этого часто связаны с необходимостью преодоления программно-аппаратных различий в технологиях межкомпьютерного и межпроцессорного взаимодействия, на которых построены соответствующие модели параллельных ГА (ПГА). Одним из технико-экономически эффективных путей разрешения такой ситуации является использование вычислительной технологии NVidia CUDA [13] для графических процессоров.

В настоящее время известны положительные примеры применения технологии CUDA в ряде ПГА МКО [14] — параллельной версии метода

роющихся частиц для определения множества Парето [15]. Эффективность этих алгоритмов исследовали при решении задач МКО с числом критериев, не превышающим трех, в ряде случаев анализировали только коэффициент сокращения времени вычислений, также использовали исключительно модель вычислений с массовым параллелизмом. Вместе с тем разработка ПГА МКО с технологией CUDA остается достаточно новым направлением в эволюционных методах МКО, и его возможности не ограничиваются уже существующими разработками. Так, рост эффективности ПГА МКО с технологией CUDA возможен за счет применения различных схем разделения задач между центральным (CPU) и графическим (GPU) процессорами, использования гибридных параллельных моделей ГА. Подход, реализующий это, рассматривается в данной работе. В нем, кроме модели с массовым параллелизмом, на определенных этапах работы ГА МКО для распределения вычислений между CPU и GPU используются модели "главный—подчиненный", а также ряд виртуальных топологий взаимодействия индивидов популяции по принципам островной модели. Эффективность реализации такого подхода исследуется при решении задач МКО различной сложности.

1. Постановка задачи многокритериальной оптимизации и применяемые генетические алгоритмы

В общем случае задача многокритериальной оптимизации [1] включает n оптимизируемых переменных, m критериев оптимальности и k ограничений. Предполагая, что требуется минимизация значений критериев, математически постановка задачи МКО может быть представлена следующим образом:

$$F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \rightarrow \min_{x \in X_\varphi},$$
$$X_\varphi = \{x \in R^n | \varphi(x) \leq 0\},$$

где $F(x)$ — векторный критерий оптимальности; $\varphi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_k(x))$ — вектор ограничивающих функций. Множество достижимости задачи МКО $Y_\varphi = \{F(x) | X \in X_\varphi\}$.

Полагаем, что решение поставленной задачи многокритериальной оптимизации принадлежит множеству Парето этой задачи, т. е. сужению множества X_φ , содержащему решения, которые нельзя улучшить одновременно по всем критериям. Отображение множества Парето в пространство критериев образует границу Парето, имеющую различные геометрические представления. Необходимым условием для получения репрезентативной аппроксимации множества Парето является участие всех критериев при сравнении решений. Для формализации этой процедуры применяется принцип Парето-доминирования [1]. Согласно ему из двух

любых решений $x^{(1)}, x^{(2)} \in X_{\phi}$ первое доминирует второе, если по всем критериям $x^{(1)}$ не хуже, но хотя бы по одному лучше, чем $x^{(2)}$. Этот принцип применяется и для оценки пригодности индивидов популяции в ГА МКО, позволяя на этапе отбора оставлять наиболее перспективные индивиды для формирования результирующего множества Парето. С этой целью проводится упорядочивание индивидов по значениям их ранга Парето [3]. Он назначается всем индивидам каждого поколения. Те из них, которые имеют ранг, равный нулю, являются недоминируемыми и принадлежат множеству Парето. Индивиды с максимальным рангом считаются доминируемыми любыми из текущей популяции, имеющими меньший ранг.

Для увеличения числа и точности Парето-оптимальных решений, кроме ранга Парето в оценке пригодности индивидов участвуют и другие показатели. Они зависят от применяемых ГА МКО, среди которых можно выделить алгоритмы NSGA-II [16], SPEA2 [17]. В первом алгоритме при оценке пригодности индивида учитываются индекс его относительного доминирования (отношение числа индивидов, доминируемых данным индивидом, к числу индивидов, которые доминируют его), плотность его соседства на границе Парето. Главное отличие SPEA2 заключается в использовании наряду с основной популяцией обновляемого Парето-архива, содержащего Парето-оптимальные решения, найденные за все время работы алгоритма. Предусмотрены процедуры обработки Парето-архива, включающие его кластеризацию, объединение с основной популяцией для выполнения операции отбора, участие при вычислении индекса относительного доминирования и др. [17]. Опыт применения этих алгоритмов подтверждает эффективность используемых ими способов управления популяцией для определения множества Парето. Поэтому они выбраны для исследования применения предлагаемой гибридной модели параллельных вычислений в ГА МКО.

2. Описание гибридной модели параллельных вычислений для ГА МКО

2.1. Некоторые сведения о технологии CUDA

CUDA (*Compute Unified Device Architecture*) представляет собой расширение языка C, позволяющее в программе распределить инструкции и данные между центральным и графическим процессорами [13]. Этим достигается параллельное выполнение программы и сокращение времени при решении вычислительных задач. Конструктивно отличаясь от CPU, большая часть площади кристалла графического процессора занята устройствами выполнения арифметико-логических операций, что делает его наиболее эффективным при выполнении именно математических вычислений.

В основе программной организации CUDA лежит так называемая модель SIMT (*Single Instruction Multiple Threads*): каждый из SIMT-процессоров выполняет одинаковую инструкцию параллельно над разными данными. Для их структурирования используется понятие потока данных, которые группируются в одно-, двух- или трехмерные блоки, а они, в свою очередь, объединяются в одно-, двухмерные сетки. Доступ к отдельному элементу данных осуществляется по набору из соответствующих индексов. Программный код с вычислениями для GPU оформляется в виде функции языка C и называется ядром. Ядро содержит операции, которые будут выполнены над данными в потоках. С каждым потоком связывается свой процессор в GPU, одновременно исполняется одна сетка и часть блока в 32 потока. Программно-аппаратные особенности технологии CUDA предполагают специальный подход к созданию программ, учитывающий распределение данных по потокам (блокам), эффективное использование памяти, синхронизацию вычислений и т.п. В общем виде параллельное решение задачи с использованием CUDA заключается в выполнении следующего набора действий: подготовка данных для расчетов в виде потоков (блоков); копирование этих данных в память GPU; вызов функции ядра для вычислений в GPU; копирование результатов расчета из GPU в память CPU; обработка полученных результатов.

2.2. Распределение операций ГА МКО между центральным и графическим процессорами

Для распределения операций ГА МКО между CPU и GPU были определены те операции, которые выполняются независимо и, следовательно, допускают распараллеливание. Кроме этого, для передачи операции GPU проверяются явное отсутствие в ней взаимодействия между потоками CUDA, конкурентного доступа к общим данным, а также наличие преимущественно арифметических вычислений. Если же операция ГА МКО содержит действия по проверке логических условий, циклических конструкций, то более эффективно ее выполнение CPU. В то же время, были определены операции, которые выполняют одновременно CPU и GPU, для чего задействуются соответствующие механизмы синхронизации. В предлагаемой гибридной модели все вычисления разделены на следующие виды:

- создание начальной популяции;
- вычисление значений критериев;
- вычисление ранга Парето индивидов, плотности соседств индивидов на границе Парето, индекса относительного доминирования и т.п.;
- обновление Парето-архива;
- вычисление пригодности индивидов;
- отбор индивидов-родителей;
- выполнение кроссингвера и мутации.

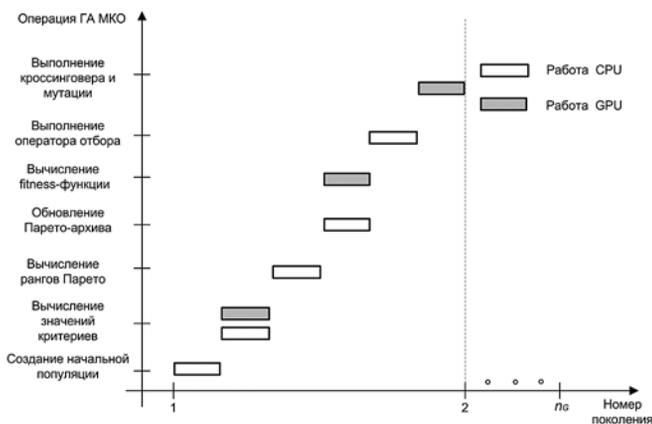


Рис. 1. Операции ГА МКО, выполняемые CPU и GPU (n_G — число поколений)

Порядок распределения между CPU и GPU указанных операций показан на рис. 1.

Представленный порядок вычислений в гибридной параллельной модели ГА МКО может быть применен для любого алгоритма этого класса. Известно, что их отличия часто связаны с особенностями вычисления дополнительных характеристик индивида, в частности, плотности соседства на границе Парето, индекса относительного доминирования, распределения по нишам (кластерам) и т.п. Поэтому для таких целей выделена отдельная операция, выполняемая на CPU и обозначенная как "Вычисление рангов Парето".

Перенос вычислений ГА МКО в среду CUDA требует добавления и корректировки механизмов взаимодействия между индивидами на уровне как структур данных, так и алгоритмических конструкций.

2.3. Особенности реализации гибридной модели параллельных вычислений для ГА МКО

Создание начальной популяции. Начальная популяция создается средствами CPU, генам всех индивидов присваиваются случайные значения из области определения переменных x_i , $i = 1, \dots, n$. Соответственно определяется соотношение размера популяции и числа генов для их распределения по блокам и потокам. Их число выбирают, исходя из технических характеристик CUDA, для достижения максимальной производительности. В памяти CPU популяция и значения критериев хранятся с использованием матричной структуры данных размером $n_p \times n$, $n_p \times t$ соответственно (n_p — размер популяции). Подобная форма используется и для других необходимых ГА МКО характеристик индивидов. При переносе данных в память видеокарты применяются другие структуры данных, учитывающие ее организацию для обеспечения максимального распараллеливания обработки информации.

Вычисление значений критериев. Данная операция выполняется для каждой новой популяции и

может оказаться самой продолжительной по времени выполнения. В то же время вычисление значений критериев, как правило, осуществляется независимо друг от друга, что позволяет распараллелить этот процесс. С этой целью в память видеокарты загружается популяция (или ее часть), а также массив с числом переменных $n_p \times t$ для хранения значений критериев.

В целях ускорения выполнения рассматриваемой операции можно применять модель "главный—подчиненный", когда CPU и GPU совместно участвуют в вычислении значений критериев. В этом случае соотношение в распределении между ними индивидов или критериев зависит от числа, сложности и особенностей вычисления последних. Так, в ряде задач для вычисления значений определенных критериев, кроме арифметических операций, может потребоваться проверка различных условий на соответствие ограничениям, а также выполнение специальных алгоритмических конструкций. В такой ситуации вычисление значений критериев целесообразно выполнять с использованием CPU, так как с подобными задачами он справляется эффективнее GPU. Принимая во внимание, что CPU и GPU работают независимо друг от друга, для одновременного завершения обоими процессорами выполнения этой операции применяют специальные функции синхронизации.

Вычисление рангов Парето. Данная процедура выполняется для любых ГА МКО, использующих принципы Парето для оценки решений. Вычисление рангов Парето представляет собой итерационную процедуру попарного сравнения значений критериев индивидов в соответствии с принципом Парето-доминирования. Преобладание в ней циклических и условных алгоритмических конструкций, а также одновременного доступа к одним значениям переменных делают целесообразной реализацию этой процедуры на CPU. Также после получения информации о значениях рангов Парето индивидов могут быть определены их дополнительные характеристики, специфические для используемого ГА МКО, например показатели плотности соседств индивидов на границе Парето, относительного доминирования, а также распределения по кластерам и нишам.

Вычисление пригодности индивидов и обновление Парето-архива. В ГА МКО *fitness*-функция является комплексным показателем, учитывающим ранг Парето индивида, его положение на (вне) границы Парето и т.п., а также качество соответствующего решения относительно субъективных предпочтений ЛПР. Определение пригодности каждого индивида можно рассматривать как независимую вычислительную операцию, которая выполняется средствами GPU. Для этого в память видеокарты передается вся необходимая для расчетов информация, общее число CUDA-потоков совпадает с раз-

мером популяции. Согласно предлагаемой модели ПГА МКО, одновременно с этой операцией может быть выполнена и другая, а именно обновление Парето-архива. Эта процедура очень часто используется в ГА МКО и состоит в определении рангов Парето индивидов относительно Парето-архива при копировании туда лучших индивидов новой популяции. Действия по обновлению Парето-архива носят не вычислительный, а упорядочивающий характер, когда требуется использование большого числа условных конструкций, одновременного доступа к данным для сравнения. Это делает целесообразным реализацию этой процедуры на CPU в виде отдельного потока команд с возможностью синхронизации с GPU. После завершения работы каждого из них выполнение переходит к следующей операции.

Отбор индивидов-родителей. После получения информации о пригодности индивидов проводится отбор из них кандидатов для формирования следующего поколения. Достаточно часто для этого используется турнирный отбор, основанный на попарном сравнении значений *fitness*-функции случайно выбранных индивидов. В предлагаемой модели эту операцию выполняет CPU. Такой выбор обусловлен отсутствием в ней вычислительных действий, а также необходимостью одновременного доступа к случайно выбранному набору индивидов. Тем не менее следует отметить и существующие примеры реализации оператора отбора средствами GPU [14, 18, 19].

Кроссинговер и мутация. Кроссинговер и мутация отвечают за изменение состава популяции, исследование новых решений в пространстве поиска. Для выполнения операций кроссинговера и мутации осуществляется доступ к двум (одному при мутации) индивидам, а точнее к их одноименным генам. Их изменение, как правило, не зависит от других генов индивида, что делает возможным достижение высокого уровня параллелизма выполнения операций кроссинговера и мутации. В настоящей модели вычислений это реализуется путем размещения этих операторов в одной CUDA-функции. Она выполняется одновременно для множества потоков, каждый из которых связан с отдельным геном. Для этого индивиды копируются в память видеокарты в виде следующей структуры (рис. 2) [14, 15], применяемой в CUDA-версиях ПГА. Она представляет собой одномерный массив генов размера $n_p \times n$, упорядоченных по номерам таким образом, что вначале располагаются первые гены всех индивидов, затем все их вторые гены и т.д. Подобная организация позволяет более эффективно выполнять доступ к памяти, использовать быструю разделяемую память.

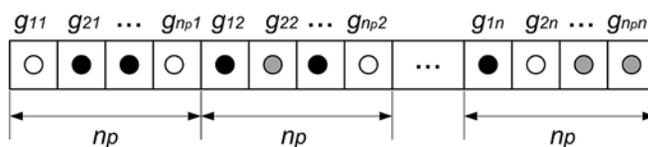


Рис. 2. Организация популяции в памяти видеокарты

Для независимой обработки каждого потока в предлагаемой модели используется вещественное представление генов. В этом случае кроссинговер и мутация представляют собой арифметические операции, максимально эффективно выполняемые GPU. Для исследования модели применялись кроссинговер BLX-alpha, формирующий одного потомка, и полиномиальная мутация.

При выполнении операции кроссинговера ген первого индивида-родителя, а также его индекс в популяции определяются по передаваемым в CUDA-функцию данным о номерах блока и потока внутри него. Выбор второго индивида-родителя проводится в зависимости от позиции его размещения в используемой виртуальной топологии популяции, моделирующей некоторую вычислительную сеть. Она ассоциируется с популяцией таким образом, что каждые n процессоров GPU, связанных с n потоками и, соответственно, n генами, моделируют компьютер, обрабатывающий одного индивида. Были исследованы три топологии: кольцо, решетка и N -мерный гиперкуб (рис. 3). Они отличаются числом компьютеров-соседей, связанных друг с другом, что оказывает большое влияние на скорость перемещения индивидов с высокой пригодностью внутри популяции, поддержание ее разнообразия, а также на объем подпопуляций, внутри которых происходит взаимодействие между индивидами.

В топологии "кольцо" выбор второго индивида-родителя возможен только слева или справа от

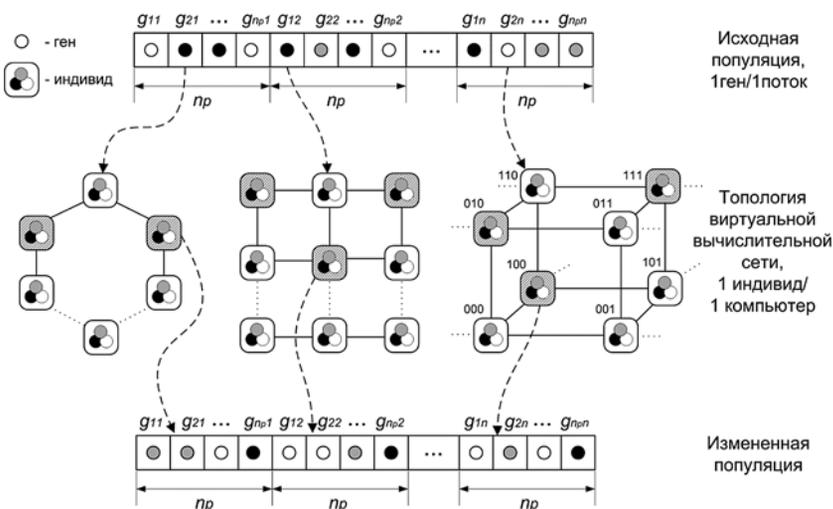


Рис. 3. Варианты топологий виртуальной вычислительной сети, моделируемой средствами CUDA

первого индивида-родителя. В топологии "решетка" второй индивид-родитель выбирается среди ближайших от первого в направлениях "север", "юг", "запад", "восток". В топологии N -мерный гиперкуб используется 2^N компьютеров, по каждой размерности имеется только два компьютера. Каждый компьютер связан ровно с такими N соседями, у которых двоичное представление их номеров имеет лишь одну различающуюся с ним позицию. В сравнении с двумя другими топологиями, у которых с увеличением размера популяции число кандидатов для второго индивида-родителя не изменяется, у гиперкуба с ростом N пропорционально увеличивается численность подпопуляций, образуемых такими кандидатами. Это позволяет исследовать на данной топологии островную модель организации популяции. При отсутствии дополнительных правил взаимодействия в ней между островами может осуществляться посредством частично перекрывающихся в гиперкубе подпопуляций размером N с постоянным перемещением между ними различных индивидов. В предлагаемой модели параллельных вычислений популяция со структурой, представленной на рис. 2, дублируется в памяти видеокарты. Эта копия используется для непосредственного изменения хромосом индивидов при выполнении генетических операторов.

Измененная популяция далее копируется обратно в память CPU, заполняя исходную матричную структуру и образуя новое поколение индивидов. Начиная с вычисления значений критериев к нему применяются все рассмотренные операции.

После выполнения условия завершения работы ГА МКО из последней версии обновленного Парето-архива формируется итоговая дискретная аппроксимация множеств Парето.

3. Исследование результатов применения гибридной модели параллельных вычислений для ГА МКО

Для анализа возможностей ПГА МКО, использующих разработанную модель параллельных вычислений, был проведен ряд экспериментов при решении тестовых задач многокритериальной оптимизации разной сложности. Исследование проводили с использованием ГА МКО NSGA-II и SPEA2 на компьютере Intel Core 2 Duo, 2,2 ГГц, с видеокарты NVIDIA GeForce 9800 GT. Эти алгоритмы выполнялись как в последовательном режиме работы, так и с использованием CUDA. По результатам их работы ставилась задача оценить степень роста скорости вычислений и влияние на качество решений использования различных виртуальных топологий. Эффективность алгоритмов оценивали по следующим показателям [20]: I_{ONVG} (мощность найденного множества Парето); I_{GD} (степень близости между полученной и заданной эталонной границами Парето); I_S (равномерность распределения решений вдоль границы Парето); I_{DE} (максимальная протяженность границы Парето по каждой из размерностей); I_{OT} (время работы ГА МКО при определении множества Парето в секундах). Во всех случаях была использована вещественная форма представления генов индивидов. Значения

Результаты исследования ПГА МКО с гибридной моделью параллельных вычислений (виртуальные топологии популяции: О — кольцо, Р — решетка, К — куб)

Задача [21]	Индикатор	SPEA2	NSGA-II	SPEA2-топологии О/Р/К	NSGA-II-топологии О/Р/К
DTLZ1 ($m = 2$)	I_{ONVG}	88	93	82/84/84	86/91/82
	I_{GD}	0,076	0,072	0,073/0,081/0,072	0,067 /0,073/0,075
	I_S	0,197	0,204	0,194/ 0,184 /0,201	0,206/0,207/0,218
	I_{DE}	0,973	0,978	0,896/0,963/0,982	0,956/ 0,987 /0,983
	I_{OT}	183,2	142,6	32,5/32,5/32,6	25,5 /25,6/25,6
DTLZ2 ($m = 4$)	I_{ONVG}	226	217	217/205/ 226	207/218/224
	I_{GD}	6,132	6,164	6,213/6,152/6,157	6,128 /6,173/6,177
	I_S	0,136	0,127	0,128/ 0,117 /0,142	0,141/0,124/0,147
	I_{DE}	1,952	1,946	1,873/1,986/1,943	1,886/ 2,037 /1,926
	I_{OT}	1531,4	243,5	221,8/223,5/225,2	36,3 /37,17/37,4
DTLZ3 ($m = 6$)	I_{ONVG}	284	347	271/277/296	293/302/338
	I_{GD}	256,452	314,613	264,762/257,352/259,105	318,247/326,405/328,117
	I_S	0,538	0,417	0,677/0,532/0,584	0,516/0,428/0,433
	I_{DE}	2,638	3,112	2,467/2,788/2,816	2,974/3,287/ 3,311
	I_{OT}	21173,4	2387,6	2499,8/2505,7/2520,6	287,6 /291,17/292,9
DTLZ6 ($m = 8$)	I_{ONVG}	439	474	392/427/431	418/463/ 476
	I_{GD}	17,423	14,637	18,541/17,428/17,611	15,147/14,852/ 14,316
	I_S	0,287	0,252	0,291/0,278/0,283	0,297/ 0,243 /0,248
	I_{DE}	1,867	1,924	1,754/2,113/ 2,287	1,873/2,016/2,218
	I_{OT}	128486,3	4776,2	10981,7/11057,3/11172,7	426,4 /434,2/442,24

управляющих параметров ГА МКО, включающих число поколений n_G , размер популяции n_p , размер Парето-архива n_A , вероятность кроссинговера p_c , вероятность мутации p_m , управляющий параметр кроссинговера α , параметр силы мутации η_m устанавливали следующими: $n_G = 1000$; $n_p = 1024$; $n_A = 500$; $p_c = 0,9$; $p_m = 1/n$; $\alpha = 0,5$; $\eta_m = 30$. При вычислении критерии равномерно распределяли между CPU и GPU.

Каждый ПГА МКО запускали многократно, для каждого индикатора определяли среднее значение. Итоговые результаты решения всех задач приведены в таблице. Наилучшие достигнутые усредненные значения показателей выделены полужирным шрифтом.

Анализ полученных значений индикатора I_{OT} позволяет сделать вывод, что использование технологии CUDA сокращает время вычислений от 5,6 до 11,7 раз в зависимости от числа критериев. Применение CUDA-модели параллельных вычислений, различных виртуальных топологий популяции оказали влияние и на качество найденных аппроксимаций множества Парето. С топологией "куб" были достигнуты наилучшие результаты по протяженности и точности границы Парето для $m = 8$. В то же время, при топологии "кольцо" по большинству индикаторов оказались наихудшие значения для $m > 4$, что связано с быстрым вырождением популяции вследствие ограниченности выбора индивидов-родителей и быстрого перемещения, копирования наилучших индивидов в популяции. При использовании топологии "решетка" в трех задачах из четырех была найдена граница Парето с наиболее плотно расположенными на ней точками. Что касается числа найденных Парето-оптимальных решений, то значения индикатора I_{ONVG} параллельных ГА МКО SPEA2 и NSGA-II оказались близкими к их версиям, выполняемым только на CPU.

Заключение

В работе представлена гибридная модель параллельных вычислений для ГА МКО. Она отличается использованием принципов организации вычислений трех известных моделей для параллельных генетических алгоритмов, а также ориентацией на графические процессорные устройства с архитектурой CUDA. В данной модели учтены возможности распараллеливания всех основных операций ГА МКО, поэтому на ее основе могут быть созданы параллельные версии различных алгоритмов этого класса. Применение разработанной модели с известными алгоритмами NSGA-II и SPEA2 показало, что в зависимости от числа критериев возможно сокращение времени вычислений с 5,6 до 11,7 раза. Также в ряде случаев было достигнуто повышение качества решений при использовании различных топологий взаимодействия индивидов популяции.

Продолжение исследований связано с разработкой на базе предложенной модели других ПГА МКО, изучением влияния на время поиска и качество решений различных размеров популяции. Также предполагается использование ПГА МКО для решения задач многокритериальной оптимизации при проектировании технических систем.

Список литературы

1. **Соболь И. М., Статников Р. Б.** Выбор оптимальных параметров в задачах со многими критериями. 2-е изд., перераб. и доп. М.: Дрофа, 2006. 175 с.
2. **Гладков Л. А., Курейчик В. В., Курейчик В. М.** Генетические алгоритмы. М.: ФИЗМАТЛИТ, 2006. 320 с.
3. **Deb K.** Multi-Objective Optimization Using Evolutionary Algorithms. Hoboken: Wiley, 2009. 536 p.
4. **Казаков П. В.** Генетические алгоритмы многокритериальной оптимизации. Обзор // Информационные технологии. 2011. № 9. С. 2—8.
5. **Карпенко А. П., Митина Е. В., Семенехин А. С.** Когенетический алгоритм Парето-аппроксимации в задаче многокритериальной оптимизации // Информационные технологии. 2013. № 1. С. 22—32.
6. **Cantu-Paz E.** Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic Publishers, 2001. 162 p.
7. **Alba E., Tomassini M.** Parallelism and Evolutionary Algorithms // IEEE Transactions on Evolutionary Computation. 2002. Vol. 6, N. 5. P. 443—462.
8. **Luna F., Nebro A., Alba E.** Parallel Evolutionary Multiobjective Optimization // Parallel Evolutionary Computations. Springer, 2006. P. 33—56.
9. **Makinen R., Periaux J., Toivanen J.** Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms // International Journal for Numerical Methods in Fluids. 1999. Vol. 30, N. 2. P. 149—159.
10. **Карпенко А. П., Овчинников В. А., Семенехин А. С.** Программная система PRADIS // FRONT для построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельного генетического алгоритма // Информационные технологии. 2009. № 8. С. 27—33.
11. **Tan K. C., Yang Y. J., Lee T. H.** A Distributed Cooperative Co-evolutionary Algorithm for Multiobjective Optimization // IEEE Transactions on Evolutionary Computation. 2006. Vol. 10, N. 5. P. 527—549.
12. **Nebro A., Durillo J., Luna F., Dorronsoro B., Alba E.** Design Issues in a Multiobjective Cellular Genetic Algorithm // Proc. of the Evolutionary Multi-Criterion Optimization, 4th International Conference (EMO 2007). 2007. P. 126—140.
13. **Боресков А. В., Харламов А. В.** Основы работы с технологией CUDA. М.: ДМКПресс, 2010. 232 с.
14. **Wong M.** Parallel Multi-Objective Evolutionary Algorithms on Graphics Processing Units // Proc. of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference (GECCO'09). 2009. P. 2515—2522.
15. **Zhou Y., Tan Y.** GPU-Based Parallel Multi-objective Particle Swarm Optimization // International Journal of Artificial Intelligence. 2011. Vol. 7, N. 11. P. 125—141.
16. **Deb K., Pratap A., Agarwal S., Meyarivan T.** A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II // IEEE Transactions on Evolutionary Computation. 2002. Vol. 6, N. 2. P. 182—197.
17. **Zitzler E., Laumanns M., Thiele L.** SPEA2: Improving the Strength Pareto Evolutionary Algorithm // Proc. of the EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, 2002. P. 95—100.
18. **Kromer P., Platos J., Snasel V., Abraham A.** A Comparison of Many-threaded Differential Evolution and Genetic Algorithms on CUDA // Proc. of the Third World Congress on Nature and Biologically Inspired Computing, 2011. P. 516—521.
19. **Pospichal P., Jaros J., Schwarz J.** Parallel Genetic Algorithm on the CUDA Architecture // Proc. of the International Conference EvoApplications. Part I. 2010. P. 442—451.
20. **Казаков П. В.** Оценка эффективности генетических алгоритмов многокритериальной оптимизации. Часть 1 // Информационные технологии. 2012. № 8. С. 2—6.
21. **Казаков П. В.** Оценка эффективности генетических алгоритмов многокритериальной оптимизации. Часть 2 // Информационные технологии. 2012. № 9. С. 42—46.

