

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 23
2017
№ 10

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Разоренов А. А. Принципы применения теории К-представлений к разработке математических основ исполнения прикладными системами естественных-языковых предписаний 699

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Филимонов А. Б., Филимонов Н. Б., Тихонов В. Ю. Оптимизация распределения ресурсов в задачах пространственного поиска объектов 707
- Низамутдинов М. М., Орешников В. В. Концепция реализации системы поддержки принятия решений в сфере управления инновационным развитием регионов на базе адаптивно-имитационной модели 714
- Федосов В. В., Федосова А. В. Многопараметрическая оптимизация в конфликтных задачах промышленной экологии 721

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Стемпковский А. Л., Гаврилов С. В., Гуров С. И., Рыжова Д. И. Методы проектирования помехозащищенных комбинационных КМОП-схем, обеспечивающие автоматическое исправление ошибок 728
- Курейчик В. М., Сафроненкова И. Б. Разработка архитектуры СППР по выбору методов решения задач компоновки 736
- Бурлов В. В., Ремонтова Л. В., Косолапов В. В., Косолапова Е. В. Моделирование кривой и поверхностей второго порядка в системе КОМПАС-3D 742
- Анисимов В. И., Васильев С. А., Журавлев Ю. И. Методы построения клиентских приложений схематических САПР на основе классов наследования 749
- Архангельский А. Н. Методы обработки топологических компонентов при проектировании и изготовлении электронной аппаратуры 753

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Юрин В. Н. Мониторинг использования компьютерного инжиниринга в инженерном образовании 761

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.

INFORMATION TECHNOLOGIES

INFORMACIONNYYE TEHNOLOGII

Vol. 23

2017

No. 10

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Razorenov A. A.** The Principles of Applying the Theory of K-representations to Developing Mathematical Foundations of Executing Natural Language Instructions by Computer Systems 699

MODELING AND OPTIMIZATION

- Filimonov A. B., Filimonov N. B., Tikhonov V. Yu.** Optimization of the Resources Distribution in the Problems of Space Search Objects 707
- Nizamutdinov M. M., Oreshnikov V. V.** Decision Support System Conception for Regions Innovative Development Management based on Adaptive Simulation Model . 714
- Fedosov V. V., Fedosova A. V.** Multiparameter Optimization Problems in Conflict Industrial Ecology 721

CAD-SYSTEMS

- Stempkovsky A. L., Gavrilov S. V., Gurov S. I., Ryzhova D. I.** Methods of Designing of Fault Tolerance CMOS Combinational Circuits which Provides Automatic Correction of Errors 728
- Kureichik V. M., Safronenkova I. B.** Decision Support System Development for the Choice of Problem-Solving Procedure of Commutation Circuit Partitioning . . . 736
- Burlov V. V., Remontova L. V., Kosolapov V. V., Kosolapova E. V.** Simulation of the Curve and Surfaces of the Second Order in the KOMPAS-3D System 742
- Anisimov V. I., Vasilev S. A., Zhuravlev Yu. I.** Methods for Constructing Client Applications, Circuit CAD Class-Based Inheritance 749
- Arkhangelskiy A. N.** Processing Methods of Topological Components in Design and Manufacturing of the Printed Circuit Boards 753

INFORMATION TECHNOLOGIES IN EDUCATION

- Yurin V. N.** Monitoring the Use of Computer Engineering in Engineering Education . . 761

Editor-in-Chief:

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

Deputy Editor-in-Chief:

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

Chairman:

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

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
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.
Ljovich 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"

Принципы применения теории К-представлений к разработке математических основ исполнения прикладными системами естественно-языковых предписаний

Статья продолжает серию публикаций автора и В. А. Фомичева (Информационные технологии 2015, № 6; 2016, № 12; 2017, № 1), посвященных применению теории К-представлений (концептуальных представлений) к разработке новых методологических основ проектирования компьютерных систем обработки естественно-языковых предписаний. Цель статьи — изложение основных идей определения бинарного отношения нового вида на семантических структурах, послужившего основой нового метода исполнения прикладными системами естественно-языковых предписаний. Это отношение называется отношением соответствия образу. Указано, что предложенный формальный подход был успешно использован при проектировании естественно-языкового интерфейса файлового менеджера NLC-2 (Natural Language Commander — Version 2).

Ключевые слова: обработка естественного языка, семантико-синтаксический анализ, теория К-представлений, СК-языки, управление программным обеспечением, естественно-языковые интерфейсы, отношение соответствия образу, Natural Language Commander

Введение

В последние семь — десять лет значительное число публикаций авторов из разных стран было посвящено проблемам семантической обработки (семантического парсинга) естественно-языковых предписаний (ЕЯП), или команд [1—10]. Эти статьи были основаны на применении различных формализмов семантических представлений (СП) и методов семантического анализа текстов на естественном языке (ЕЯ). Основными приложениями семантического парсинга ЕЯП являются управление роботами, виртуальными персонажами видеоигр, поиск в базах данных и знаний, а также управление программными системами.

Важную роль в обработке естественно-языковых текстов (ЕЯ-текстов) играет представление семантической структуры текста формальными средствами, например, формулами языка логики предикатов первого порядка (ЛППП). Применение формальных средств представления семантики текста позволяет использовать в обработке ЕЯ-текстов математические модели и методы, разработанные как в теории формальных языков, так и в других областях математики — теории алгоритмов, математической логике, теории графов и др.

Для расширения выразительных возможностей языка ЛППП за последние 50 лет был предложен ряд формальных средств для представления смысла текстов: эпизодическая логика [11], абстрактное представление смысла (Abstract Meaning Representation — AMR) [12], метод представления смысла

средствами лямбда-исчисления (Lambda Calculus Meaning Representation — LCMR) [13], а также теория К-представлений (ТКП) [23, 24, 26]. Сравнительный анализ данных формальных средств приведен в табл. 1.

Таблица 1
Сравнение характеристик средств построения СП

Критерий	ЛППП	AMR	LCMR	ТКП
Представление значений слов "некоторый", "любой", "несколько", "все" и т. п.	—	—	—	+
п-арные функциональные символы ($n \geq 1$)	+	—	+	+
п-арные отношения (предикаты)	+	+	+	+
Ссылки на ранее упомянутые объекты и события	—	+	—	+
Ссылки на смысл фраз и более крупных частей текста	—	—	—	+
Логические связки $\wedge$ и $\vee$	+	Как расширение	+	+
Кванторы $\forall$ и $\exists$	+	Только $\exists$	+	+
Составные обозначения понятий	—	Ограниченно	—	+
Составные обозначения множеств	$\pm$	—	$\pm$	+
Конечные последовательности	—	—	—	+
СП утверждений	+	+	+	+
СП вопросов	—	$\pm$	—	+
СП команд (предписаний)	—	—	—	+
Поддержка различных естественных языков	+	—	+	+

Сравнение средств преобразования ЕЯ-текстов в СП

Критерий	КСГ	ССГ	SemSynt1, SemSynt2	SemSyntRA
Сложность реализации	Middle	High	High	Low
Объем базы данных	High	High	High	Middle
Сложность заполнения БД	Middle	High	Middle	Middle
Вычислительная сложность	$O(m^2n)$	<i>Naive:</i> $O(2^n)$ <i>Optimized:</i> $O(2^d n^4)$	—	$A(n) + O(n^2m) + B(n) + O(dn)$
Поддержка омонимии	—	+	+	+

На основании анализа в проведенном исследовании для построения СП предписаний была выбрана ТКП.

Для реализации преобразования ЕЯ-текстов в СП был разработан целый ряд подходов и формальных средств, из которых наиболее широко используемыми являются:

- сопоставление с образцами команд,
- разбор регулярными выражениями,
- контекстно-свободные грамматики (КСГ),
- комбинаторные категориальные грамматики (Combinatory Categorical Grammar — CCG),
- алгоритмы, предложенные в рамках теории К-представлений — SemSyn [23], SemSynt1 [26], SemSynt2 [29],
- алгоритм SemSyntRA, обогащающий ТКП новым подходом к семантическому парсингу [34–36].

Сравнение средств преобразования ЕЯ-текстов в СП приведено в табл. 2.

В данной таблице параметр n обозначает размер входного текста, а m — число записей в базе данных (БД), используемой для работы алгоритма.

Был разработан ряд программных приложений с ЕЯ-интерфейсами, в том числе:

- *Siri* [39] — персональный ассистент для продуктов Apple;
- *Sobesednik HD* [40] — аналог Siri для русского языка;
- *Braina Project* [37] — программное обеспечение для работы с компьютером;
- *NLUI Server* [38] — сервер приложений для построения ЕЯ-интерфейсов с поддержкой широкого спектра естественных языков, включая английский, русский, французский, немецкий, японский и многие другие;
- *NLC-1* [30–33], *NLC-2* [34–36] — программное обеспечение для управления файловой системой и компьютером в целом.

Данные приложения, за исключением разработанных в рамках проведенного исследования систем NLC-1 и NLC-2, поддерживают простые команды в одно действие: "открой файл", "создай папку", "скопируй некоторый файл в папку", "скажи, сколько время". Однако перечень пользовательских команд гораздо шире. Он включает в себя сложные составные команды с условиями типа "Перемести файл "a.txt" в папку "Мои документы", если размер файла "a.txt" меньше 2 Мбайт", последовательности команд и т. п. Поддержка таких составных команд требует более глубокой теоретической базы как ос-

новы разработки алгоритмов семантического парсинга команд.

Входной язык систем NLC-1, NLC-2 значительно шире за счет применения ТКП, а в случае NLC-2 еще и за счет применения новой математической модели лингвистической базы данных [34, 35] и алгоритма SemSyntRA [34, 36]. Данный алгоритм имеет ряд существенных отличий от других алгоритмов семейства SemSynt. В первую очередь, этот алгоритм является более общим и поддерживает более широкий класс входных текстов. В то же время данный алгоритм более прост для понимания и программной реализации, чем алгоритмы семейства SemSynt. Кроме того, алгоритм SemSyntRA использует граф возможных семантико-синтаксических связей для анализа семантико-синтаксической структуры текста, в то время как другие алгоритмы семейства SemSynt для тех же целей применяют матричное семантико-синтаксическое представление (МССП) [23, 26].

Этот алгоритм обладает следующими возможностями:

- обработка сложных предписаний, в том числе с несколькими глаголами;
- семантико-синтаксические отношения во входном тексте определяются без выполнения полного синтаксического разбора;
- поддерживает обработку входных текстов на различных языках: русском, английском, немецком и многих других;
- легок для реализации и расширения.

Алгоритм SemSyntRA, изложенный в работах [34–36], описывает преобразование предписаний на ЕЯ (русском, английском или другом) в СП, являющееся К-представлением (КП) предписания. Рассмотренные в этих статьях примеры показывают, каким образом из первичного КП предписания строится вторичное КП, а затем сценарий BASH для выполнения предписания. Однако реализация данных преобразований входного текста опирается на оригинальную теоретическую основу, не рассматривавшуюся в статьях [34–36].

Основная цель данной статьи — описание основных идей этой оригинальной теоретической основы. Такой основой является определение нового отношения на СК-языке (стандартном концептуальном языке), названного отношением соответствия образцу.

Данная статья имеет следующую структуру. Сначала кратко описываются центральные идеи ТКП

и основная идея исполнения СП, а затем даются основания для определения отношения соответствия образцу.

Краткое введение в теорию К-представлений

Анализ показывает, что наиболее широкие возможности по разработке теоретических основ проектирования ЕЯ-интерфейсов приложений предоставляются теорией К-представлений (концептуальных представлений), или ТКП. Данная теория, разработанная В. А. Фомичевым, вводит, в частности, новый класс формальных языков, называемых СК-языками (стандартными концептуальными языками).

Определение класса СК-языков — часть базовой математической модели ТКП, разработанной в [17, 18, 21–24, 26] и описывающей структуру концептуального базиса (к.б.) и систему из десяти частичных операций на концептуальных структурах [21–24, 26]. Данная модель предоставляет средства для представления смысловых структур естественно-языковых предложений и сколь угодно сложных связанных текстов (дискурсов), относящихся к различным сферам профессиональной деятельности: технологии, медицине, экономике, спорту и т. д. Рассмотрим некоторые важные свойства класса СК-языков, определенного в [21–24, 26].

Множество базовых элементов произвольного СК-языка включает в себя два непересекающихся множества, являющихся составными элементами к.б.:

- X — первичный информационный универсум — множество базовых семантических единиц;
- V — множество переменных.

Первичный информационный универсум включает в себя, в частности, следующие непересекающиеся множества:

- St — множество сортов (обозначений наиболее общих понятий), они используются для ассоциирования выражений СК-языков с их семантическими характеристиками, называемыми типами;
- R — множество реляционных символов, включающее подмножество функциональных символов F .

Элементы первичного информационного универсума и переменные к.б. используются при построении СП предложений и дискурсов на ЕЯ с помощью индуктивного применения правил построения с номерами от 0 до 10, детально рассмотренных в работах [17, 18, 21–24, 26]. Язык, порожденный к.б. B , обозначается $Ls(B)$. Выражения СК-языков называются К-цепочками.

ТКП вводит понятие типа К-цепочки. Для этого в составе первичного информационного универсума выделяется подмножество сортов St , из элементов которого строятся цепочки, называемые типами. Само множество типов $Tr(S(B))$, где $S(B)$ — компонент к.б., называемый сортовой системой, определяется рекурсивно и включает в себя:

- составные типы, например: $инф.объект * дин.физ.об.$, где $инф.объект$ и $дин.физ.об.$ — обозначения типов "информационный объект", "динамический физический объект";
- типы, соответствующие понятиям, квалифицирующим объекты предметной области: $\uparrow инф.объект * дин.физ.об.$;
- типы для описания наборов объектов: $(инф.объект, число)$;
- типы для множеств объектов: $\{инф.объект * дин.физ.об.\}, \{(инф.объект, число)\}$.

Для произвольного к.б. B на СК-языке $Ls(B)$ задается отображение tpl с областью значений $Tr(S(B))$. Это отображение каждой цепочке z ставит в соответствие ее тип $tpl(z)$ [23]. Например, возможны соотношения $tpl(И. П. Павлов) = интс * дин.физ.об.$, $tpl(USB) = \uparrow инф.объект * дин.физ.об.$, где $интс$ — сорт "интеллектуальная система".

На множестве типов вводится отношение общности: тип t_1 является более общим, чем тип t_2 . Для этого сначала данное отношение определяется на первичном информационном универсуме $X(B)$, а затем по определенным правилам распространяется на все множество типов.

Например, тип $инф.объект$ является более общим для типа $инф.объект * дин.физ.об.$ — данный факт обозначается $инф.объект \vdash инф.объект * дин.физ.об.$

Кроме того, в дальнейшем нам понадобится обозначение $Tconc(B) = \{t \in Tr(S(B)) \mid t \text{ начинается с символа } \uparrow\} \cup \{\uparrow сущ\}, \{\uparrow пон\}, \{\uparrow об\}\}$ — множество типов, соответствующих понятиям.

Математические аспекты ТКП рассматриваются в работах [14, 17, 18, 21–24, 26]. Приложения моделей и методов ТКП обсуждаются, в частности, в публикациях [15, 16, 18–20, 25, 26–30, 33].

Основная идея исполнения К-представлений предписаний

Для исполнения предписаний, точнее КП предписаний, необходимо выполнить два шага: во-первых, нужно сформировать КП, содержащее только понятия, "известные" целевой системе, во-вторых, следует перевести полученное КП в язык целевой системы.

Рассмотрим предписание пользователя "Переместить на резервный диск видео с наименованием "Лекция 2016-12-15"". Чтобы не загромождать пример излишними подробностями, ограничимся рассмотрением обработки фразы "видео с наименованием "Лекция 2016-12-15"", с которой ассоциировано КП $нек видео1*(Наименование, "Лекция 2016-12-15")$.

Предположим, что у нас имеется правило трансформации

$$видео1*(Наименование, filename) \rightarrow \rightarrow файл1*(Наименование, fileName)(Расширение1, ("avi" \vee "mkv" \vee "mp4")),$$

где $fileName$ — некоторая переменная такая, что $tpl(filename) \vdash tpl("Лекция 2016-12-15")$.

Тогда, применяя данное правило преобразования к КП *нек видео1*(Наименование, "Лекция 2016-12-15")*, получаем унифицированное КП *нек файл1*(Наименование, "Лекция 2016-12-15")*(Расширение1, ("avi" $\vee$ "mkv" $\vee$ "mp4"))).

Для применения данного правила трансформации потребуется определить, что КП *нек видео1*(Наименование, "Лекция 2016-12-15")* или какая-либо его подцепочка соответствует левой части правила трансформации (*видео1*(Наименование, filename)*). Для этого потребуется сформулировать бинарное отношение соответствия образцу, обозначаемое *Match*. Для данного примера выполнено соотношение (*видео1*(Наименование, "Лекция 2016-12-15")*, *видео1*(Наименование, filename)*) $\in Match$. В следующем разделе мы рассмотрим основания для включения пар К-цепочек в отношение *Match*.

Аналогичным образом предполагается осуществлять преобразование КП в сценарии целевой системы, с той лишь разницей, что в правой части будут не К-цепочки, а шаблоны сценариев целевой системы.

Основания для включения К-цепочек в отношение соответствия образцу

Для выполнения трансформации зададим правило проверки соответствия левой части правила трансформации обрабатываемой К-цепочке. Для этого мы определим *отношение соответствия образцу* на множестве К-цепочек. Отношение построено на основе нескольких естественных предположений, названных далее основаниями.

Во-первых, соответствие элемента первичного информационного универсума u образцу z может определяться схожестью по смыслу (синонимией). Данная синонимия может задаваться некоторым рефлексивным симметричным отношением. Введем

Определение 1. Пусть B — произвольный к.б. *Отношением синонимии* согласованным с к.б. B называется симметричное рефлексивное отношение *Syn* на первичном информационном универсуме $X(B)$.

Определим на множестве $Ls(B)$ *отношение соответствия образцу Match*. Факт вхождения пары (y, z) в отношение соответствия образцу *Match* обозначается записью $y \succ z$. При этом, как будет определено дальше, отношение *Match* зависит от отношения синонимии *Syn* и правил его "продолжения" на множество $Ls(B)$, поэтому будет правильно говорить об отношении соответствия образцу с точностью до отношения синонимии *Syn*. Таким образом, можно сформулировать

Основание 1. Пусть B — произвольный к.б., u и z — произвольные элементы первичного информационного универсума $X(B)$, тогда из $(u, z) \in Syn$ следует, что $(u, z) \in Match$.

Например, пара (*напка1*, *каталог1*) может быть элементом отношения синонимии, так как информационные единицы *напка1* и *каталог1* соответствуют словам *напка* и *каталог*, которые в рамках управления файлами являются синонимами. Таким

образом, *напка1* соответствует образцу *каталог1*, т. е. (*напка1*, *каталог1*) $\in Match$ или, что то же самое, *напка1* $\succ$ *каталог1*.

Во-вторых, если образец является переменной, то соответствие образцу определяется соответствием типов образца и К-цепочки y — тип образца должен быть таким же или более общим, чем тип u . Например, если элемент некоторого первичного информационного универсума *диск1* имеет тип $\uparrow инф.объект *дин.физ.об.$, и некоторая переменная y имеет тип $\uparrow инф.объект$, то элемент *напка* соответствует образцу y . Таким образом, получаем

Основание 2. Пусть B — произвольный к.б., y является К-цепочкой из $Ls(B)$ и z — произвольная переменная из $V(B)$. Тогда пара $(y, z) \in Match$ тогда и только тогда, когда $tpl(z) \vdash tpl(y)$, где tpl — отображение СК-языка $Ls(B)$ в множество типов.

Далее, если образец z построен с помощью правил P[2]-P[10], то ему соответствуют аналогичные по структуре К-цепочки. Так как для формирования КП используются 10 правил построения, имеет смысл познакомиться с тем, как будет строиться отношение соответствия образцу в каждом из этих случаев.

Начнем рассмотрение с применения в К-цепочках интенциональных кванторов (правило построения P[1]). Например, КП *нек напка1*(Наименование, "Video")* будет соответствовать образцу *нек каталог1*, но не будет соответствовать образцу *каталог1*. Также и более простое КП *нек напка1* будет соответствовать образцу *нек каталог1*. Таким образом, имеем

Основание 3. Пусть B — произвольный к.б., u — К-цепочка вида *intq₁ c₁*, и z — К-цепочка вида *intq₂ c₂* и $(intq_1, intq_2) \in Syn$, где *Syn* — некоторое рефлексивное симметричное отношение, а пара $(c_1, c_2) \in Match$. Тогда пара $(u, z) \in Match$.

К-цепочки с функциональными и реляционными символами (т. е. с именами функций и именами n -арных отношений, где $n \geq 1$), образованные соответственно по правилам P[2] и P[4], будут соответствовать образцам с такой же структурой. То есть К-цепочка *Размер1(нек файл1*(Наименование, "a.txt"))* будет соответствовать образцу *Размер1(нек файл1)*. В более общем случае функциональные символы К-цепочки и образца должны быть синонимами, а аргументы функции (предиката) должны соответствовать аргументам образца, как аргумент *нек файл1*(Наименование, "a.txt")* соответствует образцу *нек файл*. Кроме того, если некоторая переменная n имеет тип *число*, К-цепочка *Размер1(нек файл1*(Наименование, "a.txt"))* имеет тип *целое-число* и *число* является более общим типом, чем *целое-число*, то К-цепочка *Размер1(нек файл1*(Наименование, "a.txt"))* соответствует образцу n . Аналогичные требования мы можем предъявить и к К-цепочкам, построенным с применением реляционных символов, а также к К-цепочкам вида $(a \equiv b)$. Сказанное выше позволяет нам сформулировать следующие три основания.

Основание 4. Пусть B — произвольный к.б., y — К-цепочка вида $f(a_1, \dots, a_n)$ и z — К-цепочка вида $g(b_1, \dots, b_n)$, где f и g — функциональные символы из $F(B)$, и $(f, g) \in \text{Syn}$. Пусть пары $(a_i, b_i) \in \text{Match}$ для любого i от 1 до n . Тогда пара $(y, z) \in \text{Match}$.

Основание 5. Пусть B — произвольный к.б., y — К-цепочка вида $(c \equiv d)$ и z — К-цепочка вида $(a \equiv b)$. Пусть пары (c, a) и $(d, b) \in \text{Match}$. Тогда пара $(y, z) \in \text{Match}$.

Основание 6. Пусть B — произвольный к.б., y — К-цепочка вида $r(a_1, \dots, a_n)$ и z — К-цепочка вида $p(b_1, \dots, b_n)$, где r и p — реляционные символы из $R(B) \setminus F(B)$, и $(r, p) \in \text{Syn}$. Пусть пары $(a_i, b_i) \in \text{Match}$ для любого i от 1 до n . Тогда пара $(y, z) \in \text{Match}$.

Очевидно, что применение правила P[5] к некоторой К-цепочке не приводит к каким-либо изменениям в соответствии образцу. Иными словами,

Основание 7. Пусть B — произвольный к.б., y и z являются К-цепочками из $Ls(B)$ и v — такая произвольная переменная из $V(B)$, что допустимо образование К-цепочки $y:v$ согласно правилу P[5]. Тогда из $(y, z) \in \text{Match}$ следует, что $(y:v, z) \in \text{Match}$.

Рассмотрим случай отрицания К-цепочек на простом примере: *нек файл1*(Наименование, "a.txt")* соответствует образцу *нек файл1*. Логично предположить, что и отрицание *¬нек файл1*(Наименование, "a.txt")* соответствует образцу *¬нек файл1*. Таким образом, можно определить

Основание 8. Пусть B — произвольный к.б., y и z являются К-цепочками из $Ls(B)$, тогда из $(y, z) \in \text{Match}$ следует, что $(\neg y, \neg z) \in \text{Match}$.

Более сложным является случай логических связок. Рассмотрим сначала дизъюнкцию, т. е. логическую связку "ИЛИ". К-цепочка вида $(y_1 \vee \dots \vee y_n) \succ z$ в том случае, если ему соответствует хотя бы одна из подцепочек y_i . И наоборот, $y \succ (z_1 \vee \dots \vee z_n)$, если y соответствует любой подцепочке z_i .

Основание 9. Пусть B — произвольный к.б., y — К-цепочка вида $(y_1 \vee \dots \vee y_n)$ и $z \in Ls(B)$. Пусть существует i от 1 до n такое, что $(y_i, z) \in \text{Match}$. Тогда пара $(y, z) \in \text{Match}$.

Основание 10. Пусть B — произвольный к.б., $y \in Ls(B)$ и z — К-цепочка вида $(z_1 \vee \dots \vee z_n)$. Пусть существует i от 1 до n такое, что $(y, z_i) \in \text{Match}$. Тогда пара $(y, z) \in \text{Match}$.

Основания 9 и 10 позволяют также определить отношение соответствия образцу для К-цепочки y вида $(y_1 \vee \dots \vee y_n)$ и образца z вида $(z_1 \vee \dots \vee z_m)$. Нетрудно видеть, что $y \succ z$, если существует пара (i, j) такая, что $y_i \succ z_j$.

Аналогично основаниям 9 и 10 можно определить, что К-цепочка вида $(y_1 \wedge \dots \wedge y_n)$ соответствует некоторому образцу z в том случае, если ему соответствует хотя бы одна из подцепочек y_i . Напротив, для соответствия образцу вида $(z_1 \vee \dots \vee z_n)$ логично потребовать, чтобы произвольная К-цепочка y соответствовала всем подцепочкам z_i . Таким образом, можно сформулировать

Основание 11. Пусть B — произвольный к.б., y — К-цепочка вида $(y_1 \wedge \dots \wedge y_n)$ и $z \in Ls(B)$. Пусть существует i от 1 до n такое, что $(y_i, z) \in \text{Match}$. Тогда пара $(y, z) \in \text{Match}$.

Основание 12. Пусть B — произвольный к.б., $y \in Ls(B)$ и z — К-цепочка вида $(z_1 \wedge \dots \wedge z_n)$. Пусть существует i от 1 до n такое, что $(y, z_i) \in \text{Match}$. Тогда пара $(y, z) \in \text{Match}$.

Основания 11 и 12 также позволяют определить отношение соответствия образцу для К-цепочки y вида $(y_1 \wedge \dots \wedge y_n)$ и образца z вида $(z_1 \wedge \dots \wedge z_m)$. В данном случае $y \succ z$, если для любого j от 1 до m существует i от 1 до n такое, что $y_i \succ z_j$. Этот факт менее очевиден, чем аналогичный для логической связки "ИЛИ", однако, как и в предыдущем случае, он обосновывается последовательным применением оснований 11 и 12.

Ранее мы рассматривали пример соответствия К-цепочки *нек папка1*(Наименование, "Video")* образцу *нек каталог1*. При этом факт соответствия подцепочки *папка1*(Наименование, "Video")* образцу *каталог1* принимался как данность. Сформулируем основание для данного случая.

Основание 13. Пусть B — произвольный к.б., y — К-цепочка вида $a^*(b_1, c_1) \dots (b_n, c_n)$ и z — обозначение понятия из $Ls(B)$ (т.е. $\text{tpl}(z) \in T\text{conc}(B)$). Пусть $(a, z) \in \text{Syn}$. Тогда пара $(y, z) \in \text{Match}$.

В данном основании составное понятие сопоставляется образцу, являющемуся простым понятием. Логично было бы расширить данное основание и на случай сопоставления двух составных понятий.

Основание 14. Пусть B — произвольный к.б., y — К-цепочка вида $a^*(b_1, c_1) \dots (b_n, c_n)$ и z — К-цепочка вида $z^*(u_1, v_1) \dots (u_m, v_m)$, где $n, m \geq 1$. Пусть $(a, z) \in \text{Syn}$. Пусть для любого i от 1 до m существует j от 1 до n такое, что $(b_j, u_i) \in \text{Syn}$, и пара $(c_j, v_i) \in \text{Match}$. Тогда пара $(y, z) \in \text{Match}$.

Например, *нек папка1*(Наименование, "Video") (Расположение, нек флешка1)* соответствует образцу *нек каталог1*(Наименование, "Video")*.

Для полноты покрытия всего многообразия К-цепочек нам осталось рассмотреть основания для двух правил построения.

Основание 15. Пусть B — произвольный к.б., y — К-цепочка вида $Qv(b)T$ и z — К-цепочка вида $Qv(a)S$, где Q — один из кванторов $\forall$ или $\exists$. Пусть $(b, a) \in \text{Syn}$ и пара $(T, S) \in \text{Match}$. Тогда пара $(y, z) \in \text{Match}$.

Основание 16. Пусть B — произвольный к.б., y — К-цепочка вида $\langle t_1, \dots, t_n \rangle$ и z — К-цепочка вида $\langle s_1, \dots, s_n \rangle$, где $n \geq 1$. Пусть для любого i от 1 до n пара $(t_i, s_i) \in \text{Match}$. Тогда пара $(y, z) \in \text{Match}$.

Наложим на формируемое отношение *Match* (отношение соответствия образцу с точностью до отношения синонимии *Syn*) ограничение, что любая пара $(y, z) \in \text{Match}$ удовлетворяет требованиям одного из оснований 1—16. Тогда можно говорить, что перечень оснований для определения соответствия К-цепочек образцам является исчерпывающим.

Направления продолжения исследования

Описанные в статье основания позволят в дальнейшем сформулировать математическое определение отношения соответствия КП образцу. Используя его, можно будет предложить формальный метод преобразования КП из первичной формы в унифицированную, а также метод преобразования унифицированной формы в сценарии целевой системы. В продолжении данной статьи планируется дать определение отношения соответствия образцу, исследовать его свойства и показать его применение для выполнения преобразований.

Заключение

Наряду с ранее предложенными моделью лингвистической базы данных [34, 35] и алгоритмом семантического парсинга SemSyntRA [34, 36] описанные идеи обработки предписаний на ЕЯ и исполнения СП предписаний стали основой разработки файлового менеджера с ЕЯ-Интерфейсом 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).

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. P. 1294–1303. URL: <https://aclweb.org/anthology/P/P13/P13-1127.pdf> (дата обращения: 10.03.2016).
11. Schubert L. K., Hwang C. H. Episodic Logic meets Little Red Riding Hood: A Comprehensive, Natural Representation for Language Understanding // Iwanska L., Shapiro S. C. (eds.) Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language. MIT/AAAI Press, Menlo Park, CA and Cambridge, MA. 2000. P. 111–174.
12. Banarescu L., Bonial C., Cai S., Georgescu M., Griffith K., Hermjakob U., Knight K., Koehn P., Palmer M., Schneider N. Abstract Meaning Representation for Sembanking // Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse (Sofia, Bulgaria, August 8–9 2013), 2013. P. 178–186. URL: <https://aclweb.org/anthology/W/W13/W13-2322.pdf> (дата обращения: 14.03.2017).
13. Wong Y., Mooney R. Learning Synchronous Grammars for Semantic Parsing with Lambda Calculus. // Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (Prague, Czech Republic, 2007), 2007. P. 960–967. URL: <http://aclweb.org/anthology/P/P07/P07-1121> (дата обращения: 14.03.2017).
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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., 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.
21. Фомичев В. А. Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть I // Информационные технологии. 2002. № 10. С. 16–25.
22. Фомичев В. А. Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть II // Информационные технологии. 2002. № 11. С. 34–45.
23. Фомичев В. А. Формализация проектирования лингвистических процессоров. М.: МАКС Пресс, 2005. 368 с.

24. **Фомичев В. А.** Математические основы представления содержания посланий компьютерных интеллектуальных агентов. М.: Издательский дом ГУ-ВШЭ, 2007.

25. **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.

26. **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.

27. **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.

28. **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.

29. **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.

30. **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.

31. **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.

32. **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.

33. **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.

34. **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, Part. I. Cham: Springer, 2016. Vol. 9827. P. 416–430.

35. **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.

36. **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.

37. **Briana Project.** URL: <http://www.brainasoft.com/braina/#overview> (дата обращения: 13.03.2017).

38. **NLUI Server.** URL: http://www.linguasys.net/web_production/server-item/NLUI%20Server (дата обращения: 9.02.2014).

39. **Siri.** URL: <http://www.apple.com/ios/siri/> (дата обращения: 3.03.2017).

40. **Sobesednik H. D.** URL: <http://sobesednik.idevelops.com/> (дата обращения: 3.03.2017).

A. A. Razorenov, Postgraduate Student, e-mail: arazorenov@hse.ru,

National Research University Higher School of Economics (HSE), Moscow, Russia

The Principles of Applying the Theory of K-representations to Developing Mathematical Foundations of Executing Natural Language Instructions by Computer Systems

The paper continues the series of publication written by the author and V. A. Fomichov (Informational Technologies 2015, No. 6, 2016, No. 12, 2017, No. 1) devoted to applying the theory of K-representations (knowledge representations) to developing new methodological foundations of designing computer systems processing natural language (NL) instructions. The goal of this paper is to set forth the main ideas of defining 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. This relation is called the pattern matching relation. It is indicated that the proposed formal approach has been successfully used in designing the NL interface of the file manager NLC-2 (Natural Language Commander — Version 2).

Keywords: natural language processing, semantic parsing, theory of K-representations, SK-language, software management, natural language interface, pattern matching relation, 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> (data of access 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> (data of access 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. Available at: <http://aclweb.org/anthology/N16-1089> (data of access 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> (data of access 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> (data of access 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> (data of access 13.03.2016).

7. **Marge M., Rudnicky A.** Comparing Spoken Language Route Instructions for Robots across Environment Representations. *Pro-*

ceedings 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> (data of access 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> (data of access 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> (data of access 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> (data of access 10.03.2016).

11. **Schubert L. K., Hwang C. H.** Episodic Logic meets Little Red Riding Hood: A Comprehensive, Natural Representation for Language Understanding. *Iwanska L., Shapiro S. C. (eds.) Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*. MIT/AAAI Press, Menlo Park, CA and Cambridge, MA, 2000, pp. 111–174.

12. **Banarescu L., Bonial C., Cai S., Georgescu M., Griffitt K., Hermjakob U., Knight K., Koehn P., Palmer M., Schneider N.** Abstract Meaning Representation for Sembanking. *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse* (Sofia, Bulgaria, August 8–9 2013), 2013, pp. 178–186. Available at: <https://aclweb.org/anthology/W/W13/W13-2322.pdf> (data of access 14.03.2017).

13. **Wong Y., Mooney R.** Learning Synchronous Grammars for Semantic Parsing with Lambda Calculus. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (Prague, Czech Republic, 2007), 2007, pp. 960–967. Available at: <http://aclweb.org/anthology/P/P07-1121> (data of access 14.03.2017).

14. **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.

15. **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.

16. **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.

17. **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.

18. **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.

19. **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.

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

21. **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).

22. **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).

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

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

25. **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.

26. **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.

27. **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.

28. **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.

29. **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.

30. **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.

31. **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) *Informacionnye sredstva i tekhnologii. Trudy XXII Mezhdunarodnoj nauchno-tekhnicheskoy konferencii. V 3 tomah. 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).

32. **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 molodykh 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. Leohin, B. G. L'vov, N. S. Titkova, S. U. Uvajov. (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).

33. **Fomichov V. A., Razorenov A. A.** The 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).

34. **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.

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

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

37. **Web-resource:** Briana Project. Available at: <http://www.brainasoft.com/braina/#overview> (data of access 13.03.2017).

38. **Web-resource:** NLUI Server. Available at: http://www.linguasys.net/web_production/server-item/NLUI%20Server (data of access 9.02.2014).

39. **Web-resource:** Siri. Available at: <http://www.apple.com/ios/siri/> (data of access 3.03.2017).

40. **Web-resource:** Sobesednik HD. Available at: <http://sobesednik.idelops.com/> (data of access 3.03.2017).

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 004.9; 62-52

А. Б. Филимонов, д-р техн. наук, проф., filimon_ab@mail.ru,

Московский технологический университет,

Н. Б. Филимонов, д-р техн. наук, проф., nbfilimonov@mail.ru,

Московский государственный университет им. М. В. Ломоносова,

В. Ю. Тихонов, науч. ассист., bot-32@yandex.ru,

Центр развития образования и международной деятельности "Интеробразование", Москва

Оптимизация распределения ресурсов в задачах пространственного поиска объектов*

Излагаются общие положения классической теории поиска. Для выполняющих поиск субъектов применяется термин — наблюдатели. Отмечается специфика задач дискретного поиска объектов. В континуальных задачах поиска исследуется кинематика движения наблюдателя в физическом пространстве. В задачах дискретного поиска обследуемая область представляется в виде совокупности зон, причем кинематические аспекты поиска в них не учитываются. Рассматриваются задачи оптимального распределения поисковых ресурсов по выделенным зонам поиска. Данные задачи относятся к классу задач о назначениях. Предложена их формализация в детерминированной и вероятностной постановках. Приведены модельные примеры оптимизации распределения поисковых ресурсов с применением стандартных инструментальных программных средств решения задач булева линейного программирования.

Ключевые слова: пространственный поиск объектов, дискретный поиск, планирование поисковых операций, задача о назначениях, оптимальное распределение поисковых ресурсов, вероятностные модели планирования назначений

Введение

Явление поиска представляет собой одну из важнейших сторон человеческой деятельности и встречается в самых разных областях. При этом все большую актуальность приобретают задачи пространственного поиска реальных объектов в различных средах.

Теория пространственного поиска объектов зарождалась в работах Купмана (В. О. Коорман, 1946) и к настоящему времени сложилась как самостоятельная научная дисциплина [1—5]. Основной целью пространственного поиска является обнаружение разнообразных сторонних объектов в обследуемом пространстве с определением их характера и местоположения. Объекты поиска (цели) могут быть неподвижными (статическими) либо подвижными, иметь различную природу и располагаться в разных средах. Например, это могут быть летательные аппараты, разнообразные предметы на поверхности Земли, корабли и суда, промысловая рыба и морские животные и т. д. Задачей теории поиска объектов является выработка оптимального плана поиска, обеспечивающего обнаружение объ-

екта при минимальных временных или ресурсных затратах [6—11].

Несмотря на широкие инженерные применения проблематика управления поисковыми операциями к настоящему времени остается малоизученной. Вместе с тем принципиально изменяются технологии поиска, а также его научно-технические аспекты вследствие использования мобильной робототехники в качестве эффективных технических средств поиска. Здесь особое внимание заслуживает разработка теоретических основ авиационного поиска, базирующегося на применении беспилотных летательных аппаратов (БПЛА) [12, 13], которые применяют в различных областях науки и техники при решении многих комплексных задач как мирного, так и военного характера. При этом БПЛА гражданского назначения весьма эффективно осуществляют дневной и ночной поиск различных объектов, патрулирование районов, экологический мониторинг, ретрансляцию радиосвязи, радиопеленгацию, разведку районов крупных аварий и катастроф и т. д. БПЛА военного назначения в наибольшей степени используются в качестве средств реализации операций геолокации: при исследовании мест применения оружия массового поражения; подготовке вооруженного нападения на стационарные или вражеские движущиеся объекты; операциях

* Работа выполнена при поддержке РФФИ, грант № 16-08-00313.

идентификации, распознавания, разведки, поиска, спасения и т. д.

В данной статье акцент сделан на задачах дискретного поиска целей, важнейшей особенностью которых является возможность применения методологии и математического аппарата теории расписаний [14], сводящего решаемую задачу к задачам комбинаторной оптимизации [15]. В статье развивается концепция оптимального распределения поисковых ресурсов, изложенная в работе авторов [11].

Общие положения классической теории поиска

Задача поиска возникает тогда, когда требуется определить положение объекта, находящегося в заданной области Ω физического пространства, с помощью поисковых средств. Классические результаты в данной научной области исследований отражены в монографии [1, 3, 4].

Для выполняющего поиск субъекта примем термин *наблюдатель* [1]. Задачей теории поиска объектов является выработка оптимального плана поиска, обеспечивающего обнаружение целей при минимальных временных или ресурсных затратах. Положение цели как случайной величины можно задавать с помощью некоторой плотности распределения вероятностей.

В теории поиска рассматривают два критерия оценки эффективности выбранных стратегий поиска [3]: 1) вероятность обнаружения цели; 2) средняя продолжительность процесса поиска. На основе данных критериев осуществляется оптимизационная постановка задач поиска.

Сам поиск может выполняться как одиночным наблюдателем, так и группой наблюдателей.

Если подлежащая обследованию площадь значительно больше площади, обозреваемой из некоторой фиксированной точки или просматриваемой движущимся наблюдателем за один прогон, то лучший способ добиться равномерного покрытия всей площади зоны поиска заключается в реализации последовательности перемещений по параллельным маршрутам, пролегающим на некотором фиксированном расстоянии d один от другого. Этого можно достичь двумя путями: 1) если разведку ведет один наблюдатель, то он выполняет движение по спирали или по челночному маршруту; 2) в случае нескольких наблюдателей они должны следовать параллельными курсами, так чтобы расстояния между смежными маршрутами равнялось d .

При весьма общих допущениях вероятность обнаружения цели может быть выражена формулой

$$P(\Phi) = 1 - e^{-\Phi}, \quad (1)$$

где Φ — *поисковый потенциал* наблюдателя.

Поясним смысл этой формулы. Введем функцию $p(x)$ — вероятность обнаружения цели, расстояние до которой по траверсу (направлению, перпенди-

кулярному курсу самолета, судна или его диаметральной плоскости) равняется x .

Приведенная ширина полосы обзора, охватываемая наблюдателем при его движении по заданному курсу, определяется формулой

$$W = \int_{-\infty}^{\infty} p(x) dx.$$

Согласно [3] данная величина является наиболее объективной мерой эффективности, применяемой наблюдателем разведывательной аппаратуры.

Обозначим через S_{Ω} площадь всей обследуемой области Ω . Положим, что разведывательный маршрут состоит из нескольких лежащих внутри области Ω отрезков, сумма длин которых равняется L .

Разобьем маршрут на N малых участков протяженности ΔL . Тогда

$$L = N \Delta L. \quad (2)$$

Вероятности обнаружения цели на каждом таком участке равны

$$P_1 = (\Delta L / S_{\Omega}) \int_{-\infty}^{\infty} p(x) dx = \frac{W \Delta L}{S_{\Omega}}. \quad (3)$$

Введем так называемое [3] *поисковое покрытие*:

$$E = WL.$$

Поисковый потенциал определяется формулой

$$\Phi = E / S_{\Omega}.$$

Отсюда из (2) и (3) выводим соотношение

$$P_1 = \Phi / N. \quad (4)$$

Для вероятности обнаружения цели на N участках имеем

$$P_N = 1 - (1 - P_1)^N.$$

Подставляя сюда (4) при $N \rightarrow \infty$ получаем (согласно второму замечательному пределу) формулу (1).

Поисковые потенциалы обладают свойством аддитивности — если поиск осуществляется несколькими наблюдателями, которые либо перемещаются по одному и тому же маршруту, либо следуют параллельными курсами, то их поисковые потенциалы Φ_i суммируются:

$$\Phi = \Phi_1 + \Phi_2 + \dots$$

Сущность проблемы оптимального распределения поисковых усилий раскрывает работа [3]. Так, установлено, что если цель находится внутри области Ω , но о координатах цели ничего неизвестно, то оптимальным является равномерное распределение имеющихся разведывательных ресурсов по всей площади.

В случае частичной осведомленности о местонахождении цели считаются известными априорные вероятности ее расположения в той или иной подобласти $\Omega' \subset \Omega$.

Задачи дискретного поиска

В континуальных задачах поиска исследуется кинематика движения наблюдателя в физическом пространстве. Здесь различают поиск в трехмерном пространстве, на площади (в заданном районе) или на линии (на рубеже).

В задачах *дискретного* поиска обследуемая область представляется в виде совокупности зон, причем кинематические аспекты поиска в них не рассматриваются, а процесс поиска представляется как чередование обследования выделенных зон имеющимися средствами наблюдения. Важным преимуществом задач дискретного поиска является возможность применения к ним методологии и математического аппарата теории расписаний [14], сводящего решаемую задачу к задачам комбинаторной оптимизации.

Далее полагаем, что район поиска Ω разбит на n зон:

$$\Omega = \bigcup_{i=1}^n \Omega_i \quad (5)$$

причем поисковые усилия внутри каждой зоны Ω_i распределены равномерно.

Выделение тех или иных зон поиска должно быть обусловлено топографическими условиями и элементами местности — строением рельефа местности и местными предметами; природными объектами (реки, озера, зеленые массивы, заболоченные территории и т. п.) и искусственными объектами (населенные пункты, дороги, инженерные сооружения и т. п.).

Задачи оптимального распределения поисковых ресурсов

В теории расписаний исследуются задачи составления расписаний, т. е. упорядочения некоторых работ (операций) по времени и/или по исполнителям (приборам). При этом необходимо учитывать ограничения на порядок выполнения работ. Конечная цель решения таких задач — нахождение оптимального допустимого расписания по тому или иному критерию оптимальности [16].

Частным видом задач теории расписаний является задача о назначениях [17]. Приведем традиционную формулировку данной задачи.

Задача о назначениях

Заданы:

1) некоторое множество работ (*jobs*)

$$J = \{J_1, J_2, \dots, J_n\} \quad (6)$$

с определенным набором характеристик для каждой работы: длительность, стоимость и т.д.;

2) некоторое множество исполнителей (средств — *means*)

$$M = \{M_1, M_2, \dots, M_m\}, \quad (7)$$

осуществляющих выполнение данной работы в соответствии с выбранным расписанием.

Дана также матрица себестоимости работ

$$C = \|c_{ij}\|_{n \times m}. \quad (8)$$

Здесь c_{ij} — затраты на выполнение работы J_i исполнителем M_j .