The Model of Parallel Computing Based on CUDA Technology for Multi-Objective Genetic Algorithms

There is the effective way for improving of multi-objective genetic algorithms (MOGA). It is using the technologies of parallel computing. These technologies allow to reduce time of the big population processing, the nondominated solution search, the calculation of criterion values, etc. There are three most known approaches for the implementation of the parallel computing in multi-objective genetic algorithms: the master-slave model, the island model and the massive parallel model. These models allow to obtain different goals in parallel MOGA, for example, the master-slave model reduces the computing time for criterion values calculation, the island model efforts an opportunity for independent population processing on different computers and the last model allows to obtain the extreme performance in MOGA running on multiprocessors systems. The most perspective is integrated the advantages of all models of parallel MOGA in the hybrid model. The some problems appear with software/hardware platforms for a such hybrid model. One of solutions there is to use CUDA technology for high parallel computing on graphic processing unit. In article the original hybrid parallel model based on CUDA for MOGA are considered. This model differs from other that operations of MOGA are distributed between CPU, GPU for its loading balance, crossover and mutation operators implemented on some population topologies: ring, grid, hypercube. The experiments for evaluation hybrid CUDA-model were conducted on the benchmark problems DTLZ with different number of criterions. The sequential and parallel versions of multi-objective genetic algorithms SPEA2, NSGA-II were compared. The result demonstrated that speedups of parallel MOGA range from 5,6 to 11,7 depending on number of criterions. Also it is found the use of hybrid parallel CUDA-model for MOGA allowed to improve the quality of Pareto-set on some indicators. For future work the influence of different population sizes on the search time and the quality of solutions will be studied.

Keywords: Multi-objective optimization, Pareto's principles, Pareto front, multi-objective genetic algorithms, parallel computing, CUDA technology

References

1. Sobol I. M., Statnikov R. B. Vybory optimalnykh parametrov v zadachakh so mnogimi kriteriyami, Moscow, Drofa, 2006, 175 p. (in Russian).
2. Gladkov L. A., Kureychik V. V., Kureychik V. M. Geneticheskie algoritmy. Moscow, Fizmatlit, 2006, 320 p. (in Russian).
3. Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms*, Hoboken, Wiley, 2009, 536 p.
4. Kazakov P. V. Geneticheskie algoritmy mnogokriterialnoy optimizatsiyi. Obzor, *Informacionnye tekhnologii*, 2011, no. 9, p. 2–8 (in Russian).
5. Karpenko A. P., Mitina E. V., Semenikhin A. S. Kogeneticheskiy algoritim Pareto-approximatsiyi v zadache mnogokriterialnoy optimizatsiyi, *Informatsionnye tekhnologii*, 2013, no. 1, pp. 22–32 (in Russian).
6. Cantu-Paz E. *Efficient and Accurate Parallel Genetic Algorithms*, Kluwer Academic Publishers, 2001, 162 p.
7. Alba E., Tomassini M. Parallelism and Evolutionary Algorithms, *IEEE Transactions on Evolutionary Computation*, 2002, vol. 6, no. 5, pp. 443–462.
8. Luna F., Nebro A., Alba E. Parallel Evolutionary Multiobjective Optimization, *Parallel Evolutionary Computations*, Springer, 2006, pp. 33–56.
9. Makinen R., Periaux J., Toivanen J. Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms, *International Journal for Numerical Methods in Fluids*, 1999, vol. 30, no. 2, pp. 149–159.
10. Karpenko A. P., Ovchinnikov V. A., Semenikhin A. S. Programmaya sistema PRADIS // FRONT dlya postroeniya mnozhestva Pareto v zadache mnogokriterialnoy optimizatsiyi dinamicheskikh sistem s ispolzovaniem paralelnogo geneticheskogo algoritma, *Informatsionnye tekhnologii*, 2009, no. 8, pp. 27–33 (in Russian).
11. Tan K. C., Yang Y. J., Lee T. H. A Distributed Cooperative Co-evolutionary Algorithm for Multiobjective Optimization, *IEEE Transactions on Evolutionary Computation*, 2006, vol. 10, no. 5, pp. 527–549.
12. Nebro A., Durillo J., Luna F., Dorronsoro B., Alba E. Design Issues in a Multiobjective Cellular Genetic Algorithm, *Proceedings of the Evolutionary Multi-Criterion Optimization, 4th International Conference (EMO 2007)*, 2007, pp. 126–140.
13. Boretkov A. V., Harlamov A. V. Osnovy raboty s tekhnologiyey CUDA. Moscow, DMKPress, 2010. 232 p. (in Russian)
14. Wong M. Parallel Multi-Objective Evolutionary Algorithms on Graphics Processing Units, *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference (GECCO'09)*, 2009, pp. 2515–2522.
15. Zhou Y., Tan Y. GPU-Based Parallel Multi-objective Particle Swarm Optimization, *International Journal of Artificial Intelligence*, 2011, vol. 7, no. 11, pp. 125–141.
16. Deb K., Pratap A., Agarwal S., Meyarivan T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 2002, vol. 6, no. 2, pp. 182–197.
17. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm, *Proceedings of the EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 2002, pp. 95–100.
18. Kromer P., Platos J., Snasel V., Abraham A. A Comparison of Many-threaded Differential Evolution and Genetic Algorithms on CUDA, *Proceedings of the Third World Congress on Nature and Biologically Inspired Computing*, 2011, pp. 516–521.
19. Pospichal P., Jaros J., Schwarz J. Parallel Genetic Algorithm on the CUDA Architecture, *Proceedings of the International Conference EvoApplications, Part I*, 2010, pp. 442–451.
20. Kazakov P. V. Ocenka effektivnosti geneticheskikh algoritmov mnogokriterialnoy optimizatsiyi. Chast 1, *Informatsionnye tekhnologii*, 2012, no. 8, pp. 2–6. (in Russian).
21. Kazakov P. V. Ocenka effektivnosti geneticheskikh algoritmov mnogokriterialnoy optimizatsiyi. Chast 2, *Informatsionnye tekhnologii*, 2012, no. 9, pp. 42–46. (in Russian).

В. В. Курейчик, д-р техн. наук, проф., e-mail: vkur@sfedu.ru,

В. В. Бова, доц., e-mail: vvbova@yandex.ru, **Д. В. Лещанов**, студент, leshok.dimkaa@yandex.ru,
Южный федеральный университет

Модель семантического поиска в системах управления знаниями на основе генетических процедур

Представлен отличающийся новизной подход к реализации модели семантического поиска, применимый к большим объемам распределенных источников знаний. Предложен генетический алгоритм улучшения качества поиска, позволяющий определить семантически приоритетные объекты данных и знаний для их представления в модели семантического поиска. Экспериментальные исследования подтвердили теоретическую значимость и перспективность применения такого подхода и позволили установить оптимальные значения параметров генетического алгоритма.

Ключевые слова: семантическая близость, онтологии, семантический поиск, генетические алгоритмы, системы управления знаниями

Введение

В связи с постоянным ростом информационных потоков в различных технических, экономических и социальных интеллектуальных информационных системах (ИИС) задачи разработки новых способов представления, формализации, систематизации, поиска и интеграции информации из распределенных источников знаний представляются актуальными на сегодняшний день [1–3]. Одной из основных функций управления знаниями в современных ИИС для широкого спектра проблемных областей является семантический поиск элементов знаний, имеющих распределенный и, как следствие, гетерогенный характер представления [3–5]. Актуальность и важность проблематики управления знаниями осознается большинством специалистов, занимающихся корпоративным управлением и IT-технологиями создания систем управления знаниями (СУЗ) ИИС [4, 6]. Среди основных проблем управления знаниями можно выделить следующие [4–7]:

- интеграция знаний, представленных в различных проблемно-ориентированных ИИС (например, интеграция онтологий нескольких различных функциональных областей или распределенных ресурсов знаний с устойчивыми междисциплинарными связями);
- поиск противоречий между знаниями, которые отображены в контенте распределенных ресурсов и ограничивают интерпретацию терминов и их отношения друг с другом, оценка их достоверности и надежности;
- поиск знаний, необходимых конкретному пользователю для поддержки принятия решений, интеллектуального накопления и обработки знаний;

- автоматизация создания метаданных, корректно описывающих сведения о признаках и свойствах ресурсов знаний на семантическом уровне.

Семантический подход основан на использовании взаимосвязанного набора методов и технологий по работе со смыслом, семантикой данных, информацией и знаниями [7–10]. Выбор семантического подхода в качестве платформы для модели поиска и оценки семантической близости объектов знаний из распределенных ресурсов обусловлен его эффективностью в задачах поиска, аналитической обработки и интеграции знаний в СУЗ [1–4].

Комплексные исследования в области семантического поиска направлены на разработку методов оценки и сравнения мер семантической близости, используемых в широком спектре задач информационного поиска [5–7, 11]. При разработке модели поиска и вычисления семантической близости объектов знаний должны учитываться основные параметры формирования информационного запроса:

- выбор меры близости между понятиями онтологии — определение значимости составляющих: таксономической, реляционной и атрибутивной;
- определение весов (значимости) ключевых понятий, отношений и значений (порога) частоты встречаемости в метаописаниях;
- выбор типа функции близости между элементами онтологии.

В работах [4, 8] метаописания определены как структурированная информация, характеризующая содержание документов, информационных ресурсов и баз знаний. В метаописаниях выделяют три типа: системные (служебные) метаданные; структурные метаданные; семантические метаописания. В контексте рассматриваемой проблемы семантические метаописания представляют особый вид описаний знаний, включающий концептуальное (аннотированное) изложение содержания и смысла информации об объекте. Использование метаопи-

саний позволяет существенно повысить оперативность и эффективность семантического поиска необходимых объектов знаний и информации о них при формировании семантического поискового образа (профиля) запроса.

С использованием онтологического инжиниринга в работах [7, 12] предложен метод создания особого вида семантических метаданных — триплетов (t), представляющих собой декларативные наборы высказываний "субъект (s) — предикат (p) — объект (o)". Триплеты отражают основные парадигматические отношения понятий онтологии предметной области и семантику описываемых элементов знаний из распределенных ресурсов. В работах [6—8, 12] отмечается, что такие метаописания являются ценными источниками информации для выполнения семантического поиска и с их применением возможно значительное улучшение функциональности поисковых механизмов в ИИС. Оценкой близости между элементами знаний и запросом является числовое значение, которое выражает степень сходства между ними [13]. В данной работе факторами, определяющими выбор меры семантической близости, применяемой для формального представления (профиля) пользовательского запроса, предложены:

- выбор критериев, составляющих меру близости: таксономические отношения между понятиями — характеристики онтологических структур (длина пути, глубина иерархии и др.) и ассоциативные (горизонтальные) отношения, определяющие асимметричную меру семантической близости;
- выбор значимости критериев — коэффициентов важности в комбинированной метрике вычислительной семантики.

Существующие механизмы семантического поиска [5—8] базируются на методах и подходах онтологической концептуализации предметной области знаний. Для определения мер семантического сходства и связности элементов знаний предлагается модель вычисления семантической близости, использующая набор интерпретируемых таксономических и ассоциативных зависимостей метаописаний элементов знаний, представленных в онтологии соответствующей предметной области, заданной в виде семантической сети [14]. Релевантность знаний оценивается близостью к некоторой метрике оценки подобия понятий, входящих в семантическое описание (метаданные) элементов онтологии.

Для определения близости элементов онтологии по семантическим метаданным профиля пользователя предлагается алгоритм оценки семантической близости, основанный на эволюционных процедурах и операторах генетического поиска оптимальных решений, позволяющих исключать из рассмотрения неинформативные или незначимые описания элементов знаний, а также управ-

лять скоростью обучения с помощью задания порога близости [13—15].

Постановка задачи

Основными способами повышения эффективности использования знаний в ИИС являются их систематизация (упорядочение) и использование настроенных на эту систематизацию средств поиска объектов, содержащих требуемые знания в интеллектуальном пространстве. Модель многомерного интеллектуального пространства (пространства знаний) является признанной в разных областях науки абстракцией [4] и включает такие элементы, как:

- выбор системы координат;
- задание способа описания положения объектов в выбранной системе координат;
- задание метрики (способа вычисления) близости объектов в данном пространстве.

В соответствии с этим пространство знаний СУЗ предлагается описывать следующим образом:

- онтология предметной области представляет собой систему координат;
- описания объектов знаний задаются в виде их метаописаний на основе понятий онтологии предметной области;
- семантическая близость метаописаний используется в качестве метрики (меры семантического сходства и связности объектов).

В постановке задачи данной работы используются следующие условные обозначения компонентов онтологии: онтология O представляет собой знаковую систему $O = \langle C, E, R, T \rangle$, где C — множество понятий (элементов знаний); E — множество экземпляров понятий; T — множество предикатов — типов отношений; R — множество отношений, которые задают следующие виды связи между сущностями: таксономические, атрибутивные, квантифицирующие, логические и др.

Введем следующие правила и ограничения.

1. На основе онтологии предметной области O для каждого элемента знаний c_i множества $C = \{c_i\}$ созданы семантические метаописания $m(c_i) = \{t_1, t_2, \dots, t_{n(i)}\}$, где $n(i)$ — число триплетов в логическом представлении понятия c_i ; t_i — *RDF*-триплеты (Resource Description Framework) — кортежи вида $\langle s_i, p_i, o_i \rangle$, где s_i и o_i включены в объединение C_i и E_i , а p_i включен в R .

2. Каждый запрос q , данный пользователем из множества образов запросов Q , также состоит из множества триплетов $q = \{t_1, t_2, \dots, t_{n(q)}\}$, где $n(q)$ — число триплетов, содержащихся в запросе q .

Решаемая задача заключается в нахождении весовой функция w , которая определяет значимость любого триплета $t \in T$ (T — множество возможных триплетов) при описании элементов знаний c_i и запроса q : $0 \leq w(t, c_i) \leq 1$, где $t \in T$, $c_i \in C$, $0 \leq w(t, q) \leq 1$, где $t \in T$, $q \in Q$.

Для каждого запроса q требуется определить RES — результирующее подмножество элементов знаний, которое состоит из релевантных понятий для заданного запроса q из множества C . C_i считается релевантным заданному запросу q , если и только если оценка семантической близости между ними превышает некоторую пороговую величину семантического сходства. При этом для вычисления близости между элементами знаний и запросом используются их семантические метаописания [12].

Модель комбинированной меры оценки семантической близости

Ключевым моментом в решении задач семантического поиска является определение подмножества объектов знаний из различных функциональных областей с устойчивыми междисциплинарными связями и количественная оценка семантической близости элементов знаний для вычисления мер их семантического сходства и связности [15]. Существующие методы вычислительной семантики можно условно разделить на несколько категорий:

- меры, основанные на иерархических структурах, — методы оценки концептуальной таксономической близости, использующие различные метрики нахождения длины кратчайшего пути между вершинами графа онтологии предметной области [7, 16];
- меры, использующие неиерархические отношения, — методы оценки реляционной близости [13];
- меры, учитывающие значения атрибутов [8—10].

Недостатком большинства мер, основанных на онтологических структурах, является симметричность [12]. Актуальной задачей является определение семантической близости элементов онтологии, не связанных иерархически, однако имеющих определенную проблемно-ориентированную или междисциплинарную ("горизонтальную", ассоциативную) связь [7]. В связи с этим предлагается комбинированная мера, в которой учет структуры онтологии и семантики отношений позволяет вычислять меры семантического сходства и связности между элементами онтологии (понятия, экземпляры, связи-предикаты) [11—13]. Для определения степени семантической близости между триплетами запроса M_q и триплетами множества понятий M_c введем метрики расстояния между узлами онтологии и свойствами концептов и метрики плотности и информативности связанных концептов из O . Комбинированную меру близости можно представить следующей формулой:

$$SIM(M_q, M_c) = \sum_{i=1}^n w(t, q)_i Sim^i(c_1, c_i), \quad (1)$$

где Sim^i — мера близости по определенному критерию, вес $w(t, q)_i$ определяет относительную зна-

чимость критерия триплетов запроса, сумма весов равна 1, n — число критериев.

Для расчета Sim^i из (1) введем модификацию асимметричной меры близости [9] с учетом всех видов семантических отношений R , которая является подходящей для вычисления близости между компонентами триплетов. В предлагаемой модификации ребрам графа онтологии O в зависимости от направления прохождения (возможных видов отношений) придаются различные весовые коэффициенты [16]:

- для отношения "is-a" задаются два коэффициента g и s , определяющие близость в направлении обобщения и детализации;
- для отношения *instanceOf* (связывает понятие с метаданными понятием) задаются два параметра $\delta, \gamma \in [0, 1]$, которые соответственно выражают близость метаописания понятию и близость понятия экземпляру метаданных;
- коэффициенты близости для отношения *sameAs* (синонимы) и *invertOf* (обратные отношения) соответственно равны 1 и -1 ;
- для остальных семантических отношений r_i определяется весовой коэффициент ω , который выражает семантическую близость по этим отношениям.

Для $d_j = \{c_1, \dots, c_n\}$ — путь между элементами знаний c_1 и c_n (которые могут быть понятиями, экземплярами или предикатами) — определяются следующие характеристики [12]:

- $s(d)$ — коэффициент отношения детализации;
- $g(d)$ — коэффициент отношения обобщения;
- $ic(d)$ — коэффициент отношения иерархии от экземпляра до понятия;
- $c_i(d)$ — коэффициент отношения от понятия до экземпляра;
- $inv(d)$ — коэффициент инверсного отношения;
- $oth(d)$ — коэффициент остальных отношений.

Оценка близости между c_1 и c_2 по критерию i согласно пути d_j определяется по следующей формуле:

$$Sim^i(c_1, c_2) = \max_{j=1, \dots, m} \{(-1)^{inv(d_j)} s^{s(d_j)} g^{g(d_j)} \delta^{ic(d_j)} \gamma^{ci(d_j)} \omega^{oth(d_j)}\}, \quad (2)$$

где d_j — путь между вершинами c_1 и c_2 .

Для определения плотности и информативности тематически связанных элементов и их метаописаний введем вес понятия на основании частоты встречаемости. Считается, что вес понятия запроса q тем больше, чем больше связанных с ним понятий в метаописаниях $m(c_i)$, представленных триплетами $m(c_i) = \{t_1, t_2, \dots, t_{n(i)}\}$, где $n(i)$ — число триплетов в логическом представлении понятия c_i [13]. Вес понятия запроса определим как

$$w(t, q) = 1 + \ln \left(\varphi_{t, c_i} \left(1 + \sum_{c_i \in C} \varphi_{t, c_i} SIM(c_1, c_i) \right) \right), \quad (3)$$

где φ_{t, c_j} — коэффициент частоты встречаемости триплета запроса q в метаописании $m(c_j)$, данный коэффициент задается в алгоритме расчета меры близости, $SIM(c_1, c_j)$ — мера семантической близости метаописаний вершин-понятий множества C .

Генетический алгоритм вычисления меры семантической близости

Задачи семантического поиска знаний относятся к классу NP-трудных задач оптимизации и могут быть реализованы с помощью алгоритмов генетического поиска оптимальных решений [15, 17–19]. Наиболее перспективным подходом для определения качественной меры близости является автоматическое определение весовых коэффициентов с помощью генетического алгоритма (ГА). ГА основывается на использовании аналогов с эволюционными процессами репродукции, кроссингвера, мутации и естественного отбора [19].

В работе для повышения эффективности расчета меры семантической близости и определения семантически приоритетных объектов знаний для их представления в модели поиска предлагается ГА, который позволяет эффективно находить квазиоптимальные решения за полиномиальное время [18–20]. Обобщенная структура генетического поиска приведена на рис. 1.

Для определения оптимальных значений коэффициентов по методу максимизации оценки близости в ГА определена целевая функция (ЦФ) согласно (1)–(3):

$$F = \max(SIM(M_q, M_c)). \quad (4)$$

ГА расчета значений параметров модели для оценки семантической близости представлен на рис. 2.

На первом шаге работы ГА генерируются начальные параметры элементов модели близости (размер популяции и длина хромосомы) и вводятся значения весовых коэффициентов и вероятностей для операторов кроссингвера и мутации [19].

Далее формируется начальная популяция хромосом на основе имеющихся обучающих данных из множества $C = \{c_j\}$, для которых были созданы семантические метаописания $m(c_j) = \{t_1, t_2, \dots, t_{n(i)}\}$, где каждый элемент хромосомы (ген) является триплетом в логическом представлении понятия c_j . Для оценки приспособленности каждой хромосомы вычисляется значение ЦФ согласно (4).

Селекция хромосом осуществляется детерминированным способом с использованием элитарной стратегии и с частичной заменой наименее приспособленных хромосом на наиболее приспособленные при сохранении размера популяции [19–21].

Генерирование нового набора особей для каждой пары отобранных родительских хромосом проводится с использованием операторов кроссингвера и мутации с предварительно заданной вероятностью [14]. Скрещивание хромосом выполняется случайно с ве-

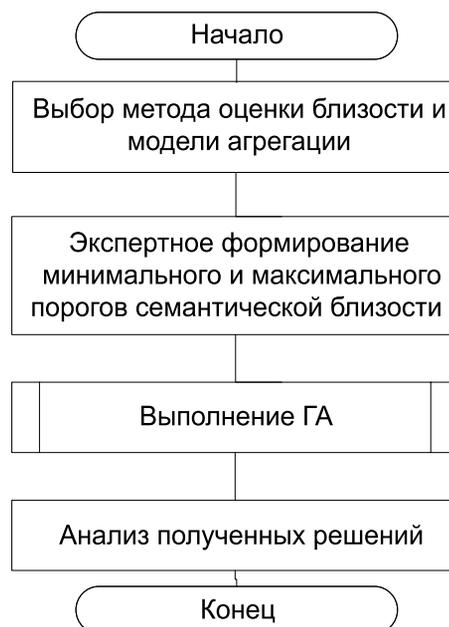


Рис. 1. Обобщенная архитектура генетического поиска

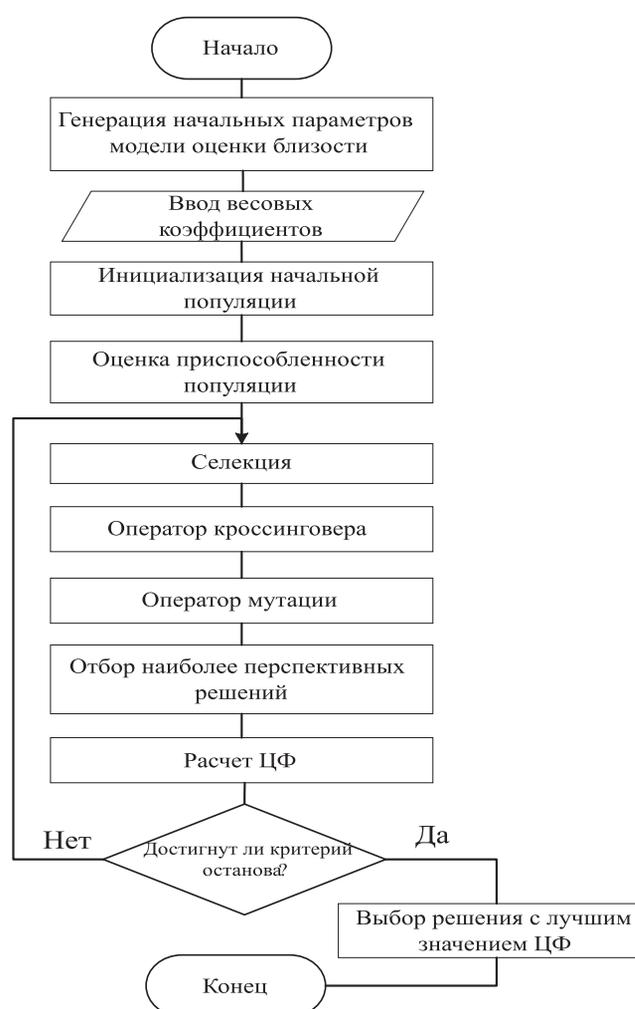


Рис. 2. Схема ГА расчета оценки близости

роятностью P_c . Точка скрещивания определяется случайным образом в заданном интервале.

Процедура мутации выполняется на популяции потомков, образованных в результате скрещивания, и заключается в изменении значения гена в хромосоме путем случайного выбора числа из интервала $[0,1]$ с вероятностью P_m .

Наиболее перспективные решения отбираются на основании вероятностей P_v , вычисленных для каждого индивидуума популяции с использованием метода пропорционального отбора [15, 19]. После оценки по формуле (4) качества каждой хромосомы в популяции и выбора лучшей из них принимается решение о продолжении эволюционной процедуры генерации следующего поколения либо о завершении процедуры обучения.

Чем больше значение ЦФ, тем выше качество хромосомы. Работа ГА завершается, если:

- функцией F достигнуто ожидаемое значение;
- выполнение заданного числа итераций (поколений) не приводит к улучшению уже достигнутого значения F ;
- истекло время, отведенное на решение задачи.

Преждевременная остановка работы ГА может произойти в случае вырождения популяции. Под вырождением понимается сокращение разнообразия хромосом, крайним проявлением вырождения является состояние популяции, в котором все ее члены имеют идентичные хромосомы [15]. В результате процесса искусственной эволюции, включающего селекцию, скрещивание, мутацию и отбор хромосом, качество решений в популяции постепенно улучшается.

Метод формирования итерационной модели семантического поиска

Анализ публикаций [6–8, 16] показывает, что в большинстве существующих механизмов формирования пользовательского запроса базы знаний

используются неявным для пользователя способом, не оставляя ему возможности самостоятельно влиять на выбор тех знаний, которые он считает важными для принятия решений. При семантическом поиске формируется информационная модель запроса пользователя, в которой отображаются [6]:

- персональные предпочтения пользователя, его знания и опыт;
- информационные потребности пользователя, особенности стоящей перед ним задачи и тех информационных объектов, обнаружение сведений о которых позволит ему решить эту задачу.

Предпочтения пользователя должны учитываться при разработке механизма поиска, позволяющего в автоматизированном режиме задавать основные параметры модели поиска [12]:

- меру близости между понятиями онтологии — определение значимости составляющих отношений: таксономической, реляционной, атрибутивной;
- вес (значимость) ключевых слов, понятий, отношений, триплетов, порог частоты встречаемости;
- критерии близости между метаданными элементов знаний и запроса.