В задаче о назначениях (ЗН) требуется распределить работы по исполнителям таким образом, чтобы выполнить работы с минимальными затратами.

Решением данной задачи является инъективная функция

$$\mu: J \rightarrow M, \quad (9)$$

которая определяет пары (работа, исполнитель) и является частичной в случае $n > m$.

Если число $m = n$, т. е. число работ и исполнителей совпадают, то задача называется *сбалансированной*, иначе — не сбалансированной. В случае сбалансированной ЗН удовлетворяются два условия: каждый исполнитель выполняет только одну работу и каждая работа выполняется только одним исполнителем.

Заметим, что отображение (9) является биекцией в случае сбалансированной задачи: оно ставит в соответствие каждой работе определенного исполнителя.

Несбалансированная ЗН может быть приведена к сбалансированному виду введением недостающего числа фиктивных работ либо исполнителей. Так, если имеется избыток исполнителей, то следует назначить недостающее число работ с заведомо большой ценой для каждого исполнителя и после решения задачи игнорировать их. Аналогично поступаем в случае избытка работ.

Задачу о назначениях можно представить как задачу булева линейного программирования. Далее $\mathbf{B} = \{0, 1\}$ — булево множество.

Введем *матрицу назначений*:

$$\mathbf{X} = \|x_{ij}\|_{n \times m}, \quad (10)$$

где $x_{ij} \in \mathbf{B}$ — булева переменная: $x_{ij} = 1$, если исполнитель M_j назначен на работу J_i , и $x_{ij} = 0$ — в противном случае. Таким образом,

$$x_{ij} = 1 \Leftrightarrow j = \mu(i).$$

Суммарные затраты на выполнение всех работ:

$$C(\mathbf{X}) = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}. \quad (11)$$

Решается оптимизационная задача с булевыми переменными $x_{ij} \in \mathbf{B}$:

$$C(\mathbf{X}) \rightarrow \min \quad (12)$$

при ограничениях:

$$\sum_{j=1}^m x_{ij} = 1, i = 1:n; \quad (13)$$

$$\sum_{i=1}^n x_{ij} = 1, j = 1:m. \quad (14)$$

Заметим, что ограничение (13) требует, чтобы для каждой работы был назначен один исполнитель, а ограничение (14) — чтобы каждому исполнителю была назначена в точности одна работа.

Одним из наиболее популярных алгоритмов решения данной задачи является венгерский алгоритм, обеспечивающий решение сбалансированной задачи о назначениях за полиномиальное время от числа работ [17, 18].

Формализация задачи распределения поисковых ресурсов

Организация поиска пространственных объектов сопряжена с распределением поисковых ресурсов (усилий, средств) по зонам поиска. При этом под поисковыми ресурсами понимаются наблюдатели со всеми их техническими средствами и качествами. В условиях ограниченности поисковых ресурсов стоит задача их оптимального распределения (см., напр., [19]).

Задачу оптимального распределения поисковых ресурсов сформулируем как задачу о назначениях, исходя из разбиения района поиска на зоны (5). Тогда выполняемые работы (6) — это поиск цели в заданной зоне, а исполнителями работ (7) являются наблюдатели.

Любому наблюдателю может быть назначена любая зона поиска. Дана матрица затрат ресурсов на выполнение поисковых работ (8): здесь c_{ij} — затраты на выполнение поиска цели наблюдателем M_j в зоне Ω_i .

Задача состоит в распределении наблюдателей по зонам таким образом, чтобы выполнить поисковые работы с минимальными затратами.

Дальнейшее усложнение данной задачи включает два аспекта: 1) учет возможности совместного действия нескольких наблюдателей в каждой зоне поиска; 2) переход от детерминированной к вероятностной постановке задачи.

Первый аспект учитывается посредством исключения ограничений (13). Второй аспект сопряжен с введением вероятностных характеристик как расположения цели, так и технических возможностей наблюдателей.

Вероятностная постановка задачи

В излагаемой вероятностной постановке задачи распределения поисковых ресурсов считаются заданными априорные вероятности нахождения цели в выделенных зонах поиска и условные вероятности

обнаружения целей в этих зонах используемыми средствами поиска.

Введем обозначения:

- w_i — вероятность нахождения цели в зоне Ω_i ($i = 1:n$);
- w_{ij} — вероятность обнаружения цели в случае, когда она находится в зоне Ω_i , а поиск в этой зоне осуществляет наблюдатель M_j , т. е. $x_{ij} = 1$;
- $\mathbf{w}_3 = \|\mathbf{w}_i\|_{1 \times n}$, $\mathbf{w}_{3/n} = \|\mathbf{w}_{ij}\|_{n \times m}$;
- $P(\mathbf{X})$ — полная вероятность обнаружения цели для принятого плана назначений поисковых ресурсов.

Ситуации нахождения цели в одной из зон образуют полную группу несовместных событий, поэтому согласно формуле сложения вероятностей

$$\sum_{i=1}^n w_i = 1.$$

Утверждение. Вероятность обнаружения цели вычисляется по формуле

$$P(\mathbf{X}) = \sum_{i=1}^n \sum_{j=1}^m p_{ij} x_{ij}, \quad (15)$$

где

$$p_{ij} = w_i w_{ij}. \quad (16)$$

Доказательство. Пусть распределение наблюдателей по зонам описывается функцией (9). Согласно формуле полной вероятности

$$P = \sum_{i=1}^n w_i w_{i, \mu(i)}. \quad (17)$$

Очевидно,

$$w_{i, \mu(i)} = \sum_{j=1}^m w_{ij} x_{ij}. \quad (18)$$

Подстановка (18) в (17) с учетом обозначения (16) дает (15).

Пусть задана нижняя граница вероятности обнаружения цели $\underline{p}$:

$$P(\mathbf{X}) \geq \underline{p}. \quad (19)$$

Вероятностную постановку задачи распределения поисковых средств с целевой функцией (12) дает экстремальная задача (10)—(15), (19).

Дуальная постановка задачи: вводится верхнее ограничение на затраты

$$C(\mathbf{X}) \leq \bar{c}, \quad \bar{c} = \text{const} \quad (20)$$

и ищется матрица назначений $\mathbf{X}$, обеспечивающая максимум вероятности обнаружения цели:

$$P(\mathbf{X}) \rightarrow \max. \quad (21)$$

Рассматриваемые задачи условной оптимизации (12), (19) и (21), (20) не могут быть решены венгерским методом и требуют привлечения других, более общих методов решения задач булева линейного программирования.

В следующих модельных примерах предполагается применение БПЛА для визуального поиска пропавшего объекта на местности (в частности, это может быть потерпевший аварию спортивный планер), осуществляемого с помощью технических средств дистанционного видеомониторинга.

Пример 1. Топографическая карта района поиска в масштабе 1:200 000 представлена на рисунке. Район разбит на $n = 6$ равновеликих зон. Площадь каждой зоны 36 км^2 ($6 \times 6 \text{ км}$).

Поиск осуществляется тремя группами однотипных мобильных средств (МС) соответственно с номерами: 1) M_1, M_2 ; 2) M_3, M_4, M_5 ; 3) M_6 . Таким образом, всего МС $m = 6$.

Матрица относительных (топливно-временных) затрат при облете зон в процессе их сканирования:

$$C = \|c_{ij}\|_{6 \times 6} = \begin{bmatrix} 4 & 4 & 5 & 5 & 5 & 3 \\ 1 & 1 & 2 & 2 & 2 & 1,5 \\ 3 & 3 & 4 & 4 & 4 & 2 \\ 1,5 & 1,5 & 2,5 & 2,5 & 2,5 & 2 \\ 1,5 & 1,5 & 2,5 & 2,5 & 2,5 & 1,5 \\ 3 & 3 & 4 & 4 & 4 & 2 \end{bmatrix}.$$

Для решения задачи воспользуемся средствами математического пакета MATLAB. Венгерский метод реализуется библиотечной функцией `munkres()`.

Код соответствующего m-скрипта:

```
[Y, Q] = munkres(C);
X = Y';
[W, J] = find(Y);
disp(X);
disp(J'); %jobs
disp(M'); % means
```

Здесь J и M — соответственно массивы номеров работ (зон) и приписанных им исполнителей (наблюдателей) в полученном решении.

Результат решения ЗН представлен в табл. 1.

Таблица 1

Решение ЗН — минимизация суммарных затрат

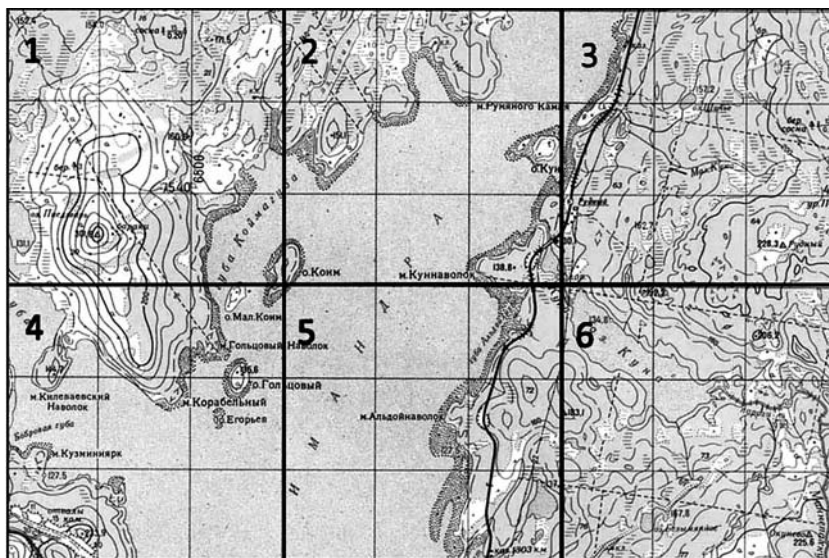
J	1	2	3	4	5	6
M	5	4	3	2	1	6

Оптимальные суммарные затраты $Q = 16$.

Пример 2. Включим вероятностные факторы в постановку задачи распределения поисковых ресурсов из примера 1.

Вектор априорных вероятностей нахождения цели в зонах:

$$w_3 = \|w_i\|_{1 \times 6} = (0,08, 0,22, 0,08, 0,30, 0,20, 0,12).$$



Разбиение области поиска на зоны

Матрица условных вероятностей обнаружения целей:

$$w_{3/n} = \|w_{ij}\|_{6 \times 6} = \begin{bmatrix} 0,55 & 0,55 & 0,50 & 0,50 & 0,50 & 0,70 \\ 0,85 & 0,85 & 0,75 & 0,75 & 0,75 & 0,90 \\ 0,70 & 0,70 & 0,60 & 0,60 & 0,60 & 0,75 \\ 0,75 & 0,75 & 0,75 & 0,75 & 0,75 & 0,90 \\ 0,75 & 0,75 & 0,70 & 0,70 & 0,70 & 0,90 \\ 0,70 & 0,70 & 0,55 & 0,55 & 0,55 & 0,75 \end{bmatrix}.$$

Для решения задачи в среде пакета MATLAB воспользуемся библиотечной функцией `bintprog()`, предназначенной для решения задач булева линейного программирования.

Размерность решаемой оптимизационной задачи

$$N = n \times m = 36.$$

Для применения функции `bintprog()` необходимо преобразовать матрицу переменных $X \in \mathbf{B}^{n \times m}$ в вектор-столбец $x \in \mathbf{B}^N$, а также соответственно изменить форму представления функций (11), (15) и уравнений (13), (14).

Положим, что вектор неизвестных переменных x является вертикальной конкатенацией вектор-столбцов матрицы X . Построим векторы $c, p \in \mathbf{B}^{N \times 1}$ посредством операции вертикальной конкатенации вектор-столбцов соответственно матриц C и P . В системе MATLAB данная операция выполняется функцией `reshape()`:

$$c = \text{reshape}(C, N, 1); p = \text{reshape}(P, N, 1);$$

Для замены в уравнениях (13) и (14) матрицы X на вектор x перепишем данные уравнения в другой векторно-матричной форме:

$$A_1 x = b_1, A_2 x = b_2, \quad (22)$$

где

$$A_1 = J_{1,m} \otimes E_n, b_1 = J_{n,1}; A_2 = E_m \otimes J_{1,n}, b_2 = J_{m,1}.$$

Здесь символ $\otimes$ обозначает кронекерово умножение матриц; E_k — единичная матрица k -го порядка, а $J_{k,l}$ — матрица единиц размера $k \times l$, — каждый ее элемент равен единице.

Объединим уравнения (20):

$$Ax = b.$$

Здесь матрица A образована вертикальной конкатенацией матриц A_1 и A_2 , а вектор b образован вертикальной конкатенацией векторов b_1 и b_2

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

Соответствующий фрагмент кода на MATLAB:

```
A1 = kron(ones(1, m), eye(n, n)); b1 = ones(n, 1);
A2 = kron(eye(m, m), ones(1, n)); b2 = ones(m, 1);
A = [A1; A2]; b = [b1; b2];
```

Наконец, для решения задачи (12), (19) исполняются команды

```
[x, Fval] = bintprog(c, -p', -Pl, A, b);
C = Fval;
```

а задачи (21), (20) — команды

```
[x, Fval] = bintprog(-p, c', Cu, A, b);
P = -Fval;
```

Приведем результаты решения трех вариантов постановки задачи оптимизации поисковых ресурсов.

1. Решение задачи из примера 1, оптимальное по критерию стоимости поисковых работ, дает вероятность обнаружения цели $P_1 = 0,7180$.

2. В случае отсутствия ограничений на стоимость работ получаем предельную вероятность обнаружения объекта $P_{\max} = 0,7690$.

Результат решения задачи представлен в табл. 2.

Таблица 2

Решение ЗН — максимизация вероятности обнаружения цели

J	1	2	3	4	5	6
M	5	2	3	6	4	1

Плановые затраты на поисковые работы $C = 17,5$.

3. Решим задачу минимизации затрат (12) при ограничении снизу (19) на вероятность обнаружения цели. Полагаем $\underline{p} = 0,99P_{\max} = 0,7613$.

Оптимальные назначения представлены в табл. 3.

Таблица 3

Решение ЗН — оптимизация затрат с учетом допустимой вероятности обнаружения цели

J	1	2	3	4	5	6
M	5	2	4	3	6	1

Затраты $C = 17,0$. Вероятность обнаружения $P = 0,7640$.

Заключение

Работа посвящена проблеме оптимального распределения ресурсов в задачах пространственного поиска объектов. Изложены общие положения классической теории поиска. Дана характеристика задачам дискретного поиска: их важнейшим методологическим аспектом является возможность применения концептуального и математического аппарата теории расписаний, сводящего решаемую задачу к задачам комбинаторной оптимизации. Предложена формализация задач оптимального распределения поисковых ресурсов в детерминированной и вероятностной постановках. Приведены примеры решения задачи оптимального распределения поисковых ресурсов с применением стандартных инструментальных средств математического пакета MATLAB.

Список литературы

1. Абчук В. А., Суздаль В. Г. Поиск объектов. М.: Сов. радио, 1977. 334 с.
2. Головинский О. Б., Лавинский Г. В. Поисковые системы. Киев: Техника, 1979. 103 с.
3. Морз Ф. Теория поиска // Исследование операций. Т. 1. М.: Мир, 1981. С. 549—629.
4. Хеллман О. Введение в теорию оптимального поиска. М.: Наука, 1985. 248 с.
5. Ким Д. П. Методы поиска и преследования подвижных объектов. М.: Наука, 1989. 336 с.
6. Беседин В. М., Крамарчук М. Г. Оптимальный взаимный поиск в системе управления роботами // Управление в робототехнических комплексах и гибких автоматизированных производствах: Межвуз. сб. науч. трудов. М.: МИРЭА, 1987. С. 151—156.
7. Строцев А. А. Оптимальный поиск неподвижной цели многопозиционной информационной системой // Журнал радиоэлектроники. 2004. № 4. С. 1.
8. Савич А. В., Кириллов И. Г. Бурковский С. И. Оптимизация автономного поиска целей многофункциональной РЛС многоканального зенитно-ракетного комплекса средней дальности // Системы обработки информации, 2004. Вып. 9 (37). С. 158—163.
9. Маркушин Н. А. Использование имитационного моделирования для поиска морских подвижных объектов // Сб. докл. Третьей всерос. науч.-практ. конф. "Имитационное моделирование. Теория и практика". Т. II. СПб.: ЦНИИ технологии судостроения, 2007. С. 124—129.
10. Кокуев А. А., Клитров С. В. Оптимизация способов свободного поиска воздушных целей истребителями в заданном районе // Военная мысль. 2013. № 11. С. 54—60.
11. Филимонов А. Б., Филимонов Н. Б., Тихонов В. Ю. Планирование операций в задачах пространственного поиска объектов // Известия ЮФУ. Технические науки. Темат. выпуск "Перспективные системы и задачи управления". 2017. № 1—2. С. 185—197.
12. Sarris Z. Survey of UAV applications in civil markets. June 2001 [Электронный ресурс]. URL: http://med.ee.nd.edu/MED9/Papers/Aerial_vehicles/med01-164.pdf
13. Абдулов Р. Н., Абдуллаев Н. А., Асадов Х. Г. Вопросы оптимизации применения БПЛА для поиска и слежения объектов // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 1 (83). С. 45—49.
14. Конвей Р. В., Максвелл В. Л., Миллер Л. В. Теория расписаний. М.: Наука, 1975. 360 с.
15. Корте Б., Фиген И. Комбинаторная оптимизация. Теория и алгоритмы. М.: МЦНМО, 2015. 720 с.
16. Лазарев А. А., Гафаров Е. Р. Теория расписаний. Задачи и алгоритмы. М.: МГУ им. М. В. Ломоносова, 2011. 222 с.
17. Таха Х. А. Задача о назначениях // Введение в исследование операций. Гл. 5, п. 5.4. М.: Вильямс, 2001. С. 206—213.
18. Kuhn H. W. The Hungarian for the Assignment Problem // Naval Research logistics Quarterly, 1955. Vol. 2. P. 83—97.
19. Аркин В. И. Задачи оптимального распределения поисковых усилий // Теория вероятностей и ее применения. 1964. Т. 9, № 1. С. 179—180.

A. B. Filimonov, D. Sc., Professor, filimon_ab@mail.ru,
Moscow Technological University, 78, Vernadsky Avenue, Moscow, 119454, Russia,
N. B. Filimonov, D. Sc., Professor, nbfilimonov@mail, M. V. Lomonosov Moscow State University,
Leninskie Gory, Moscow, 119991, Russia,
V. Yu. Tikhonov, Research Assistant, Centre for Educational Development
and International Activities, 8, Bolshoi Chudov per., Moscow, 119021, Russia

Optimization of the Resources Distribution in the Problems of Space Search Objects

One of the most important areas of modern science and technology is the space of object. The task of search arises when it is required to determine the position of some object (goal) being in the given area of physical space with the help of search means. The theory of search is one of the sections of the operations theory. The subject of its study are the mathematical models of search processes, methods of organization and optimal planning of search operations, providing the detection of goals with minimal time and resource inputs.

The general positions of the classical theory of search are presented in the paper. The term "observers" is admissible for subjects fulfilling the search. It is noted the specific character of tasks discrete search of objects. In the continual tasks of search the kinematics of the motion of the observer in the physical space is investigated. In the problems of discrete search the investigated area is represented as the rotation of the investigation of chosen zones by having means of observation.

The important advantage of the discrete search problems is the possibility of the application to them the methodology and mathematical device of scheduling theory, reducing the solved problem to the problems of the combinatorial optimization.

The problems of optimal distribution of search resources by the distinguished areas of search is considered. The given problems belong to class of the assignment problems. Their formalization in the determined and probability formulations is proposed. The model examples of the optimization of the distribution of search resources with use of standard instrumental software of the problems of Boolean linear programming are reduced.

Keywords: space object search, discrete search, planning search operations, assignment problems, optimal distribution of search resources, probabilistic models for planning purposes

References

1. **Abchuk V. A., Suzdal' V. G.** *Poisk ob'ektov* (Search of objects), Moscow, Sov. radio, 1977. 334 p.
2. **Golovinskij O. B., Lavinskij G. V.** *Poiskovyie sistemy* (Search systems). Kiev: Tekhnika, 1979. 103 p.
3. **Morz F.** *Teoriya poiska*, Issledovanie operatsij (Theory of search) / Survey operations]. v. 1. Moscow, Mir, 1981, pp. 549–629.
4. **Khellman O.** *Vvedenie v teoriyu optimal'nogo poiska* (Introduction to the theory of optimal search), Moscow, Nauka, 1985. 248 p.
5. **Kim D. P.** *Metody poiska i presledovaniya podvizhnykh ob'ektov* (Methods of search and pursuit of moving objects), Moscow, Nauka, 1989. 336 p.
6. **Besedin V. M., Kramarchuk M. G.** *Optimal'nyj vzaimnyj poisk v sisteme upravleniya robotami* (Optimal mutual search system robot control), *Upravlenie v robototekhnicheskikh kompleksakh i gibkikh avtomatizirovannykh proizvodstvakh*, Mezhdvuz. sbornik nauch. trudov, Moscow, MIEHRA, 1987, pp. 151–156.
7. **Strotsev A. A.** *Optimal'nyj poisk nepodvizhnoj tseli mnogo-pozitsionnoj informatsi-onnoj sistemoj* (Optimal search for a stationary target multi-position information system), *Zhurnal radioelektroniki*, 2004, no. 4, pp. 1.
8. **Savich A. V., Kirillov I. G., Burkovskij S. I.** *Optimizatsiya avtonomnogo poiska tselej mnogofunktsional'noj RLS mnogokanal'nogo zenitno-raketnogo kompleksa srednej dal'nosti* (Optimization of Autonomous search for targets multi-function radar multi-channel anti-aircraft missile complex medium-range), *Sistemi obrabki informatsii*, 2004, is. 9 (37), pp. 158–163.
9. **Markushin N. A.** *Ispol'zovanie imitatsionnogo modelirovaniya dlya poiska morskikh podvizhnykh ob'ektov* (The use of simulation to search for sea mobile objects) // *Sbornik dokl. Tre'ej vsereoss. nauch.-prakt. ronf. "Imitatsionnoe modelirovanie. Teoriya i praktika"*, vol. II, Saint Petersburg: TSNIi tekhnologii sudostroenie, 2007, pp. 124–129.
10. **Kokuev A. A., Ktitrov S. V.** *Optimizatsiya sposobov svobodnogo poiska vozdushnykh tselej istrebitelyami v zadannom rajone* (Optimization methods free search of air targets by fighters in a given area), *Voennaya mys'*, 2013, no. 11, pp. 54–60.
11. **Filimonov A. B., Filimonov N. B., Tihonov V. Yu.** *Planirovanie operacij v zadachah prostranstvennogo poiska ob'ektov* (Operations scheduling in tasks spatial search objects), *Izvestiya YUFU. Tekhnicheskije nauki. Temat. vypusk "Perspektivnye sistemy i zadachi upravleniya"*, 2017, no. 1–2, pp. 185–197.
12. **Sarris Z.** *Survey of UAV applications in civil markets*, June 2001, URL: http://med.ee.nd.edu/MED9/Papers/Aerial_vehicles/med01-164.pdf.
13. **Abdulov R. N., Abdullaev N. A., Asadov KH. G.** *Voprosy optimizatsii primeneniya BPLA dlya poiska i slezheniya ob'ektov* (The optimization of the use of drones for search and tracking of objects), *Nauchno-tehnicheskij vestnik informatsionnykh tekhnologij, mekhaniki i optiki*, 2013, no. 1 (83), pp. 45–49.
14. **Konvej R. V., Maksvell V. L., Miller L. V.** *Teoriya raspisanij* (Theory of scheduling). Moscow, Nauka, 1975, 360 p.
15. **Korte B., Figen I.** *Kombinatornaya optimizatsiya. Teoriya i algoritmy* (Combinatorial optimization. Theory and algorithms), Moscow, MTSNMO, 2015, 720 p.
16. **Lazarev A. A., Gafarov E. R.** *Teoriya raspisanij. Zadachi i algoritmy* (Theory. Problems and algorithms), Moscow, MGU im. M. V. Lomonosova, 2011. 222 p.
17. **Takha Kh. A.** *Zadacha o naznacheniakh* (Task assignment), *Vvedenie v issledovanie operatsij*, part. 5, p. 5.4, Moscow; Vil'yams, 2001, pp. 206–213.
18. **Kuhn H. W.** *The Hungarian for the Assignment Problem*, *Naval Research logistics Quarterly*, 1955, vol. 2, pp. 83–97.
19. **Arkin V. I.** *Zadachi optimal'nogo raspredeleniya poiskovykh usilij* (The problem of optimal distribution of search efforts), *Teoriya veroyatnostej i eyo primeneniya*, 1964, vol. 9, no. 1, pp. 179–180.