В работе предлагается способ интерактивного взаимодействия пользователя при формировании запроса на основе трехэтапной модели расчета близости, обеспечивающий итеративную процедуру настройки следующих параметров модели семантического поиска:

- меры близости между понятиями онтологии (меры, основанные на иерархических отношениях, неиерархических отношениях, атрибутах; гибридные меры);
- меры вычисления частной близости (степень, в которую возводится значение семантической близости между понятиями; тип агрегации близостей между понятиями — максимум);
- меры вычисления близости между профилями (степень, в которую возводится значение семантической близости между частными близостями; тип агрегации частных близостей).

Пользователь настраивает агрегирующую функцию (ГА расчета семантического подобия) при предоставлении ему результатов запроса в каждой итерации, отображающих его предпочтения и ранжирование результатов поисковой процедуры. На рис. 3 представлен интерфейс настройки параметров модели и выбора мер вычисления семантической близости.

Предложенный способ формирования модели запроса позволяет решать задачи поиска по тематической близости с помощью формальных методов определения семантической бли-

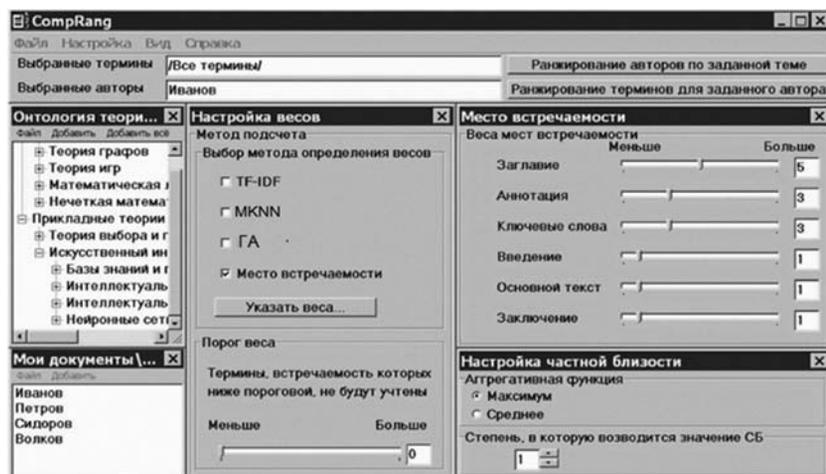


Рис. 3. Интерфейс настройки параметров итерационной модели поиска

зости элементов знаний, представленных в их метаописаниях.

Вычислительный эксперимент

В целях исследования производительности разработанного ГА были проведены вычислительные эксперименты. Для оценки разработанного алгоритма проведен сравнительный анализ с алгоритмом, основанным на мере подобия TF-IDF, который рассматривает каждый триплет как отдельное понятие, а для вычисления близости — косинусную меру [8] и алгоритмом MKNN (Mutual KNN) [6], поддерживающим метод вычисления близости отношений между ближайшими соседями — триплетами понятий и отношений.

Для проведения экспериментальных исследований работы предложенного ГА была разработана программная среда, позволяющая реализовать итеративную процедуру настройки параметров модели семантической близости рассматриваемыми алгоритмами.

Результаты экспериментальных исследований позволили определить зависимость времени работы алгоритмов от входных параметров модели близости: n — число хромосом в популяции; хромосома

представляет собой $m(c_i) = \{t_1, t_2, \dots, t_{n(i)}\}$, где $n(i)$ — число триплетов в логическом представлении понятия c_i . Графики зависимости времени работы разработанного алгоритма, а также TF-IDF и MKNN от количества входных данных модели оценки близости представлены на рис. 4. Временная сложность разработанного алгоритма ориентировочно составляет $O(n^2)$.

Проведена серия вычислительных экспериментов на полноту и точность извлечения релевантных понятий алгоритмами MKNN и ГА двумя описанными метриками близости с различным числом элементов онтологии — триплетов S_k , представленными их метаописаниями.

Полученные результаты свидетельствуют о практически линейном росте числа найденных понятий в зависимости от параметра k для обоих алгоритмов (рис. 5).

ГА извлекает большее число релевантных понятий с точки зрения полноты и точности запроса. Это происходит потому, что MKNN удаляет пары элементов, которые не являются взаимными соседями [12].

Во время поискового процесса ГА применяются правила сопоставления триплетов элементов онтологий триплетам запроса. Точность выполнения зависит от качества полученных на каждой итерации эффективных решений ГА, взвешивающего результаты определения критериев подобия элементов знаний в онтологии.

Заключение

Предложенный в статье подход представляет оригинальный механизм семантического поиска, использующий ГА для определения близости элементов онтологии по семантическим метаданным описания пользовательского запроса и имеет потенциал для дальнейшего совершенствования. Предложенный подход целесообразно применять для развития перспективных технологий семантического поиска, онтологического проектирования и интеграции знаний в ИИС, формирования семантически сконцентрированных знаний в условиях распределенности информации. Подводя итог, можно определить основные направления для внедрения предложенных решений, которые могут быть направлены на реализацию проектов:

- в области управления структурированной и неструктурированной информацией для обеспечения возможности эффективной работы с неструктурированной информацией и любыми формами контента;
- в области интеллектуального ведения и сопровождения бизнеса для поиска и извлечения ценной информации в целях принятия продуманных и качественных решений;

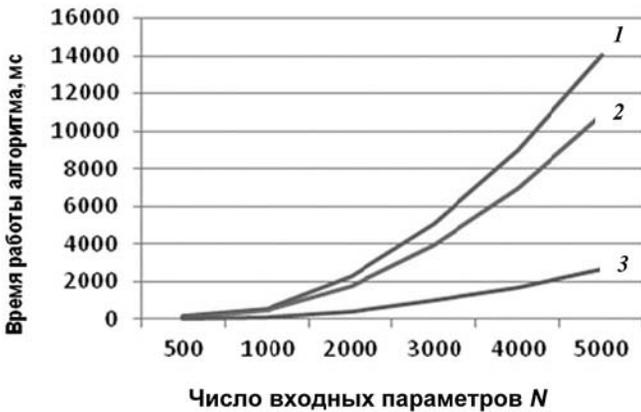


Рис. 4. Графики зависимости времени решения от числа входных параметров: 1 — MKNN, 2 — ГА, 3 — TF-IDF

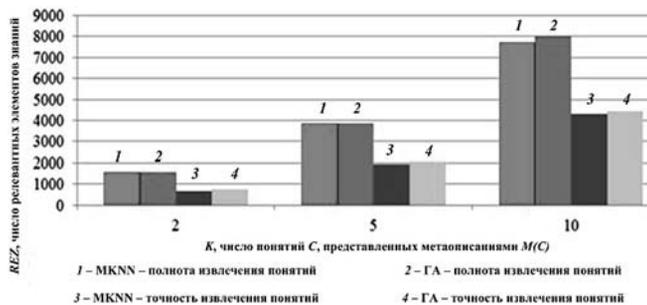


Рис. 5. Графики зависимости точности и полноты извлечения понятий алгоритмами MKNN и ГА

- в области интеграции данных для обеспечения интеграции распределенной информации в разнородных средах и системах.

Работа выполнена при поддержке гранта РФФИ № 17-07-00446.

Список литературы

1. **Bova V. V., Kureichik V. V., Legebokov A. A.** The integrated model of representation model of representation oriented knowledge in information systems // Proceeding of the 8th IEEE International Conference "Application of Information and Communication Technologies". 2014. P. 111–115.
2. **Бова В. В.** Концептуальная модель представления знаний при построении интеллектуальных информационных систем // Известия ЮФУ. 2014. № 7 (156). С. 109–117.
3. **Бова В. В.** Модель поиска и анализа решений для управления знаниями в интеллектуальных информационных системах // Известия ЮФУ. Технические науки. 2013. № 7 (144). С. 120–125.
4. **Тузовский А. Ф., Чириков С. В., Ямпольский В. З.** Системы управления знаниями (методы и технологии). Томск: Изд-во НТЛ, 2005. 60 с.
5. **Kravchenko Y. A., Kuliev E. V., Kursity I. O.** Information's semantic search, classification, structuring and integration objectives in the knowledge management context problems // Proceeding of the 10th IEEE International Conference on "Application of Information and Communication Technologies". 2016. P. 136–141.
6. **Castano S., Ferrara A., Montanelli S., Racca G.** Semantic information interoperability in open networked systems // Proceedings of the International Conference "SNW". 2004. P. 215–230.
7. **Kravchenko Y. A., Kursity I. O., Bova V. V.** Models for Supporting of Problem-Oriented Knowledge Search and Processing // Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry". 2016. Vol. 1. P. 287–297.
8. **Нгуен Б. Н., Тузовский А. Ф.** Обзор подходов семантического поиска // Известия Томского государственного университета систем управления и радиоэлектроники. 2010. № 2. С. 234–237.
9. **Крюков К. В., Панкова Л. А., Шипилина Л. Б.** Меры семантической близости в онтологиях // Проблемы управления. 2010. № 2. С. 2–14.
10. **Semenova A. V., Kureychik V. M.** Multi-objective particle swarm optimization for ontology alignment // Proceeding of the 10th International Conference on "Application of Information and Communication Technologies". 2016. P. 141–148.
11. **Кравченко Ю. А., Марков В. В., Новиков А. А.** Семантический поиск в semantic web // Известия ЮФУ. Технические науки. 2016. № 6 (179). С. 65–75.
12. **Нгуен Б. Н., Тузовский А. Ф.** Модель информационного поиска на основе семантических метаописаний // Управление большими системами. 2013. № 41. С. 51–92.
13. **Кравченко Ю. А., Курситыс И. О., Кулиев Э. В.** Разработка генетического алгоритма расчета семантической близости в задачах управления знаниями // Известия ЮФУ. Технические науки. 2016. № 6 (167). С. 75–88.
14. **Bova V. V., Zaporozhets D. Y., Kureichik V. V.** Integration and Processing of Problem-Oriented Knowledge Based on Evolutionary Procedures // Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry". 2016. Vol. 1. P. 239–251.
15. **Bova V. V., Kureichik V. V., Zaruba D. V.** Data and knowledge classification in intelligence informational systems by the evolutionary method // Proceeding of the 6th International Conference "Cloud System and Big Data Engineering (Confluence)". 2016. P. 6–11.
16. **Zhu H., Zhong J., Li J., Yu Y.** An approach for semantic search by matching RDF graphs // Proceedings LAIRS Conference. 2012. P. 450–454.
17. **Zaporozhets D. Y., Zaruba D. V., Kureichik V. V.** Hybrid bionic algorithms for solving problems of parametric optimization // World Applied Sciences Journal. 2013. N. 23. P. 1032–1036.
18. **Бова В. В.** Подход к созданию модели управления знаниями на основе методов генетического поиска // Известия ЮФУ. Технические науки. 2016. № 6 (167). С. 54–65.
19. **Гладков Л. А., Курейчик В. В., Курейчик В. М.** Генетические алгоритмы. М.: Физматлит, 2010. 368 с.
20. **Rodzin S., Rodzina L.** Theory of bioinspired search for optimal solutions and its application for the processing of problem-oriented knowledge // Proceeding of the 8th IEEE International Conference "Application of Information and Communication Technologies". 2014. P. 142–147.
21. **Bova V. V., Legebokov A. A., Gladkov L. A.** Problem-Oriented Algorithms of Solutions Search Based on the Methods of Swarm Intelligence // World Applied Sciences Journal. 2013. N. 27 (9). P. 1201–1205.

V. V. Kureichik, Professor, e-mail: vkur@sfedu.ru,

V. V. Bova, Assistant Professor, e-mail: vvbova@yandex.ru,

D. V. Leshchanov, Student, e-mail:leshok.dimkaa@yandex.ru,

Southern Federal University

Semantic Search Model for Knowledge Management Systems Based on Genetic Procedures

Nowadays, one of the main functions of knowledge management in modern intelligent information systems is the semantic search for knowledge elements that have a distributed character of representation. The article presents the innovative approach to the implementation of the semantic search model. Such approach is applicable to large volumes of distributed sources of knowledge. Authors propose a model of semantic search and evaluation of the proximity of knowledge elements in the ontology of the subject domain. Particularly, the ontology is defined as a semantic network. The relevance of knowledge is assessed in the way of the proximity to some metric of the similarity estimation of knowledge objects. The objects are included in the meta-descriptions of ontology elements in intelligent information systems. On the one hand, semantic meta-descriptions are considered as the sets of triplets "subject-predicate-object".

On the other hand, it is presented in terms of the ontology of the subject area of problem-oriented information systems, and in the terms of the search image of the user's query as well. Authors developed a complex model for calculating semantic proximity for calculating measures of semantic similarity and coherence of knowledge. The model uses a set of interpreted metrics of taxonomic and associative dependencies, objects in the meta-descriptions of the search query. The algorithm for evaluating semantic proximity is based on evolutionary procedures and operators of the genetic search for optimal solutions. Such procedures allow excluding non-informative or insignificant descriptions of knowledge elements from consideration, as well as control the speed of learning by setting the magnitude of the proximity threshold. The article considers a genetic algorithm based on the use of analogs with evolutionary reproduction processes, crossing-over, mutation, and natural selection. A series of experiments have been performed for analyzing of the developed algorithm. The obtained data have confirmed the theoretical significance and prospects of the proposed approach application and allowed to determine the optimal values of the parameters of the algorithm. The proposed approach is useful in the following tasks: development of advanced semantic search technologies, ontological design, and integration of knowledge to support the task of forming semantically concentrated knowledge in knowledge management systems in the conditions of information distribution.

Keywords: semantic similarity, ontology, semantic network, semantic meta-model, genetic algorithms, genetic operators, knowledge management systems

References

1. Bova V. V., Kureichik V. V., Legebokov A. A. The integrated model of representation model of representation oriented knowledge in information systems, *Proceeding of the 8th IEEE International Conference "Application of Information and Communication Technologies"*, 2014, pp. 111–115.
2. Bova V. V. Konceptualnaja model predstavljenija znanij pri postroenii intellektualnyh informacionnyh system (A conceptual model of representation of knowledge when building intelligent information systems), *Izvestija SFU*, 2014, no. 7 (156), pp. 109–117 (in Russian).
3. Bova V. V. Model poiska i analiza reshenij dlja upravlenija znanijami v intellektualnyh informacionnyh sistemah (Model search and analysis solutions for knowledge management in intelligent information systems), *Izvestija SFU*, 2013, no. 7 (144), pp. 120–125 (in Russian).
4. Tuzovskij A. F., Chirikov S. V., Jampolskij V. Z. Sistemy upravlenija znanijami (metody i tehnologii) (The knowledge management system (methods and technologies)), *Tomsk: Izd-vo NTL*, 2005, 60 p. (in Russian).
5. Kravchenko Y. A., Kuliev E. V., Kursityis I. O. Information's semantic search, classification, structuring and integration objectives in the knowledge management context problems, *Proceeding of the 10th IEEE International Conference on "Application of Information and Communication Technologies"*, 2016, pp. 136–141.
6. Castano S., Ferrara A., Montanelli S., Racca G. Semantic information interoperability in open networked systems, *Proceedings of the International Conference "SNW"*, 2004, pp. 215–230.
7. Kravchenko Y. A., Kursityis I. O., Bova V. V. Models for Supporting of Problem-Oriented Knowledge Search and Processing, *Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry"*, 2016, vol. 1, pp. 287–297 (in Russian).
8. Nguen B. H., Tuzovskij A. F. Obzor podhodov semanticheskogo poiska (Overview of approaches to semantic search), *Izvestija Tomskogo gosudarstvennogo universiteta sistem upravlenija i radioelektroniki*, 2010, no. 2, pp. 234–237 (in Russian).
9. Krjukov K. V., Pankova L. A., Shipilina L. B. Mery semanticheskoy blizosti v ontologijah (Measures of semantic closeness in the ontology), *Problemy upravlenija*, 2010, no. 2, pp. 2–14 (in Russian).
10. Semenova A. V., Kureychik V. M. Multi-objective particle swarm optimization for ontology alignment, *Proceeding of the 10th International Conference on "Application of Information and Communication Technologies"*, 2016, pp. 141–148.
11. Kravchenko Y. A., Markov V. V., Novikov A. A. Semanticheskij poisk v semantic web (Semantic search in semantic web), *Izvestija SFU*, 2016, no. 6 (179), pp. 65–75 (in Russian).
12. Nguen B. H., Tuzovskij A. F. Model informacionnogo poiska na osnove semanticheskikh metaopisanij (Model of information retrieval based on semantic meta descriptions), *Upravlenie bolshimi sistemami*, 2013, no. 41, pp. 51–92 (in Russian).
13. Kravchenko Y. A., Kuliev E. V., Kursityis I. O. Razrabotka geneticheskogo algoritma rascheta semanticheskoy blizosti v zadachah upravlenija znanijami (The development of genetic algorithm for the calculation of the semantic proximity in the knowledge management tasks), *Izvestija SFU*, 2016, no. 6 (167), pp. 75–88 (in Russian).
14. Bova V. V., Zaporozhets D. Y., Kureichik V. V. Integration and Processing of Problem-Oriented Knowledge Based on Evolutionary Procedures, *Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry"*, 2016, vol. 1, pp. 239–251.
15. Bova V. V., Kureichik V. V., Zaruba D. V. Data and knowledge classification in intelligence informational systems by the evolutionary method, *Proceeding of the 6th International Conference "Cloud System and Big Data Engineering (Confluence)"*, 2016, pp. 6–11.
16. Zhu H., Zhong J., Li J., Yu Y. An approach for semantic search by matching RDF graphs, *Proceedings LAIRS Conference*, 2012, pp. 450–454.
17. Zaporozhets D. Y., Zaruba D. V., Kureichik V. V. Hybrid bi-ionic algorithms for solving problems of parametric optimization, *World Applied Sciences Journal*, 2013, no. 23, pp. 1032–1036.
18. Bova V. V. Podhod k sozdaniju modeli upravlenija znanijami na osnove metodov geneticheskogo poiska (The approach to creation of model of knowledge management based on the methods of genetic search), *Izvestija SFU*, 2016, no. 6 (167), pp. 54–65 (in Russian).
19. Gladkov L. A., Kureichik V. V., Kureichik V. M. Geneticheskie algoritmy (Genetic algorithms), *Moscow, Fizmatlit*, 2010, 368 p. (in Russian).
20. Rodzin S., Rodzina L. Theory of bioinspired search for optimal solutions and its application for the processing of problem-oriented knowledge, *Proceeding of the 8th IEEE International Conference "Application of Information and Communication Technologies"*, 2014, pp. 142–147.
21. Bova V. V., Legebokov A. A., Gladkov L. A. Problem-Oriented Algorithms of Solutions Search Based on the Methods of Swarm Intelligence, *World Applied Sciences Journal*, 2013, no. 27 (9), pp. 1201–1205.

УДК 004:378

Е. Н. Решетова, преподаватель, e-mail: ereshetova@hse.ru,
Национальный исследовательский университет "Высшая школа экономики" (НИУ ВШЭ)

Опыт преподавания учебной дисциплины "Групповая динамика и коммуникация в профессиональной практике программной инженерии"

Статья посвящена определению целей, задач и содержания обучения по профессионально-ориентированной дисциплине "Групповая динамика и коммуникация в профессиональной практике программной инженерии", которая входит в учебный план подготовки бакалавров по направлению "Программная инженерия". Предложен инновационный подход к преподаванию дисциплины, в основу которого положена командная разработка программных проектов.

Ключевые слова: программная инженерия, командный проект, групповая динамика, командная разработка

Введение

Во всем мире индустрия программного обеспечения становится ключевой движущей силой общественного и экономического развития. Можно утверждать, что мы живем в век программного обеспечения. Оно становится основным элементом, обеспечивающим людям требуемые возможности и качество жизни. Специалисты, знающие, как лучше всего разрабатывать программные системы, получают шанс изменить мир к лучшему. При подготовке таких специалистов должны быть представлены актуальные тенденции рынка программной разработки.

В процессе обучения специалистов по разработке программных средств привыкли использовать техническую метафору (программная инженерия) и инженерную модель образования. Помогает ли это профессионалам решать практические вопросы, находить выход из сложных ситуаций в реальных проектах? Не всегда. Проблема "мифического человеко-месяца" [1] жива до сих пор, ведь, рассуждая о разработке программного обеспечения, и в современном мире часто продолжают рассматривать людей как "заменимые компоненты для программирования" [2] и нанимать "группу Java-программистов" [3], забывая об истинной природе программирования — деятельности социальной и интеллектуальной. Мой личный опыт работы в софтверных компаниях сформировал убежденность в человекоцентричности процесса разработки программного обеспечения. Вслед за Алистэром Коуберном, одним из немногих методологов, который

все свои изыскания подкрепляет многолетним опытом работы среди программистов, повторяю: главное в любом проекте — люди, они решают все. Именно они могут вытащить проект из глубокого кризиса, и именно они создают проблемы, не желая и не умея договариваться, не выполняя правила и предписания. А потому в профессиональной практике программной инженерии ценятся не только сугубо технические компетенции, но определенные "нетехнические" знания и навыки.

Тема формирования "нетехнических" компетенций для технических специалистов тождественна активно развиваемой в образовательном сообществе теме формирования так называемых универсальных компетенций — надпрофессиональных умений и навыков, позволяющих повысить эффективность и важных для любого специалиста самых разных отраслей [4]. "Универсальные компетенции признаны необходимым условием успешной работы современного специалиста вне зависимости от отраслевой специализации" [5]. В "Атласе новых профессий" как результате проекта, инициированного Агентством стратегических инициатив (АСИ) и Московской школой управления СКОЛКОВО (масштабного исследования "Форсайт компетенций 2030"), представлены надпрофессиональные навыки и компетенции, такие как: системное мышление, клиентоориентированность, междотраслевая коммуникация, знание нескольких языков и разных типов культур, навыки управления проектами и людьми (от локальных групп до сообществ), навыки программирования, умение работать в режиме многозадачности и в постоянно меняющихся

условиях, навыки бережливого производства, навыки художественного творчества.

Задачи программы обучения современных специалистов на образовательном направлении "Программная инженерия" в НИУ ВШЭ построены с учетом:

- современных тенденций и требований рынка;
- рекомендаций международных стандартов:
 - SWECOM [6];
 - профессионального — SWEBOOK [7], отражающего общепринятое представление о том, что должен знать работник, имеющий степень бакалавра и четырехлетний опыт работы;
 - образовательного — рекомендаций Software Engineering 2014 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (SE2014) [8], содержащих руководящие принципы создания учебных планов по программной инженерии для студентов-выпускников.

В эти задачи наравне с сугубо техническими компетенциями входит развитие надпрофессиональных (межпредметных) компетенций, необходимых студентам для того, чтобы быстро интегрироваться в профессиональную деятельность сразу после выпуска и быть успешнее, чем студенты других альтернативных программ в нашей профессиональной сфере.

Цель и задачи дисциплины

Инженерия программного обеспечения — это область компьютерной науки, которая занимается построением программных систем, настолько больших и/или сложных, что для этого требуется участие команды или команд разработчиков. Дэвид Парнас [9] определил инженерию программного обеспечения как "коллективное проектирование многовариантного программного обеспечения". В нем точно ухватывается суть термина и подчеркивается различие между программированием и программной инженерией. Программист пишет программу целиком, а программный инженер создает программный компонент, который входит в программную систему, стыкуясь с компонентами, созданными другими такими же инженерами. Компонент, написанный одним разработчиком, может быть переписан или использован другими разработчиками для создания новых версий системы, даже если автор компонента давно покинул проект. Программирование — это дело, главным образом, индивидуальное, а программная инженерия — всегда коллективное. Мысли, сформулированные еще 80-х годах, напрямую откликаются в современном мире, где индустрия разработки коммерческого программного обеспечения практически полностью ориентирована на групповую (командную) разработку. Лучшие проекты по разработке

программных продуктов, сложной высокотехнологичной продукции или систем планируются и исполняются (от создания концепции, разработки и вывода на рынок) в результате объединенных усилий разнородной группы людей — проектной команды. Человек уже не способен в одиночку обновлять свои знания и навыки, чтобы соответствовать текущему моменту и использовать изменения. Он просто перестает удивляться и реактивно реагирует. Наиболее способными к проактивному существованию в VUCA [10] мире оказываются команды, создающие знания и навыки "на лету" в процессе выполнения работы. Адаптироваться к этим изменениям в одиночку — все время отставать и недоиспользовать открывающиеся возможности. Развитие группы с опорой на коллективные, лучше — командные точки силы — сильный современный концепт. В настоящее время, когда жизнеспособность организаций определяется их возможностью приспособиться к частым и непредвиденным изменениям, которые должны выполняться быстрее, чем у конкурентов, на специалистов программной инженерии ложится огромная ответственность за обеспечение качества разрабатываемых программных систем и предоставляемых ими сервисов. Только команда, действующая слаженно и эффективно, может обеспечить должное качество и своевременность разработки. Обучение навыкам организации высокопродуктивной командной деятельности заложено в дисциплине "Групповая динамика и коммуникации в профессиональной практике программной инженерии", изучаемой на втором курсе бакалавриата.

Что является основной целью дисциплины? Формирование у будущих специалистов в области программной инженерии системного представления о коллективной разработке, состоянии, механизмах и основах продуктивной деятельности проектных команд в жизненном цикле разработки программного продукта.

Основными задачами курса являются:

- изучение основополагающих теоретических положений о совокупности возникающих в группах разработчиков программного обеспечения процессов и явлений, затрагивающих групповую динамику, и особенностей межличностного и межгруппового общения в процессе разработки программного обеспечения;
- изучение методов работы в группе, представление о практических вопросах и проблемах, возникающих в процессе разработки продуктов в команде;
- получение практических навыков применения методов управления групповой динамикой и коммуникацией в жизненном цикле программного проекта и проектных командах профессионального взаимодействия.

В контексте социально-психологической подготовки студентов мы рассматриваем пять основных элементов групповой динамики:

- цели команды;
- нормы команды;
- структура команды и проблема лидерства;
- сплоченность команды;
- стадии развития команды.

Одними из самых заметных феноменов, которые студенты должны "прожить" на себе, являются смена динамики группы и появление эффекта сотрудничества, или полезного взаимодействия. "Взаимоотношения людей влияют на их продуктивность", "по отдельности мы обладаем определенными навыками и умениями, однако при объединении для работы в команде возникают новые свойства и специфические явления" — так говорят студенты. Высокая или низкая производительность может рассматриваться как эмерджентное свойство группы людей, работающих вместе особым образом. "Команда с высоким коллективным интеллектом знает, что всякая великая команда — это процесс, а не вещь в себе. Понятно, что каждый участник также является процессом, а не статичным объектом. Характеристики личности, индивидуальные soft skills — это генеральные руководства, но каждый стремится развиваться и меняться к лучшему. Участники понимают, что поведение в значительной степени определяется ситуацией, и осмысленно создают такие, в которых сотрудничество становится легким и плодотворным. Вне зависимости от типа личности участника всегда есть возможность для расширения сотрудничества" [11].