М. М. Низамутдинов, канд. техн. наук, доц., зав. сектором, e-mail: marsel_n@mail.ru,
В. В. Орешников, канд. экон. наук, ст. науч. сотр., e-mail: VOresh@mail.ru,
ФГБУН Институт социально-экономических исследований Уфимского научного центра РАН

Концепция реализации системы поддержки принятия решений в сфере управления инновационным развитием регионов на базе адаптивно-имитационной модели¹

Рассматриваются вопросы разработки системы поддержки принятия решений (СППР) в области управления инновационным развитием региональных социально-экономических систем. Проведен содержательный анализ известных подходов и инструментальных средств поддержки принятия решений, сформулированы основные требования к организации поддержки принятия решений для исследуемой задачи. Предложена концепция разработки СППР для обоснования параметров инновационной стратегии развития региона на основе использования адаптивно-имитационной модели. Охарактеризована логика взаимосвязи и структура функциональных блоков СППР, позволяющие организовать информационную поддержку рассматриваемого процесса.

Ключевые слова: поддержка принятия решений, инновационное развитие, имитационное моделирование, региональное управление

Введение

Стратегическая для регионов России задача выхода на инновационную модель развития требует сегодня масштабного внедрения передовых технологий и новаций, обеспечивающих выход на новый технологический уклад с производством товаров и услуг с высокой добавленной стоимостью. Не менее важная задача — формирование непосредственно инновационно ориентированной системы стратегического планирования, которая могла бы обеспечить встраивание потенциальных инновационных центров в существующую, крайне инерционную модель воспроизводственного процесса, просчитывать влияние инновационных факторов на развитие экономической системы в целом, взаимно адаптировать и синхронизировать эти процессы в рамках принятого горизонта стратегического планирования.

В связи с этим актуализируются вопросы выработки подходов к прогнозированию и планированию развития регионов, учитывающих данные особенности. На сегодняшний день решение этой проблемы сопряжено с рядом сложностей практического характера, обусловленных, в том числе, несогласованностью целей (по уровням, горизонтам и субъектам управления), недостаточной системностью реализуемой экономической политики, отсутствием четкой системы анализа и эффективного использования управленческой информации, а также недостаточной методической разработанностью вопросов стратегического планирования на региональном уровне. Как результат этого наблюдается низкое качество формируемых прогнозных оценок по основным параметрам развития ре-

гионов, в том числе в рамках программ и стратегий инновационного развития, что обуславливает необходимость их постоянного пересмотра уже в ходе реализации стратегий.

В этих условиях дальнейшее развитие методов стратегического планирования на региональном уровне объективно требует их качественного переосмысления, обусловленного, в первую очередь, необходимостью учета постоянных изменений параметров и условий внутри глобальной конкурентной рыночной среды, их влияния на инновационное развитие. В таких условиях формирование инновационных программ даже на среднесрочную перспективу требует внедрения механизмов взаимного учета интересов и адаптации стратегий поведения экономических агентов разного уровня, наличия процедур предварительного согласования, а затем — совместной реализации, мониторинга и итеративной корректировки стратегических параметров инновационного развития территорий. В данном контексте эффективным инструментарием совершенствования методов стратегического планирования могут стать современные технологии поддержки принятия решений на основе адаптивных методов управления и имитационного моделирования, позволяющие обеспечить качественную информационную поддержку при разработке и реализации программ инновационного развития на региональном уровне.

Анализ подходов и инструментальных средств

Анализ современного состояния исследований по рассматриваемой проблематике подразумевает комплексное рассмотрение подходов, посвященных стратегическому управлению региональным развитием, вопросам теории инновационной экономики, а также системам поддержки принятия решений.

¹ Исследование проводится при поддержке РФФИ, № 16-29-12883.

Кроме того, отдельно следует выделить вопросы построения экономико-математических моделей, применяемых в указанных сферах и являющихся, в определенной степени, связующим звеном.

Выделенные направления, каждое в отдельности, имеют достаточно обширную научную базу. Так, основы теории инновационной экономики были созданы еще в начале XX века австрийским экономистом Й. Шумпетором, который дал определение и классификацию инноваций. Инновационным процессом уделено внимание в исследованиях длинных волн Кондратьева, диффузий инноваций К. Фримена и др. На сегодняшний день ведущие в научно-техническом отношении страны мира создали постиндустриальное общество, в котором сектор инновационной экономики стал доминирующим (Ф. Фукуяма, Дж. Нейсбитт и др.).

Отдельного упоминания заслуживает также теория технологических укладов, рассматриваемая во взаимоотношенности с вопросами инновационного развития экономики (С. Ю. Глазьев, Д. С. Львов, К. Перес, Ю. В. Яковец и др.). Следует отметить, что проводимые в данном направлении исследования носят в значительной степени характер качественного анализа и диагностики инновационных процессов и не всегда ориентированы на количественную оценку взаимовлияния инноваций и социально-экономического развития территории и, в частности, формирования соответствующих стратегических прогнозов и планов развития.

Сегодня становится понятным, что с технологической точки зрения решение данной задачи требует разработки специализированных методов поддержки принятия решений, в том числе систем класса DSS (ERP, OLAP, Data Mining, CRM и т. д.). Однако и здесь большинство имеющихся на сегодняшний день разработок направлено на решение проблем микроэкономического уровня, в частности, обеспечение деятельности отдельных корпораций, начиная от планирования ресурсов предприятия и управления проектами и заканчивая поддержкой принятия решения в части работы со слабоструктурированной информацией. На макроэкономическом же уровне, в том числе в рамках разработки стратегий инновационного развития регионов, подобные системы практически не применялись. При этом существующие отдельные исследования в данной сфере, например, информационно-аналитическая система поддержки принятия решений (СППР) по управлению инновационной деятельностью в регионе [1], СППР по стратегии инновационного развития региона [2] и т. д., не в полной степени отражают такое значимое свойство региональной системы, как адаптивность поведения основных экономических субъектов. Имеется также ряд исследований, которые носят сугубо теоретический и/или отраслевой характер [3] и не могут быть применены в реальной практике управления на макроуровне.

В области экономико-математического моделирования региональных процессов и систем также выделяют различные направления, в частности, модели общеэкономического равновесия; модели, основанные на использовании вероятностно-статистических методов; модели системной динамики и прочие модели имитационного типа; модели, основанные на использовании различных интеллектуальных информационных технологий. При этом в большинстве своем реальные модели комбинируют различные подходы. В качестве примера зарубежных моделей, построенных на основе балансового и эконометрического методов, можно привести "Уортонскую годовую модель экономики США" (Wharton model), модель "LIFT" (Long-term Interindustry Forecasting Tool), "Брукингскую модель экономики США", "Модель налогового-бюджетной политики США".

Среди отечественных моделей следует выделить модельно-программные комплексы "СИРЕНА" и "СИРЕНА-2" (Институт экономики и организации промышленного производства Сибирского отделения РАН), модельный комплекс "ПОЛИГОН-2" (Новосибирский государственный университет), модель "Russian Interindustry Model" (Институт народнохозяйственного прогнозирования РАН), "Модель межотраслевых взаимодействий" (ИЭП НТП АН СССР, Ю. В. Яременко), CGE-модель "RUSEC", "Эконометрическая модель экономики России", "CGE-модель социально-экономической системы России со встроенными нейронными сетями" (ЦЭМИ РАН), "Модель региона Российского Севера" (Сыктывкарский государственный университет), модель "Губернатор" и "Агент-ориентированная модель г. Москвы" (ЦЭМИ РАН) и т. д. На основе имитационных моделей также построены многие комплексные информационно-аналитические системы, в том числе широко используемый в реальной практике регионального управления информационно-аналитический комплекс "Прогноз" (ЗАО "Прогноз").

Вместе с тем представленные модели зачастую не отражают особенности инновационного развития регионов, поскольку не содержат описания параметров инновационной деятельности как управляемого адаптивного процесса. Кроме того, многие модели, построенные, в частности, на базе эконометрических уравнений, теряют свои прогностические способности в условиях изменения структуры экономики ввиду недостаточного учета адаптивных свойств элементов региональной системы, а потому не являются (а в большинстве случаев и не могут являться) основой для формирования систем поддержки принятия решений. Таким образом, несмотря на обширное число исследований по данной тематике остается нерешенным ряд вопросов, касающихся как методических, так и практических аспектов поддержки принятия решений, особенно в части моделирования инновационного развития

сложных социально-экономических систем регионального уровня и задействования адаптивных механизмов согласования интересов экономических агентов.

Концептуальные аспекты и требования к реализации СППР

Рассмотренные выше методологические проблемы в сфере стратегического планирования на региональном уровне обосновывают необходимость разработки гибкого и адаптивного инструментария управления, позволяющего, с одной стороны, учитывать фундаментальные изменения принципов взаимодействия субъектов в рамках инновационной экономики, а с другой — оперативно оценивать последствия принимаемых решений. Исходя из этого необходимо качественное развитие методической и инструментальной базы формирующейся сегодня системы управления инновационным развитием региональных систем за счет интеграции в рамках единого инструментария таких методов, как стратегическое управление, экономико-математическое моделирование, теория поддержки принятия решений и имитационное моделирование, а также использование современных компьютерных и информационных технологий.

На наш взгляд, такой синтез различных подходов может быть реализован в рамках использования СППР на базе адаптивных имитационных моделей [4], в структуру которых будут интегрированы проблемно-ориентированные базы знаний с механизмом логического вывода, а также интеллектуальные технологии обработки слабоструктурированной информации, используемые для поиска решений в процессе формирования и корректировки параметров управления инновационным развитием региона.

В целом теоретическое обоснование разработки подобной проблемно-ориентированной СППР должно включать реализацию нескольких последовательно взаимосвязанных этапов, определяющих основные конструктивные особенности.

В рамках *концептуального этапа исследований* необходимо четко сформулировать базовые принципы и требования к разработке инструментария поддержки принятия решений применительно к задаче формирования стратегий инновационного развития регионов. Необходимо также сформировать общую методологию исследования, обосновать подходы, методы и технологии, которые будут задействованы при разработке инструментария поддержки принятия решений, а также сформулировать непосредственно концепцию построения адаптивно-имитационной модели региона. При этом методология исследования должна определять системный взгляд на региональную социально-экономическую систему с выделением места, роли и взаимосвязей экономических, социальных и инновационных процессов в рамках единого воспроиз-

водственного процесса, формализуемых на базе адаптивной имитационной модели экономики региона как функционального ядра системы поддержки принятия решений.

В рамках *этапа проектирования и математической формализации адаптивной имитационной модели региона* должно осуществляться встраивание структурных элементов и функциональных связей, необходимых для моделирования стратегий инновационного развития на региональном уровне. Необходимо на базе предложенной методологии исследования и концепции моделирования разработать последовательно логические, информационные и математические модели для всего множества экономических агентов регионального уровня. Также необходимо разработать и встроить в общий контур подсистему управления региональным развитием, ориентированную на решение задачи инновационного развития, осуществить системную интеграцию всех функциональных блоков и подсистем.

Этап конструирования управляющей подсистемы должен включать разработку модели и механизмов формирования индикативного плана развития региональной системы и обоснование множества управляющих и регулирующих параметров в целях формирования стратегий экономического роста на основе задействования факторов инновационного развития. Механизм формирования целевых индикаторов и параметров управления в рамках обоснования параметров стратегий инновационного развития на региональном уровне позволит также интегрировать в единой информационной среде процедуры целеполагания и регулирования. При этом наличие нечеткого алгоритма классификации ситуаций и корректировки индикативного плана позволит вырабатывать регулирующие воздействия для достижения параметров индикативного плана в условиях взаимной адаптации целей функционирования объекта и системы управления. Формирование такого механизма должно предполагать сочетание особенностей поведения экономических агентов и правил принятия решений органами государственного управления при реализации ими инновационной стратегии развития региона.

На *этапе конструирования непосредственно программного обеспечения СППР* необходимо системно интегрировать разработанные модели и алгоритмы в единый инструментарий, включая базы статистических данных, программный и пользовательский интерфейс, модуль управления с возможностью конструирования сценарных карт, графическую систему вывода и отображения результатов экспериментальных и сценарных расчетов и т. д.

На *этапе методического обеспечения СППР* необходимо также предусмотреть разработку методики формирования стратегий инновационного развития региона с применением адаптивно-имитационной модели. Содержательно это включает, в том

числе, процедуры формирования индикативного плана, определения регулирующего воздействия и проведения комплекса вычислительных экспериментов. Данная методика позволит увязывать ресурсные возможности и интересы экономических агентов и подсистемы управления региональной социально-экономической системы в рамках единой процедурной, расчетной, методической базы, что обеспечит возможность системного моделирования стратегий инновационного развития.

И наконец, на *этапе практической апробации СППР* необходимо провести комплекс экспериментальных и сценарных расчетов с использованием разработанной имитационной модели, в рамках которых будет обосновываться выбор наиболее предпочтительной стратегии инновационного развития региона в среднесрочной перспективе.

Важно отметить, что базовым принципом исследования на всех его этапах становится реализация системного подхода, определяющего необходимость рассмотрения всех ключевых элементов и взаимосвязей региональной системы, формирующих базис инновационного развития, а также описание их основных свойств и детерминирующих факторов инновационного развития. В рамках предлагаемого подхода существенными являются также принципы адаптивности и согласованности процессов принятия решений как экономическими агентами, так и управляющей системой региона при реализации мер государственной экономической политики.

При этом в отличие от существующих информационно-аналитических систем предлагаемый к разработке инструментарий поддержки принятия управленческих решений будет базироваться на использовании адаптивной имитационной модели региона, ориентированной на формирование инновационной стратегии развития экономики. Такой класс моделей позволяет целостно отражать адаптивное поведение экономических агентов и региональной системы управления в процессе реализации их стратегий, он отличается наличием встроенных контуров взаимодействия субъектов для различных классов экономических ситуаций и позволяет адаптировать их ресурсные стратегии при согласовании общих целей функционирования.

Практические аспекты реализации СППР на базе адаптивно-имитационной модели

В самом общем виде конструируемая СППР в области управления региональным инновационным развитием, на наш взгляд, должна решать следующие базовые задачи:

- сбор и накопление фактографической информации, структурированной информации в виде правил принятия решений, а также слабоструктурированной информации в виде онтологий и базы прецедентов;
- анализ ретроспективной, текущей и прогнозной ситуации с использованием проблемно-ориен-

тированной базы знаний с механизмом логического вывода;

- формирование прогноза развития ситуации в различных макроэкономических условиях на базе адаптивной имитационной модели;
- выработка комплекса рекомендаций по формированию и корректировке параметров управления инновационным развитием региона.

При этом, учитывая специфику объекта управления, особенности его функционирования и развития, а также принимая во внимание цели и задачи, ставящиеся перед разрабатываемой СППР, в качестве ее ядра в данном случае целесообразным представляется выбрать такую конструкцию имитационной модели, которая реализует возможность адаптации поведения различных экономических агентов как к изменению условий макросреды, так и к поведению других агентов. Кроме того, следует обратить внимание на то, что существенное влияние оказывают и характеристики самого рассматриваемого агента, правила принятия им решений, которые могут основываться не только на "собственном опыте", но и включать правила, выявленные при анализе поведения иных агентов. Решению подобных задач способствует использование возможностей адаптивной имитационной модели региона, логическая структура которой, на наш взгляд, должна интегрировать в единой иерархически организованной конструкции три ключевых уровня — экономические агенты, органы регионального управления, макросреда.

Стратегия поведения экономического агента в рамках данной модели представляет собой набор условий и соответствующих им правил поведения. Тем самым учитывается ситуационный характер принятия решений. Необходимо отметить, что рассматриваются как "внутренние" условия (в том числе состояние экономического агента, объем имеющихся запасов, потребности экономического агента), так и "внешние" условия (в том числе возможность получения ресурсов от других экономических агентов и внешней среды, изменение макроэкономических показателей). Изменение условий способно привести не только к количественным, но и к качественным изменениям в функционировании экономического агента. Такая структура построения адаптивно-имитационной модели (АИМ) позволяет получать сбалансированные прогнозно-плановые оценки показателей регионального развития. Однако следует отметить, что сама по себе имитационная модель не позволяет решить все задачи, стоящие перед СППР, и требует дополнения комплексом функциональных блоков. Состав и взаимосвязь между блоками определяется исходя из поставленных задач и общей логики формирования управленческого решения (см. рисунок).

Первым этапом является формирование комплекса исходных данных. Блок ввода данных обеспечивает наполнение базы данных различной ин-

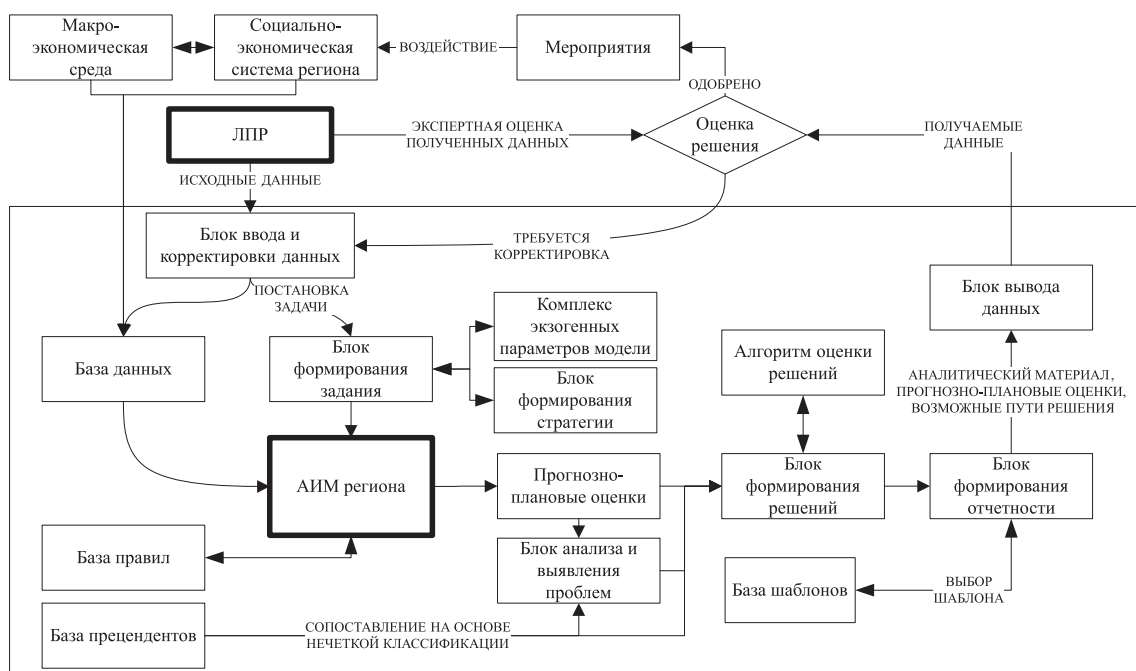


Схема организации СППР на базе адаптивно-имитационной модели

формацией о состоянии объекта управления и внешней среды. В связи с тем, что данные должны обновляться в системе периодически, необходимым условием ее функционирования является организация мониторинга. Это позволяет не только иметь представление о текущем и ретроспективном состоянии региона, но и отслеживать последствия принимаемых управленческих решений, а соответственно расширять и совершенствовать имеющиеся базы правил и базы прецедентов. В этом контексте следует принимать во внимание масштабность данной работы. База данных хранит в себе информацию о региональной социально-экономической системе, а также о значимых характеристиках внешней среды. Необходимо учесть отражение в ней разнородной информации, относящейся к различным объектам с учетом изменения их характеристик во времени. Обработка огромных массивов информации требует применения современных информационных технологий обработки данных и интеллектуального анализа. В связи с этим хранение данных предлагается организовать на базе концепции OLAP (в том числе за счет свойств многомерной базы) и Data Mining. Как отмечают эксперты в данной области, с помощью Data Mining можно выявить скрытые закономерности в больших объемах информации [5].

Кроме того, блок ввода служит для постановки задачи пользователем системы — лицом, принимающим решения (ЛПР). На рисунке данный этап представлен блоком формирования задания. Необходимость подобного блока обусловлена, с одной стороны, разнообразием практических задач и направлений применения СППР, начиная от анализа ретроспективной информации и заканчивая выра-

боткой альтернативных решений при планировании инновационного развития [6]), а с другой стороны — необходимостью ввода не только отчетных данных, но и экзогенных параметров модели, а также выбором стратегии развития. Комплекс экзогенных параметров модели включает в себя сценарные параметры макроэкономической среды, управляемые параметры, а также иные показатели, которые используются в расчетах, но не моделируются в рамках самого инструментария. Блок формирования стратегии должен позволять формировать общую стратегию развития региональной системы и определять ряд параметров модели до проведения модельного эксперимента в целях выбора приоритетного направления развития. В частности, могут быть выделены социально ориентированная, инвестиционно ориентированная, инновационная [7], комплексная и иные стратегии. Таким образом, блок ввода данных является интерфейсом ввода статистической информации, управляющих и сценарных параметров.

Совокупность указанных выше данных будет являться входной информацией для адаптивной имитационной модели региональной системы. В рамках данной статьи не будем подробно останавливаться на внутреннем устройстве адаптивной модели и алгоритмах расчета итоговых показателей. Однако следует отметить, что имитационное моделирование подразумевает возможность не только выбора правил поведения из имеющейся базы правил, но и корректировку самой этой базы. В зависимости от поставленной задачи такая модель может решать задачи как прогнозирования (построение прогноза развития социально-экономической системы без задействования управляющих алгоритмов), так и планирования (с учетом управленческих установок).

В то же время постановка задачи формирования параметров инновационного развития региона [8] накладывает определенные требования к процедуре проведения модельных экспериментов. В частности, необходимым является определение показателя инновационного развития региона. Исследователями отмечается, что при решении данной задачи могут быть выдвинуты следующие требования [9]:

- агрегирование многих критериев, имеющих различную размерность и направленность изменений;
- учет весов критериев, т. е. их важности в интегральной оценке;
- формализация нечетких понятий для обеспечения эффективной обработки качественной информации наравне с четкими количественными данными;
- привязка интегрального показателя к целевым ориентирам стратегического развития региона и т. д.

Полученные прогнозно-плановые оценки поступают в блок анализа и выявления проблем. На данном этапе осуществляется структурный анализ, анализ динамики и иные процедуры, позволяющие выявить как непосредственно проблемы, так и предпосылки для их возникновения. Для этого применяется база прецедентов, содержащая информацию о возможности применения накопленного опыта решения проблем в процессе выработки решения новых задач. Основным понятием здесь является прецедент как структурированное представление накопленного опыта в виде данных и знаний, обеспечивающее его последующую автоматизированную обработку с помощью специализированных программных систем. Общая структура прецедента включает два основных компонента — идентифицирующую часть (описывает имеющийся опыт), обучающую часть (описывает решения и их характеристики). На основе прогнозно-плановых оценок, выявленных в ходе анализа проблем развития и имеющихся прецедентов, СППР формирует комплекс альтернативных решений, которые должны быть оценены по определенным критериям.

Одной из наиболее важных составляющих СППР является блок формирования решений, в рамках которого предполагается осуществлять классификацию ситуаций и сопоставление с имеющейся базой прецедентов для выбора возможных путей решения. При этом представляется целесообразным использовать методы теории нечеткой логики. Такой подход дает возможность отслеживать плавные изменения свойств объекта управления, а также задействовать качественные характеристики. При этом нечеткие переменные также наилучшим образом подходят для планирования факторов во времени, когда их будущая оценка затруднена. Здесь важнейшим этапом является построение функций принадлежности нечетких множеств, которые описывают семантику используемых базовых значений нечет-

ких и лингвистических переменных. Существует несколько методов построения функций принадлежности. Однако для разработки инструментария поддержки принятия решений в области управления региональным развитием необходимо учитывать такую особенность, как наличие разнородных источников информации (предприятия, население, общественные организации, органы государственного и муниципального управления и т. д.). Кроме того, выбор решений должен учитывать специфику инновационного развития.

Завершающим этапом является формирование отчета о результатах моделирования и предлагаемых способах решения выявленных проблем. Целесообразным при этом представляется использование готовых шаблонов унифицированных отчетов. Подобные отчеты позволяют ЛПР быстрее ориентироваться в получаемых данных. Однако следует понимать, что СППР является лишь инструментом, позволяющим подготовить решение, но не подменяет собой ЛПР, на которое возлагается ответственность. В связи с этим проводится экспертная оценка полученных данных. В случае одобрения предложенного решения на его основе разрабатывается мероприятие (комплекс мероприятий), реализация которого приводит к изменению состояния региональной системы. В случае его неодобрения осуществляется корректировка постановки задачи и повторный запуск СППР.

Следует отметить, что для ЛПР в рамках его практической деятельности использование СППР не должно быть связано с непосредственной настройкой отдельных параметров экономико-математической модели, проведением математических расчетов или изучением особенностей математического аппарата нечеткой логики. Взаимодействие с системой осуществляется через блоки ввода и вывода информации посредством привычного графического интерфейса, а вся внутренняя работа системы скрыта от пользователя.

Кроме указанных функциональных блоков и адаптивно-имитационной модели, для использования СППР потребуется также разработка методического обеспечения по практическому использованию инструментария, включающего, в том числе:

- методику сбора и обработки данных о состоянии региональной социально-экономической системы;
- методику проведения эксперимента с имитационной моделью социально-экономического развития региона;
- методику формирования управленческого решения на базе использования СППР.

В целом, предлагаемая СППР может стать универсальным средством, обеспечивающим ЛПР необходимой для принятия решений информацией о ретроспективном, текущем и будущем состоянии региональной системы, позволяющим проводить анализ и выявлять возможные пути разрешения сложившихся проблем.

В рамках проведенного исследования предложена концептуальная схема реализации СППР в области управления инновационным развитием региона, а также определено место адаптивной имитационной модели в структуре системы, показаны возможности ее использования при формировании прогнозно-плановых оценок развития региона, а также оценке эффективности применения альтернативных управляющих воздействий. Описана процедура формирования решения и раскрыты назначения функциональных блоков, которые позволяют работать с модельным комплексом для конечного пользователя, обеспечивая удобное использование инструментария для лица, не являющегося специалистом в области экономико-математического моделирования.

С теоретической точки зрения предложенный инструментарий позволит расширить возможности совместного применения как методов теории управления и поддержки принятия решения, интеллектуальных информационных технологий, так и экономико-математических методов, современных технологий компьютерного имитационного моделирования для задач стратегического планирования развития социально-экономических систем макро- и мезоуровня. При этом сам программный инструментарий потенциально имеет важную практическую направленность. Предложенная конструкция СППР может быть использована органами государственного управления в рамках определения параметров среднесрочного развития субъектов Российской Федерации, а также иных региональных систем при разработке социальной, экономической и бюджетной политики.

1. Палюх Б. В., Какатунова Т. В. Нечеткая когнитивная карта как инструмент моделирования инновационной деятельности на региональном уровне // Программные продукты и системы. 2012. № 4. С. 128—131.
2. Захарова А. А. Некоторые аспекты разработки информационной системы поддержки принятия стратегических решений об инновационном развитии региона // Современные наукоемкие технологии. 2007. № 9. С. 27—29.
3. Антамошкин А. Н., Антамошкина О. И., Ходос Д. В. Моделирование основных этапов формирования программы инновационного развития // Вестник Сибирского государственного аэрокосмического университета имени академика М. Ф. Решетнева. 2010. № 4 (30). С. 204—206.
4. Низамутдинов М. М., Орешников В. В. Инструментарий прогнозирования изменения параметров регионального развития на основе адаптивно-имитационного подхода // Труды 3-й Международной конференции "Информационные технологии и интеллектуальная поддержка принятия решений — ITIDS'2015". Уфа: Изд. УГАТУ, 2015. С. 211—215.
5. Фаттахов Р. В., Иванова Е. И., Сметанина О. Н. О роли информационных ресурсов при поддержке принятия управленческих решений на региональном уровне // Вестник Уфимского государственного авиационного технического университета. 2007. Т. 9, № 2. С. 82—87.
6. Строев П. В. Анализ факторов, оказывающих влияние на инновационную активность в экономике России // Вестник Финансового университета. 2012. № 2 (68). С. 143—148.
7. Гажва В. О., Губанова Е. В. Инновационное развитие Калужского региона // Материалы Международного научно-практического круглого стола "Направления социально-экономического развития региональной экономики". 2016. С. 122—125.
8. Тюшняков В. Н., Жертовская Е. В., Якименко М. В. Информационно-аналитическое обеспечение ситуационного центра как основа разработки стратегий инновационного развития региона // Фундаментальные исследования. 2015. № 11-6. С. 1253—1257.
9. Захарова А. А. Интегральная оценка инновационного развития региона на основе нечетких множеств // Научное обозрение. Технические науки. 2014. № 1. С. 161—168.

M. M. Nizamutdinov, Ph. D., Associate Professor, e-mail: marsel_n@mail.ru,
Head of Economic and Mathematical Modeling Sector,
V. V. Oreshnikov, Ph. D., Senior Researcher, e-mail: VOresh@mail.ru,
Institute of Social and Economic Research, Ufa Scientific Center

Decision Support System Conception for Regions Innovative Development Management based on Adaptive Simulation Model

The article discusses the development of decision support system (DSS) in the innovative development management for regional socio-economic systems. The meaningful analysis of known approaches and decision support tools has been made. Basic requirements for decision support organization have been formulated. The DSS conception is proposed to justify the innovation strategy parameters for region development based on adaptive simulation model. The logic and the structure of DSS functional blocks are characterized that allow organizing information support.

Keywords: decision support, innovative development, simulation model, regional management

References

1. Palyukh B. V., Kakatunova T. V. Nchetkaya kognitivnaya karta kak instrument modelirovaniya innovatsionnoy deyatelnosti na regional'nom urovne, *Programmnyye produkty i sistemy*, 2012, no. 4, pp. 128—131 (in Russian).
2. Zakharova A. A. Nekotoryye aspekty razrabotki informatsionnoy sistemy podderzhki prinyatiya strategicheskikh resheniy ob inno-

vatsionnom razvitii regiona, *Sovremennyye naukoymkiye tekhnologii*, 2007, no. 9, pp. 27—29 (in Russian).

3. **Antamoshkin A. N., Antamoshkina O. I., Khodos D. V.** Modelirovaniye osnovnykh etapov formirovaniya programmy innovatsionnogo razvitiya, *Vestnik Sibirskogo gosudarstvennogo aerokosmicheskogo universiteta imeni akademika M. F. Reshetneva*, 2010, no. 4 (30), pp. 204—206 (in Russian).

4. **Nizamutdinov M. M., Oreshnikov V. V.** Instrumentariy prognozirovaniya izmeneniya parametrov regional'nogo razvitiya na osnove adaptivno-imitatsionnogo podkhoda, *Trudy 3-ey Mezhdunarodnoy konferentsii "Informatsionnyye tekhnologii i intellektual'naya podderzhka prinyatiya resheniy — ITIDS'2015"*, Ufa: UGATU, 2015, pp. 211—215 (in Russian).

5. **Fattahov R. V., Ivanova E. I., Smetanina O. N.** O roli informatsionnykh resursov pri podderzhke prinjatija upravlencheskikh resheniy na regional'nom urovne (Role of information resources in decision making support at the regional level), *Vestnik Ufimskogo gosudarstvennogo aviacionnogo tekhnicheskogo universiteta*, 2007, vol. 9, no. 2, pp. 82—87 (in Russian).

6. **Stroev P. V.** Analiz faktorov okazyvajushhih vlijanie na innovatsionnuju aktivnost' v jekonomike Rossii (Analysis of the factors influencing the innovation activity in the Russian economy), *Vestnik Finansovogo universiteta*, 2012, no. 2 (68), pp. 143—148 (in Russian).

7. **Gazhva V. O., Gubanova E. V.** Innovatsionnoe razvitie Kaluzhskogo regiona (Innovative development of the Kaluga region), *Materialy mezhdunarodnogo nauchno-prakticheskogo "kruglogo stola" 'Napravlenija social'no-jekonomicheskogo razvitiya regional'noj jekonomiki'*, 2016, pp. 122—125 (in Russian).

8. **Tjushnjakov V. N., Zhertovskaja E. V., Jakimenko M. V.** Informacionno-analiticheskoe obespechenie situacionnogo centra kak osnova razrabotki strategij innovatsionnogo razvitiya regiona (Information and analytical support of the situational center as a basis for realize region's innovative development strategies), *Fundamental'nye issledovaniya*, 2015, no. 11-6, pp. 1253—1257 (in Russian).

9. **Zaharova A. A.** Integral'naja ocenka innovatsionnogo razvitiya regiona na osnove nechetkikh mnozhestv (Integral assessment of region's innovative development on the basis of fuzzy sets), *Nauchnoe obozrenie. Tekhnicheskie nauki*, 2014, no. 1, pp. 161—168 (in Russian).

УДК 519.683, 519.856

В. В. Федосов, канд. техн. наук, доц., e-mail: vlr.fdsu@gmail.com, Москва,

А. В. Федосова, канд. физ.-мат. наук, доц., e-mail: afedosova@unal.edu.co

Национальный университет, Богота, Колумбия (Universidad Nacional de Colombia)

Многопараметрическая оптимизация в конфликтных задачах промышленной экологии

Рассмотрена задача конфликта избыточных выбросов группы источников с нормативами загрязнения зон области. Критерием оптимизации определен численный интеграл многоэкстремальной функции превышения выбросов источников над нормативами области. Оптимизацию вели по четырем параметрам, три из которых принадлежат параболическим функциям, моделирующим выбросы источников. В качестве алгоритма применена процедура Нелдера—Мида. Информация для расчета стартового симплекса включала 2400 элементов. Численным экспериментом показана устойчивая сходимость к нулю критерия оптимизации, что соответствует разрешению конфликта. На каждой итерации фиксировались параметры оптимизации, а также результаты их взаимодействия в 2D—3D-графике. Установлена тенденция снижения концентрации избыточных выбросов вплоть до покидания источниками контролируемой области. Обсуждены результаты многопараметрической оптимизации.

Ключевые слова: загрязнение области, выбросы источников, нормативы загрязняющей нагрузки, многопараметрическая оптимизация, алгоритм Нелдера—Мида, многоэкстремальные функции, трансграничный перенос загрязнений

Введение

Целью многих задач промышленной экологии является разрешение или понижение степени конфликта между выбросами источников и допустимыми нормативами загрязнения области [1, 2]. Карту загрязнения области определяет общая функция выбросов источников, которая зависит от их числа, мощностей, параметров разброса, координат размещения и т. д. Карту нормативов области формируют количество, конфигурации и уровни нормативов зон [3].

Такие задачи формулируют как полубесконечные, так как исходное число точек области бесконечно [2, 4, 5] (*SIP*, *Semi-Infinite Programming*).

Известны эффективные алгоритмы, обеспечивающие качественные решения, при которых нормативы загрязнения выполняются для каждой точки области. *SIP* является сложным алгоритмом и в версии внешних аппроксимаций состоит из ряда процедур: стохастического поиска обычных точек невыполнения ограничений, поиска критических точек невыполнения максимизацией функций ограничения, нелинейного программирования матрицы критических точек [6, 7]. Этот алгоритм обычно реализует монопараметрическую задачу (например, ограничение только мощностей источников), что приводит к достаточно кардинальным (жестким) решениям, так как иные параметры влияния из оптимизации исключены [3]. Экономически такие решения довольно затратны, так как значительно

понижают промышленный потенциал области. Работа с такими алгоритмами требует достаточной квалификации, а авторское программное обеспечение не всегда доступно.

Когда в системе действует ряд параметров, его пытаются упорядочить, выделить главный параметр и по нему вести оптимизацию. Такие выделения и особенно ранжирование могут быть ошибочными, так как ведутся на основе ограниченного опыта вероятных оценок, интуиции, литературных рекомендаций, возможно не переносимых на данный случай. При альтернативной многопараметрической оптимизации алгоритм "видит" функцию критерия оптимизации сразу по всем параметрам ряда, автоматически их ранжирует и корректирует в сторону наилучших сочетаний.

Оптимизация множества параметров necessarily связана со сложными алгоритмами. Если необходимо выявить тенденции влияния параметров быстро, более простыми средствами и в многопараметрическом исполнении, то достаточными могут стать известные алгоритмы прямого действия. Просмотрев ленту итераций, можно оценить удачность выбранного старта, понять траектории трендов параметров, выбрать наилучшие сочетания и даже заложить их в последующие проектные решения.

В работе осуществлена многопараметрическая оптимизация для задачи понижения загрязняющей нагрузки области. Применен неградиентный алгоритм Нелдера—Мида (Н—М), организующий движение симплекса в пространстве многих параметров. Программирование и вывод графики вели в *MATLAB*.

Постановка задачи и алгоритм

Условная двумерная область (территория) раз-
мерами $\mathbf{d}$ разделена на Ω зон с границами произвольной конфигурации. В зонах действуют разные нормативы загрязнения $\mathbf{Norm}$.

На территории размещены N источников выбросов точечного типа. Характер выбросов: максимум в точке размещения источника и понижение до нуля в границах пятна разброса загрязнения.

Для упрощения разбросы загрязнений источниками описаны функциями параболического типа, хотя пригодны многие иные (например, Гаусса), включая не симметричные и не одинаковые для множества источников. Собственно моделирование разброса частиц загрязнений относят к особо сложным проблемам. Они находятся на стыках многих дисциплин и наряду с математикой требуют привлечения данных мониторинга или прогноза. Исследователь должен постоянно контролировать и допускать адекватность своих моделей.

В любую точку s с координатами $[X, Y]$ территории от каждого источника поступает $\mathbf{h}_{s,j}$ загрязнения (у.е.):

$$\mathbf{h}_{s,j} = -\mathbf{a}_j \mathbf{R}_{s,j}^2 + \mathbf{H}_j,$$

где j — номер источника в группе; $\mathbf{H}$ — вектор мощностей источников; $\mathbf{a}$ — вектор коэффициентов парабол разброса источников; $\mathbf{R}$ — матрица удалений источников от точки территории:

$$\mathbf{R} = \sqrt{(\mathbf{tN}_{1,j} - X)^2 + (\mathbf{tN}_{2,j} - Y)^2},$$

где $\mathbf{tN}$ — матрица координат размещения источников.

Источник j считаем активным для точки s , если для предельного радиуса разброса источником $\mathbf{R}_{\max_j}$

выполняется неравенство $\mathbf{R}_{s,j} < \mathbf{R}_{\max_j} = \sqrt{\mathbf{H}_j / \mathbf{a}_j}$.

Функцию общего загрязнения области $F(\mathbf{H}, \mathbf{tN}, \mathbf{a})$ образует суммирование локальных функций разброса источников. При достаточно разреженном размещении источников выбросов $F(\sim)$ многоэкстремальна.

Если понизить $F(\sim)$ за счет вектора допустимых нормативов зон области $\mathbf{Norm}$, получим функцию превышения (нарушения) нормативов $G(\mathbf{H}, \mathbf{tN}, \mathbf{a}, \mathbf{Norm})$. Обычно $G(\sim)$ сохраняет многоэкстремальный тип функции $F(\sim)$. Аналитическое представление $G(\sim)$ недоступно, однако после объявления дискретности сетки области для выявления $G(\sim)$ применимы численные методы.

Поскольку при любых изменениях параметров разброса вид функции превышения $G(\sim)$ уже иной, критерием оптимизации принято численное значение интеграла функции $G(\sim)$ на $\mathbf{d}$:

$$G(\sim) \rightarrow \min \left(\sum_{s=1}^{d_1 d_2} \sum_{j=1}^N (-\mathbf{a}_j \mathbf{R}_{s,j}^2 + \mathbf{H}_j) - \mathbf{Norm}(\omega_s) \right), \quad (1)$$

где $\mathbf{Norm}(\omega_s)$ — норматив загрязнения в точке s , принадлежащей зоне ω области.

В практике многомерной нелинейной оптимизации получил распространение симплексный алгоритм Нелдера—Мида [8], развивающий более раннюю идею Ф. Химсворта. Считается достаточно экономичным при поиске лучших направлений по сравнению с другими планами. Направление движения симплекса определяют только соотношения значений целевой функции в его вершинах, а не их абсолютные значения. В ходе оптимизации легко вводятся дополнительные или исключаются утратившие роль параметры. Алгоритм применим в случаях дрейфа характеристик объекта, работает на негладких, зашумленных и даже разрывных функциях. Находит ближайший локальный экстремум, что предположительно допустимо в данном случае. Некоторые недостатки алгоритма: отсутствие тео-

рии сходимости (самый серьезный), желательность корректировки шагов.

Число источников выбросов в задаче может быть значительным, что повышает мерность симплекса до $N + 1$, а таблицу расчета координат вершин до $(N + 1)N$, причем отдельно для каждого из m параметров оптимизации. Входная размерность элементов информации симплекса достигает $m(N + 1)N$. Проектирование старта и движения симплекса вели в собственном программном модуле. Это обеспечило контроль всех этапов прохождения оптимизации, графический вывод результатов, а также управление смещениями параметров для разных источников выбросов.

Симплекс рассчитывали по стандартной таблице коэффициентов смещений параметров, включенных в оптимизацию.

Несмотря на исходное размещение источников в области **d**, в модели не исключен трансграничный перенос загрязнений. Он происходит в полосе шириной $2R_j$, центральная линия которой совпадает с границами **d** и индивидуальна для каждого источника. При пересечении любой из границ трансграничный перенос источника равен 50 %. После полного пересечения собственной полосы $2R_j$ выбросы источника j уже не достигают области **d** (источник теряет активность), а его достигнутые параметры "замораживаются". Все это отслеживается алгоритмом автоматически.

Суммарное загрязнение области не зависит от расположения источников, только если удаление источников от четырех границ области **d** составляет не менее R_j . При удалениях меньших R_j начинается трансграничный сброс загрязнений и функции $F(\sim)$, $G(\sim)$ и интеграл $G(\sim)$ уже зависят от позиционирования.

Предположительная логика понижения конфликта (критерия оптимизации) за счет трендов параметров следующая: понижение мощностей источников выбросов; ослабление (повышение) защитных нормативов зон области; переход к более равномерному распределению источников по области; вытеснение источников сначала в полосу трансграничного переноса, а далее до полного исключения досягаемости выбросами области **d**. Все эти тренды снижают загрязняющую нагрузку области.

Сила и направление действия эффектов взаимодействия параметров непредсказуемы. Алгоритм задействует сразу все включенные в симплекс параметры оптимизации, поэтому решения по каждому из них должны быть менее жесткими.

Прекратить оптимизацию можно разными способами: а) безусловно, т. е. доведя до нуля значение превышения (целевой функции); б) формулируя условия, например, ограничив покидание области источниками: любыми, некоторыми, по сумме мощностей любых источников; по снижению мощности

определенного источника; по суммарной мощности выборки источников; по возрастанию норматива определенной зоны, выборки зон или всех зон области. Естественно, разрешены многие варианты совместных условий, причем в логике И, ИЛИ. Взамен прекращения допустима остановка по выполнению заданного для параметра условия, после чего параметр просто исключают из симплекса, переводя в статус фиксированных.

В работе многопараметрическая оптимизация проведена в варианте без ограничений.

Численный эксперимент

В табл. 1 приведены параметры источников ($N = 24$) выбросов загрязнений на ограниченной территории **d** с сеткой [800, 400]. План территории содержит $\Omega = 6$ зон с разными нормативами загрязнения **Norm**. Эта база достаточна для расчетов целевой функции (1). В расчетах стартового симплекса выбраны интервалы варьирования: $\Delta x = 100$, $\Delta y = 50$, $\Delta H = 300$, $\Delta Norm = 400$. Общее число элементов при формировании симплекса составило $4 \times 25 \times 24 = 2400$.