Следующий важный элемент опыта в команде — задачи коммуникационных связей между членами команды. Коммуникация — это не просто языковое, письменное или индивидуальное общение, а взаимосвязанные акты передачи и восприятия информации во взаимодействии специалистов в профессиональной практике программной инженерии, основа структуры группы, показатель физической и социальной близости. Очевидно, что не все виды получения и распределения информации между членами команды могут приводить к эффективной работе. Многие из них отрицательно сказываются на ее сплоченности и возможности совместных действий. Это должно учитываться при управлении деятельностью группы. Эффективная коммуникация быстрее приводит к индивидуальной удовлетворенности деятельностью группы, позволяет ее членам наиболее полно выразить себя. Правильная организация коммуникации может способствовать, с одной стороны, повышению производительности, а с другой — достижению удовлетворенности в групповом взаимодействии.

Таким образом, ключевыми для всех команд являются вопросы, как "складывать" индивидуаль-

ные навыки, какие создавать условия, как организовывать процесс, чтобы команда становилась высокопроизводительной? Как использовать различные способы коммуникации, позволяющие эффективнее решать практические проблемы в зависимости от ситуации и целей команды? В "Групповой динамике и коммуникации в профессиональной практике программной инженерии" в "живой" групповой работе рождаются ответы на эти вопросы.

Принципы реализации учебной дисциплины и описание процесса

Учебный поток делится на небольшие по численности команды (не более 4—5 человек), которые будут, работая вместе, развивать идею и создавать программный продукт. Каждая команда проходит через определенные "вехи" (milestones) — ключевые точки, характеризующие достижение какого-либо существенного результата. Этот результат может быть оценен и проанализирован, что подразумевает ответ на вопрос: "А достигли ли мы целей, поставленных на этом шаге?". Методически мы выделяем девять обязательных встреч команды.

На первом стартовом собрании (Kick-off—встреча) команда инициирует проект. Происходит оценка существующей ситуации, разработка видения программного продукта, концепция решения и оценка возможности его реализации. Определяется состав команды. Обсуждаются цели и структура проекта. Kick-off — встреча, на которой в реальном мире обычно встречаются с высшим руководством, заказчиками, маркетологами и другими ключевыми стейкхолдерами. Руководство сообщает команде о проекте, причинах возникновения, необходимости, целях управления проектом. Представитель маркетинга объясняет маркетинговую потребность в продукте, важные конкурентные преимущества и особые соображения клиента, которые команда должна знать. В нашем курсе некоторые команды встречаются с реальными заказчиками. Некоторые команды генерируют идеи проектов самостоятельно. Цель первой встречи — информировать всех членов команды о работе, о целях для команды и убедить членов команды в важности проекта.

На второй встрече происходит "запуск" команды: команда формулирует цели. Задача команды — договориться о процессе и результатах работы, коммуникации, ролях, установить единые прозрачные и четкие правила работы. Команда разрабатывает Устав проектной команды — неформальный документ ("базовые правила"), понятный и принимаемый всеми членами команды, разъясняющий пять "К": кто, что, когда/по наступлении какого события, в какой форме, кому предоставляет информацию и отчетность по основным моментам деятельности. Также команда распределяет роли членов команды. Стандартные роли команды — лидер

группы, разработчики, UX-разработчик, аналитик, тестировщик. Другие возможные роли могут быть присвоены по мере необходимости. Студенты — участники команды должны понять и договориться, что в современном проекте вполне нормально (и ожидаемо), что человек будет одновременно играть несколько ролей. Иными словами, пусть аналитик знает, что разработчик имеет полное право беседовать с клиентом (это даже приветствуется). Пусть тестировщики не удивляются тому, что разработчику придется написать немало автоматизированных тестов. А если в проекте не будет специального разработчика пользовательских интерфейсов, это не означает, что никто не собирается заниматься юзабилити и дизайном. Разумеется, все это будет сделано, только другими людьми, которые исполняют в проекте сразу несколько ролей. Команда обязательно выявляет поддающиеся оценке цели и задачи.

На третьей и четвертой встречах команда составляет общую стратегию проекта и план. Студенты создают концептуальный дизайн, разрабатывают стратегию развития, определяют подробный процесс, инструменты и средства поддержки, которые им понадобятся. Они достаточно много знают о продукте, который будут производить, оценивают его размер и время, необходимое для каждого шага процесса. Студенты прогнозируют трудозатраты для выполнения каждой задачи: оценивают число часов, которые каждый член команды будет тратить на проект каждую неделю. Из оценки задач и еженедельных часов команда создает график — по-недельное расписание, в котором указано, какие задачи должны быть выполнены, и время, которое на это отводится.

После того как команда составит общий план, она формирует план контроля качества — на пятой встрече. В этом плане определяются измеримая основа для отслеживания качества работы и действия, которые команда планирует совершить для достижения качества. При составлении плана качества члены группы оценивают число дефектов, которые они будут вводить и удалять на каждой фазе: сколько недостатков будет оставлено для тестирования системы, приемочных испытаний клиентов и окончательного тестирования.

Шестая встреча посвящена подробным планам следующего этапа (разработки). Просматривая всю рабочую нагрузку, команда гарантирует равномерность распределения задач среди участников. В результате получается так называемый сбалансированный командный план.

Во время седьмой встречи команда определяет основные риски проекта и оценивает их по вероятности и воздействию. Команда также назначает члена команды для отслеживания каждого риска и готовит план мероприятий по смягчению последствий для наиболее значительных рисков.

Восьмая встреча посвящена обсуждению вопросов управления проектом.

На девятой встрече команда проводит обзор с руководством (ключевыми стейкхолдерами, присутствовавшими на первой встрече). Задача команды — представить промежуточный результат и получить рекомендации по развитию продукта, его функциональности и пр. Содержание командной презентации может быть следующим:

- Название команды. Название проекта.
 - Vision/концепция продукта:
 - Кто будет использовать продукт? Кто целевая аудитория?
 - Какие нужды пользователей продукт удовлетворяет? Какую пользу приносит продукт? Как он облегчает жизнь пользователям?
 - Описание рынка продукта. Как эти проблемы решаются в настоящее время?
 - Каковы критические атрибуты, чтобы удовлетворить эти нужды и сделать продукт успешным?
 - Чем продукт похож на существующие аналоги и чем он отличается? Сравнительная характеристика конкурирующих продуктов по параметрам.
 - Описание продукта команды. Достоинства продукта. Анализ востребованности продукта.
 - Цели проектной команды и задачи проекта.
 - Коммерческое обоснование. Способы монетизации.
 - Технические вопросы:
 - Визуализированная архитектура решения. Внутренняя структура (карта) продукта, из каких блоков он состоит и какой информацией должны наполняться блоки.
 - Как предполагается масштабировать.
 - Управление требованиями. Бэк-трекинг.
 - Главные препятствия и риски реализации проекта. Мероприятия по их устранению.
 - Имена членов команды, основные принципы команды, роли, ответственность.
 - Распределение работ, план-график.
- Команда демонстрирует свой подготовленный план, описывает, как он был подготовлен, и демонстрирует, что все члены соглашаются и с этим планом. В финале присутствующие стейкхолдеры должны согласиться с тем, что представленный командой материал отвечает потребностям бизнеса, скорректировать в случае надобности и разрешить продолжить проект.

Технология построения команды

Команды проходят типичные стадии развития и сталкиваются с определенными проблемами, опасностями и возможностями, характеризующими те или иные стадии. Проживание стадий — обязательный процесс для каждой команды, каким бы

образом он ни проходил — сознательно, бессознательно, пассивно, активно. Смысловой процесс не состоится, если студенты не смогут последовательно пройти через различные стадии развития команды и спроектировать три важных момента внутренней динамики, образующей базовую систему командной работы. Первый момент касается задачи, которая стоит перед командой. Второй момент касается людей, из которых команда состоит. Третий момент касается взаимоотношений членов группы [12].

Фактически апогей командной работы — это достижение цели, заключающейся в конструктивной совместной деятельности, при взаимодействии данных трех аспектов. Командное развитие состоит в понимании, наполнении смыслом, проживании, развитии:

- аспекта личностного развития и опыта;
- содержательного аспекта;
- командного аспекта.

Что конкретно означают эти три аспекта?

Для каждой из стадий командного развития мы можем задать определенные вопросы и получить направляющие ответы.

- Что испытывают отдельные члены группы? Какие вопросы их волнуют?
- Какие типичные процессы протекают в группе? Что характеризует эту стадию?
- Какую позицию занимают отдельные участники или группа по отношению к задаче/цели проекта?
- Какие специфические особенности присущи группе в каждой стадии?
- Какие важные шаги должна предпринять группа, чтобы конструктивно продвинуться в своем развитии? Какие шаги в отношениях и шаги по отношению к задаче?
- Какую роль играет лидер команды на каждой стадии и как он должен реагировать на происходящее или его предупреждать?
- Какие виды групповой деятельности в случае необходимости нужно модерировать, чтобы ускорить процесс командного развития?

Используемая модель развития команды является производной от классических пяти стадий группового развития, которые были определены еще в 1950—1960 гг. и которые можно также встретить в более поздней литературе по групповой динамике [13—18]:

- ориентировочная стадия;
- стадия бурления, столкновения и позиционирования;
- стадия экспериментирования и структурирования;
- стадия конструктивной совместной деятельности;
- стадия прощания и расформирования.

В ходе выполнения командного проекта студенты проходят через стадии развития команды, проживают процессы и явления, затрагивающие груп-

повую динамику из роли рядового участника или лидера, понимают особенности межличностного и внутригруппового общения, практикуют механизмы управления группой, совершая развивающие мероприятия по отношению к задаче и к участникам команды.

Характеристика стадий является методологической основой для понимания и изучения происходящего в команде и проведения мероприятий с командой. В качестве примера рассмотрим ориентировочную стадию развития команды и мероприятия, включаемые в учебный процесс.

Характеристика ориентировочной стадии.

В команде доминирует сильная неуверенность. Участники сталкиваются со многими неизвестными аспектами. Ведут себя в отношении других скорее подчеркнуто вежливо. Коммуникация состоит из множества ритуалов, так как в этом случае реакция предсказуема. Например, на вопрос: "Как у вас дела?", в ответ можно услышать: "Спасибо, хорошо". Люди демонстрируют минимальную степень участия и подтверждают свое желание установить контакты, не особенно углубляясь. Единой группы пока не существует. Участники команды еще не считают себя связанными друг с другом, и, как правило, имеется сильная центробежная тенденция: прочь от общего плена в направлении небольших группок или пар. Многие с интересом ожидают какую-то новую информацию друг о друге. Поставленная задача лишь в самых общих чертах известна членам команды — как тема проекта или общее намерение.

Вопросы членов команды на ориентировочной стадии:

- Как это получится?
- Что представляют собой другие члены команды?
- Что происходит?
- Чего от меня ожидают?
- Куда мы движемся и почему?
- Кто здесь лидер?
- Как конкретно звучит задача?
- Какие у нас цели?
- Как я вписываюсь в команду?
- Каков объем предстоящей работы?
- Сбудутся ли мои ожидания?
- Каковы представления об этом у других?

Шаги развития. Важнейшим шагом развития на данной стадии является раскрытие. Нужно позаботиться о том, чтобы отдельные члены команды повернулись лицом к другим, проявили себя и что-то узнали о коллегах. Участникам необходимо раскрыться для темы. Важно сформировать у них готовность к чему-то новому, что, возможно, окажется не таким, как они пока себе представляют. И это должно происходить без страха. Последний пункт касается личной работоспособности по отноше-

нию к групповой задаче. Для этого требуются, прежде всего, информация, серьезные межличностные отношения и договоренности о временных правилах игры. Это придает уверенность и структурированность, снимает страх и предоставляет первую возможность проявить себя на деле.

Действия в области решения задач:

- Определение и донесение предпосылок создания команды: цели, ближайшие задачи, ограничения.
- Установление первичных норм и правил командной работы.
- Регулировка и первичное распределение обязанностей.
- Стимулирование распределения и поддержания ролей.
- Концентрация внимания команды на совместном планировании и организации труда.

Действия в области отношений:

- Знакомство членов команды друг с другом.
- Создание обстановки взаимопонимания и доброжелательности.
- Наблюдение за каждым членом команды, помощь в поиске своего места в команде и в демонстрации профессионального потенциала.
- Организация обучения с выработкой командных норм и правил сосуществования и поведения.

Командные мероприятия. У участников имеется ярко выраженная потребность в структуре. Им необходимы ориентиры и уверенность, чтобы оформиться в команду и развиваться в направлении взаимодоверия. А потому командные мероприятия на этой стадии преследуют следующие цели:

- предоставить объективную информацию;
- обеспечить четкую структуру (рамочные условия, временную структуру, последовательность);
- разъяснить контекст и цель совместной задачи;
- продемонстрировать профессиональную компетенцию и личную мотивацию участников;
- сделать прозрачными личность и стиль лидера;
- дать в команде свободу шутке, мотивации и уверенности.

Стартовое собрание команды (Kick-off — встреча), вторая встреча команды (см. выше раздел "Принципы реализации и описание процесса") являются средой для достижения этих целей. Ориентировочную стадию, как правило, завершают конкретизацией Устава команды.

Дискуссионные вопросы и ограничения

Первый вопрос — о длительности курса. Какой по длительности? С одной стороны, достаточно большой объем изучаемого материала из-за междисциплинарности с трудом вмещается в один семестр. С другой стороны, в силу характерной для студенческого сообщества динамики сложно поддерживать стабильный состав группы дольше, чем один семестр, — а нам важно сохранить команды,

сформированные с начала работы. В случае курса длительностью более 1 семестра (у нас курс длится 1,5 семестра — три модуля) команда должна учитывать дополнительные риски — возможного выбывания членов команды после зимней сессии.

Второй вопрос связан с квалификацией преподавателя дисциплины. Какими компетенциями он должен обладать? С одной стороны, преподаватель должен уметь формировать и развивать группы: отбирать членов группы; распределять роли; проводить ретроспективы, давать конструктивную обратную связь и оценивать успехи, т. е. обеспечивать содержательное консультирование и эффективное руководство командой. С другой стороны, преподаватель должен обучать корректному использованию инженерных методов, таких как оценка и планирование проектов, анализ требований и спецификации, высокоуровневому и детальному проектированию. В третьих, содержание должно затрагивать практические вопросы реализации проекта, например, планирование задач и контроль их выполнения, оценку качества и слежение за ним, поэтапную разработку и руководство ею. Преподаватель также должен не просто читать лекции, а вести практическое обучение. При этом не только объяснять методы и процедуры выполнения поставленных задач, но также контролировать и анализировать работу группы и производительность отдельных ее членов, помогая добиваться большей эффективности. К сожалению, обеспечить качественное практическое руководство во всех этих областях трудно. При отсутствии такого руководства студенческие проекты могут превратиться в иллюстрацию того, как не нужно разрабатывать программные продукты.

Третий вопрос связан с возможностью обеспечения четкой дисциплины. Это необходимо — нужно обеспечить дисциплину процесса (поэтапная разработка, сбор данных, стандарты на документацию, анализ и проверка и т.д.). Но насколько это возможно в учебном процессе? Но только принципиальное позиционирование преподавателем четкой дисциплины, формирование мотивации и помощи студентам в осознании необходимости дисциплины приведут группу к четкому рабочему процессу. Любое попустительство и согласие с критическим отношением к дисциплине могут привести к тому, что группа потеряет веру в адекватность самого процесса, и в результате сложится хаотическая среда, где каждый делает то, что хочет.

Анализ полученных результатов

Дисциплина "Групповая динамика и коммуникация в профессиональной практике программной инженерии" с опытом создания командных проектов была проведена в первый раз.

По окончании курса среди студентов проводился опрос. Студенты, особенно те, кто принимал учас-

тие в предыдущих групповых проектах, в основном положительно оценили полученный опыт. Около 75 % высоко оценили курс, и свыше 85 % считают, что работа над групповым проектом была весьма полезной. В подавляющем большинстве студенты выше всего оценили четкое определение ролей и процесс, который описывает задачи для каждого этапа проекта. Самое сильное недовольство вызвал объем необходимой бумажной работы. Хотя большинство студентов подтвердили, что документация, сбор и анализ данных крайне важны, особого восторга все это у них не вызвало.

Вот некоторые отзывы студентов о дисциплине:

- Дает очень хороший материал для того, чтобы помочь студентам формировать группу: определение ролей, четкие обязательства, организация взаимодействия друг с другом, планирование и т. д.
- Хорошее средство обучения разработке проектов при условии, что студенты хорошо владеют "персональной разработкой". Предоставляет множество полезных методик, которые можно взять, опробовать и перенести из процесса обучения на свою реальную работу.

Наиболее положительную оценку новой программе давали студенты, которые на первом курсе серьезно овладевали персональным процессом разработки программного обеспечения [19] и научились:

- учитывать время, затраченное на работу над проектом;
- учитывать найденные дефекты;
- классифицировать типы дефектов;
- оценивать размер задачи;
- осуществлять систематический подход к описанию результатов тестирования;
- планировать программные задачи;
- распределять их по времени и составлять график работы;
- выполнять индивидуальную проверку проекта и архитектуры;
- осуществлять индивидуальную проверку кода;
- выполнять регрессионное тестирование.

Студенты, интегрировавшие в персональный процесс "кухню" качественной разработки, смогли и в командной разработке сохранить и приумножить класс работы и прочувствовать синергетический эффект от работы в группе или, наоборот, регрессию. Некоторые студенты, уже мнящие себя программистами и никогда не сталкивавшиеся с проблемами управления и обеспечения качества, демонстрировали определенный скепсис и недовольство ответственностью и за эти вопросы. Самое серьезное недовольство в курсе вызвали процессы сбора требований, планирования и управления качеством. По мнению студентов, решать эти вопросы должен менеджер.

Все команды (а всего была создана 21 команда) смогли пройти по ключевым вехам процесса разработки и в том или ином виде к концу курса представили прототип продукта. При этом восемь команд не смогли пройти весь жизненный цикл развития команды — кто-то "застрял" на третьей стадии, а кто-то вернулся на вторую.

Студенты поделились "прожитыми" выводами-условиями, необходимыми для эффективной работы в команде, подчеркнув, что "теоретически эти условия не очевидны" и что "курс обеспечивает явное руководство — как построить эффективные технические команды". Участники команд отметили, что действительно поняли важные вещи об успешных командах, а именно:

- чтобы сделать то, что они предполагают сделать, нужно договориться, как сделать работу, и понимать, что их план достижим;
- члены команды устанавливают общие цели и определенные роли;
- команда развивается по согласованной стратегии;
- все члены команды участвуют в разработке плана, и каждый участник знает его и понимает личную роль;
- члены команды определяют персональный фронт работы, ориентируясь на общий процесс;
- члены команды делают работу способом, которым они запланировали сделать это;
- члены команды общаются свободно и часто;
- команда формирует связную группу: участники сотрудничают и все преданы удовлетворению цели;
- участники знают свой статус, получают обратную связь о своей работе;
- лидер и все участники команды несут ответственность за поддержание открытого и эффективного командного общения. Когда члены команды не знают статус проекта, не понимают, что их товарищи по команде делают, или не знают, какие задачи стоят впереди, им трудно оставаться мотивированными. Общее информационное поле является ключевой частью поддержания энергии команды;
- участники команды могут брать на себя роль лидера;
- опыт проигрыша при качественном анализе ситуации дает больше чем равномерное достижение целей.

Заключение

Реализация дисциплины "Групповая динамика и коммуникация в профессиональной практике программной инженерии" с ведением "живых" командных проектов оказалась делом довольно сложным и трудоемким. Тем не менее, 21 студенческая команда, состоящая из студентов 2-го курса бакалавриата по направлению "Программная инже-

рия", инициировала и реализовала 21 проект в рамках курса.

Какие проекты были реализованы? Различные по степени сложности, новизне используемых технологий, трудоемкости, предметной области. Например:

- Telegram-бот для поиска свободных аудиторий в учебных корпусах НИУ ВШЭ;
- Информационная фармацевтическая система;
- Агрегатор медицинских учреждений Москвы, проводящих лабораторные исследования.

Четыре команды на Межвузовской научно-технической конференции студентов, аспирантов и молодых специалистов им. Е. В. Арменского рассказали о работе над своими проектами:

- AIMS — клиент-серверное приложение-мотиватор для Android.
- Кроссплатформенный сервис удобного чтения "Read-out".
- Квест-гид для жителей Москвы "Welcome to Moscow".
- Факторозависимый персонализированный советчик досуга.

У четырех команд были реальные заказчики:

- Основным стекхолдером проекта "Мобильное приложение компании "Подворье"" был генеральный директор компании.
- Проект "Система для проверки работ по программированию" курировал преподаватель департамента программной инженерии.
- Постановку задачи в проекте "Формирование расписания занятий на основе данных из РУЗ ВШЭ" осуществлял руководитель Департамента информационных технологий НИУ ВШЭ.
- Проект по созданию "Программы со статистическими данными по загруженности аудиторий в НИУ ВШЭ" был реальной задачей для администрации университета.

Важным индикатором, говорящим о достижениях запланированных в курсе целей, является, на мой взгляд, активизация командной проектной деятельности вне рамок дисциплины. Так, впервые на образовательном направлении "Программная инженерия" были инициированы и реализованы командные курсовые работы (6 работ). Интересно, что некоторые команды выбрали и сформулировали более сложные, объемные, связанные с изучением новых технологий, интересные работы, чем при изучении дисциплины. Например:

- Сервис организации экспертной деятельности на базе блокчейн платформы Ethereum;
- Сервис анализа загруженности аудиторного фонда НИУ ВШЭ.

Студенты активней стали участвовать в хакатонах, при этом не только генерируя идеи, а доводя проекты до реализации. Так, команда с проектом "Программа "Weddingchain" для регистрации брака

с использованием технологии блокчейн" заняла 2-е место на хакатоне Сбербанка весной 2017 г.

Курс, ориентированный на реализацию командных проектов, позволил заинтересованным студентам без лишних административных проволочек "окунуться", находясь в образовательном процессе, в тему блокчейн-технологий — актуальнейшую тему, формирующую, в том числе, поле цифровой экономики [20], и не только изучить основные принципы блокчейна, но и попробовать реализовать технологию на практике. Эти студенты в новом учебном году станут менторами для всех желающих приобщиться и разобраться в блокчейне.

Весной 2017 г. бакалаврская программа по направлению "Программная инженерия" прошла профессионально-общественную аккредитацию АПКИТ — профессиональной ассоциации, объединяющей крупнейшие IT-компании России [21]. Аккредитация, выданная на шесть лет, подтвердила, что мы обеспечиваем нужный бизнесу набор и высокий уровень компетенций, среди которых умение работать в команде, обеспечивая ее сплочение, партнерское взаимодействие с коллегами, лидерство, готовность брать ответственность за работу членов команды и за результат выполнения заданий, является одним из приоритетных. В связи с этим такое наполнение и такая реализация учебной дисциплины "Групповая динамика и коммуникации в профессиональной практике программной инженерии" являются правильным вектором в подготовке наших студентов, отвечает как требованиям профессиональных стандартов, так и современным требованиям рынка. Формирование умений командного взаимодействия уже в учебном процессе университета является важным условием повышения конкурентоспособности выпускника.

Список литературы

1. **Brooks Jr. F. P.** The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition. Boston: Addison-Wesley, Longman, Inc., 1995, 322 p.
2. **Cockburn A.** Characterizing People as Non-Linear, First-Order Components in Software Development // Alistair Cockburn [Official website]. URL: <http://alistair.cockburn.us/Characterizing+people+as+non-linear,+first-order+components+in+software+development> (дата обращения: 09.07.2017).
3. **Cockburn A.** Agile Software Development: The Cooperative Game (2nd Edition). Montgomery: Published by Addison-Wesley, Professional, 2006. 504 p.
4. **Решетова Е. Н.** Универсальные компетенции студента образовательного направления "Программная инженерия" // Информационные технологии. 2016. Т. 22. № 12. С. 938—945.
5. **Атлас новых профессий 2.0** / Под ред. Павла Лукши. М.: Олимп-Бизнес, 2016. 288 с.
6. **Software Engineering Competency Model** // IEEE. 2014. 168 p.
7. **Guide to the Software Engineering Body of Knowledge Version 3.0 SWEBOK**. // IEEE. 2014. 335 p.
8. **Software Engineering 2014. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering.** Joint Task Force on Computing Curricula // IEEE Computer Society, Association for Computing Machinery. 2015. 134 p.

9. **Parnas D. L.** Some software engineering principles // *Structured Analysis and Design, State of the Art Report*, INFOTECH International, 1978. P. 237–247.

10. **Nick Petrie.** *Future Trends in Leadership Development*. Center for Creative Leadership, 2014. 36 p.

11. **Джойс С.** Эффект муравейника. Успешная работа команды и коллективный разум. СПб: Символ-Плюс, 2010. 208 с.

12. **Томпсон Лей.** Создание команды. М.: Вершина, 2006. 544 с.

13. **Tannenbaum S. T., Beard R. L., Salas E.** Chapter 5. Team Building and its influence on team effectiveness: an examination of conceptual and empirical developments. // *Advances in Psychology*. Published by Elsevier B. V., 1992. Vol. 82. P. 117–153.

14. **Курт Левин.** Теория поля в социальных науках. СПб.: Сенсор, 2000. 368 с.

15. **William Scott.** Organization Theory: An Overview and Appraisal // *Academy of Management Journal*. 1961. Vol. 4. P. 7–26.

16. **Katzenbach J. R., Smith D. K.** *The wisdom of teams: creating the high performance organization*. New York: Harper Business, 1994. 84 p.

17. **Манфред Геллерт, Клаус Новак.** Все о командообразовании: руководство для тренеров. М.: Вершина, 2006. 352 с.

18. **Клаус Фопель.** Создание команды. Психологические игры и упражнения. М.: Генезис, 2003. 400 с.

19. **Watts S. Humphrey.** *The Personal Software ProcessSM (PSPSM)*. Pittsburgh: Carnegie Mellon University, 2000. 55 p.

20. **Заседание** Совета по стратегическому развитию и приоритетным проектам. URL: <http://kremlin.ru/events/president/news/54983> (дата обращения: 09.07.2017)

21. **Реестр** аккредитованных образовательных программ в области информационных технологий. URL: <http://it-poa.ru/reestr.html> (дата обращения: 09.07.2017)

E. N. Reshetova, Teacher, e-mail: ereshetova@hse.ru,
National Research University Higher School of Economics (HSE)

Teaching Experience the Academic Discipline "Group Dynamics and Communication in Professional Software Engineering Practice"

The article is devoted to defining goals, objectives and content of training on the professional-oriented discipline "Group dynamics and communication in professional software engineering practice", which is included in the curriculum of preparation of bachelors in the direction "Software engineering". We propose an innovative approach to teaching the discipline, which was based on team development of software projects.

Keywords: software engineering, team project, group dynamics, team development

References

1. **Brooks Jr. F. P.** *The Mythical Man-Month: Essays on Software Engineering*, Anniversary Edition. Boston, Addison-Wesley Int., 1995, 322 p.
2. **Cockburn A.** Characterizing People as Non-Linear, First-Order Components in Software Development, *Alistair Cockburn*. Accessed: [http://alistair.cockburn.us/Characterizing + people + as + non-linear, + first-order + components + in + software + development](http://alistair.cockburn.us/Characterizing+people+as+non-linear,+first-order+components+in+software+development) (accessed: 09.07.2017).
3. **Cockburn A.** *Agile Software Development*, The Cooperative Game (2nd Edition). Montgomery. Published by Addison-Wesley Professional, 2002, 504 p.
4. **Reshetova E. N.** Universalnie kompetenzii studenta obrazovatel'nogo napravleniya "Programnaya inzheneriya", *Informazionnie tehnologii*, 2016, vol. 22, no. 12, pp. 938–945 (in Russian).
5. **Atlas novih professiy 2.0.** Pod red. Pavla Lukshi, Moscow: Olimp-Biznes, 2016, 288 p. (in Russian).
6. **Software Engineering Competency Model**, *IEEE*, 2014, 168 p.
7. **Guide** to the Software Engineering Body of Knowledge Version 3.0 SWEBOOK, *IEEE*, 2014, 335 p.
8. **Software Engineering 2014.** Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Joint Task Force on Computing Curricula, *IEEE Computer Society, Association for Computing Machinery*. 2015, 134 p.
9. **Parnas D. L.** Some software engineering principles, *Structured Analysis and Design, State of the Art Report*, INFOTECH International, 1978. pp. 237–247.
10. **Nick Petrie.** *Future Trends in Leadership Development*. Center for Creative Leadership, 2014, 36 p.
11. **Dgois S.** *Effekt muraveinika. Uspeshnaya rabota komandi I kollektivnii razum*, Saint Peterburg, Simvol-Plyus, 2010, 208 p. (in Russian).
12. **Tompson Ley.** *Sozdaniye komandi*, Moscow, Vershina, 2006, 544 p. (in Russian).
13. **Tannenbaum S. T., Beard R. L., Salas E.** Team Building and its influence on team effectiveness: an examination of conceptual and empirical developments. *Advanced in Psychology*. Published by Elsevier B. V. 1992, vol. 820, pp. 120–131.
14. **Kurt Levin.** *Teoriya polya v sozialnykh naukakh*. Saint Peterburg, Sensor, 2000, 368 p. (in Russian).
15. **William Scott.** Organization Theory: An Overview and Appraisal, *Academy of Management Journal*, 1961, vol. 4, pp. 7–26.
16. **Katzenbach J. R., Smith D. K.** *The wisdom of teams: creating the high performance organization*, New York, Harper Business, 1994, 84 p.
17. **Manfred Gellert, Claus Novak.** *Vse o komandoobrasovanii: rukovodstvo dlya trenerov*, Moscow, Vershina, 2006, 352 p. (in Russian).
18. **Klaus Fopel.** *Sozdanie komandi. Psikhologicheskiye igry i upravleniya*, Moscow, Genesis, 2003, 400 p. (in Russian).
19. **Watts S. Humphrey.** *The Personal Software ProcessSM (PSPSM)*. Pittsburgh, Carjnegie Mellon University, 2000, 55 p.
20. **Zasedaniye** Soveta po strategicheskomu razvitiyu i prioritetnym proektam. accessed: <http://kremlin.ru/events/president/news/54983> (accessed: 09.07.2017). (in Russian).
21. **Reyestr** akkreditovannykh obrazovatel'nykh programm v oblasti informatsionnykh tekhnologiy // Podtverzheniye kachestva professional'nykh obrazovatel'nykh programm v oblasti informatsionnykh tekhnologiy (professional'no-obshchestvennaya akkreditatsiya, POA). accessed: <http://it-poa.ru/reestr.html> (accessed: 09.07.2017) (in Russian).

УДК 502.084, 574.3

Р. А. Коротченко, канд. техн. наук, ст. науч. сотр., e-mail: romankor@mail.ru,
Тихоокеанский океанологический институт им. В. И. Ильичева ДВО РАН, Владивосток

Применение сингулярного спектрального анализа для обнаружения акустических сигналов от серых китов

Представлены методика и алгоритм обработки данных гидроакустического мониторинга, выполняющего задачу автоматизированного обнаружения сигналов серых китов. Обработка акустических сигналов с помощью сингулярного спектрального анализа позволяет эффективно разделить нестационарную импульсную составляющую, характерную для сигналов серых китов, и естественный акустический фон. Рассмотрены примеры обработки реальных сигналов. Отмечается возможность применения методики для поиска подобных локализованных сигналов.

Ключевые слова: сингулярный спектральный анализ, гидроакустика, нестационарные сигналы, серые киты

Введение

Длительный гидроакустический мониторинг на прибрежном шельфе позволяет накапливать огромное количество данных, требующее впоследствии тщательного анализа в целях получения информации определенного направления [1, 2]. Возможности гидроакустики для наблюдений за активностью морских млекопитающих и, в частности, мониторинга занесенной в Красную книгу малочисленной популяции серых китов, присутствующих в летне-осенний период на северо-восточном шельфе Сахалина, предполагают развитие эффективных алгоритмов автоматической обработки информации, позволяющие обнаруживать появление животных в зоне чувствительности приемных акустических систем и выполнять постобработку данных и статистический анализ по результатам сезонных наблюдений. Для достижения указанной цели необходимо решить ряд задач, одной из которых является поиск и локализация сигналов животных в накопленных акустических данных. Сложность решения задачи обусловлена рядом факторов, из которых наиболее значимыми являются краткость импульсных сигналов серых китов, невысокие амплитуды сигналов относительно уровня шума и в целом редкое появление сигналов в записях акустических данных.

Регулярное присутствие естественного акустического шума предполагает применения алгоритмов фильтрации. Однако в силу того, что частотные полосы акустического фона и голосов животных пересекаются, алгоритмы фильтрации и обнаружения должны использовать специфические структурные особенности сигналов китов и применять

подходящую модель данных. Одним из способов решить данную задачу является применение сингулярного спектрального анализа (ССА), который позволяет выполнить декомпозицию акустического сигнала на периодическую, осциллирующую и шумовую составляющие и выделить информационное подпространство с высоким вкладом от импульсоподобных сигналов китов. В этом случае признаком, указывающим на возникновение импульсных нестационарных сигналов, является появление характерных особенностей собственной 2-й и 3-й главных функций и изменение "энергетического" вклада от них в составе разложения. Построение сигнальной функции, выделяющей наличие подобного вклада в составе акустических данных, дает возможность реализовать автоматизированную обработку данных и поиск сигналов.

Основные принципы сингулярного спектрального анализа

В отечественной литературе метод ССА известен как "Гусеница" [3, 4]. В основе метода лежит переход от одномерного временного ряда $F = (f_1, \dots, f_n, \dots)$ к многомерному представлению с помощью множества векторов задержек, формирующих "траекторную" матрицу:

$$X = [X_1 \dots X_K], \text{ где } X_j = (f_{j-1}, \dots, f_{j+L-2})^T.$$

Основными параметрами отображения являются длина окна L и число векторов задержки K . Длину окна L следует подбирать исходя из возможности разложения временного ряда на идентифицируемые слагаемые, что предполагает выделение трендов и периодических компонент.

Далее выполняется сингулярное разложение траекторной матрицы

$$\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2 + \dots + \mathbf{X}_d, \mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T,$$

где $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d > 0$ — упорядоченные ненулевые собственные числа ковариационной матрицы $\mathbf{X}\mathbf{X}^T$, $\{U_i \in \mathbf{R}^L\}_{i=1}^d$ — соответствующие им собственные векторы, а $\{V_i = \lambda_i^{-1/2} \mathbf{X}^T U_i \in \mathbf{R}^K\}_{i=1}^d$ — факторные векторы.

На третьем шаге проводится группировка компонент разложения. Из разбиения $\{1, \dots, d\}$ на m непересекающихся подмножеств I_j получается

$$\mathbf{X} = \mathbf{X}_{I_1} + \mathbf{X}_{I_2} + \dots + \mathbf{X}_{I_m}, \mathbf{X}_{I_j} = \sum_{k \in I_j} \mathbf{X}_k.$$

Число и смысл разбиения на отдельные подмножества отражает требуемую степень сглаживания ряда и разделение на интерпретируемые составляющие. Поиск компонент для требуемой группировки, главным образом на основе анализа собственных троек, будем называть процедурой идентификации. Как правило, при правильном выборе окна L амплитуды собственных значений в одной группе близки. Это хорошо видно, если построить график логарифмов собственных значений.

Последним шагом является восстановление рядов $F_N^{(j)}$ по сгруппированным матрицам \mathbf{X}_{I_j} с помощью усреднения вдоль побочных диагоналей:

$$f_s = \begin{cases} \frac{1}{s} \sum_{i=1}^s x_{i, s-i+1}, & 1 \leq s \leq r; \\ \frac{1}{r} \sum_{i=1}^r x_{i, s-i+1}, & r \leq s \leq c; \\ \frac{1}{N-s+1} \sum_{i=1}^{N-s+1} x_{i+s-c, c-i+1}, & c \leq s \leq N, \end{cases}$$

где r и c — число строк и столбцов в матрице \mathbf{X}_{I_j} , N — длина обрабатываемого сегмента данных. Индекс s указывает смещение вдоль побочной диагонали, например, для третьей диагонали это суммирование $x_{13} + x_{22} + x_{31}$, $N = 3$.

Таким образом формируется сглаженное представление ряда $F_N = F_N^{(1)} + \dots + F_N^{(m)}$ с декомпозицией на аддитивные компоненты, называемые модами. В случае нестационарных временных рядов эти моды позволяют разделить исходные данные на трендовые, периодические и шумовую составляющие.

Применение ССА не предполагает знания априорной информации о системе, порождающей данные, или построения специфической модели динамики ряда, а выбор числа главных компонент, участвующих в восстановлении ряда, эффективно управляет фильтрацией случайных возмущений, повышая соотношение сигнал/шум.

Особенности акустических сигналов от серых китов

Отчетливые акустические сигналы от серых китов, иллюстрирующие амплитудно-временные характеристики импульсов, представлены на рис. 1. Характерными особенностями сигналов является их групповой характер: послышки идут наборами по 3...10 импульсов в пачке с длительностью отдельного импульса около 0,2 с. Общая длительность послышки составляет 1,5...3 с.

Представление о частотно-временной структуре импульса дает рис. 2. Важным фактом, иллюстрируемым на рисунке, является присутствие в частотном составе импульса тональных компонент, сосредоточенных в полосе 100...400 Гц. Следует отметить, что в серии импульсов, продуцируемых китом, от сигнала к сигналу частота основного тона существенно изменяется, что указывает на своеобразие каждого импульса в пачке. Нестационарность им-

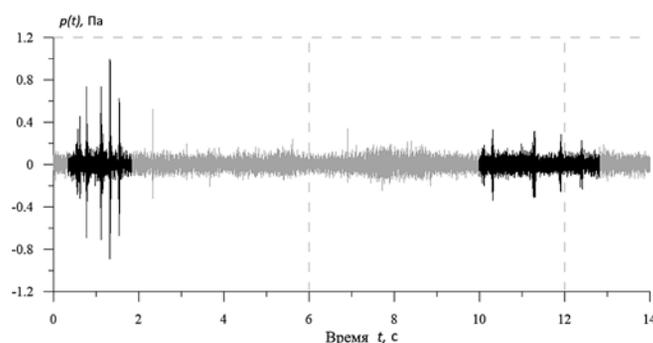


Рис. 1. Коммуникационные сигналы от пары серых китов; $p(t)$ — акустическое давление

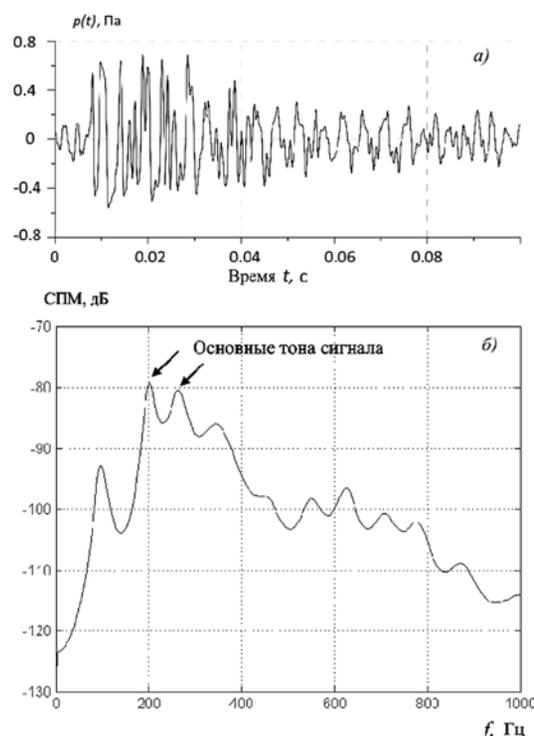


Рис. 2. Отдельный импульс (а) и оценка его мгновенной спектральной плотности мощности (СПМ) (б)

пульсных сигналов приводит к формированию локализованных осциллирующих составляющих фрагмента временного ряда с выделением частот, которые отсутствуют в фоновом шуме.

Адаптация ССА к поиску импульсных сигналов от кита

При поиске импульсных сигналов от серого кита выполняется ССА последовательных фрагментов ряда вариаций акустического давления. Каждый фрагмент ряда трансформируется в траекторную матрицу, и далее, с помощью сингулярного разложения, из него выделяются аддитивные составляющие, так называемые моды, интерпретируемые как осциллирующие составляющие и шум. Появление в разложении заметного вклада осциллирующей составляющей, согласованной во временном масштабе с импульсными сигналами от серого кита, является регистрируемым признаком. Повторяемость появления признака в течение нескольких секунд указывает на серийный характер импульсного сигнала и повышает достоверность процедуры поиска и идентификации.

Метод эффективно работает при низком соотношении сигнал/шум за счет того, что происходит фильтрация и разделение сигнальной и шумовой составляющих. В подпространстве сигнала, выделяемом второй и третьей модами разложения, зашумление незначительно, и отношение сигнал/шум существенно возрастает. Использование второй моды разложения и ее собственного значения как "энергетического" критерия наличия сигнала позволяет обнаруживать сигналы, которые визуально или "на слух" в акустических записях не выделяются.

На рис. 3 иллюстрируется процесс анализа фрагмента данных с помощью разработанного программного обеспечения. Для сравнения приведены результаты ССА двух фрагментов: в присутствии импульсного сигнала и без сигнала. Для каждого фрагмента восстанавливались три старшие моды разложения исходного сигнала. Принципиальным является различие величин 2-го и 3-го собственных значений A_k сингулярного разложения, отражаемых на верхней и нижней гистограммах. Собственные значения 2-й и 3-й мод возрастают при наличии сигнала и остаются малыми при его отсутствии. Такое поведение собственного значения можно интерпретировать как меру информации о наличии импульсного сигнала в исходных данных.

Чтобы повысить надежность выделения из шума сигналов, изменяющихся по длительности и частоте, оказалось эффективным использовать одновременно две или три модели разложения исходного сигнала, отличающиеся различным параметром окна L , определяющим размерность "траекторной матрицы". Произведение логарифмов собственных значений, рассчитанных по разным моделям, увеличивает информационную состоятельность признака сигнала. Применение трех моделей иллюстрируется

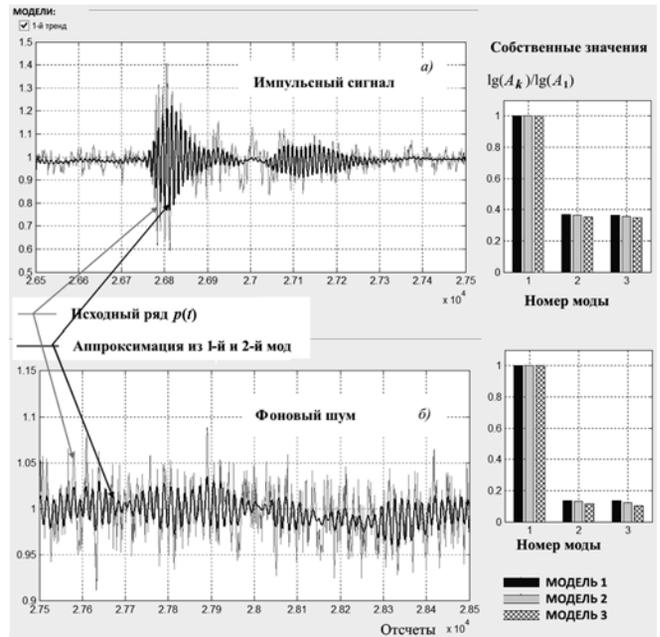


Рис. 3. Фрагменты данных и результаты анализа вклада модовых составляющих в случае присутствия сигнала (а) и без сигнала (б)

на рис. 3. На гистограммах представлено отношение логарифмов собственных значений трех старших модовых функций к логарифму 1-го собственного значения. В случае наличия импульсных сигналов амплитуда собственных значений значительно выше, чем в записи фонового шума.

Последовательный ряд значений логарифмов собственных значений для второй моды (или их произведение для набора моделей), рассчитанный при проходе по фрагментам временного ряда акустических данных, представляет "сигнальную функцию", поскольку наличие "пиков" на ее графике указывает на присутствие импульса. Другой версией сигнальной функции является разностная производная ряда логарифмов собственных значений, поскольку она

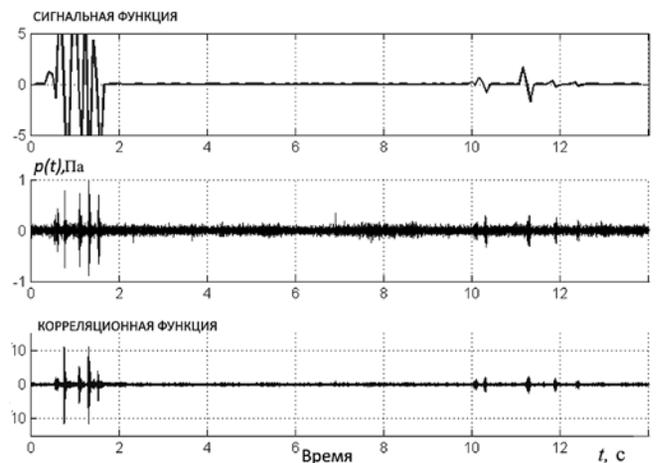


Рис. 4. Обнаружение импульсов с помощью сигнальной и корреляционной функций (верхний и нижний графики) при низком уровне шума в данных (средний график)

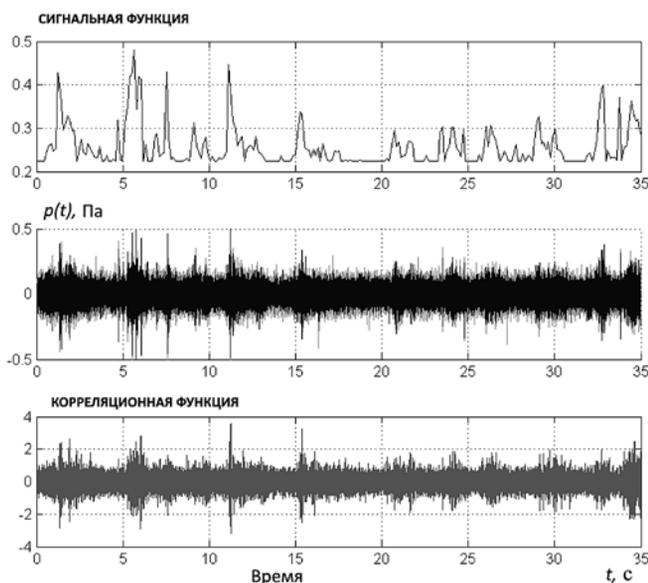


Рис. 5. Обнаружение импульсов с помощью сигнальной и корреляционной функций (верхний и нижний графики) при повышенном уровне шума в данных (средний график)

более отчетливо выделяет изменение амплитуды исходной сигнальной функции и наличие серии.

На рис. 4, 5 представлены практические примеры поиска и выделения импульсных групп в акустической записи. Для сравнения приводится обработка с помощью взаимокорреляционной функции акустической записи с отчетливым импульсом, представленным на рис. 2. Для сигналов с высоким отношением к уровню шума методы дают близкие результаты (рис. 4), а в случаях, когда уровень шума возрастает, сигнальная функция на основе логарифма собственного значения второй моды ССА

дает более отчетливую информации о присутствии импульсов (рис. 5).

Заключение

Применение сингулярного спектрального анализа позволило выполнить фильтрацию сигналов особого вида, характерных для серых китов, из суммарного звукового фона за счет анализа вклада 2-й и 3-й главных мод разложения. Сравнение с корреляционным методом поиска сигналов показало более высокое качество обнаружения и выделения импульсов в ситуациях с повышенным уровнем естественного шума.

Описанная методика поиска и выделения импульсов на фоне естественных акустических шумов решила задачу автоматической обработки больших объемов данных гидроакустического мониторинга в целях извлечения информации об изменении сезонной активности животных.

Следует отметить, что аналогичная обработка применима к поиску других типов локализованных сигналов на фоне псевдостационарных шумов в данных, представленных временными рядами.

Список литературы

1. Rutenko A. N., Borisov S. V., Gritsenko A. V., Jenkerson M. R. Calibration and monitoring the gray whale mitigation zone and estimation acoustic transmission during 3-D seismic survey, Sakhalin Island, Russia. // *Environmental monitoring and Assessment*. 2007. Vol. 134. P. 21–44.
2. Рутенко А. Н., Гаврилевский А. В., Коротченко Р. А. и др. Мониторинг сейсмоакустических сигналов и антропогенных шумов на шельфе о. Сахалин // *Акустический журнал*, 2012. Т. 58, № 2. С. 248–257.
3. Golyandina N., Nekrutkin V., and Zhigljavsky A. *Analysis of Time Series Structure: SSA and Related Techniques*. London: Chapman & Hall/CRC, 2001.
4. Голяндина Н. Э. Метод "Гусеница" SSA: анализ временных рядов. СПб.: СПб ГУ, 2004. 74 с.

R. A. Korotchenko, Ph. D., Senior Researcher, e-mail: romankor@mail.ru,
V. I. Il'ichev Pacific Oceanological Institute, Vladivostok

Application of Singular Spectral Analysis to Detection of Acoustic Signals of Gray Whales

The paper presents a methodology and an algorithm for processing hydroacoustic monitoring data that are used for automated detection of gray whale signals. Using singular spectrum analysis in acoustic signals processing makes it possible to efficiently separate the nonstationary impulse components that are typical for gray whale signals from the ambient noise. A number of examples are given for real signals processing. The possibility of using the methodology to detect other localized signals is shown.

Keywords: singular spectral analysis, hydroacoustic signals, monitoring, gray whales

References

1. Rutenko A. N., Borisov S. V., Gritsenko A. V., Jenkerson M. R. Calibration and monitoring the gray whale mitigation zone and estimation acoustic transmission during 3-D seismic survey, Sakhalin Island, Russia, *Environmental monitoring and Assessment*, 2007, vol. 134, pp. 21–44.
2. Rutenko A. N., Gavriljevskij A. V., Korotchenko R. A. et al. Monitoring sejsmoakusticheskikh signalov i antropogennyh шумов на

- shel'fe o. Sahalin (Rutenko A. N., Gavriljevskij A. V., Kovzel' D. G., Korotchenko R. A. Monitoring of Seismoacoustic Signals and Anthropogenic Noise on the Shelf of Sakhalin Island). *Akusticheskij Zhurnal*, 2012, vol. 58, no. 2, pp. 248–257 (in Russian).
3. Golyandina N., Nekrutkin V., and Zhigljavsky A. *Analysis of Time Series Structure: SSA and Related Techniques*. London: Chapman & Hall/CRC, 2001.
4. Golyandina N. Je. *Metod "Gusenica" SSA: analiz vremennyh rjadov*. SPb.: SPb GU, 2004. 74 p. (in Russian).

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ CRYPTOSAFETY INFORMATION

УДК 621.396

Б. И. Филиппов, канд. техн. наук, доц., e-mail: filippov-boris@rambler.ru,
ФГБОУ ВПО Новосибирский государственный технический университет, 630073, г. Новосибирск

Разработка и исследование устройства защиты от ошибок для системы передачи изображений по гидроакустическому каналу связи

Целью работы является разработка принципов и методов защиты от ошибок в гидроакустических каналах связи. Для обоснования выбора параметров разрабатываемого устройства защиты от ошибок были выполнены исследования статистических характеристик потока ошибок в гидроакустических каналах связи существующей аппаратуры гидроакустической системы связи путем обработки предоставленных записей (файлов) образцов кадров экспериментальной передачи изображений с использованием этой аппаратуры. Полученные зависимости вероятности ошибок на выходе модуля устройства защиты от ошибок от качества дискретного канала связи позволяют определить требования к значению сигнал/шум на входе приемного тракта судовой части и, соответственно, требования к значению мощности акустического сигнала передающего устройства в зависимости от глубины его нахождения.

Ключевые слова: гидроакустический канал связи, гидроакустическая система связи, передача цифровой информации, повышение достоверности, помехоустойчивое кодирование, статистические характеристики потока ошибок, устройство защиты от ошибок

Введение

В соответствии с теоремой Клода Шеннона для дискретных каналов связи (ДКС) с помехами информация может передаваться со сколь угодно высокой степенью достоверности при условии, что скорость передачи информации не превышает пропускной способности канала. Помехоустойчивость системы связи определяется, как известно [1], ее способностью противостоять вредному влиянию помех или, другими словами, это способность системы связи функционировать с требуемым качеством передачи информации в условиях неблагоприятных внутренних и внешних воздействий. Количественной мерой помехоустойчивости при передаче дискретных сообщений, как правило, является вероятность ошибки в принимаемой последовательности.

Наиболее эффективным средством обеспечения высокой помехоустойчивости сложной системы является введение избыточности, необходимой для обнаружения и исправления ошибок, возникающих при работе системы и ее элементов. Теоретической базой эффективного использования вводимой избыточности является теория помехоустойчивого кодирования.

Помехоустойчивые (или корректирующие) коды имеют различные приложения. Так, эти коды используются для защиты данных в вычислительных устройствах и сетях, системах хранения данных, спутниковых и навигационных системах, в сотовых телефонах и пейджерах, они могут быть применимы для получения надежной связи даже тогда,

когда мощность принимаемого сигнала близка к мощности шумов. В специальных приложениях коды используют для защиты информации против преднамеренно организованных помех.

Как показывают экспериментальные исследования, статистические характеристики гидроакустических каналов связи (ГАКС) имеют свои аналоги в коротковолновых, УКВ и других радиоканалах с переменными параметрами. Поэтому разработанные для этих каналов принципы и методы защиты от ошибок могут быть использованы и в системах передачи информации, использующих ГАКС, конечно, с учетом специфических свойств распространения акустических сигналов в водной среде [2–11]. В работах [8, 10, 11] речь идет о моделировании гидроакустического канала, это в общем-то понятно, если учесть, что получение экспериментальных данных для таких каналов — весьма трудоемкая и затратная операция. В работе [7] предлагается для повышения пропускной способности использовать в гидроакустическом канале технологию MIMO—OFDM. По мнению авторов, в этом случае возможна передача исследовательских данных и изображения с видеокамер. Практическое применение технологии MIMO—OFDM для гидроакустических каналов представляется спорным, и, как будет показано ниже, есть другие способы передачи изображения по таким каналам.

Наибольший интерес представляет работа [9], которая в основном подтверждает результаты, полученные в работах [2] (статистические характерис-

тики сигналов и помех в гидроакустическом канале) и [14] (применение помехоустойчивого кодирования в таких каналах).

1. Статистические особенности потока ошибок в аппаратуре ГАСС

Для обоснования выбора параметров разрабатываемого устройства защиты от ошибок были выполнены исследования статистических характеристик потока ошибок в гидроакустическом канале связи существующей аппаратуры гидроакустической системы связи (ГАСС) путем обработки предоставленных записей (файлов) образцов кадров экспериментальной передачи изображений с использованием этой аппаратуры. Путем сравнения образцов переданных и принятых кадров изображения были получены массивы ошибок в канале существующей аппаратуры гидроакустической системы связи. Для этого пиксели принятого и переданного изображений преобразовывались в первичные потоки двоичных символов и путем сложения этих потоков по модулю два формировался массив ошибок в данном кадре изображения. В процессе преобразования кадров изображений строки с нарушением синхронизации из рассмотрения исключались.