На рис. 1 (см. вторую сторону обложки) показана карта начального загрязнения области по параметрам табл. 1 для вектора уровней 1020—4830 ед., что для диапазона нормативов зон 1250—2088 означает значительную избыточность загрязнения

Таблица 1
Начальные параметры источников выбросов загрязнений и нормативы зон области

Источники	Параметры				
	H	tN		a	Norm
1	1515	102	365	0,0621	1775
2	1254	224	219	0,0201	1875
3	1111	127	388	0,0652	1875
4	1719	640	58	0,0402	1425
5	953	634	384	0,0626	2088
6	1288	679	374	0,0169	2088
7	1321	595	157	0,0544	2088
8	1287	565	14	0,0239	1425
9	746	79	330	0,0174	1775
10	1344	760	15	0,0315	1250
11	977	613	318	0,0348	1875
12	617	357	259	0,0405	1875
13	1364	222	272	0,0542	1875
14	1287	96	200	0,0235	1875
15	1722	469	90	0,0327	1425
16	1424	405	280	0,0283	1875
17	1624	438	56	0,0649	1425
18	563	673	102	0,0284	1425
19	1514	743	141	0,0277	2088
20	631	493	190	0,0281	1425
21	853	469	220	0,0582	1425
22	1662	606	302	0,0299	1875
23	894	62	23	0,0445	1600
24	1109	747	53	0,0555	1250

области. Функция загрязнения имеет не менее двух максимумов. Уровень загрязнений 4830 охватывает не менее трети площади области. Наименее загрязненной выглядит только юго-западная часть области, однако объявить ее, соответствующей нормативам, преждевременно.

На рис. 2 (см. вторую сторону обложки) показан в цвете план нормативных зон, уровни нормативов в зонах и размещение источников. Концентрация источников в зонах области достаточно различна (от 0 до 8), что с учетом параметра **a** не исключает взаимной досягаемости зон. Наибольшее число источников сосредоточено в зонах 2, 5, что коррелируется с расположением максимумов загрязнения на рис. 1.

В отличие от взаимного сброса загрязнений между зонами, трансграничный перенос означает сброс источниками части выбросов вне границ области **d**. Параметры **H**, **tN** и **a** по табл. 1 формируют заметные объемы трансграничного переноса уже в начальном плане размещения источников:

- */ север (9, 1, 3, 13, 2, 16, 11, 22, 5, 6);
- */ восток (6, 22, 19, 24, 10);
- */ юг (19, 18, 24, 10, 4, 8, 15, 17, 23);
- */ запад (23, 14, 9, 1).

Только источники 7, 12, 20 и 21 не имеют трансграничных выбросов в начальном плане.

На рис. 3 (см. вторую сторону обложки) показана начальная функция превышения загрязнения территории над нормативами зон. Превышения явно значительные, в ряде мест, возможно, кратные. Наибольшие, с уровнями более 4000 ед., превышения в правом нижнем квадранте сформированы группой мощных выбросов источников 4, 7, 8, 10, 24, 15, 17. Возвышения также имеют место на северо-западе и северо-востоке. Юго-запад имеет меньшие превышения.

Начальная функция превышения имеет сложный многоэкстремальный вид. Целевая функция задачи (1) соответствует численному интегралу функции превышения. Этот интеграл вычисляли во всех вершинах стартового симплекса и последовательным отражением наихудших (максимальное значение) вершин продвигались в пространстве параметров оптимизации.

Можно видеть определенную корреляцию функции превышения (рис. 3) с функцией загрязнения

области (см. рис. 1). Понятно, что интегральное по области представление у функции загрязнения будет выше.

Интеграл функции превышения на рис. 3 принимаем за точку отсчета 100 %, от которой на каждой итерации будут появляться новые функции превышения, по интегральной оценке которых определяем эффективность движения симплекса.

Результаты оптимизации

Оптимизацию провели в составе 301 итерации.

В табл. 2 приведен укрупненный протокол движения симплекса. Протокол демонстрирует достаточно уверенное убывание целевой функции (млн ед.). Итерации с нарушением убывания означают повороты симплекса.

На каждой итерации фиксировали критерий превышения, параметры оптимизации, а также графику в составе плана размещения источников, карты загрязнения области и 3D вида функции нарушения нормативов загрязнения области.

Устойчивое понижение критерия оптимизации (1) на итерациях в диапазоне 100...9,9 % означает его унимодальную связь с параметрами, несмотря на то, что сама общая функция превышения на итерациях выражено многоэкстремальная. Для такого случая применение алгоритма локальной оптимизации Н-М оправдано.

Детально рассмотрены результаты итераций 142 (лучшее значение критерия превышения в первой половине серии) и 300 (лучшее значение критерия превышения во второй половине серии). На итерации 142 критерий превышения опустился примерно до 48,3 %, а на итерации 300 — примерно до 9,9 % от начального значения.

На рис. 2 (см. вторую сторону обложки), 4, 5 видим перестановку источников выбросов и значительное возращание трансграничного переноса.

На рис. 3, 8, 9 (см. вторую, третью и четвертую сторону обложки, соответственно) наблюдаем последовательное "размывание" функции превышения.

На рис. 1 (см. вторую сторону обложки), 6, 7 (см. третью сторону обложки) видим последовательное снижение загрязняющей нагрузки области.

Все изменения сопровождаются корректировкой нормативов загрязнения зон области.

Таблица 2

Протоколы движения симплекса по критерию оптимизации (1)

Итерации	0	1	11	21	31	41	51	61	71
G(H, tN, Norm)	729,8	683,6	667,4	603,6	597,6	581,5	572,9	537,1	496,8
Итерации	81	91	101	111	121	131	141	142	151
G(H, tN, Norm)	476,1	483,4	489,9	472,7	394,7	393,8	357,0	352,9	360,9
Итерации	161	171	181	191	201	211	221	231	241
G(H, tN, Norm)	355,6	331,6	280,7	268,8	257,4	240,7	214,9	178,4	157,5
Итерации	251	261	271	281	291	300	301		
G(H, tN, Norm)	145,9	147,5	121,3	94,4	93,6	68,9	72,3		

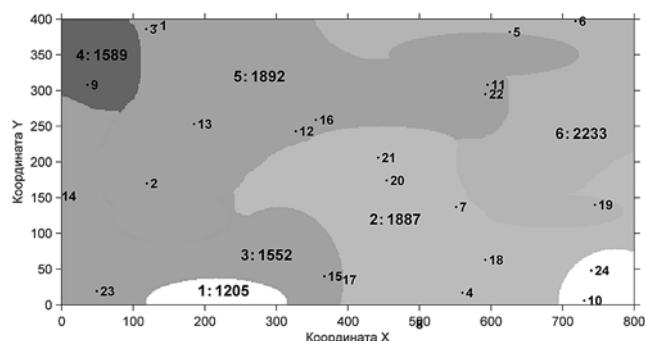


Рис. 4. План размещения источников выбросов и нормативы зон на итерации 142

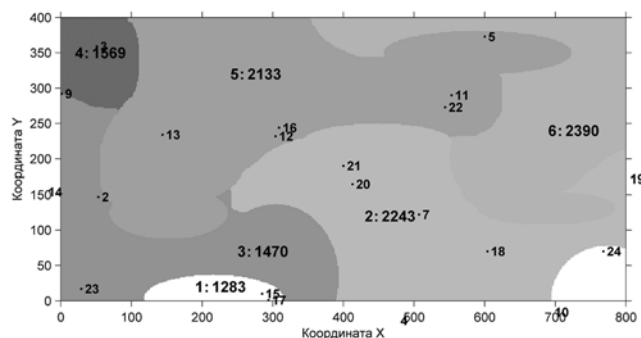


Рис. 5. План размещения источников выбросов и нормативы зон на итерации 300

Исходно наиболее нагруженными по загрязнению зонами были 5, 2, 6 (см. рис. 2 на второй стороне обложки). Алгоритм предлагает новые планы размещения источников, понижая их концентрацию в проблемных зонах (см. рис. 4, 5). Одновременно заметно снижение требований безопасности в этих зонах.

Карты загрязнения области на итерациях показывают последовательное ослабление загрязняющей нагрузки (см. рис. 1, 6, 7 на обложке). В первую очередь видны усилия симплекса подавить два начальных возвышения (см. рис. 1 на второй стороне обложки). Действия сопровождаются выравниванием загрязнений области до уровней их соответствия скорректированным нормативам зон.

Сложность многоэкстремальных функций нарушения с ростом числа итераций сохраняется (см. рис. 3, 8, 9 на второй, третьей и четвертой стороне обложки, соответственно), но интегралы функций, судя по размерности шкалы значений загрязнений (Excess), заметно понижаются.

Если рис. 3 (см. вторую сторону обложки) уже фиксировал начальные трансграничные выбросы источников, то рис. 8 (см. третью сторону обложки) показывает их возрастание. Это хорошо видно на южной границе области: приближение совместного пика (источники 15, 17, см. рис. 4) и пика источника 10 (см. рис. 4) со стороны внутренней границы области (рис. 8) и последующее удаление

их от внешней границы (рис. 5, см. третью сторону обложки).

Переход к трансграничной схеме наблюдали для всех направлений. Если на итерации 142 (см. рис. 4) область покинули только два источника (8 (юг) и 14 (запад)), то на итерации 300 (см. рис. 5) — уже шесть: 1, 6 (север), 4, 8 (юг), 14 (запад), 19 (восток). Источник 14, первоначально расположенный в зоне 5, перед выводом из области **d** пересек всю зону 3. Для источника 19 [801, 172] итерация 300 зафиксировала момент покидания области. Перемещения источников подтверждают активность параметров размещения (координат), необходимость их ввода в симплекс.

Возрастание трансграничных выбросов загрязнений приводит к понижению интеграла превышения.

На рис. 10 (см. четвертую сторону обложки) показаны тренды мощностей источников выбросов по трем сопоставляемым итерациям (0—142—300). Предполагалось сокращение мощностей и стремление не ослаблять начальные нормативы зон области. Однако к итерации 300 сокращение зафиксировано всего у семи источников, у пяти источников мощности почти не изменились, а у 12 источников заметно возросли. Интерпретировать это уже сложно, так как параметры влияют не только напрямую, но также в смешанных взаимодействиях. Можно предположить, например, удвоение мощности источника 5 тем, что он остался почти один на северо-востоке, причем вблизи 6 с самым ослабленным нормативом. То же можно отнести к источнику 23 на юго-западе. Источник 19 заметно увеличил свою мощность, но за счет пересечения восточной границы области реально потерял половину своего выброса. Подобное произошло с источником 9 на южной границе. Вместе с тем почти двухкратное падение мощности источника 15 возможно произошло вследствие перемещения в зону 1 с наиболее жестким нормативом загрязнения.

Если в табл. 1 начальных параметров источников им соответствовали зональные нормативы (столбец 5), то при переходе к оптимальным нормативам выполняли обратную свертку локальных нормативов до $\text{Norm}(\omega)$:

$$\text{Norm}_{it, \omega} = \max(\text{Norm1}_{it, \omega}, \text{Norm2}_{it, \omega}, \text{Norm3}_{it, \omega}, \dots, \text{Normk}_{it, \omega}),$$

где k — число источников, попавших в зону ω , на итерации it со своими вариациями нормативов.

На рис. 11 (см. четвертую сторону обложки) показаны тренды защитных нормативов зон области по сопоставляемым итерациям (0—142—300). Четыре зоны ослабили нормативы, а две (3, 4) — немного усилили. По средним оценкам ($\text{Norm}_0 = 1669$, $\text{Norm}_{142} = 1726$, $\text{Norm}_{300} = 1848$) на итерациях нормативы последовательно понижаются, что ведет к ослаблению конфликта.

Объяснение усиления нормативов в двух других зонах может быть лишь предположительным. Возможность трендов нормативов в обоих направлениях заложена в алгоритм оптимизации. Обе зоны исходно по числу источников мало нагружены. В зоне 3 вначале располагался только источник 23, причем с частично трансграничным сбросом загрязнения. На 1, 6, 7 и рис. 3, 8, 9 (см. обложку) видно, что в юго-западной части области превышений почти нет. К итерации 300 источник 23 еще больше увеличил трансграничный сброс. Правда, в зоне добавилось примерно половина выбросов источника 2 и немного от трансграничных сбросов источников 15, 17, которые почти вдвое сократили свои мощности. В этих условиях алгоритм принял решение понизить (ужесточить) норматив зоны 3.

В зоне 4 исходно присутствовали источники 1, 9 с суммарной мощностью около 2250 ед. К итерации 300 оба источника покинули зону, но появился источник 3 с мощностью около 1600. Кроме того, присутствовало примерно треть трансграничного сброса источника 9 (около 1500 ед.). Решение алгоритма немного понизить норматив зоны выглядит оправданным.

Можно предположить, что алгоритм стремится не только к ослаблению нормативов зон, но также к выравниванию вектора нормативов зон. Однако такой вывод на примере одной задачи пока не имеет достаточного объема подтверждающих данных.

Планы размещения источников на завершающих итерациях показывают стремление алгоритма к более равномерной плотности размещения источников, что, видимо, позволяет избежать кардинальных изменений нормативов зон.

К итерации 300 на подавляющей площади области (~90 %) нормативы выполнены. Ясно, что продолжение итераций доведет выполнение до 100 % площади.

Если тенденции мощностей источников и нормативов зон, находящихся в прямом конфликте, ясны: $H \downarrow$ и $Norm \uparrow$, то по параметру размещения источников просматривается тенденция их вытеснения из контролируемой нормативной области и усиление доли трансграничного переноса. Приемлемость таких схем разрешается уже в компетенции проектировщиков.

В целом поведение симплекса можно характеризовать как достаточно рациональное. Действительно, ему удалось разместить в области более высокий промышленный потенциал ($H_0 = 28\,779$, $H_{142} = 37\,080$, $H_{300} = 33\,466$), в среднем почти сохранить нормативы зон (экологический статус области) и провести не всегда понятные, но довольно активные перестановки (включая вывод) источников выбросов. Общий эффект интегрального понижения функции нарушения нормативов загрязнения области доведен до 9,9 % от исходного и без сомнения может быть продолжен до 0.

Никаким самостоятельным варьированием параметров или случайным поиском достижение хотя бы близких результатов в такой задаче будет нереальным.

Включение параметра нормативов показывает возможности алгоритма в разрешении достаточно жесткого конфликта. Если исходить из приоритета экологии, то нормативы следует фиксировать, а оптимизации подвергать только мощности и размещение источников.

На основе данного подхода сложность модели может быть значительно повышена за счет расширения перечня параметров оптимизации, причем без принципиальных ограничений. Иногда напротив интересна более узкая постановка. Например, зафиксировав стартовые координаты источников, выяснить конфликт мощностей источников только с нормативами зон, или зафиксировать мощности источников и нормативы зон, а оптимизации подвергнуть только координаты размещения источников.

Дополнительными достоинствами многопараметрической оптимизации являются:

- учет меняющихся на итерациях трансграничных переносов загрязнений;
- назначение "неправильных" границ нормативным зонам области;
- возможность замены вектора нормативов зон области функцией нормативов области;
- оптимизация до полного прекращения сбросов загрязнения отдельными источниками на d .

Сравнительное сопоставление ролей параметров оптимизации в погашении конфликта показало большую активность изменения координат источников. Алгоритм за счет активного вытеснения источников из d фактически размывает (увеличивает размеры) первоначальной области. Это делает возможной постановку задачи, при которой трансграничный перенос в модели отменяется, а размеры d на итерациях уже не ограничивают.

При более сложном представлении функций описания выбросов источников ряд параметров оптимизации, включенных в симплекс, может быть соответственно расширен.

Многопараметрическая оптимизация представляет собой интересный и мощный инструмент для проектирования новых или реконструкции сложившихся территорий со сложной инфраструктурой. В работе удалось показать последовательное ослабление конфликта одновременно по четырем параметрам модели.

Результаты моделирования конфликта позволяют получить разнообразные производные технико-экономические оценки совместных трендов параметров. Разработанные модули ПО можно встраивать в более мощные системы проектирования областей с насыщенной инфраструктурой.

Заключение

Показаны возможности многопараметрической оптимизации в задаче конфликта выбросов источников с защитными нормативами загрязнения области. Критерий оптимизации (уровень конфликта) определен численным интегралом многоэкстремальной функции превышения.

При удачном проектировании симплекса в составе четырех параметров (алгоритм Нелдера—Мида) критерий оптимизации уверенно сходится к нулю, что определяет эту функцию как унимодальную.

Содержание протоколов итераций наглядно отражает одновременные тренды параметров оптимизации, позволяет выбрать наилучшие или приемлемые сочетания и в целом подчиняется логике модели.

Программный комплекс позволяет помимо табличных данных наблюдать результаты итераций в 2D—3D-графике.

Список литературы

1. Замай С. С., Якубайлик О. Э. Модели оценки и прогноза загрязнения атмосферы промышленными выбросами в информационно-аналитической системе природоохранных служб крупного города: учеб. пособие. Красноярск: Краснояр. гос. ун-т, 1998. 109 с.
2. Fang F., Zhang T., Pavlidis D., Pain C. C., Buchan A. G., Navon I. M. Reduced order modelling of an unstructured mesh air pollution model and application in 2D/3D urban street canyons // ELSEVIER. Atmospheric Environment. 2014. Vol. 96. P. 96—106.
3. Федосов В. В., Федосова А. В. Оптимизация в системе групповых выбросов-заборов загрязнений для производственной площадки (территории) // Программная инженерия. 2014. № 4. С. 19—25.
4. Stein O. How to solve a semi-infinite optimization problem // European Journal of Operations Research, 2012. Vol. 223. P. 312—320.
5. Bhattacharjee B., Lemonidis P., Green W. H., Barton P. I. Global solution of semi-infinite programs // Mathematical Programming. 2005. Vol. 103. P. 283—307.
6. Volkov Y. V., Zavriev S. K. A General Stochastic Outer Approximation Methods // SIAM Journal on Control and Optimization. 1997. Vol. 35. P. 1387—1421.
7. Завриев С. К., Новикова Н. М., Федосова А. В. Стохастический алгоритм решения выпуклых задач полубесконечной оптимизации с ограничениями равенствами и неравенствами // Вестник Московского университета. Сер. 15. Вычислит. матем. и кибернетика. 2000. № 4. С. 30—35.
8. Nelder J. A., Mead R. A simplex method for function minimization // The Computer Journal, 1965. Vol. 7, N. 4. Oxford: Oxford University Press.

V. V. Fedosov, Moscow, Russia, e-mail: vlr.fdsv@gmail.com,

A. V. Fedosova, Universidad Nacional de Colombia, Bogotá, Colombia

Multiparameter Optimization Problems in Conflict Industrial Ecology

The problem of the conflict of excess emissions of sources from zones of pollution regulations in the area. The criterion adopted to optimize the numerical integral function multiextremal exceeded total emissions sources on regulations field zones. Optimization conducted simultaneously on four parameters, three of which belong to the parabolic functions, modeling the emissions sources. The Nelder—Mead procedure is applied as an algorithm. Information for the calculation of the starting simplex consisted of 2400 elements. The numerical experiments show a steady convergence to zero optimization criterion, which corresponds to the resolution of the conflict. At each iteration, the optimization parameters were recorded, and the results of their interaction in the 2D—3D graphics. The tendency of displacement of excess emissions power beyond the controlled area, which means the increase in the volume of cross-border transport of pollution. Derivatives discussed technical and economic evaluation of multiparameter optimization.

Keywords: contamination area sources emissions of polluting loads, multiparametric optimization, Nelder—Mead algorithm, multiextremal function, transboundary pollution

References

1. Zamaj S. S., Jakubajlik O. Je. Modeli ocenki i prognoza zagrnaznenija atmosfery promyshlennymi vybrosami v informacionno-analiticheskoy sisteme prirodoohrannyh sluzhb krupnogo goroda: Ucheb. posobie, Krasnojarsk, Krasnojar. gos. un-t. 1998. 109 p. (in Russian).
2. Fang F., Zhang T., Pavlidis D., Pain C. C., Buchan A. G., Navon I. M. Re-duced order modelling of an unstructured mesh air pollution model and application in 2D/3D urban street canyons, *ELSEVIER, Atmospheric Environment*, 2014, vol. 96, pp. 96—106.
3. Fedosov V. V., Fedosova A. V. Optimizacija v sisteme grup-povyh vybro-sov-zaborov zagrnaznenij dlja proizvodstvennoj plosh-hadki (territorii), *Programmnaja inzhenerija*, 2014, no. 4, pp. 19—25 (in Russian).
4. Stein O. How to solve a semi-infinite optimization problem, *Eu-ropean Journal of Operational Research*, 2012, vol. 223, pp. 312—320.
5. Bhattacharjee B., Lemonidis P., Green W. H., Barton P. I. Global solution of semi-infinite programs, *Mathematical Program-ming*, 2005, vol. 103, pp. 283—307.
6. Volkov Y. V., Zavriev S. K. A General Stochastic Outer Ap-proximation Methods, *SIAM Journal on Control and Optimization*, 1997, vol. 35, pp. 1387—1421.
7. Zavriev S. K., Novikova N. M., Fedosova A. V. Stohastichesk-ij algoritm reshenija vypuklyh zadach polubeskonechnoj optimizacii s ograničenijami ravenstvami i neravenstvami, *Vestnik Moskovskogo Universiteta. Ser. 15. Vychislit. matem. i kibernetika*, 2000, no. 4, pp. 30—35 (in Russian).
8. Nelder J. A., Mead R. A simplex method for function mini-mization, *The Computer Journal*, 1965, vol. 7, no. 4. Oxford: Oxford University Press, pp. 441—461.

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ CAD-SYSTEMS

УДК 621.3.049.771.14

А. Л. Стемповский, академик РАН, д-р техн. наук, проф., директор,
С. В. Гаврилов, д-р техн. наук, проф, зав. отд., sergey_g@ippm.ru,
Институт проблем проектирования в микроэлектронике РАН, Москва
С. И. Гуров, канд. физ.-мат. наук, доц., sgur@cs.msu.ru,
Московский государственный университет имени М. В. Ломоносова,
Д. И. Рыжова, канд. техн. наук, мл. науч. сотр., ryzhova_d@ippm.ru,
Институт проблем проектирования в микроэлектронике РАН, Москва

Методы проектирования помехозащищенных комбинационных КМОП-схем, обеспечивающие автоматическое исправление ошибок

Работа посвящена основным методам проектирования комбинационных КМОП-схем, обеспечивающим автоматическое исправление сбоев. Рассмотрена специфика проектирования самокорректируемых схем и проанализированы подходы к автоматической коррекции ошибок на основе методов избыточного кодирования. Также рассмотрены важные особенности задачи исправления ошибок схем, вызванных сбоями. В заключительной части приведено краткое описание перспективного подхода к исправлению ошибок с использованием обратимой схемотехники.

Ключевые слова: комбинационные схемы, проектирование, помехозащищенность, САПР

Введение

Повышение надежности функционирования интегральных микросхем (ИМС) остается актуальной проблемой синтеза. Важной стороной этой проблемы является устойчивость схем к кратковременным самоустраняемым сбоям. Считается, что интенсивность отказов (потери способности устройства выполнить требуемую функцию). Причиной сбоев является воздействие на схему различных видов помех: радиационных, скачков напряжений питания, деградаций сигналов во времени и др. Исследования показывают, что вероятность ошибки от указанных причин может варьироваться в очень широких пределах: от менее чем 0,1 до 30 % для больших и очень малых схем соответственно [1]. Поэтому сбоеустойчивость является важнейшим требованием к аппаратуре, работающей в экстремальных условиях космоса.

Обоснованными представляются предположения о следующем характере результатов сбоев комбинационных ИМС:

- одиночная неисправность того или иного элемента схемы может привести к кратной ошибке на выходе схемы, в результате чего типичной является ситуация либо полного отсутствия ошибки (маскирование), либо наличия сразу

нескольких ошибок на выходе схемы (конус ошибок);

- наиболее частыми являются ошибки константного (*stick-at faults*) типа.

В данной работе рассматриваются автоматические способы коррекции ошибок, возникающие при сбоях комбинационных схем (выходные сигналы которых в некоторый момент времени однозначно определяются входными сигналами, имеющими место в данный момент времени). Возможны следующие подходы к решению этой задачи:

- 1) использование самокорректируемых схем;
- 2) применение избыточного кодирования информации;
- 3) использование обратимой схемотехники.

1. Самокорректируемые схемы

Построение самопроверяемых схем является традиционным подходом к решению проблемы проектирования комбинационных схем, обеспечивающих автоматическое исправление ошибок [2–4]. Самокорректируемость функциональных блоков — это свойство обнаруживать и исправлять неисправности как в основной, так и в контролирующей аппаратуре (когда аппаратные затраты контролирующего устройства становятся больше затрат контролируемого устройства). Самокорректируемость обеспечивается созданием контролирующей схемы

в дополнение к основной, в этом случае говорят о схемной избыточности [5]. При возникновении ошибки контролирующая схема обеспечивает пересчет значений выходного вектора основной схемы, причем такой пересчет может повторяться неоднократно (CED, Concurrent Error Detection). Таким образом, исправление ошибки связано также с временной избыточностью в процессе обработки информации. Очевидным недостатком является слишком большое время получения результата.

Другим широко используемым методом является N -модульная избыточность, как один из типов резервирования (понятие резервирования включает в себя различные методы повышения надежности путем введения структурной избыточности). Резервирование выполняют как на уровне отдельных узлов, так и на уровне различных компонентов схемы. При этом методе основная схема дублируется N раз, и решение о том, какой результирующий вектор является правильным, определяется схемой голосования (процедура мажорирования). Одним из самых распространенных подходов является тройное модульное резервирование (*TMR, Triple Modular Redundancy*), основанное на предположении, что ошибка может возникнуть только в одном блоке из трех. Очевидно применение аппаратного резервирования приводит к проигрышу в площади схемы (при троировании — более 200 %), но практически не влияет на временные задержки.

2. Избыточное кодирование

Одним из перспективных способов автоматической коррекции ошибок являются методы избыточного кодирования [4—14]. Здесь следует говорить об информационной, или кодовой избыточности. При этом любой избыточный код может обнаружить или исправить не все ошибки, а только ошибки определенного типа, и для обнаружения ошибок требуется значительно меньшая избыточность, чем для их исправления.

Отметим общие свойства задачи. Прежде всего, как правило, считают, что исходная схема спроектирована и не подлежит изменению.

Обычно искажение информации, возникающее при сбоях, не описывается в рамках традиционной модели двоичного симметричного канала, которая используется при разработке избыточных кодов.

Эффективность и сложность применения того или иного метода избыточного кодирования существенно зависит от требуемого уровня обеспечения надежности исправления ошибки. Важным является то, что обычные оценки качества кода (кодвое расстояние, избыточность и т. д.) не являются определяющими при выборе кода для решения рассматриваемой задачи, поэтому на практике часто выбирают коды с худшими характеристиками, но позволяющими более просто выполнить декодирование и исправление выходного вектора схе-

мы. Иными словами, реализация кодирования/декодирования выбранного кода не должна существенно снижать быстродействие построенной схемы.

Известны данные по кратности ошибок для типовых узлов дискретных вычислительных устройств, которые оказываются близки к данным в целом по всему устройству [15]. Вероятность возникновения одиночной ошибки в схеме составляет 75...80 %, двойной ошибки — 15...20 %, тройной ошибки — 2...5 %, вероятность возникновения четырех и более ошибок крайне мала (менее 1 %).

Очевидно, что при использовании корректирующих кодов важно заметить и исправить ошибку именно в информационных, а не в проверочных символах, это особенно существенно для нашей задачи. Основным недостатком метода применения избыточного кодирования для синтеза схем с исправлением ошибок является увеличение избыточности в зависимости от числа исправлений ошибок в битах [7].

При разработке средств исправления ошибок схем, вызванных сбоями, следует учитывать следующие особенности задачи, из которых будут вытекать требования к свойствам выбираемого кода.

1. Проверочные биты вычисляет специальная корректирующая схема параллельно и одновременно с информационными битами, вычисляемыми основной схемой, в то время как при классическом подходе предполагается кодирование уже имеющегося сообщения.

2. Выигрыш в суммарной площади основной и корректирующей схем при описываемом подходе получается вследствие того, что число проверочных разрядов значительно меньше числа информационных, а методы оптимизации позволяют синтезировать корректирующую схему существенно меньшей площади, чем основная схема.

3. Поскольку коды с оптимальными характеристиками избыточности требуют кодирующих и декодирующих устройств повышенной сложности, при выборе кода необходимо учитывать все факторы, влияющие на сложность реализации схемы.

4. Представляется естественным строить корректирующие схемы для комбинационных схем также комбинационными, что означает ограничение на реализацию алгоритмов декодирования, заключающееся в отказе от использования последовательностных элементов, прежде всего от сдвиговых регистров.

5. Проведенные эксперименты показали, что наиболее вероятный одиночный сбой функционального элемента схемы, как правило, либо маскируется, либо приводит к инвертированию единственного разряда на выходе схемы, т. е. кратные ошибки маловероятны [8].

В некоторых случаях для обеспечения практической достоверности наличия не более, чем одной ошибки, возможно использование естественной

избыточности кода. Совокупность возможных выходных k -векторов данной комбинационной схемы существенно меньше мощности множества 2^k , наименьшее расстояние между ними велико [16]. Например, в дешифраторе должен меняться только один выходной разряд, и другое свидетельствует об ошибке.

Для исключения появления ошибки уже в процессе проверки и исправления выходного вектора корректирующую схему выполняют в специальном сбоеустойчивом варианте, обеспечивающем защиту от сбоев с помощью схемотехнических и технологических методов, поэтому этот процесс можно считать свободным от ошибок [17]. Это соответствует принципу выбора так называемого "доверенного сторожа", который предполагает, что средства контроля менее уязвимы к дестабилизирующим воздействиям, чем основные функциональные устройства [15].

Платой за такую безошибочность служит увеличение площади корректирующей схемы. Понятно, что успех будет лишь в случае, когда площадь последней менее чем в 2 раза превосходит площадь основной схемы, иначе исправление одиночной ошибки методом тройного резервирования с мажорированием выходов по правилу "2 из 3" становится более выгодным [18].

На практике обычно используют коды, исправляющие одиночную и обнаруживающие двойную ошибку (схемы SEC/DED, Single error correction/Double error detection). В схемах такого типа применяют либо модифицированные коды Хемминга, либо LDPC-коды. Заметим, что применение последних упрощает реализацию кодирования и декодирования.

При высокой степени асимметрии и множественности ошибок целесообразно использовать коды, ориентированные на коррекцию ошибок в несимметричном канале: коды Бергера, коды с постоянным весом и др. В устройствах, реализующих арифметические операции с переносом, целесообразно использовать арифметические корректирующие коды [19].

В разд. 2 рассмотрены основные линейные коды общего вида, учитывающие перечисленные выше особенности задачи. Это блочные линейные коды для исправления малого числа ошибок, прежде всего, одиночной ошибки. Нелинейные коды строят для исправления ошибок определенного типа, и их использование связано с трудоемкими исследованиями свойств конкретного синтезируемого устройства [20]. Применение же циклических кодов нецелесообразно, поскольку их эффективная реализация требует использования последовательных схем.

2.1. Линейные блочные коды. Коды с проверкой на четность

При систематическом блочном кодировании кодовые слова (n, k) -кода имеют длину $n = k + m$, где k — число информационных разрядов, повторяющих символы сообщения, а m — число проверочных разрядов. Проверочные разряды являются некоторой функцией от информационных разрядов в линейных кодах — различные линейные комбинации информационных.

Систематическим кодом с проверкой на четность называется двоичный блочный код, в котором каждому сообщению $u = (u_1, \dots, u_k)$ соответствует кодовое слово $v = (v_1, \dots, v_n)$, определяемое следующим образом:

$$v_i = \begin{cases} u_i, & 1 \leq i \leq k; \\ \sum_{j=1}^k u_j g_{i-k-1,j}, & k+1 \leq i \leq n, \end{cases} \quad (1)$$

где Σ означает сумму по модулю 2, а множество двоичных символов $\{g_{i,j}\}$, $0 \leq i \leq m-1$, $1 \leq j \leq k$, которое задает метод кодирования, фиксировано [1]. Таким образом происходит размещение символов сообщения в первых позициях кодового слова.

Общий код с проверкой на четность определяется следующим выражением:

$$v_i = \sum_{j=1}^k u_j g_{i,j}, \quad 1 \leq i \leq n,$$

которое задает кодирование общего вида, в том числе несистематическое.

Кодирование линейными кодами описывается умножением вектора сообщения (в нашей задаче это выходной вектор комбинационной схемы) на порождающую матрицу G :

$$v = uG.$$

Искажение в результате сбоя описывается вектором ошибок e :

$$w = v + e,$$

где $+$ обозначает сумму по модулю 2, w — полученный, возможно искаженный, вектор.

Декодирование линейных кодов можно проводить с помощью метода дизъюнкции сфер, синдромного и мажоритарного декодирования.

Метод дизъюнкции сфер связан с построением таблицы результатов, полученных при распространении ошибок определенного типа на все входные векторы, и требует значительных аппаратных затрат по сравнению с двумя другими, поэтому его применение для коррекции ошибок нецелесообразно [4].

Синдромное декодирование основано на вычислении вектора s синдромов (исправляющего вектора) по проверочной матрице H :

$$s = Hw,$$

порождающая и проверочная матрицы связаны соотношением $GH^T = 0$.

Если ошибок нет, то $s = 0$, т. е. синдром является нулевым вектором. В противном случае для определения неизвестного вектора ошибок необходимо применять те или иные алгоритмы декодирования линейных кодов. Для этого заранее определяются синдромы для возможных вариантов ошибок и определение наиболее вероятной позиции ошибки $\hat{e}$ проводится по полученному синдрому. Далее выполняется коррекция кодовой комбинации $\hat{u} = v + \hat{e}$, и ошибка считается исправленной (тогда и только тогда, когда $\hat{u} = u$). Очевидно, что процесс декодирования линейных кодов в общем случае остается достаточно трудоемким.

Мажоритарное декодирование (принцип голосования) базируется на системе проверочных равенств для каждого информационного бита. В качестве проверочных равенств используются линейные комбинации строк проверочной матрицы H . Любой информационный бит выражается различными независимыми способами в виде линейных комбинаций других символов. Результаты вычислений подаются на соответствующий этому символу мажоритарный элемент. Если ошибки отсутствуют, то проверочные равенства не нарушаются, и на выходе мажоритарного элемента получаем истинное значение символа. Если число проверок равно $d^{2s} + 1$, где d — кодовое расстояние, а появление ошибки кратности s и менее не приводит к нарушению более s проверок, то правильное решение может быть принято по большинству неискаженных проверок. Чтобы указанное условие выполнялось, любой другой информационный символ не должен входить более чем в одно проверочное равенство. В этом случае мы имеем дело с системой разделенных проверок.

В настоящее время известно значительное число кодов, допускающих мажоритарную схему декодирования, а также сформулированы некоторые подходы при конструировании таких кодов. Для линейных кодов, рассчитанных на исправление многократных ошибок, часто более простыми оказываются декодирующие устройства, построенные по мажоритарному принципу.

Заметим, что во всех указанных кодах длина кодовых слов определяется однозначно. Это обстоятельство накладывает существенные ограничения на число информационных разрядов k в кодовом слове. Для обхода этого ограничения переходят к усеченным кодам [21].

2.2. Коды Хэмминга

Код Хэмминга принадлежит классу SEC/SED-кодов (Single error correcting/Single error detecting) и требует минимального числа проверочных битов. Этот код имеет минимальное кодовое расстояние $d = 3$, что дает возможность определять и исправ-

лять однократную ошибку в кодовом слове, но не позволяет находить все двойные ошибки. Вектор синдрома кода Хэмминга при одиночной ошибке является двоичным представлением номера искаженного разряда.

Для того чтобы реализовать возможность детектирования двоичных ошибок, к проверочным битам кода Хэмминга добавляют еще один общий бит четности, вычисляемый из всех информационных и проверочных битов, и получают модифицированный код Хэмминга. Использование дополнительного общего бита четности позволяет увеличить минимальное кодовое расстояние до четырех. Таким образом, можно вычислить любую двукратную ошибку, кроме тех, которые затрагивают сам общий бит четности. При проектировании декодера на основе модифицированного кода Хэмминга необходимо предусмотреть ситуацию, когда в считываемом кодовом слове будет присутствовать однократная ошибка в проверочном бите четности.

2.3. Коды Хсяо

Коды Хсяо широко применяют для помехоустойчивого кодирования субмикронных КМОП оперативных запоминающих устройств. Их также называют кодами с нечетным весом столбца, и они являются подклассом кодов Хэмминга. Порождающая матрица кода Хсяо для k информационных разрядов обладает следующими свойствами:

- каждый столбец содержит нечетное число единиц;
- общее число единиц в проверочной матрице минимально;
- разница числа единиц в любых двух рядах не превышает единицы;
- никакие два столбца не повторяются.

Столбцы порождающей матрицы могут быть получены следующим образом: сначала выбирают все столбцы веса 1, затем все столбцы весов 3, 5 и т. д., пока не будут сформированы все $n = 2^m - 1$ столбцов. При этом минимизируется число единиц в проверочной матрице, что приводит к уменьшению задержек вычисления синдрома. Вместе с тем усложняется алгоритм вычисления номера искаженного разряда. При этом число функциональных элементов и связей в схемах кодера/декодера, обеспечивающих обнаружение двойных ошибок, больше, чем в схемах кодера/декодера Хэмминга, что сказывается на быстродействии итоговой схемы.

Проверочную матрицу кода Хсяо можно получить путем линейных операций над матрицей модифицированного кода Хэмминга, приведя ее к каноническому виду (с единичной матрицей в правой части проверочной матрицы). Коды Хсяо имеют минимальное кодовое расстояние $d = 4$. Это позволяет обнаруживать и исправлять однократную ошибку, а также детектировать все двойные ошибки.

Модифицированный код Хсяо имеет два дополнительных бита четности, поэтому его провероч-

ная матрица содержит меньшее число единиц. По заявлению авторов, это позволяет проектировать быстродействующие кодеры/декодеры [22].

2.4. Итеративные коды (коды-произведения)

Итеративные коды имеют минимальное кодовое расстояние $d = 3$ и позволяют определять и исправлять однократную ошибку в кодовом слове, но не позволяют обнаруживать все двойные ошибки, в частности, ошибки в проверочных битах.

Процедура построения двумерного итеративного кода состоит в следующем. Сначала заданную совокупность информационных битов делят на группы информации по t разрядов в каждой группе. Таким образом, формируются q групп. Полученные группы информации представляют в виде информационной матрицы размерностью $t \times q$. Из каждой строки и каждого столбца по методу контроля четности вычисляется проверочный бит. При декодировании в случае возникновения одиночной ошибки ее позиция в информационной матрице будет однозначно выражена через соответствующие проверочные биты.

Основной недостаток применения данного типа кодов к задаче исправления ошибок в схеме состоит в том, что итеративные коды требуют больше проверочных битов, чем коды Хэмминга или Хсяо.

2.5. Низкоплотностные коды. Коды Галлагера

Линейные коды с малой плотностью проверок на четность (*Low Density Parity-check Codes, LDPC*), предложенные Р. Галлагером, являются блоковыми линейными кодами с проверкой четности [23]. Данные коды описываются разреженными проверочными матрицами H , что уменьшает число символов, входящих в проверочные соотношения, поскольку число единиц в строках проверочной матрицы усложняет структуру декодера. Некоторые коды данного типа могут быть получены из кодов Хэмминга с помощью укорачивания.

Важной характеристикой порождающей матрицы *LDPC*-кода является отсутствие циклов определенного размера (кратности), а присутствие таких циклов свидетельствует о заложенной в структуру матрицы избыточности, не приводящей к улучшению помехоустойчивых свойств кода. Существенной положительной стороной таких кодов является то, что не только кодирование, но и декодирование их выполняется достаточно быстро. Параметры *LDPC*-кодов для случая исправления одиночной ошибки приведены в работе [24]. На сегодняшний день они считаются оптимальными для применения при синтезе помехозащищенных схем при реализации декодирования мажоритарным методом.

2.6. Спектральный код на основе функций Радемахера

Спектральный код на основе функций Радемахера представляет собой систематический код с проверками на четность, имеющий произвольное число k информационных разрядов.

Для $k = 2^q$ положим, что

$$g_{0,j} \equiv 1, g_{i,j} = \frac{1}{2} (1 - \text{Rad}(i, k, j)); 1 \leq i \leq q, 1 \leq j \leq k,$$

где $\text{Rad}(i, k, j) = \text{sgn} \sin\left(\frac{2^i \pi j}{k}\right)$ — значение i -й дискретной функции Радемахера на интервале определения $[0, k]$ в j -й точке отсчета, а $\text{sgn } x$ — это сигнум-функция. Таким образом будут получены $m = q + 1$ проверочных разрядов и образован блоковый систематический код длины $n = 2^q + q + 1$ с проверкой на четность.

Система функций Радемахера ортогональна, ортонормирована, но не полна и используется для построения функций Уолша, по которым осуществляют спектральное разложение кусочно-постоянных булевых функций [25]. В силу этого полученный код предлагается называть R -спектральным.

Ясно, что порождающая матрица G_R данного кода есть единичная матрица I_k k -го порядка с присоединенной справа транспонированной порождающей матрицей $G_{RM}(k, 1)$ кода Рида—Маллера длины 2^k первого порядка:

$$G_R = [I_k | G_{RM}^T(k, 1)].$$

Также матрица G_R может быть получена из кода Макдональда.

Покажем, что построенный код способен исправлять ошибку инвертирования некоторого разряда в исходном сообщении u . Для этого примем еще одно допущение, что одиночная ошибка может произойти только в информационных битах, т. е.

$$w' = (w_1, \dots, w_k) = u + e, e = (e_1, \dots, e_k),$$

где w' — начальная часть полученного вектора, содержащая информационные разряды; e — вектор ошибки, содержащий не более одной единицы.

Вычислим синдромы $s = (s_0, \dots, s_q)$ для вектора w' :

$$s_i = \sum_{j=1}^k w_j g_{i,j}, 0 \leq i \leq q.$$

Если $s_0 = 0$, то считаем, что ошибки не произошло. В противном случае вычисляем значения $r_i = w_{k+i} + s_i; i = 1, \dots, q$.

Нетрудно видеть, что для каждой ошибки номер искаженного разряда может быть получен дешифрированием вектора $r = (r_1, \dots, r_q)$, поскольку инвертирование какого-либо одиночного разряда в

$u = (u_1, \dots, u_k)$ приводит к одной из 2^k комбинаций вектора r , и данное отображение биективно. Это свойство есть следствие того, что совокупность векторов $(g_{1,j}, \dots, g_{q,j}), j = \overline{1, 2^q}$, совпадает со всеми двоичными наборами длины q . Ясно, что вычисление вектора ошибки построенного кода происходит по укороченной схеме: входом является синдром ошибки разрядности $n - k$, а выходом — вектор ошибки разрядности k .

Построенный спектральный линейный нециклический R -код имеет кодовое расстояние $d = 2$. Нулевой вектор и вектор, имеющий единицу только в первом и $(k + 1)$ -м разрядах, — его кодовые слова. Тем не менее R -код позволяет исправлять одиночную ошибку в информационных разрядах полученного слова, а для определения разряда, в котором произошла ошибка, может быть использована стандартная схема дешифратора.

Рассмотрим пример с $q = 3$. Значения коэффициентов $g_{i,j}, 0 \leq i \leq 3, 1 \leq j \leq 8$ представлены ниже в таблице, являющейся порождающей матрицей первого порядка длины 8.

Таблица значений $g_{i,j}$ для случая $q = 3, k = 8$

Позиции информационных разрядов, j	1	2	3	4	5	6	7	8
$g_{0,j}$	1	1	1	1	1	1	1	1
$g_{1,j}$	0	0	0	0	1	1	1	1
$g_{2,j}$	0	0	1	1	0	0	1	1
$g_{3,j}$	0	1	0	1	0	1	0	1

Значения вектора $(g_{1,j}, g_{2,j}, g_{3,j})$ есть двоичный код числа $j - 1$. Значения проверочных $(v_9, \dots, v_{12})$ разрядов для сообщения $u = (u_1, \dots, u_8)$ будут определяться следующими соотношениями:

$$\begin{aligned} v_9 &= u_1 + u_2 + u_3 + u_4 + u_5 + u_6 + u_7 + u_8; \\ v_{10} &= u_1 + u_2 + u_3 + u_4; \\ v_{11} &= u_1 + u_2 + u_5 + u_6; \\ v_{12} &= u_1 + u_3 + u_5 + u_7. \end{aligned}$$

Если ошибка произошла, например, в 5-м разряде, то $r_1 = 1, r_2 = r_3 = 0$, при ошибке в 8-м разряде будем иметь $r_1 = r_2 = r_3 = 1$ и т. д.

Ограничение $k = 2^q$ можно снять следующим образом. При $2^{q-1} < k \leq 2^q$ строим код для q , как указано выше, и отбрасываем последние $2^q - k$ значений v и коэффициентов $g_{i,j}$. В результате получим $(\lceil \log k \rceil + 1 + k, k)$ -код, исправляющий одиночную ошибку в информационных битах. На основе использованной идеи могут быть построены и коды для исправления кратных ошибок.