Всего было получено четыре файла с массивами ошибок: *errMass 1.txt*, *errMass 2.txt*, *errMass 3.txt*, *errMass 4.txt*. Каждый из них содержит около $3 \cdot 10^5$ двоичных символов. Эти файлы использовались для определения статистических характеристик потока ошибок в реальном гидроакустическом канале связи, а также для статистических испытаний разрабатываемого устройства защиты от ошибок.

Определялись следующие характеристики [1]:

- средняя вероятность ошибки p_{cp} ;
- распределение кратностей ошибок t в кодовом слове длины $n - P_n(t)$;
- распределение длин пакетов ошибок $P(t_{п})$;
- распределение длин интервалов между пакетами ошибок $P(t_{инт})$.

Результаты статистической обработки полученных массивов ошибок приводятся ниже.

В табл. 1—4 приведены распределения кратностей ошибок для различных значений средней вероятности ошибки в канале связи, при этом массивы ошибок были разбиты на кодовые слова длиной $n = 100$ символов.

Для сравнения было получено распределение кратностей ошибок в канале связи с независимым характером ошибок в предположении, что вероятность ошибки равна средней вероятности ошибки в файле *errMass1.txt* ($p_{cp} = 0,005$). Результаты приведены в табл. 5.

Сравнение результатов табл. 1 и 5 показывает, что ошибки в файле *errMass1.txt* ($p_{cp} = 0,005$) существенно пакетируются, следствием чего является увеличение вероятностей ошибок большой кратности и вероятности приема кодового слова без ошибок. Аналогичный характер имеют потоки

ошибок и в других файлах: *errMass2.txt*, *errMass3.txt*, *errMass4.txt*. По характеру ошибок на анализируемых кадрах изображений можно предположить, что причинами пакетирования ошибок являются:

- занижение уровня принимаемого сигнала, обусловленное качкой обслуживающего судна и про-

Таблица 1

Распределение кратностей ошибок $P_n(t)$ в файле *errMass1*, $p_{cp} = 0,005$

Длина пакетов ошибок	Кратность ошибок				
	0	1	2	3	4
0	0,84691	0,02135	0,04468	0,04895	0,01018
5	0,00558	0,01051	0,00361	0,00099	0,00230
10	0,00131	0,00033	0,00164	0,00066	0,00033
15	0,00000	0,00033	0,00033	0,00000	0,00000

Таблица 2

Распределение кратностей ошибок $P_n(t)$ в файле *errMass2*, $p_{cp} = 0,0019$

Длина пакетов ошибок	Кратность ошибок				
	0	1	2	3	4
0	0,91524	0,03909	0,02825	0,00723	0,00296
5	0,00230	0,00131	0,00033	0,00033	0,00000
10	0,00099	0,00033	0,00033	0,00066	0,00033
15	0,00033	0,00000	0,00000	0,00000	0,00000

Таблица 3

Распределение кратностей ошибок $P_n(t)$ в файле *errMass3*, $p_{cp} = 0,0061$

Длина пакетов ошибок	Кратность ошибок				
	0	1	2	3	4
0	0,77957	0,08574	0,05486	0,03351	0,00591
5	0,00591	0,00887	0,00361	0,00197	0,00526
10	0,00131	0,00164	0,00033	0,00230	0,00066
15	0,00033	0,00066	0,00000	0,00033	0,00000
20					

Таблица 4

Распределение кратностей ошибок $P_n(t)$ в файле *errMass4*, $p_{cp} = 0,073$

Длина пакетов ошибок	Кратность ошибок				
	0	1	2	3	4
0	0,19054	0,06110	0,06932	0,07454	0,05683
5	0,04796	0,05552	0,04468	0,04763	0,04369
10	0,03318	0,02825	0,02562	0,02070	0,01938
15	0,02168	0,01643	0,01610	0,01577	0,01183
20	0,01347	0,00788	0,01150	0,00657	0,00591
25	0,00460	0,00263	0,00361	0,00230	0,00131
30	0,00197	0,00197	0,00033	0,00066	0,00033

Таблица 5

Распределение кратностей ошибок $P_n(t)$ в канале с независимыми ошибками, $p_{cp} = 0,005$

Длина пакетов ошибок	Кратность ошибок				
	0	1	2	3	4
0	0,60354	0,30511	0,07672	0,01279	0,00161
5	0,00020	0,00003	0,00000	0,00000	0,00000

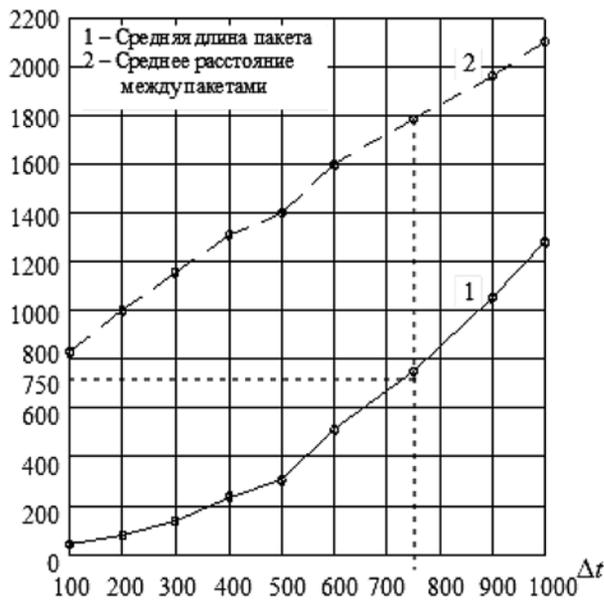


Рис. 1. Зависимость средней длины пакета ошибок и интервала между пакетами от Δt

явлением эффекта многолучевости распространения сигнала вследствие отражения от неоднородностей водной среды;

— помехи импульсного характера (возможно, от гребного винта обслуживающего судна), что особенно характерно для файлов *errMass3.txt*, *errMass4.txt* (заметна периодическая помеха с интервалом около 0,1 с).

Для выяснения характеристик пакетирования ошибок были рассчитаны распределения длин пакетов и длин интервалов между пакетами ошибок.

На рис. 1 приведена зависимость средней длины пакета ошибок $\bar{l}_п$ и средней длины интервала между пакетами ошибок $\bar{l}_{инт}$ от базы Δt (расстояния в двоичных символах между соседними ошибками в пакете) для файла *errMass1.txt* ($p_{ср} = 0,005$). Фактическая средняя длина пакета ошибок и средняя длина интервала между пакетами определены по рис. 1 как значения, соответствующие равенству базы Δt и средней длины пакета ошибок (в рассматриваемом случае это точка, соответствующая $\Delta t = \bar{l}_п = 750$ двоичных символов, при этом средняя длина интервала между пакетами $\bar{l}_{инт} = 1800$).

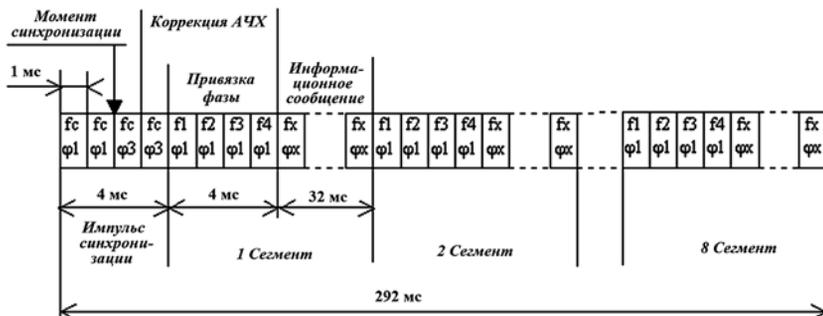


Рис. 2. Структура излучаемой информационной строки ГАСС

Аналогично в файле *errMass3.txt* ($p_{ср} = 0,0061$) средняя длина пакета ошибок уменьшается до 480 символов, а среднее расстояние между пакетами ошибок — до 945 символов. В файле *errMass4.txt* ($p_{ср} = 0,073$) можно выделить только сгущения ошибок с периодом около 350 двоичных символов.

Полученные результаты статистической обработки файлов с массивами ошибок в существующей ГАСС дают основание считать, что в рассматриваемом случае имеет место дискретный канал связи с пакетированием ошибок, который может быть описан математической моделью ДКС с памятью [12].

По средней длительности пакета ошибок можно определить параметр Q , который [13] характеризует пакетирование ошибок и память в дискретном канале связи и имеет следующие значения:

$$Q_{\bar{l}_п}^1 = 0,00125 \text{ для массива ошибок в файле } errMass1.txt;$$

$$Q = 0,0021 \text{ для массива ошибок в файле } errMass3.txt.$$

2. Выбор помехоустойчивого кода для реализации УЗО

Реализация УЗО для передачи изображений в существующей гидроакустической системе связи должна выполняться программным способом в соответствии с требованиями структуры и утвержденного протокола обмена разрабатываемого УЗО с аппаратурой ГАСС.

Структура информационной строки ГАСС, передаваемой автономным подводным роботом в направлении обслуживающего судна, приведена на рис. 2. В соответствии с принятой структурой информационной строки не подлежат кодированию помехоустойчивым кодом УЗО импульсы синхронизации и импульсы привязки фазы, поскольку функционально их прием и обработка приемником должны предшествовать приему информационных символов соответствующей строки или сегмента. Получается, что строка содержит 256 информационных четырехзначных сигналов (от 0 до 15-го уровня градаций), которые в дальнейшем будем называть полубайтами.

Предварительные испытания ряда кодеров циклических и сверточных кодов в условиях, когда статистика ошибок соответствует реальным потокам ошибок, записанным в файлах: *errMass1.txt*, *errMass2.txt*, *errMass3.txt* (рис. 2), показали, что для исправления ошибок в существующем канале ГАСС с учетом пакетирования ошибок необходимы помехоустойчивые коды, скорость которых $R \leq 1/2$.

В качестве кода, исправляющего ошибки, в разрабатываемом УЗО может быть применен циклический код БЧХ с параметрами $(n, k) = (204, 96)$, исправляющий ошибки кратности

$t_u \leq 14$, который при $p_{\text{вх}} = 6 \cdot 10^{-3}$ (память в канале связи $Q = 0,002$) обеспечивает вероятность ошибки декодирования $P_d \approx 5 \cdot 10^{-4}$. Сверточный код (404,202), рассмотренный в работе [14], с пороговым итерационным декодером в тех же условиях обеспечивает $P_d \approx 2 \cdot 10^{-4}$.

Для борьбы с пакетированием ошибок в состав УЗО должен быть введен перемежитель символов, который, как известно, позволяет приблизить характеристики канала к характеристикам канала с независимыми ошибками.

При установке перемежителя символов канала связи и циклический код (204,96), и сверточный код (404,202) обеспечивают требуемые качественные показатели ($P_d < 10^{-4}$), при этом для циклического кода $P_d \approx 5 \cdot 10^{-6}$, а для сверточного кода $P_d \approx 2 \cdot 10^{-7}$.

Исходя из полученных результатов, для практической реализации УЗО в аппаратуре ГАСС выбран сверточный код (404,202) с итерационным пороговым декодером блочного типа, применяемый в сочетании с перемежителем символов канала связи.

При использовании данного кода следует иметь в виду следующие особенности.

1. Сверточный код с итерационным пороговым декодером блочного типа в отличие от циклического кода позволяет использовать кодовое слово произвольной длины $n \geq n_0 \cdot \max[\deg g(x)]$. Это дает возможность для кодирования информационного сообщения использовать кодовые слова, соответствующие длине строки $256 \times 4 = 1024$ или длине столбца $192 \times 4 = 768$ двоичных символов.

2. Вычислительная сложность декодирования сверточного кода, равная $\Phi_k^{(св)} = c_k \cdot J \cdot (y + 1)$, меньше в $10 \div 15$ раз, чем у циклического кода с декодером по алгоритму Берликемпа — Мессе [15] для двоичных кодов $\Phi_n^{(п)} = c_k \cdot t_u \cdot n$ (c_k — коэффициент пропорциональности, зависящий от типа ЭВМ).

3. Качество декодирования сверточного кода с итерационным пороговым декодером несколько лучше, чем у циклического кода, что объясняется дополнительным эффектом внутреннего перемежителя, свойственного непрерывным сверточным кодам.

4. Недостатком сверточных кодов, как непрерывных, является необходимость в передаче в конце каждого кадра "хвостовика" в виде произвольного кодового слова для того, чтобы декодер выдал на выход последнее информационное слово. Иначе приходится мириться с тем, что это слово остается в памяти декодера и не выдается получателю либо будет принудительно считано без исправления ошибок.

Выбранный сверточный код (404,202) с производящим многочленом $g(x) = 1 + x^7 + x^{27} + x^{76} + x^{113} + x^{137} + x^{155} + x^{156} + x^{170} + x^{202}$ и скоростью кода $R = 1/2$ является одним из лучших канонических самоортогональных кодов, имеющих $J = 10$ ортогональных проверок и позволяющих осуществлять итерационное пороговое декодирование. Это код средней длины с длиной регистров памяти декодера в 202 ячейки.

Были проведены сравнительные испытания выбранного сверточного кода (404,202) с аналогичным по скорости кода $R = 1/2$ сверточным кодом (1218,609), имеющим $\max[\deg g(x)] = 609$. Результаты испытаний представлены на рис. 3.

Видно, что в рабочей области значений $p_{\text{вх}}$ (менее $7 \cdot 10^{-2}$) код (1218,609) уступает коду (404,202), который позволяет вместе с перемежением символов в канале обеспечить сравнительно лучшие условия исправления независимых ошибок. Кроме того, выбранный сверточный код (404,202) с перемежителем практически не уступает в отношении импульсных помех каскадным кодам на основе коротких сверточных кодов и циклических кодов Рида — Соломона, обычно используемых в аппаратуре цифровой связи при более низкой вероятности ошибок в канале, чем в аппаратуре ГАСС. На рис. 4

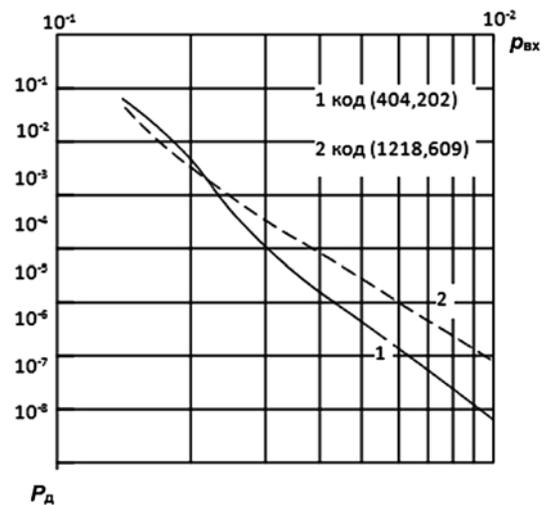


Рис. 3. Зависимость вероятности ошибки декодирования от вероятности ошибок на входе декодера $p_{\text{вх}}$

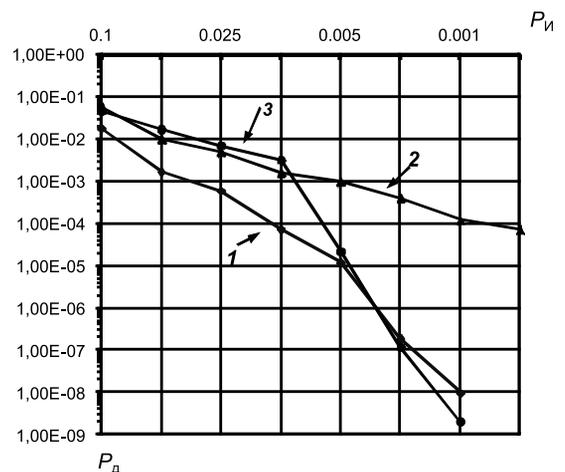


Рис. 4. Зависимость вероятности ошибки декодирования от вероятности появления импульсной помехи: 1 — сверточный код (404,202) с перемежителем, декодер пороговый, 5 итераций; 2 и 3 — каскадный код: внутренний код сверточный $R = 1/2$, декодер Витерби, внешний код — код Рида — Соломона (255,239), в варианте 3 внутренний код — тот же код Рида — Соломона с перемежителем

приведены результаты сравнительных статистических исследований этих кодов на модели канала с импульсными помехами.

В процессе этих исследований на ЭВМ предполагалось, что длительность отдельного импульса помехи примерно равна длительности 300 двоичных символов, а вероятность ошибки при действии импульсной помехи равна 0,5; при этом длительность импульсной помехи примерно соответствует длительности периодических пакетов ошибок в файлах *errMass3.txt*, *errMass4.txt*.

Видно, что сверточный код (404,202) с перемежителем в диапазоне рабочих значений вероятности ошибки системы ГАСС не уступает каскадному коду, а с учетом сложности и скорости декодирования каскадного кода сверточный код с итерационным пороговым декодированием является предпочтительным.

Приведенные результаты подтверждают обоснованность выбора сверточного кода (404,202) в сочетании с перемежителем символов, потому что причиной почти периодических помех в системе ГАСС могут быть именно импульсные помехи.

3. Результаты испытаний УЗО

Проверку функционирования устройства защиты от ошибок для автономного подводного робота проводили путем контроля приема тестовой информации, передаваемой от модуля УЗО автономного подводного робота на модуль УЗО обслуживающего судна с использованием модели канала гидроакустической связи в виде:

- программного имитатора канала с независимыми ошибками;
- программного имитатора канала с реальными потоками ошибок, полученными путем сравнения кадров изображения, переданных и принятых аппаратурой ГАСС.

Контроль правильности функционирования модулей УЗО осуществляли с помощью аппаратуры контроля работоспособности на основе ПЭВМ с программным обеспечением "TestCodec". При этом обеспечивались автоматизированная регистрация, обработка и представление измерительной информации, а также управление экспериментом.

Воздействие ошибок на приемное устройство модуля УЗО обслуживающего судна обеспечивалось в имитационном режиме.

На рис. 5—9 (см. третью сторону обложки) приведены примеры передачи кадров цифровой информации с восстановлением изображения после помехоустойчивого кодирования и передачи по каналу связи с имитацией помех (рис. 6 и 8, см. третью сторону обложки). После декодирования цифровой сигнал кадра снова восстанавливался в виде изображения (рис. 7 и 9, см. третью сторону обложки). Видно, что искажения на рис. 6 исправляются декодером не полностью (рис. 7), в то время как искажения на рис. 8 полностью исправляются

(рис. 9). Как будет показано в разд. 4, вероятность искажения пикселя на рис. 6 находится на границе исправляющей способности кода, использованного в разработанном УЗО.

4. Оценка достоверности передачи цифровой информации

Оценка достоверности передачи цифровой информации от модуля УЗО—Р автономного подводного робота на судовой модуль УЗО—С осуществлялась путем многократной передачи в каждом сеансе связи в качестве кадра данных известной тестовой псевдослучайной последовательности. На выходе приемного устройства оценивалась частота ошибок для оценки качества функционирования УЗО.

Общий объем измерений выбирали с учетом требуемого объема испытаний для каждого значения вероятности ошибки на входе декодера. Ошибки в дискретном канале связи с заданной вероятностью $p_{вх}$ имитировались программно.

Независимые ошибки в дискретном канале связи в соответствии с заданным значением $p_{вх}$ имитировали с помощью датчика случайных чисел ПЭВМ. Результаты измерений приведены на рис. 10.

Оценку достоверности передачи в канале с реальным потоком ошибок проводили путем однократной передачи информационного кадра объемом 307 200 дв. символов (или 76 800 пикселей). Объем измерений определялся объемом файлов ошибок, выделенных из реальных кадров изображений, полученных на судовой части аппаратуры ГАСС. Использовали четыре кадра изображений в виде файлов *errMass1.txt*, *errMass2.txt*, *errMass3.txt*, *errMass4.txt* с различной средней вероятностью ошибки. Измерения проводили в условиях, когда в канале был использован матричный перемежитель и когда перемежитель не использовался.

Результаты измерений приведены в табл. 6.

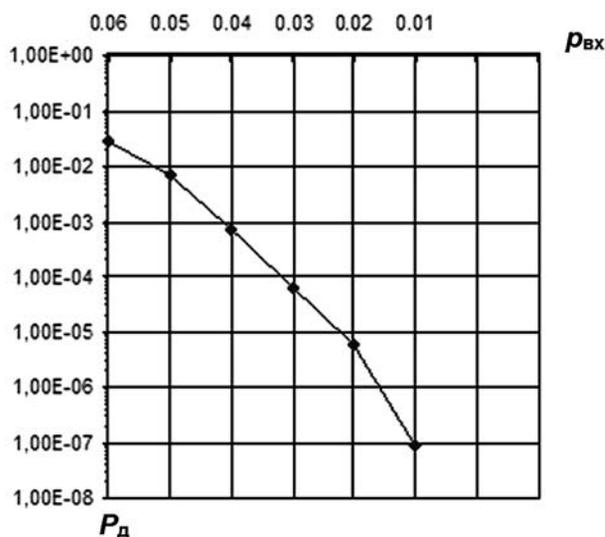


Рис. 10. Зависимость вероятности ошибки декодирования от вероятности ошибки на входе

Таблица 6

Вероятности ошибки декодирования для массивов ошибок в реальном канале при передаче изображений

Сверточный код (404,202), декодер пороговый, 6 итераций				
Файл ошибок→	<i>errMass1</i>	<i>errMass2</i>	<i>errMass3</i>	<i>errMass4</i>
$P_{вх}$	$4,99 \cdot 10^{-2}$	$1,87 \cdot 10^{-2}$	$6,1 \cdot 10^{-2}$	$7,25 \cdot 10^{-2}$
$P_{д}$ без перемежения	$3,2 \cdot 10^{-4}$	$3,1 \cdot 10^{-4}$	$3,8 \cdot 10^{-4}$	$8,36 \cdot 10^{-2}$
$P_{д}$ с перемежением	$< 2 \cdot 10^{-5}$	$< 2 \cdot 10^{-5}$	$< 2 \cdot 10^{-5}$	$8,68 \cdot 10^{-2}$

Таблица 7

Зависимость вероятности ошибки декодирования от отношения сигнал/шум на модели канала с переменными параметрами

Параметры	$H_p = 6,66$	$H_p = 10,00$	$H_p = 15,00$	$H_p = 20,00$
	$H_c = 1,00$	$H_c = 1,50$	$H_c = 2,25$	$H_c = 3,00$
$P_{вх}$	$1,4 \cdot 10^{-2} \pm 2,0 \cdot 10^{-4}$	$6,1 \cdot 10^{-2} \pm 1,6 \cdot 10^{-5}$	$4,4 \cdot 10^{-3} \pm 2,0 \cdot 10^{-5}$	$1,4 \cdot 10^{-3} \pm 1,2 \cdot 10^{-5}$
$P_{д}$ без перемежения	$3,1 \cdot 10^{-3} \pm 3,0 \cdot 10^{-5}$	$1,0 \cdot 10^{-3} \pm 4,1 \cdot 10^{-6}$	$3,2 \cdot 10^{-5} \pm 2,0 \cdot 10^{-6}$	$2,3 \cdot 10^{-5} \pm 1,1 \cdot 10^{-6}$
$P_{д}$ с перемежением	$1,0 \cdot 10^{-4} \pm 2,0 \cdot 10^{-6}$	$1,4 \cdot 10^{-5} \pm 1,0 \cdot 10^{-6}$	$4,0 \cdot 10^{-7} \pm 1,0 \cdot 10^{-7}$	$< 1 \cdot 10^{-7}$

Для измерений были использованы результаты анализа статистических свойств потока ошибок в реальном канале и определены параметры дискретного канала для моделирования процесса передачи информации. Для моделирования использованы характеристики файла ошибок *errMass3.txt*, наиболее характерного для существующей аппаратуры ГАСС.

Результаты измерений приведены в табл. 7.

В этом случае (табл. 7) использовали сверточный код (404,202), декодер пороговый, 6 итераций; $P_{д}$ — вероятность декодирования; H_p , H_c — регулярная и случайная составляющие отношения сигнал/шум; память в канале $Q = 0,0021$, что соответствует массиву *errMass3*. Измерялось полное время кодирования и декодирования (включая запись и считывание файлов) для 100 кадров изображения.

Заключение

В результате исследования УЗО для цифровой передачи изображений по гидроакустическому каналу связи с размером кадра $256 \times 192 \times 4$ двоичных символов были проанализированы:

— статистические характеристики потока ошибок при цифровой передаче изображений по ГАКС в существующей аппаратуре ГАСС;

— методы повышения достоверности передачи блоков цифровой информации по ГАКС при наличии копии передаваемого блока в пункте приема.

Результаты испытаний УЗО для цифровой передачи изображений в аппаратуре гидроакустической связи дают основание сделать следующие выводы:

- устройство обеспечивает передачу цифровой информации с модуля автономного подводного робота на модуль обслуживающего судна с требуемой достоверностью;
- анализ результатов табл. 6, 7 зависимости вероятности ошибки на выходе модуля УЗО от качества дискретного канала связи позволяет определить требования к отношению сигнал/шум на входе приемного тракта судовой части и, соответственно, требования к мощности акустического сигнала передающего подводного устройства в зависимости от глубины его нахождения.

Список литературы

1. Зюко А. Г., Фалько А. И., Панфилов В. Л., Банкет В. Л., Ивашенко П. В. Помехоустойчивость и эффективность систем передачи информации / под ред. А. Г. Зюко. М.: Радио и связь, 1985. 272 с.
2. Филиппов Б. И., Чернецкий Г. А. Анализ статистических характеристик сигналов и помех в гидроакустических каналах связи // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. 2015. № 3. С. 78—84.
3. Филиппов Б. И. Определение наклонной дальности между судном и донной станцией // Вестник РГРТУ. 2016. № 55. С. 33—40.
4. Филиппов Б. И. Энергетический расчет гидроакустических линий связи // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. 2016. № 3. С. 67—77.
5. Евтютов А. П., Колесников А. Е., Яликов А. П. Справочник по гидроакустике. Л.: Судостроение, 1982. 344 с.
6. Bluen S., Heard G. J., Pecknold S. Autonomy and networking challenges of future under water systems // Proceedings of the IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE) 2015. IEEE, 2015, pp. 1514—1519.
7. Федосов В. П., Легин А. А., Ломакина А. В. Алгоритмы, основанные на технологии MIMO-OFDM, для реализации цифрового гидроакустического канала связи // Известия ЮФУ. Технические науки. 2015. № 7 (168). С. 148—158.
8. Литвинцева А. В., Оболонин М. А. Использование линейного предсказателя речи в программной модели низкоскоростного вокодера для передачи речи по гидроакустическому каналу связи // Современные проблемы науки и образования. 2013. № 3, С. 232—240.
9. Kekal K. G., Kebkal V. K., Kebkal A. G., Petroccia R. Experimental Estimation of Delivery Success of Navigation Data Packages transmitted via Digital Hydroacoustic Communication Channel // Gyroscopy and navigation. 2016. Vol. 7, N. 4. P. 343—352.
10. Khan A. I. R., Gang Q., Mehboob K. Investigation of channel modeling and simulation of ofdm based communication near northern regions of arabian sea // Res. J. Appl. Sci. Eng. Technol. 2013. Vol. 5, N. 4. P. 1169—1182.
11. Huang J., Babeau M., Blouin S., Hamm C., Taillefer N. Simulation and Modeling of Hydro Acoustic Communication Channels with Wide Attenuation and Ambient Noise // International Journal of Parallel, Emergent and Distributed Systems. 2017. N. 2. P. 1—21.
12. Анализатор характеристик гидроакустических каналов связи. Отчет / НЭИС им. Н. Д. Псурцева. Научные руководители: А. А. Макаров, Г. А. Чернецкий. № ГР 0184000588, Инв. № 02860021414, Новосибирск, 1985. 77 с.
13. Макаров А. А. АРМ исследования и проектирования систем передачи информации: учеб. пособие. Новосибирск: СибГУТИ, 2001. 78 с.
14. Filippov B. I., Chernetsky G. A. Increase of Reliability of Transfer of Blocks the Digital Information on the Hydroacoustic Communication Channel // Journal of Federal University. Engineering & Technologies. 2016. Vol. 9, N. 4. P. 489—499.
15. Filippov B. I., Chernetsky G. A. Application of Error — Correcting Coding in Hydroacoustic Communication Channels // Journal of Siberian Federal University. Engineering & Technologies. 2017. Vol. 10, N. 3. P. 377—387.

Research and Development of the Device of Protection against Mistakes for System of Transfer of Images on the Hydroacoustic Communication Channel

The most effective remedy of providing a high noise stability of difficult system is entering of the redundancy necessary for the detection and error correction arising during the work of system and its elements. Theoretical base of effective use of the entered redundancy is the theory of noiseproof coding. As show pilot studies, statistical characteristics of hydroacoustic communication channels (HACC) have the analogs in short-wave, VHF and other radio channels with variable parameters. Therefore the principles and methods of protection against mistakes developed for these channels can be used and in the systems of information transfer using GAKS is final taking into account specific properties of distribution of acoustic signals in the water environment. Therefore the subject of work is represented urgent. For reasons for the choice of parameters of the developed device of protection against mistakes (DPM) researches of statistical characteristics of a flow of mistakes in HACC of the existing equipment of the hydroacoustic system of communication (HASC) by handling of the provided records (files) of samples of a personnel of experimental transfer of images with use of this equipment were executed. By comparison of samples of the transferred and accepted personnel of the image massifs of mistakes in the channel of the existing GASS equipment were received. In total four files with massifs of mistakes were received: *errMass 1.txt*, *errMass 2.txt*, *errMass 3.txt*, *errMass 4.txt*. Each of them contains about $3 \cdot 10^3$ binary symbols. These files were used for determination of statistical characteristics of a flow of mistakes in a real hydroacoustic communication channel, and also for statistical testing of the developed DPM. In work are considered: the principles and algorithms of increase in reliability of digital transmission of images in HASC; the algorithm of implementation of DPM in the conditions of hindrances in a HASC equipment communication channel is developed; the state-of-the-art review of methods of increase in reliability of transfer of blocks of digital information on HACC is made. Results of testing of DPM for digital transmission of images in HASC give the grounds to draw the following conclusions: offered DPM provides digital data transmission from the module of the autonomous underwater robot on the module of the serving vessel with required reliability; the received dependences of probability of mistakes at the DPM module exit from quality of a discrete communication channel allow to determine requirements to the relation signal/noise on an entrance of a reception path of a ship part and, respectively, the requirement to the size of capacity of an acoustic signal of the sending device depending on depth of its stay.

Keywords: hydroacoustic communication channel, hydroacoustic system of communication, digital data transmission, increase in reliability, noiseproof coding, statistical characteristics of a stream of mistakes, device of protection against mistakes

References

1. Zjuko A. G., Fal'ko A. I., Panfilov I. P., Banket V. L., Ivashchenko P. V. *Pomehoustojchivost' i jeffektivnost' sistem peredachi informacii* (Noise stability and efficiency of systems of information transfer) / pod red. A. G. Zjuko. M.: Radio i svjaz', 1985. 272 p.
2. Filippov B. I., Cherneckij G. A. *Vestnik AGTU. Serija: Upravlenie, vychislitel'naja tehnika i informatika*, 2015, no. 3, pp. 78–84.
3. Filippov B. I. Opredelenie naklonnoj dal'nosti mezhdju sudnom i donnoj stanciej, *Vestnik RGRTU*, 2016, no. 55, pp. 33–40.
4. Filippov B. I. Jenergeticheskij raschjot gidroakusticheskikh linij svjazi (Power calculation of hydroacoustic communication lines), *Vestnik AGTU. Serija: Upravlenie, vychislitel'naja tehnika i informatika*, 2016, no. 3, pp. 67–77.
5. Evtjutov A. P. *Spravochnik po gidroakustike* (Reference book on hydroacoustics). Leningrad, Sudostroenie, 1982. 344 p.
6. Bluen S., Heard G. J., Pecknold S. Autonomy and networking challenges of future underwater systems, *Proceedings of the IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE) 2015*, IEEE, 2015, pp. 1514–1519.
7. Fedosov V. P., Legin A. A., Lomakina A. V. Algorithms based on MIMO-OFDM technology for realization of digital hydroacoustic communication channel. *News of SFedU. Technical science*, 2015, no. 7 (168), pp. 148–158.
8. Litvintseva A. V., Obolonin M. A. Using a linear predictor of speech in a software model of a low-speed vocoder for voice transmission over a hydroacoustic communication channel, *Modern problems of science and education*, 2013, no. 3, pp. 232–240.
9. Kekal K. G., Kebkal V. K., Kebkal A. G., Petroccia R. Experimental Estimation of Delivery Success of Navigation Data Packages transmitted via Digital Hydroacoustic Communication Channel, *Gyroscopy and navigation*, 2016, vol. 7, no. 4, pp. 343–352.
10. Khan A. I. R., Gang Q., and Mehboob K. Investigation of channel modeling and simulation of ofdm based communication near northern regions of arabian sea, *Res. J. Appl. Sci. Eng. Technol.*, 2013, 5 (4), pp. 1169–1182.
11. Huang J., Babeau M., Blouin S., Hamm C., Taillefer N. Simulation and Modeling of Hydro Acoustic Communication Channels with Wide Attenuation and Ambient Nois., *International Journal of Parallel, Emergent and Distributed Systems*, 2017, no. 2, pp. 1–21.
12. Analizator karakteristik gidroakusticheskikh kanalov svjazi. Otchet (Analyzer of characteristics of hydroacoustic communication channels. Report) / NJeIS im. N. D. Psurceva. Nauchnye rukovoditeli: A. A. Makarov, G. A. Cherneckij. no. GR 0184000588, Inv. no. 02860021414, Novosibirsk, 1985. 77 p.
13. Makarov A. A. ARM issledovanija i proektirovanija sistem peredachi informacii (Automated workplace of a research and design of systems of information transfer) / ucheb. posobie, SibGUTI, Novosibirsk. 2001. 78 p.
14. Filippov B. I., Cherneckij G. A. Increase of Reliability of Transfer of Blocks the Digital Information on the Hydroacoustic Communication Channel, *Journal of Federal Universiti. Engineering & Technologies*, 2016, vol. 9, no. 4, pp. 489–499.
15. Filippov B. I., Cherneckij G. A. Application of Error — Correcting Coding in Hydroacoustic Communication Channels, *Journal of Siberian Federal University, Engineering & Technologies*, 2017, no. 10 (3), pp. 377–387.

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ NEUROTECHNOLOGIES

УДК 519.7

Е. А. Андреева, д-р физ.-мат. наук, проф., e-mail: andreeva.tvgu@yandex.ru,
И. С. Храмов, аспирант, e-mail: igorhramov@yahoo.com,
Тверской государственный университет

Алгоритм построения приближенного оптимального решения задач на основе искусственных нейронных сетей с учетом запаздывания

Рассматривается дискретная задача оптимального управления с запаздыванием по векторам состояния и управления для искусственных нейронных сетей, приводится алгоритм построения оптимального решения для данной задачи.

Ключевые слова: искусственные нейронные сети, оптимизация управления, необходимые условия оптимизации

Введение

Интерес к искусственным нейронным сетям значительно вырос за последние несколько лет. Это вызвано как теоретическими, так и прикладными достижениями нейроинформатики, что вернуло возможности использования вычислений в сферах, до этого относящихся лишь к области человеческого интеллекта, и возможности создания машин, способность которых учиться и запоминать в точности напоминает мыслительные процессы человека.

Искусственные нейронные сети предложены для огромного числа задач, в том числе для распознавания лиц, геномной инженерии, робототехники и др.

Суть всех подходов нейроинформатики — разработка методов создания (синтеза) нейронных схем, решающих те или иные задачи. Нейрон при этом выглядит очень просто: устройство (сумматор с активационной функцией) с большим числом входов и одним выходом. Различие между подходами и методами — в деталях представлений о работе нейрона и, конечно, в представлениях о работе связей. Устройства нейроинформатики — это связевые системы. В отличие от цифровых микропроцессорных систем, являющихся сложными комбинациями процессорных и запоминающих блоков, нейропроцессоры содержат память, распределенную в связях между нейронами. Тем самым основная нагрузка на выполнение конкретных функций процессорами ложится на архитектуру системы, детали которой, в свою очередь, определяются межнейронными связями.

В настоящее время большое внимание уделяется моделированию и исследованию искусственных нейронных сетей, программа управления которыми должна работать в реальном времени, обладать

способностью к обучению, максимально полно использовать поступающую извне информацию, иметь память о прошлых ситуациях, обладать способностью непрерывного обобщения и классификации поступающей информации (рис. 1).

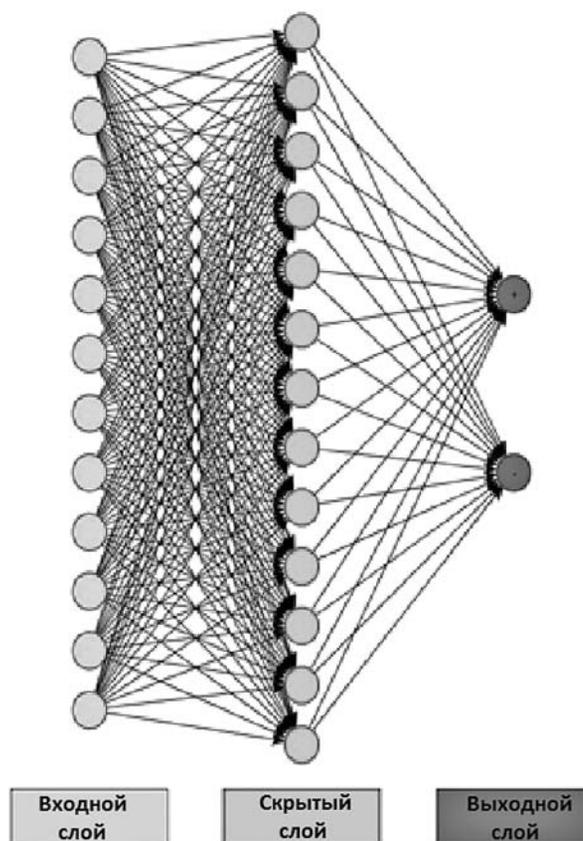


Рис. 1. Пример архитектуры нейронной сети

В статье рассмотрена проблема моделирования и обучения искусственной нейронной сети (ИНС) достаточно общей топологии, описываемой системой рекуррентных соотношений с запаздыванием. Для решения этой задачи используется аппарат математической теории оптимального управления, позволяющий для различных критериев определить оптимальные весовые коэффициенты нейронной сети и методы нелинейного программирования [2].

Математическая постановка задачи

Прежде чем перейти к исследованию дискретных моделей ИНС (рис. 2), рассмотрим дискретную задачу оптимального управления с запаздыванием по векторам состояния и управления.

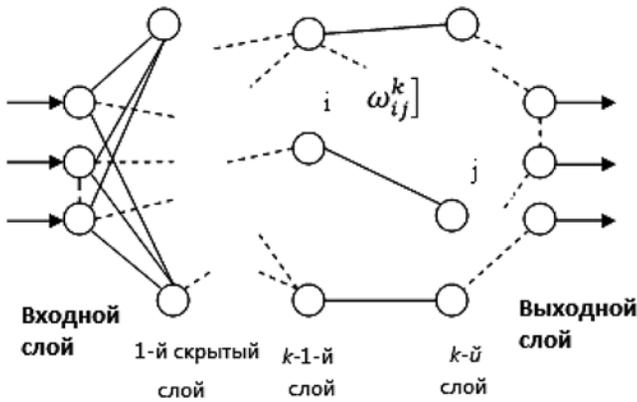


Рис. 2. Рассматриваемая модель ИНС

Предположим, что в системе имеется q векторов. Для удобства расчетов будем обозначать индексы пересчета i, j, n и m . Большими буквами обозначим матрицы и векторы, малыми — соответствующие элементы матриц и векторов.

Дискретная система на каждом шаге k характеризуется набором векторов x^{k-v}, x^{k-v+1}, x^k и векторов управления u^{k-p}, u^k . Терминальная функция $\Phi(x^q)$ характеризует состояние системы в момент времени q :

$$I([u]) = \sum_{k=0}^{q-1} f_k^0(x^{k-n}, \dots, x^k, u^{k-m}, \dots, u^k) + \Phi(x^q). \quad (1)$$

Переход из k -го состояния в $(k+1)$ -е описывается рекуррентными соотношениями

$$x^{k+1} = f_k(x^{k-n}, \dots, x^k, u^{k-m}, \dots, u^k), \quad k = \overline{0, q-1}; \quad (2)$$

$$x^k = b^k \in \mathbb{R}^l, \quad k = -v, \dots, 0; \quad (3)$$

$$u^k = u^k \in \mathbb{R}^r, \quad k = -p, \dots, 0 \quad (4)$$

с заданными начальными условиями

$$z^k = (x^{k-n}, \dots, x^k, u^{k-m}, u^k); \quad [u] = [u^0, \dots, u^{q-1}];$$

$$[\tilde{u}] = [u^0, \dots, u^{i-1}, u^i + \Delta, u^{i+1}, \dots, u^{q-1}];$$

$$[x] = [x^0, \dots, x^q];$$

$$[\tilde{x}] = [x^0, \dots, x^{i-1}, x^i + \Delta, x^{i+1}, \dots, x^q].$$

Дадим приращение Δ вектору x^i , тогда последующие векторы $\tilde{x}^{i+1}, \dots, \tilde{x}^q$ вычисляются согласно формуле (2):

$$\tilde{x}^{i+1} = f_i(x^{i-n}, \dots, x^{i-1}, x^i + \Delta, \dots, x^{q-1}).$$

Определим сопряженный вектор

$$p^i = \frac{dI([x], [u])}{dx^i} = \lim_{\Delta \rightarrow 0} \frac{I([\tilde{x}][\tilde{u}]) - I([x][u])}{\Delta}. \quad (5)$$

Из (5) следует, что если $i = q$, то имеет место равенство

$$p^q = \frac{dI([x][u])}{dx^q} = \frac{\partial \Phi(x^q)}{\partial x^q}; \quad (6)$$

если $i = q-1$, то

$$\begin{aligned} p^{q-1} &= \frac{\partial I}{\partial x^{q-1}} + \left(\frac{\partial x^q}{\partial x^{q-1}} \right)^T \frac{\partial I}{\partial x^q} = \\ &= \frac{\partial}{\partial x^{q-1}} [f_{q-1}^0(z^{q-1}) + (p^q, f_{q-1}(z^{q-1}))]. \end{aligned} \quad (7)$$

В общем случае имеем

$$\begin{aligned} p^i &= \frac{\partial I}{\partial x^i} + \left(\frac{\partial x^{i+1}}{\partial x^i} \right)^T \frac{\partial I}{\partial x^{i+1}} + \dots + \left(\frac{\partial x^{i+n+1}}{\partial x^i} \right)^T \frac{\partial I}{\partial x^{i+n+1}} = \\ &= \frac{\partial f_i^0(z^i)}{\partial x^i} + \dots + \frac{\partial f_{i+n}^0(z^{i+n})}{\partial x^i} + \left[\frac{\partial f_i(z^i)}{\partial x^i} \right]^T p^{i+1} + \dots \\ &\quad \dots \left[\frac{\partial f_{i+n}^0(z^{i+n})}{\partial x^i} \right] p^{i+n+1} = \\ &= \sum_{l=0}^n \frac{\partial}{\partial x^i} [\partial f_{i+l}^0(z^{i+l}) + (p^{i+l+1}, f_{i+l}(z^{i+l}))], \\ &\quad i = \overline{1, q-1}, \quad p^j = 0, \quad j > q. \end{aligned} \quad (8)$$

Введем функцию $H_k(z^k, p^{k+1}) = \lambda_0 f_k^0(z^k) + (p^{k+1}, f_k(z^k))$, где λ_0 — соответствующий коэф-

фициент. Тогда (8) с помощью этой функции преобразуется следующим образом:

$$p^i = \sum_{l=0}^v \frac{\partial H_{i+l}(z^{i+l}, p^{i+l+1})}{\partial x^i}, \quad i = \overline{q-1, l}. \quad (9)$$

Множество векторов $[x] = [x^1, \dots, x^q]$ зависит от начального набора состояния x^0 и допустимых векторов $[u] = [u^0, \dots, u^{q-1}]$ или $[x] = [x(u)]$, и нам необходимо минимизировать сложную функцию $I = I([x(u)], [u])$ аргумента u , полная производная функции которой по вектору u^i определяется выражением

$$\begin{aligned} \frac{dI([x(u)], [u])}{du^i} &= \\ &= \frac{dI(x(u), [u])}{du^i} + \sum_{l=0}^m \frac{\partial x^{i+l+1}[u]}{\partial u^i} \frac{dI([x(u)], [u])}{dx^{i+l+1}} = \\ &= \sum_{l=0}^m \frac{\partial f_{i+l}^0(z^{i+l})}{\partial u^{i+l}} + \left[\frac{\partial f_{i+l}^0(z^{i+l})}{\partial u^{i+l}} \right]^T p^{i+l+1} = \\ &= \sum_{l=0}^m \frac{\partial}{\partial u^i} H_{i+l}(z^{i+l}, p^{i+l+1}). \end{aligned} \quad (10)$$

Функция Лагранжа для дискретной задачи оптимального управления (1)–(4) имеет вид

$$\begin{aligned} L([x], [u], [p], \lambda_0) &= \\ &= \sum_{i=0}^{q-1} [\lambda_0 f_i^0(z^i) + (p^{i+1}, f_i(z^i) - x^{i+1})] + \lambda_0 \Phi(x^q). \end{aligned}$$

Чтобы получить необходимые условия оптимальности, вычислим производные функции Лагранжа по переменным $x^i, i = \overline{1, q}, u^i, i = \overline{1, q-1}$:

$$\begin{aligned} \frac{\partial L}{\partial x^i} &= \sum_{l=0}^v \frac{\partial H_{i+l}(z^{i+l}, p^{i+l+1})}{\partial x^i} - p^i = 0, \quad i = \overline{1, q-1}, \\ \frac{\partial L}{\partial x^q} &= \frac{\partial \Phi(x^q)}{\partial x^q} - p^q = 0; \\ \frac{\partial L}{\partial u^i} &= \sum_{l=0}^p \frac{\partial H_{i+l}(z^{i+l}, p^{i+l+1})}{\partial u^i} = 0, \quad i = \overline{0, q-1}. \end{aligned} \quad (11)$$

Сравнивая формулы (10) и (11), получим следующее равенство:

$$\frac{\partial L}{\partial u^i} = \frac{dI([x(u)], [u])}{du^i}, \quad i = \overline{0, q-1}. \quad (12)$$

Формула (12) может использоваться для получения приближенного оптимального решения, например, градиентным методом.

Если в задаче (1)–(4) присутствуют ограничения на вектор управления $u^i \in U_i, i = \overline{0, q-1}$, например

$$\begin{aligned} U_i &= \{v \in \mathbb{R}^r: g_l^i(v) \leq 0, \\ &h_l^i(v) = 0, \quad l = \overline{k+1, s}\}, \end{aligned} \quad (13)$$

то можно использовать метод Лагранжа с новой функцией Лагранжа. В этом случае новая функция Лагранжа \tilde{L} для задачи (1)–(4) определяется выражением

$$\tilde{L} = L + \sum_{i=0}^{q-1} \left[\sum_{l=0}^k \mu_l^i g_l^i(v) + \sum_{l=k+1}^s \nu_l^i h_l^i(v) \right].$$

Для решения задачи (1)–(4) в этом случае можно использовать метод штрафных функций [3].

Рассмотрим модель нейронной сети, в которой присутствует запаздывание по вектору состояния, а управляющими функциями являются весовые коэффициенты нейронной сети [4].

Пусть переход из k -го состояния в $(k+1)$ -е осуществляется по следующему правилу:

$$x^{k+1} = h(x^k, x^{k-v}) + (W_0 + W_k)g(x^{k-v}) \quad (14)$$

или в покомпонентном виде —

$$\begin{aligned} x_i^{k+1} &= h_i(x^k, x^{k-v}) + \sum_{j=1}^n [\omega_{ij}^0 + \omega_{ij}^k] g_j(x^{k-v}), \\ &i = \overline{1, n}, \end{aligned} \quad (15)$$

с начальными условиями $x^i = a^i, i = -v, \dots, 0$; здесь $a^i \in \mathbb{R}^n, i = -v, \dots, 0$ — заданные векторы в \mathbb{R}^n, h, g — заданные n -мерные векторные непрерывно дифференцируемые функции, W_0 — заданная по-

стоянная матрица, $W_k = \{\omega_{ij}^k\}, i, j = \overline{1, n}$ — $n \times n$ -матрица весовых коэффициентов или управлений на k -м шаге, принимающих значения в заданном множестве, например

$$|\omega_{ij}^k| \leq A_{ij}^k, \quad k = 0, \dots, q-1; \quad i, j = \overline{1, n}. \quad (16)$$

Весовые коэффициенты ω_{ij}^k выбираются из условия минимума функции

$$\begin{aligned} I(\omega) &= \Phi(x^q) + \\ &+ \sum_{k=0}^{q-1} E_k(x^k, x^{k-v}) + \frac{1}{2} \sum_{k=0}^{q-1} \sum_{i,j=1}^n r_{ij}^k (\omega_{ij}^k)^2. \end{aligned} \quad (17)$$

В выражении (17) коэффициенты $r_{ij}^k > 0$ при каждом $k = \overline{0, q-1}$ образуют положительно определенную матрицу, которую мы обозначим через $R_0, \omega_{ij}^0, \omega_{ij}^q$ — набор $n \times n$ матриц, $E_k(x^k, x^{k-v})$ — задан-

ные скалярные функции, $x^{k-v} = (x_1^{k-v}, \dots, x_n^{k-v})$, т. е. каждая компонента вектора x^{k-v} имеет свое запаздывание.

В модели (14)–(17) оптимальные весовые коэффициенты $\bar{\omega}_{ij}^k$ определяются равенством

$$\bar{\omega}_{ij}^k = p_i^{k+1} [r_{ij}^k]^{-1} g_j(x^{k-v}), \quad (18)$$

сопряженные векторы удовлетворяют следующим рекуррентным соотношениям:

$$p_i^k = \left[\frac{\partial E_k(\bar{x}^k, \bar{x}^{k-v})}{\partial x_i^k} - \frac{\partial E_{k+v}(\bar{x}^{k+v}, \bar{x}^k)}{\partial x_i^k} \right] + p_i^{k+1} \frac{\partial h_i(x^k, x^{k-v})}{\partial x_i^k} + p_i^{k+1+v} \left[\frac{\partial h_{i+v}(x^{k+v}, x^k)}{\partial x_i^k} + \sum_{j=1}^n \omega_{ij}^{k+1} \frac{\partial g_j(x^k)}{\partial x_i^k} \right] \quad (19)$$

с граничными условиями

$$p_i^q = \lambda_0 \frac{\partial \Phi(x^q)}{\partial x_i^q}, \quad i = \overline{1, n}; \quad p_i^s = 0, \quad s > q; \quad i = \overline{1, n}. \quad (20)$$

Алгоритм построения приближенного оптимального решения

Алгоритм построения приближенного оптимального решения приведен ниже.

1. Зададим произвольный набор матриц

$$(W^k)^{(0)} = \{\omega_{ij}^k\}^{(0)}, \quad i, j = \overline{1, n}, \quad k = \overline{0, q-1},$$

на диагоналях которых стоят нулевые элементы $\omega_{ii} = 0$.

2. По рекуррентным соотношениям (15), с использованием начальных значений $x^i = a^i, i = -v, \dots, 0$ (здесь $a^i \in \mathbb{R}^n, i = -v, \dots, 0$, — заданные векторы в \mathbb{R}^n) и набора матриц $(W^k)^{(0)}$ определим $x_i^k, i = \overline{1, n}, k = \overline{1, q}$. В результате получим набор векторов x^1, \dots, x^q , соответствующий выбранным $(W^k)^{(0)}$, который обозначим $(x^k)^{(0)}$.

3. Вычислим значение функции (17), используя $(x^k)^{(0)}$ и $(W^k)^{(0)}$, и обозначим эту величину $I^{(0)}$. Здесь верхний индекс в скобках соответствует номеру итерации.

4. Найдем сопряженные векторы по формулам (20) и (19) начиная с индекса q .

5. Вычислим коэффициенты $\omega_{ij}^k, i, j = \overline{1, n};$

$k = \overline{0, q-1}$, соответствующие первой итерации

$(W^k)^{(1)} = \{\omega_{ij}^k\}^{(1)}$, по формуле (18) (в общем случае

пользуемся методом градиентного спуска с дроблением шага). С полученными значениями весовых коэффициентов переходим ко второму шагу, используя вместо матриц $(W^k)^{(0)}$ улучшенные матрицы $(W^k)^{(1)}$, элементы которых получены по формуле

$$(\omega_{ij}^k)^{(l+1)} = (\omega_{ij}^k)^{(l)} - \alpha^{(0)} \left(\frac{\partial L}{\partial \omega_{ij}^k} \right)^{(l)}, \quad (21)$$

где $\alpha^{(0)} > 0$ — величина шага градиентного спуска.

6. Аналогично, согласно (15) строим $(x^k)^{(1)}$. Найдем значение минимизируемой функции (17), используя $(x^k)^{(1)}$ и $(W^k)^{(1)}$, и обозначим эту величину $I^{(1)}$. Вычислим приращение.

7. Если $\Delta I^{(1)} > 0$, то заменяем на втором шаге $(W^k)^{(0)}$ на $(W^k)^{(1)}$; если $\Delta I^{(1)} \leq 0$, то при использовании метода градиентного спуска с дроблением шага уменьшаем шаг в два раза или переходим к новому алгоритму. Вычисления продолжаем до тех пор, пока $\Delta I^{(l)} < \varepsilon$, где $\varepsilon > 0$ — заданная точность.

Дискретная задача оптимального управления с запаздыванием

Следующая модель используется по той же схеме. Для построения приближенного оптимального решения может быть использован метод проекции градиента, в котором изменение весовых коэффициентов на двух последовательных итерациях осуществляется по формуле (21).

Рассмотрим следующую дискретную модель нейронной сети с запаздыванием:

$$I = \sum_{l=0}^{q-1} B^l(x^l, \omega^l) + \Phi(x^q) \rightarrow \inf,$$

$$x_i^{l+1} = F_i^l \left(\sum_{j=1}^n \omega_{ij}^{l-v} x_j^{l-v}, \sum_{j=1}^n \omega_{ij}^{l-v-1} x_j^{l-v-1}, \dots, \sum_{j=1}^n \omega_{ij}^l x_j^l \right), \quad x^{-v}, x^{-v+1}, \dots, x^0 \in R;$$

$$\omega_{ij}^{-v}, \dots, \omega_{ij}^{-1} \text{ заданы, } |\omega_{ij}^l| \leq A_{ij}^l,$$

$$l = \overline{0, q-1}, \quad i, j = \overline{1, n}.$$

Для определенности в данном случае обозначим активационные функции нейронной сети как $B(x, \omega)$.

Функция Лагранжа для этой задачи имеет следующий вид:

$$L = \lambda_0 \sum_{l=0}^{q-1} B^l(x^l, \omega^l) + \lambda_0 \Phi(x^q) + \sum_{l=0}^{q-1} \sum_{i=1}^n p_i^{l+1} \left(x_i^{l+1} - F_i^l \left(\sum_{j=1}^n \omega_{ij}^{l-v} x_j^{l-v}, \sum_{j=1}^n \omega_{ij}^{l-v-1} x_j^{l-v-1}, \dots, \sum_{j=1}^n \omega_{ij}^l x_j^l \right) \right).$$

Вычислим производные функции Лагранжа:

$$\begin{aligned} \frac{\partial L}{\partial x_m^k} &= \lambda_0 \frac{\partial B^k(x^k, \omega^k)}{\partial x_m^k} + p_m^k - \\ &- \sum_{i=1}^n p_i^{k+v+1} \frac{\partial F_i^{k+v}(z_i^k, \dots, z_i^{k+v})}{\partial z_i^k} \frac{\partial z_i^k}{\partial x_m^k} - \\ &- \sum_{i=1}^n p_i^{k+v} \frac{\partial F_i^{k+v-1}(z_i^{k-1}, z_i^k, \dots, z_i^{k+v-1})}{\partial z_i^k} \frac{\partial z_i^k}{\partial x_m^k} - \\ &- \dots - \sum_{i=1}^n p_i^{k+1} \frac{\partial F_i^k(z_i^{k-v}, \dots, z_i^k)}{\partial z_i^k} \frac{\partial z_i^k}{\partial x_m^k}, \\ &k = 1, \dots, q-1, m = \overline{1, n}, \end{aligned}$$

где

$$z_i^k = \sum_{j=1}^n \omega_{ij}^k x_j^k; \quad \frac{\partial z_i^k}{\partial x_m^k} = \omega_{im}^k; \quad \frac{\partial z_s^k}{\partial x_{sp}^k} = x_p^k.$$

Учитывая определения z_i^k , получим следующие выражения для производных функции Лагранжа:

$$\begin{aligned} \frac{\partial L}{\partial x_m^k} &= \lambda_0 \frac{\partial B^k(x^k, \omega^k)}{\partial x_m^k} + p_m^k - \\ &- \sum_{i=1}^n p_i^{k+v+1} \frac{\partial F_i^{k+v}(z_i^k, \dots, z_i^{k+v})}{\partial z_i^k} \omega_{im}^k - \\ &- \sum_{i=1}^n p_i^{k+v} \frac{\partial F_i^{k+v-1}(z_i^{k-1}, z_i^k, \dots, z_i^{k+v-1})}{\partial z_i^k} \omega_{im}^k - \\ &- \dots - \sum_{i=1}^n p_i^{k+1} \frac{\partial F_i^k(z_i^{k-v}, \dots, z_i^k)}{\partial z_i^k} \omega_{im}^k, \end{aligned}$$

$$k = 1, \dots, q-1, i, m = \overline{1, n}, p_i^q = -\lambda_0 \frac{\partial \Phi(x^q)}{\partial x_i^q},$$

$$p_i^l = 0, l > q, i = \overline{1, n};$$

$$\begin{aligned} \frac{\partial L}{\partial \omega_{sp}^k} &= \lambda_0 \frac{\partial B^k(x^k, \omega^k)}{\partial \omega_{sp}^k} + p_m^k - \\ &- \sum_{i=1}^n p_i^{k+v+1} \frac{\partial F_i^{k+v}(z_i^k, \dots, z_i^{k+v})}{\partial z_i^k} \frac{\partial z_i^k}{\omega_{sp}^k} - \\ &- \dots - \sum_{i=1}^n p_i^{k+1} \frac{\partial F_i^k(z_i^{k-v}, \dots, z_i^k)}{\partial z_i^k} \frac{\partial z_i^k}{\omega_{sp}^k} = \\ &= \lambda_0 \frac{\partial B^k(x^k, \omega^k)}{\partial \omega_{sp}^k} - p_s^{k+v+1} \frac{\partial F_s^{k+v}(z_s^k, \dots, z_s^{k+v})}{\partial z_s^k} x_p^k - \\ &- p_i^{k+v} \frac{\partial F_s^{k+v-1}(z_s^{k-1}, z_s^k, \dots, z_s^{k+v-1})}{\partial z_s^k} x_p^k \dots - \\ &- p_s^{k+1} \frac{\partial F_s^k(z_s^{k-v}, \dots, z_s^k)}{\partial z_s^k} x_p^k, k=1, \dots, q-1, s, p = \overline{1, n}. \end{aligned}$$

Заключение

Обобщая вышесказанное, можно сделать заключение, что описанные методы моделирования и обучения искусственной нейронной сети общей топологии, описываемой системой рекуррентных соотношений с запаздыванием, являются универсальными, что позволяет использовать их в различных областях для решения широкого спектра прикладных задач.

Список литературы

1. Андреева Е. А., Евтушенко Ю. Г. Численные методы решения задач оптимального управления для систем, описываемых интегро-дифференциальными уравнениями типа Фредгольма // Модели и методы оптимизации. 1989. № 1. С. 4—13.
2. Андреева Е. А., Пустырникова Ю. А. Численные методы обучения искусственных нейронных сетей с запаздыванием // ЖВМ и МФ. 2002. Т. 42. С. 1383—1391.
3. Евтушенко Ю. Г. Оптимизация и быстрое автоматическое дифференцирование. М.: ВЦ РАН, 2013. 144 с.
4. Андреева Е. А., Мазурова И. С. Численный метод обучения искусственной нейронной сети с учетом запаздывания при передаче сигнала // ТЕОРЕТИЧЕСКИЕ И ПРАКТИЧЕСКИЕ ВОПРОСЫ НАУКИ XXI ВЕКА. Сб. статей Международной научно-практической конференции, 2014. С. 68—72.
5. Галушкин А. И. Нейронные сети: основы теории. Научное издание, М.: Горячая Линия — Телеком, 2015. 496 с.
6. Андреева Е. А., Мазурова И. С. Применение метода быстрого автоматического дифференцирования для оптимизации систем интегро-дифференциальных уравнений // Вестник Кемеровского государственного университета. 2014. № 4—2 (60). С. 47—53.

Optimization of the Artificial Neural Network with Allowing for Delay

This article discusses the discrete optimal control problem with delay in state and control vectors for artificial neural networks, the algorithm for constructing an optimal solution for this problem. In the process of work, a neural network model is considered, in which there is a lag in the state vector, and the control functions are the weights of the neural network. To construct an approximate optimal solution, it is proposed to use the gradient projection method. The described methods of modeling and learning the artificial neural network of the general topology described by the system of recurrence relations with delay are universal, which allows them to be used in various fields to solve a wide range of applied problems.

Keywords: artificial neural networks, optimization of management, necessary conditions for optimization

References

1. **Andreeva E. A., Evtushenko Yu. G.** Chislennyye metody resheniya zadach optimal'nogo upravleniya dlja sistem, opisyvaemyh integro-differencial'nymi uravnenijami tipa Fredgol'ma. [Numerical methods for solving optimal control problems for systems being described's integral-differential equations of Fredholm type], *Models and methods of optimization*, 1989, no. 1, pp. 4–13 (in Russian).
2. **Andreeva E. A., Pustyrnakova Y. A.** Chislennyye metody obuchenija iskusstvennyh neyronnyh setej s zapazdyvanijem. [Numerical methods for training artificial neural networks with delay], *Zh. GVM I MF*. 2002, vol. 42, pp. 1383–1391 (in Russian).
3. **Evtushenko Yu. G.** *Optimizacija i bystroje avtomaticheskoe differencirovanie*. [Optimization and fast automatic differentiation.], Scientific publication CCAS, Moscow, 2013, 144 p. (in Russian).
4. **Andreeva E. A., Mazurova I. S.** Chislennyyj metod obuchenija iskusstvennoj neyronnoj seti s uchetom zapazdyvanija pri peredache signala. [Numerical method of training artificial neural network based on delay during signal transmission], *TEORETICHESKIE I PRAKTICHESKIE VOPROSY NAUKI XXI VEKA*. Sbornik statej Mezhdunarodnoj nauchno-prakticheskoj konferencii, 2014, pp. 68–72 (in Russian).
5. **Galushkin A. I.** *Nejronnye seti: osnovy teorii*. [Neural networks: basic theory]. Scientific publication. Moscow, Goryachaya Liniya — Telecom, 2015. 496 p. (in Russian).
6. **Andreeva E. A., Mazurova I. S.** Primenenie metoda bystrogo avtomaticheskogo differencirovanija dlja optimizacii sistem integro-differencial'nyh uravnenij. [Application of fast automatic differentiation for optimization systems of integral-differential equations.], *Vestnik of Kemerovo State University*, 2014, no. 4–2 (60), pp. 47–53 (in Russian).

Указатель статей, опубликованных в журнале "Информационные технологии" в 2017 г.

ОБЩИЕ ВОПРОСЫ

Мусаев А. А. К оцениванию эффективности информационных систем. Методологические аспекты № 5

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Алгулиев Р. М., Набибекова Г. Ч.** Методы определения степени интеграции стран в среде электронного государства . . . № 5
- Астафуров В. Г., Евсюткин Т. В., Аксенов С. В.** Эффективный алгоритм формирования нечетких систем на основе модели Питтсбургского классификатора № 7
- Вишнеков А. В., Иванова Е. М.** Применение методов поддержки принятия решений при построении сетей доступа . № 9
- Волобуев М. Ф., Мальцев А. М., Соколов И. В.** Экспериментальные исследования макета резервированной системы, управляемой нечетким контроллером № 7
- Ермаков А. Е.** Язык семантических трансформаций для компьютерной интерпретации текста № 6
- Зак Ю. А.** Об одном алгоритме Fuzzy-регрессионного анализа № 12
- Коляда А. А.** Обобщенная интегрально-характеристическая база модулярных систем счисления № 9
- Наумова В. В., Дьяков С. Е., Платонов К. А.** Технологии поиска и представления информации в научной инфраструктуре "Геология Дальнего Востока России" № 1
- Никифоров Д. А., Корж Д. В., Сиваков Р. Л.** Обзор инструментов для валидации XML-документов с помощью правил контроля, описанных на объектном языке ограничений (OCL) № 5
- Полещук О. М.** Использование нечетких логических функций для поддержки принятия решений по результатам рейтингового оценивания № 2
- Разоренов А. А.** Математические основы применения теории К-представлений к разработке алгоритмов исполнения прикладными системами естественно-языковых предписаний № 12
- Разоренов А. А.** Принципы применения теории К-представлений к разработке математических основ исполнения прикладными системами естественно-языковых предписаний № 10
- Разоренов А. А., Фомичев В. А.** Новый подход к формализации семантической обработки предписаний на основе теории К-представлений № 1
- Романов А. Ю., Ломотин К. Е., Козлова Е. С.** Применение методов машинного обучения для решения задачи автоматической рубрикации статей по УДК № 6
- Савченко В. В., Акатьев Д. Ю.** Информационная технология психокоррекции эмоционального состояния пользователя на основе его устного чтения № 11
- Сорокин А. Б., Смольянинова В. А.** Концептуальное проектирование экспертных систем поддержки принятия решений № 9
- Стемповский А. Л., Левченко Н. Н., Окунев А. С., Климов А. В., Змеев Д. Н.** Программирование задачи "молекулярная динамика" в потоковой модели вычислений № 12
- Ульянов М. В., Сметанин Ю. Г.** Об одном характеристическом функционале слов над конечным алфавитом № 5
- Цветков В. Я.** Когнитивные технологии № 2
- Яхьяева Г. Э., Карманова А. А., Ершов А. А., Савин Н. П.** Вопросно-ответная система для управления информационными рисками на основе теоретико-модельной формализации предметных областей № 2

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Акимкина Э. Э.** Инструментальный подход к организации сбора данных в хранилище систем поддержки принятия решений № 6
- Акопов А. С., Бекларян А. Л., Хачатрян Н. К., Фомин А. В.** Система прогнозирования динамики добычи нефти с использованием имитационного моделирования № 6
- Аникин В. И., Аникина О. В.** Визуальное компонентное моделирование систем нечеткого вывода в электронных таблицах № 8
- Бульчев Г. Г.** Метод пространственных характеристик в задачах механики деформируемого твердого тела. Часть 1 № 1
- Бульчев Г. Г.** Метод пространственных характеристик в задачах механики деформируемого твердого тела. Часть 2 . № 2
- Бульчев Г. Г.** Моделирование динамики полого цилиндра методом пространственных характеристик № 6
- Валеева А. Ф., Гончарова Ю. А.** Об одном подходе к решению задач операционного планирования по доставке однородной продукции различным клиентам. Часть 2 № 4
- Вересников Г. С., Панкова Л. А., Пронина В. А.** Использование неопределенного программирования в проектировании . № 4
- Волосатова Т. М., Воронина П. А., Козарь А. А.** Имитационное моделирование угроз информационной безопасности систем модульной авионики № 11
- Гридин В. Н., Рыжов Н. Г., Анисимов В. И., Абухазим М. М.** Методы повышения эффективности процессов моделирования динамических режимов нелинейных систем . . № 11
- Деменков Н. П., Микрин Е. А., Мочалов И. А.** Нечеткое преобразование Лапласа в задачах нечеткого математического моделирования. Часть 1 № 4
- Деменков Н. П., Микрин Е. А., Мочалов И. А.** Нечеткое преобразование Лапласа в задачах нечеткого математического моделирования. Часть 2 № 5
- Димов Э. М., Маслов О. Н., Сухова С. В.** Имитационное моделирование деятельности отдела менеджмента качества на базе платформы AnyLogic № 3
- Казанов П. В.** Модель параллельных вычислений для генетических алгоритмов многокритериальной оптимизации, ориентированная на графические процессорные устройства с архитектурой CUDA № 12
- Кишлаков Д. Л., Тараканов П. В., Шашурин Г. В., Берчун Ю. В.** Эффективность облачных вычислений в моделировании кинетики трещин в наводороженных элементах конструкций № 2
- Курейчик В. В., Бова В. В., Лещанов Д. В.** Модель семантического поиска в системах управления знаниями на основе генетических процедур № 12
- Левин В. И.** Моделирование систем с неопределенностью с помощью полиинтервальной математики № 5
- Левин В. И.** Метод раздетерминизации для труднорешаемых задач математического моделирования № 3
- Левин В. И.** Многомерные интервалы, их исчисление и применение № 7
- Литвин В. Г., Литвин Ю. В.** Об оценке времени разработки крупномасштабной информационно-управляющей системы, создаваемой в структуре системы систем № 7
- Литвин Ю. В., Ноздрин К. С., Полков А. А.** Система виртуального прототипирования и моделирования информационно-управляющих систем реального времени № 5
- Маничев В. Б., Митенкова Е. Ф., Жук Д. М., Кожевников Д. Ю., Соловьев А. В., Соловьева Е. В.** Использование AL-устойчивых методов решения систем ОДУ для задач изотопной кинетики реакторных систем № 3
- Низамутдинов М. М., Орешников В. В.** Концепция реализации системы поддержки принятия решений в сфере управления инновационным развитием регионов на базе адаптивно-имитационной модели № 10
- Прилуцкий М. Х., Власов В. С., Кривошеев О. В.** Задачи оптимального планирования как задачи распределения ресурсов в сетевых канонических структурах № 9

- Светушков Н. Н.** Интегральное описание процесса теплопередачи для численного моделирования высокоскоростной заделки прокатных валков № 11
- Соснина О. А., Филинских А. Д., Коротаева А. С.** Сравнение способов создания низкополигональной 3D-модели. . . № 8
- Титов Д. В., Ретинская И. В., Ретинский В. С.** Моделирование инфраструктуры центра сертификации. № 7
- Третьяков Р. В., Димов Э. М., Маслов О. Н., Халимов Р. Р.** Совершенствование управления процессом разработки программного обеспечения на основе математических моделей и технологии имитационного моделирования. № 11
- Федосов В. В., Федосова А. В.** Многопараметрическая оптимизация в конфликтных задачах промышленной экологии . . . № 10
- Филимонов А. Б., Филимонов Н. Б., Тихонов В. Ю.** Оптимизация распределения ресурсов в задачах пространственного поиска объектов. № 10

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Горбаченко В. И., Катков С. Н.** Мемристорные клеточные сети для решения краевых задач № 4
- Инютин С. А.** Метод вычисления характеристики отношения порядка для параллельных форматов данных № 8
- Родионов А. Н.** Некоторые синтаксические и семантические ограничения сценарных моделей бизнес-процессов. . . № 4

СИСТЕМЫ

АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Анисимов В. И., Васильев С. А., Журавлев Ю. И.** Методы построения клиентских приложений схемотехнических САПР на основе классов наследования. № 10
- Архангельский А. Н.** Методы обработки топологических компонентов при проектировании и изготовлении электронной аппаратуры № 10
- Бурлов В. В., Ремонтова Л. В., Косолапов В. В., Косолапова Е. В.** Моделирование кривой и поверхностей второго порядка в системе КОМПАС-3D № 10
- Курейчик В. М., Сафроненкова И. Б.** Разработка архитектуры СППР по выбору методов решения задач компоновки. № 10
- Печенко И. С., Петров А. Б.** Модели данных для архитектурных спецификаций систем на кристалле № 7
- Стемпковский А. Л., Гаврилов С. В., Гуров С. И., Рыжова Д. И.** Методы проектирования помехозащищенных комбинационных КМОП-схем, обеспечивающие автоматическое исправление ошибок № 10
- Тельпухов Д. В., Соловьев Р. А.** Метод поиска эквивалентных ошибок в логических схемах № 8

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Белим С. В., Ларионов С. Б.** Алгоритм выявления поврежденных пикселей изображения на основе выявления сообществ на графах № 4
- Доморацкий Е. П.** Метод статистической малоракурсной реконструкции геометрических характеристик трехмерных объектов по их дискретным проекционным изображениям. . . № 3
- Кугаевских А. В.** Сравнение параметров фильтра Габор для эффективного извлечения краев. № 8
- Потехин А. С., Стрельников А. В.** Методика оценки транспортного потока на перекрестке по данным видеонаблюдения . № 2
- Стюгин М. А.** Метод построения неразличимого исполняемого кода. № 1
- Терехин А. В.** Алгоритм вычисления инвариантных признаков невыпуклых изображений проекций произвольно расположенных трехмерных объектов. № 11
- Терехин А. В.** Подход к классификации невыпуклых проекций трехмерных объектов по диагональным отрезкам с использованием кортежей. № 9
- Устинов А. А., Дворников С. В., Агеева Н. С.** Научно-методический аппарат адаптивного ортогонального преобразования видеоданных № 2
- Филиппов Б. И.** Передача телеметрической информации по гидроакустическому каналу связи. № 9

- Чан Ван Нгиа.** Разработка параллельной архитектуры кодеров БЧХ и LDPC в стандартах систем цифрового телевизионного вещания второго поколения с регулируемыми параметрами кодирования на базе ПЛИС № 5

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Авдошин С. М., Лазаренко А. В.** Пиринговые ботнеты. . . № 4
- Алгулиев Р. М., Имамвердиев Я. Н., Сухостат Л. В.** Киберфизические системы: основные понятия и вопросы обеспечения безопасности. № 7
- Асратян Р. Э., Лебедев В. Н.** Интернет-служба конвейерной обработки защищенных информационных запросов в мульти-сетевой среде № 8
- Дворников С. В., Дворников С. С., Иванов Р. В., Гулидов А. А., Чихонадских А. П.** Защита от структурных помех радиоканалов с частотной манипуляцией № 3
- Дворников С. В., Пшеничников А. В., Манаенко С. С., Дворников С. С.** Метод формирования многопозиционных помехозащищенных сигнальных конструкций № 9
- Коляда А. А., Коляда Н. А., Протасеня С. Ю., Шабинская Е. В.** Мультипликативно-субтрактивный метод вычисления денормирующего коэффициента для криптографических RSA-преобразований в модулярном коде № 2
- Костин В. Н.** Синтез оптимального размещения технических средств систем физической защиты критически важных объектов. № 1
- Машкина И. В., Куприянов А. О.** Классификационная схема и модели современных вирусных программ. № 6
- Филиппов Б. И.** Разработка и исследование устройства защиты от ошибок для системы передачи изображений по гидроакустическому каналу связи № 12
- Щеглов К. А., Щеглов А. Ю.** Подход к моделированию параметров и характеристик интегрированной информационно-эксплуатационной безопасности № 4
- Щеглов К. А., Щеглов А. Ю.** Сложность реализации угрозы безопасности и математическая модель потенциального нарушителя. № 2
- Щеглов К. А., Щеглов А. Ю.** Эксплуатационная безопасность. Моделирование реализации угроз атак потенциальным нарушителем № 1

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Гаврилова И. Л.** Опыт разработки и внедрения в учебный процесс вуза электронного учебно-методического комплекса . № 6
- Загидуллин Р. Ш., Черников А. С., Чибисов А. А.** Решение проблемы объединения сред MOODLE и Электронный Университет МГТУ им. Н. Э. Баумана № 6
- Решетова Е. Н.** Опыт преподавания учебной дисциплины "Групповая динамика и коммуникация в профессиональной практике программной инженерии" № 12
- Юрин В. Н.** Мониторинг использования компьютерного инжиниринга в инженерном образовании № 10
- Юсупова Н. И., Сметанина О. Н., Гаянова М. М.** Модели и методы обработки информации при анализе образовательных программ (на примере подготовки информатиков) . . . № 6

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИОМЕДИЦИНСКИХ СИСТЕМАХ

- Голов А. В., Тимме Е. А., Козлов А. В.** Алгоритм автоматизированной оценки параметров работоспособности человека при выполнении нагрузочных тестов № 4
- Ефимов Ю. С., Матвеев И. А.** Выделение точных границ радужки на изображении глаза № 4

БАЗЫ ДАННЫХ

- Аскерова Б. Г.** Метод оптимальной энтропийной кластеризации на базе нового определения интересности атрибута состояния элементов сообщения. Часть 1. Теоретическое обоснование № 9
- Саак А. Э.** Диспетчеризация массивов с заявками параболического типа в Grid-системах. № 6

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Беляков С. Л., Белякова М. Л., Савельева М. Н.** Интеллектуальное управление диалогом при визуальном анализе сложных карт и схем № 11
- Головинов Е. Э., Аминев Д. А., Романова И. И.** Алгоритм формирования, передачи и обработки мониторинговой информации от дождевальной машины на основе геоинформационных систем № 9
- Кудряшов А. П., Соловьев И. В.** Реконструкция городской обстановки с учетом рельефа местности с использованием топографического плана и сервисов Google Maps № 5
- Михеева Т. И., Федосеев А. А., Михеев С. В., Головин О. К.** Метод синтеза тематического слоя объектов транспортной сети на основе материалов космической съемки № 11
- Ружичка З. З., Музалевский К. В., Якубайлик О. Э., Швецев Е. Г.** Информационно-технологическое обеспечение задач обработки и представления радиометрических спутниковых данных № 7
- Цветков В. Я.** Сходимость линейных пространственных задач № 9

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОРГАНИЗАЦИОННЫХ И СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ

- Парфенова М. Я.** Интегративный подход к оценке технологической эффективности стратегического профиля информационных технологий организации № 3
- Сумин В. И., Смоленцева Т. Е.** Формирование структуры иерархических многоуровневых организационных систем № 6

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Долженко А. И., Глушенко С. А.** Динамическая валидация данных в WPF-приложениях № 3
- Фонин Ю. Н.** Применение языка PDDL для автоматической генерации средств разработки процессоров № 8

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Авдошин С. М., Песоцкая Е. Ю.** Интернет вещей: Металлургия № 3
- Авдошин С. М., Песоцкая Е. Ю.** Интернет вещей: Ритейл № 8
- Басов И. В., Грошев С. В., Карпенко А. П., Шайтан К. В., Шуров Д. Л.** Метод построения и кластерного анализа карт вероятности заселенности конформаций дипептидов № 9
- Владова А. Ю.** Алгоритмическое обеспечение информационной системы геотехнического мониторинга транспорта углеводородов в условиях многолетней мерзлоты № 3
- Коротченко Р. А.** Применение сингулярного спектрального анализа для обнаружения акустических сигналов от серых китов № 12

- Кряжев С. А., Кузнецова Е. В., Макаров Е. Н.** Управление портфелем ИТ-проектов как инструмент реализации ИТ-стратегии № 11
- Левин В. И.** Моделирование и количественная оценка эффективности научных исследований № 11
- Маликов С. Н.** Концепция сервис-ориентированного подхода к управлению информационными технологиями на основе аксиоматического метода № 3
- Печенко И. С., Венгер О. В., Плоткин Д. А.** AGES Tables: среда для создания и использования табличных спецификаций систем на кристалле № 5
- Полищук Ю. В.** О способе мониторинга информационной энтропии больших технических систем № 1
- Скороходов А. В.** Применение метода Виолы-Джонса для обнаружения атмосферных гравитационных волн над водной поверхностью по спутниковым снимкам MODIS № 1

ДИСКУССИОННЫЙ КЛУБ

- Лосик Г. В.** Антропологическая информация о вариативности сообщения № 2

Журнал в журнале

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Каширина И. Л., Львович Я. Е., Сорокин С. О.** Нейросетевое моделирование формирования кластерной структуры на основе сетей ART № 3
- Михайлюк Т. Е., Жернаков С. В.** Об одном подходе к выбору оптимальной архитектуры ПЛИС в нейросетевом логическом базисе № 3
- Пекунов В. В.** Извлечение информации из нейронных сетей прямого распространения в виде простых алгебраических моделей № 1
- Харламов А. А.** Ассоциативная память — нейросетевая среда для формирования единого пространства знаний № 1

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Андреева Е. А., Храмов И. С.** Алгоритм построения приближенного оптимального решения задач на основе искусственных нейронных сетей с учетом запаздывания № 12
- Барский А. Б.** Нейросетевой метод ускоренного принятия решений операционной системой суперкомпьютера EPIC-архитектуры № 5
- Гейдаров П. Ш.** Нейронные сети прямого распространения с вычисляемыми параметрами № 7
- Кадыева Л. М., Левин С. Е., Окрент Я. Н., Лоскутов Д. И., Романенков А. М.** Использование нейронечетких вычислений для диспетчеризации на железнодорожной станции со сложной топологией № 8
- Печников Д. А.** Применение методов интервальной оценки результатов контроля знаний в компьютерных системах тестирования № 8

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова.*

Корректор *З. В. Наумова.*

Сдано в набор 10.10.2017. Подписано в печать 22.11.2017. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ1217. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

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

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

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