Можно также избавиться от последнего существенного ограничения на одиночную ошибку — предположения о ее появлении лишь в информационных битах. Для этого достаточно удвоить в кодовом слове символ v_{k+1} , лишь незначительно увеличив избыточность кода и сложность декодирования. Этого, однако, можно не делать при использовании R -кодов для задачи повышения надежности цифровых управляющих устройств методами избыточного кодирования.

R -коды во многом схожи с кодами Хэмминга. Кодирование и декодирование проходит по одной общей схеме. Асимптотически эти коды имеют одинаковую избыточность k/n . В неукороченном случае R -код уступает коду Хэмминга по избыточности на пренебрежимо малую величину, а при большом укорачивании эти коды могут иметь и одинаковую избыточность. При этом, как отмечалось, укорачивание может существенно усложнить реализацию кодирования/декодирования кода Хэмминга. Известно, что при исправлении одиночной ошибки кодом Хэмминга площадь корректирующей схемы может составлять 130 % от площади основной [26]. Можно предположить, что использование предлагаемых R -кодов позволит существенно улучшить данный результат, поскольку защиты схемы на схемотехническом уровне можно будет обеспечивать функциональные элементы корректирующей схемы, вычисляющие не все m проверочных битов, а только единственный $(k + 1)$ -й бит. При этом R -коды очевидно значительно экономнее многомерных итеративных кодов и LDPC-кодов.

Существенной особенностью R -кода с проверкой на четность является простота его реализации для произвольного числа информационных символов. Кодирование и декодирование R -кода проходит по той же схеме, что и у кода Хэмминга. Близость кода к кодам Рида—Маллера позволяет предполагать существование для них методов мажоритарного декодирования. Избыточности данных кодов при одинаковой длине практически совпадают. Структура R -кода позволяет предполагать возможность конструирования кодов, исправляющих более одной ошибки и построенных по той же идее.

3. Обратимая схемотехника

Другим перспективным подходом к коррекции ошибок в схеме является использование обратимой схемотехники. Обратимые схемы представляют собой новую технологию, основанную на подходе, существенно отличающемся от традиционного: обратимые схемы реализуют исключительно обратимые функции. Это позволяет использовать обратимость для синтеза сбоеустойчивых схем. Как и с проблемой обнаружения, диагностика неисправностей может быть проще для обратимых схем по сравнению с необратимыми схемами.

В настоящее время ведутся исследования по созданию сбоеустойчивых обратимых схем, способных обнаружить широкий класс ошибок на выходах [27, 28]. Для этого вводится класс обратимых логических вентилей, состоящий из элемента Фредкина и модифицированного двойного затвора Фейнмана.

Небольшое количество публикаций в литературе основано на довольно специальных методах формирования обратимых кодов, которые, как правило, связаны с большими накладными расходами. Следует надеяться, что методы, основанные на защищенном контроле бита четности, сами по себе и в сочетании с другими методами окажутся полезными для обеспечения устойчивости обратимых логических схем в различных областях их применения.

Заключение

В данной работе рассмотрены основные методы проектирования комбинационных КМОП-схем, обеспечивающие автоматическое исправление ошибок. Это такие методы, как использование самопроверяемых и самокорректируемых схем, применение избыточного кодирования информации, использование обратимой схемотехники.

Работа выполнена при поддержке РФФИ (проект № 15-07-02065 А).

Список литературы

1. Huang H.-M., Wen H.-P. W. Fast-yet-accurate statistical soft-error-rate analysis considering full-spectrum charge collection // IEEE Design & Test, 2013. P. 77–86.
2. Согомонян Е. С., Слабаков Е. В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
3. Варшавский В. И., Розенблюм Л. Я., Таубин А. Р. Полностью самопроверяемые асинхронные комбинационные схемы и свойство индицируемости // Автоматика и телемеханика, 1982. Вып. № 5. С. 138–146.
4. Щербаков Н. С. Самокорректирующиеся дискретные устройства. М.: Машиностроение, 1975. 216 с.
5. Коц Ч. Коды с исправлением ошибок и их реализация в цифровых системах // Методы введения избыточности для вычислительных систем / Под ред. В. С. Пугачева. М.: Сов. радио, 1966. С. 179–229.
6. Стемповский А. Л., Гаврилов С. В., Иванова Г. А., Рыжова Д. И. Методы повышения надежности комбинационных микроэлектронных схем на основе мультиинтервального анализа быстродействия // Системы высокой доступности. 2015. № 4. С. 69–76.
7. Poolakkaparambil M., Mathew J. BCH code based multiple bit error correction in finite field multiplier circuits // Proc. of the 12th International Symposium on Quality Electronic Desing, ISQED, 2011. Santa Clara, California, USA, 14–16 March 2011. P. 1–6.
8. Гаврилов С. В., Гуров С. И., Жукова Т. Д., Рыжова Д. И. Применение теории кодирования для повышения надежности

комбинационных схем // Информационные технологии. 2016. № 12. С. 931–937.

9. Гаврилов С. В., Гуров С. И., Жукова Т. Д., Рыжова Д. И., Тельпухов Д. В. Методы повышения сбоеустойчивости комбинационных ИМС методами избыточного кодирования // Прикладная математика и информатика: Труды факультета Вычислительной математики и кибернетики. М.: Изд-во факультета ВМК МГУ, 2016. № 53. С. 93–102.

10. Иваськов Ю. Л., Рякин О. М. Об одной информационной модели ненадежных комбинационных схем // Кибернетика. 1966. Вып. 2. № 6. С. 41–46.

11. Стемповский А. Л., Гаврилов С. В., Иванова Г. А., Рыжова Д. И., Соловьев А. Н. Методы синтеза помехозащищенных комбинационных блоков // Информационные технологии. 2015. Т. 21. № 11. С. 821–826.

12. Hsiao M. Y. A class of optimal minimum odd-weight-column SEC-DED codes // IBM J. Res. Develop. 1970. Vol. 14. P. 395–401.

13. Стемповский А. Л., Гаврилов С. В., Иванова Г. А., Соловьев А. Н. Оптимизация схем кодирования на основе выбора варианта коммутаций с учетом логических корреляций между выходами комбинационной схемы // Известия ЮФУ. Технические науки. 2015. № 6 (167). С. 255–262.

14. Щербаков Н. С. Достоверность работы цифровых устройств. М.: Машиностроение, 1989. 224 с.

15. Кулик А. С. Элементы теории рационального управления объектами. Харьков: Изд. ХАИ, 2016. 255 с.

16. Савченко Ю. Г. Использование естественной информационной избыточности для автокоррекции ошибок в логических сетях // Кибернетика. 1970. № 6. С. 85–87.

17. Юдинцев В. Радиационно-стойкие интегральные схемы. Надежность в космосе и на земле // Электроника: Наука, технология, бизнес. 2007. № 5. С. 72–77.

18. Стемповский А. Л., Тельпухов Д. В., Соловьев Р. А., Тельпухова Н. В. Исследование вероятностных методов оценки логической уязвимости комбинационных схем // VII Всероссийская научно-техническая конференция "Проблемы разработки перспективных микро- и наноэлектронных систем — 2016": сб. науч. тр. под общей ред. А. Л. Стемповского. Часть IV. С. 121–126.

19. Колесник В. Д., Мирончиков Е. Т. Коды с исправлением ошибок при арифметических операциях // Проблемы передачи информации. 1965. С. 20–28.

20. Стемповский А. Л., Тельпухов Д. В., Соловьев Р. А., Рыжов В. С. Тестовая система для сравнения алгоритмов, увеличивающих надежность комбинационных схем // Информационные технологии. 2015. Т. 21, № 12. С. 910–915.

21. Вернер М. Основы кодирования. М.: Техносфера, 2004. 288 с.

22. Anwar M. T., Lala P. K., Thenappan P. Decoder Design for a New Single Error Correcting/Double Error Detecting Code // Proceeding of world academy of science, 2007. Vol. 22. P. 247–251.

13. Gallager R. G. Low density parity check codes. Cambridge: M.I.T. Press, 1963. 90 p.

24. Кодирование информации (двоичные коды): справочник / Под ред. проф. Н. Т. Березнюка. Харьков: Вища школа, 1978. 252 с.

25. Карповский М. Г., Москалев Э. С. Спектральные методы анализа и синтеза дискретных устройств. Л.: Энергия, 1973. 144 с.

26. Kaur J., Kaur H., Thapar Sh. Reversible and Fault Tolerant Reversible Gates // IJETTCS, 2012. Vol. 1. Issue 3. URL: <http://www.ijettcs.org/Volume1Issue3/IJETTCS-2012-10-25-095.pdf>.

27. Parhami B. Fault-Tolerant Reversible Circuits // Proc. 40th Asilomar Conf. Signals, Systems, and Computers, Pacific Grove, 2006. P. 1726–1729.

28. Boykin P. O., Roychowdhury V. P. Reversible Fault-Tolerant Logic // Proc. Int. Conf. Dependable Systems and Networks, 2005. P. 444–453.

A. L. Stempkovsky, Member of RAS, D. Sc., Professor, Director,
Institute for Design Problems in Microelectronics of Russian Academy of Science (IPPM RAS)],
S. V. Gavrilov, D. Sc., Professor, Head of Department,
Institute for Design Problems in Microelectronics of Russian Academy of Science (IPPM RAS)],
S. I. Gurov, Ph. D., Assistant Professor,
Lomonosov Moscow State University,
D. I. Ryzhova, Ph. D., Junior Research Scientist,
Institute for Design Problems in Microelectronics of Russian Academy of Science (IPPM RAS)]

Methods of Designing of Fault Tolerance CMOS Combinational Circuits which Provides Automatic Correction of Errors

This article is devoted to the main methods of CMOS combinational circuits design, which provides automatic correction of errors. In the article we considered the specifics of self-correcting circuits design and analyzed approaches to automatic error correction based on redundant coding methods. Also important features of the correcting errors problem caused by failures, are considered. In the final part we briefly describe a perspective approach to correcting errors using reversible circuits.

Keywords: combinational circuits, design, fault tolerance, CAD

References

1. **Huang H.-M., Wen H.-P. W.** Fast-yet-accurate statistical soft-error-rate analysis considering full-spectrum charge collection, *IEEE Design & Test*, 2013, pp. 77–86.
2. **Sogomonyan E. S., Slabakov E. V.** *Samoproveryaemye ustroystva i otkazoustoichivye sistemy*. Moscow, Radio i svyaz', 1989, 208 p. (in Russian).
3. **Varshavskii V. I., Rozenblyum L. Ya., Taubin A. R.** Polnost'yu samoproveryaemye asinkhronnye kombinatsionnye skhemy i svoystvo inditsiruemosti, *Avtomatika i telemekhanika*, 1982, no. 5, pp. 138–146 (in Russian).
4. **Shcherbakov N. S.** *Samokorrektruyushchiesya diskretnye ustroystva*, Moscow, Mashinostroenie, 1975. 216 p. (in Russian).
5. **Kots Ch.** Kody s ispravleniem oshibok i ikh realizatsiya v tsifrovyykh sistemakh / V kn.: *Metody vvedeniya izbytochnosti dlya vychislitel'nykh sistem*, V. S. Pugachev, Moscow, Sov. radio, 1966, pp. 179–229 (in Russian).
6. **Stempkovskii A. L., Gavrilov S. V., Ivanova G. A., Ryzhova D. I.** Metody povysheniya nadezhnosti kombinatsionnykh mikroelektronnykh skhem na osnove mul'tiinterval'nogo analiza bystrodeistviya, *Sistemy vysokoi dostupnosti*, 2015, no. 4, pp. 69–76 (in Russian).
7. **Poolakkaparambil M., Mathew J.** BCH code based multiple bit error correction in finite field multiplier circuits, *ISQED*, 2011, pp. 1–6.
8. **Gavrilov S. V., Gurov S. I., Zhukova T. D., Ryzhova D. I.** Primenenie teoriya kodirovaniya dlya povysheniya nadezhnosti kombinatsionnykh skhem, *Informatsionnye tekhnologii*, 2016, no. 12, pp. 931–937 (in Russian).
9. **Gavrilov S. V., Gurov S. I., Zhukova T. D., Ryzhova D. I., Telpukhov D. V.** Metody povysheniya sbestoichivosti kombinatsionnykh IMS metodami izbytochnogo kodirovaniya, *Prikladnaya matematika i informatika: Trudy fakul'teta Vychislitel'noi matematiki i kibernetiki*, Moscow, Izd. VMK MGU, 2016, no. 53, pp. 93–102 (in Russian).
10. **Ivas'kov Yu. L., Ryakin O. M.** Ob odnoi informatsionnoi modeli nenadezhnykh kombinatsionnykh skhem, *Kibernetika*, 1966, issue 2, no. 6, pp. 41–46 (in Russian).
11. **Stempkovskii A. L., Gavrilov S. V., Ivanova G. A., Ryzhova D. I., Soloviev A. N.** Metody sinteza pomekhozashchishchennykh kombinatsionnykh blokov, *Informatsionnye tekhnologii*, 2015, vol. 21, no. 11, pp. 821–826 (in Russian).
12. **Hsiao M. Y.** A class of optimal minimum odd-weight-column SEC-DED codes, *IBM J. Res. Develop.*, 1970, vol. 14, pp. 395–401.
13. **Stempkovskii A. L., Gavrilov S. V., Ivanova G. A., Soloviev A. N.** Optimizatsiya skhem kodirovaniya na osnove vybora varianta kommutatsii s ucheto logicheskikh korrelyatsii mezhdu vykhodami kombinatsionnoi skhemy, *Izvestiya YuFU. Tekhnicheskie nauki*, 2015, no. 6 (167), pp. 255–262 (in Russian).
14. **Shcherbakov N. S.** *Dostovernost' raboty tsifrovyykh ustroystv*, Moscow, Mashinostroenie, 1989, 224 p. (in Russian).
15. **Kulik A. S.** *Elementy teorii ratsional'nogo upravleniya ob"ektami*, Khar'kov, KhAI, 2016. 255 p. (in Russian).
16. **Savchenko Yu. G.** Ispol'zovanie estestvennoi informatsionnoi izbytochnosti dlya avtokorrektsii oshibok v logicheskikh setyakh, *Kibernetika*, 1970, no. 6, pp. 85–87 (in Russian).
17. **Yudintsev V.** Radiatsionno-stoikie integral'nye skhemy. Nadezhnost' v kosmose i na zemle, *Elektronika: Nauka, tekhnologiya, biznes*, 2007, no. 5, pp. 72–77 (in Russian).
18. **Stempkovskii A. L., Telpukhov D. V., Soloviev R. A., Telpukhova N. V.** Issledovanie veroyatnostnykh metodov otsenki logicheskoi uyazvimosti kombinatsionnykh skhem, *VII Vserossiyskaya nauchno-tekhnicheskaya konferentsiya "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem — 2016": sb. nauchn. tr.*, ed. A. L. Stempkovskii, chast' IV, pp. 121–126 (in Russian).
19. **Kolesnik V. D., Mironchikov E. T.** Kody s ispravleniem oshibok pri arifmeticheskikh operatsiyakh, *Problemy peredachi informat-sii*, 1965, pp. 20–28 (in Russian).
20. **Stempkovskii A. L., Telpukhov D. V., Soloviev R. A., Rukhlov V. S.** Testovaya sistema dlya sravneniya algoritmov, uvelichivayushchikh nadezhnost' kombinatsionnykh skhem, *Informatsionnye tekhnologii*, 2015, vol. 21, no. 12, pp. 910–915 (in Russian).
21. **Verner M.** *Osnovy kodirovaniya*, Moscow, Tekhnosfera, 2004, 288 p. (in Russian).
22. **Anwar M. T., Lala P. K., Thenappan P.** Decoder Design for a New Single Error Correcting/Double Error Detecting Code, *Proceeding of world academy of science*, 2007, vol. 22, pp. 247–251.
23. **Gallager R. G.** *Low density parity check codes*, Cambridge, M. I.T. Press, 1963. 90 p.
24. **Kodirovanie informatsii (dvoichnye kody)**, Spravochnik, ed. N. T. Berezhnyuk, Khar'kov, Vishcha shkola, 1978, 252 p. (in Russian).
25. **Karpovskii M. G., Moskalev E. S.** *Spektral'nye metody analiza i sinteza diskretnykh ustroystv*, Leningrad, Energiya, 1973, 144 p.
26. **Kaur J., Kaur H., Thapar Sh.** Reversible and Fault Tolerant Reversible Gates, *IJETTCs*, 2012, vol. 1, issue 3. URL: <http://www.ijettcs.org/Volume1Issue3/IJETTCs-2012-10-25-095.pdf>.
27. **Parhami B.** Fault-Tolerant Reversible Circuits, *Proc. 40th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, 2006, pp. 1726–1729.
28. **Boykin P. O., Roychowdhury V. P.** Reversible Fault-Tolerant Logic, *Proc. Int. Conf. Dependable Systems and Networks*, 2005, pp. 444–453.

В. М. Курейчик, д-р техн. наук, проф., зав. каф., kur@tgn.sfedu.ru,

И. Б. Сафроненкова, аспирант, safronenkova050788@yandex.ru,

Федеральное государственное автономное образовательное учреждение высшего образования
"Южный федеральный университет", г. Таганрог

Разработка архитектуры СППР по выбору методов решения задач компоновки

Описана разработка архитектуры системы поддержки принятия решений, направленной на оптимизацию выбора метода решения задач компоновки при проектировании электронно-вычислительной аппаратуры (ЭВА). Проведен анализ формальных постановок задачи компоновки с учетом различных критериев. Предложена математическая модель рассматриваемой задачи, принципиальным отличием которой является учет критерия числа межблочных связей и критерия задержки сигнала. Синтезирована структура пространства признаков классификации задач компоновки.

Ключевые слова: система поддержки принятия решений, онтология, класс, свойство, экземпляр, компоновка, многокритериальная задача оптимизации, гиперграф

Введение

В условиях рыночной экономики конкурентная борьба за потребителей требует от предприятия постоянного обновления выпускаемой продукции, повышения ее качества, максимального удовлетворения пожеланий заказчиков, для чего необходимо сокращать сроки и стоимость инженерной подготовки производства, качественно совершенствовать разрабатываемые проекты. Зачастую инженер-конструктор вынужден исходить из своих субъективных представлений об эффективности возможных альтернатив и важности различных критериев в процессе проектирования. Такой подход далеко не всегда гарантирует принятие верного решения [1]. В условиях увеличения объема информации, усложнения решаемых задач, необходимости учета большого числа взаимосвязанных факторов и быстро меняющихся требований к проектируемому объекту актуальность систем поддержки принятия решений (СППР) в проектной деятельности инженера растет. Автоматизировать в полном объеме творческие операции невозможно. Однако современные методы искусственного интеллекта, использующие в своей работе базы знаний (БЗ), позволяют оказать инженеру существенную помощь в принятии решений, предлагая ему на выбор допустимые альтернативы в задачах, не поддающихся формализации [2, 3]. Задача компоновки коммутационной схемы (КС) является первой на этапе конструкторского проектирования, служит источником информации для последующих этапов (задач размещения и трассировки), определяет качество и эффективность процесса проектирования в целом. Ключевым фактором здесь становится оптимальный выбор метода решения задачи компоновки, позволяющий получить предпочтительный для лица принимающего

решения (ЛПР) вариант с учетом временных затрат. Каждый из существующих методов ориентирован на решение определенного круга задач, имеет свои преимущества и недостатки. В настоящее время на рынке интеллектуальных технологий имеется большое число программ ECAD (англ. Electronics Computer Aided Design — САПР электронных устройств, радиоэлектронных средств, интегральных схем, печатных плат и т. п.), различных по функционалу, мощности и цене, используемым методам решения. Зачастую инженеру трудно разобраться во всем многообразии предоставляемых продуктов. Хотя полное исключение человека из процесса принятия решений невозможно, создание СППР, направленной на поиск и рекомендацию эффективных или новых методов решения, предполагает повысить качество и сократить сроки процесса проектирования. Ввиду недостаточности подобных разработок и отсутствию аналогов в современной отечественной научной среде авторами предлагается разработка СППР, направленной на оптимизацию выбора метода решения задачи компоновки КС (далее — СППР ВМ САПР), на этапе проектирования. Принцип работы СППР ВМ САПР заключается в следующем. На вход системы поступает формальное описание произвольной задачи компоновки и выполняется ее классификация в соответствии с определенным ранее пространством признаков. Результатом классификации является отнесение произвольной задачи к определенному классу в онтологии задач. Далее происходит соотнесение классифицированной задачи компоновки с возможными методами ее решения и рекомендацией оптимального.

В разделах данной работы будут рассмотрены следующие вопросы.

1. Анализ формальных постановок задачи компоновки КС с учетом критерия межблочных связей

и критерия задержки распространения сигнала. Формулировка задачи компоновки как многокритериальной задачи оптимизации.

2. Синтез структуры пространства признаков задач компоновки для дальнейшей процедуры классификации.

3. Синтез обобщенной архитектуры СППР ВМ САПР.

Данное исследование направлено на формирование общей концепции и принципа работы СППР ВМ САПР, разработку математической модели оптимизационной задачи компоновки и синтез классификационных признаков задачи компоновки.

1. Анализ формальных постановок задачи компоновки КС с учетом различных критериев. Формулировка задачи компоновки как многокритериальной

Компоновкой КС ЭВА на конструктивно законченные части называется процесс распределения элементов низшего конструктивного уровня в высший в соответствии с выбранным критерием [4]. При решении такого вида задач, ввиду увеличения числа элементов, уменьшения размеров компонентов БИС и СБИС, повышения их быстродействия, наряду с критерием числа межблочных связей и функциональной законченности блоков, явилась необходимость учета такого критерия как длительность задержки распространения сигнала.

Сформулируем задачу компоновки КС как задачу разрезания ориентированного гиперграфа $H = (X, E)$ со взвешенными вершинами и ребрами на m подсхем. Требуется разбить множество X гиперграфа H на подмножества $X_1, X_2, \dots, X_m$, такие что:

$$X_i \neq \emptyset, i \in I = \{1, 2, \dots, m\};$$

$$X_k \cap X_l = \emptyset; \bigcup_{k \in I} X_k = X; |X_k| = n_k; k, l \in I.$$

Формулировка задачи компоновки с учетом критерия межблочных связей. Пусть S_j — j -й маршрут от одного из начальных ребер гиперграфа H к одному из конечных ребер; l_j — условная длина j -го маршрута, которая определяется числом частей гиперграфа H , через которые проходит маршрут S_j ,

$$l_j = \sum_{i=1}^m \sum_{k=2}^t \alpha_{ki}^{(j)}, \quad (1)$$

где

$$t = |E_j|, E_j = \{e_j | e_j \in S_j\};$$

$$\alpha_{ki}^{(j)} = \begin{cases} 1, & \text{если } X_{k-1} \subset P_i, \text{ но } \exists x \in X_k: x \notin P_i \\ 0, & \text{в противном случае.} \end{cases}$$

Таким образом, некоторое фиксированное разбиение f характеризуется величиной $\alpha_f = l_{j\max}$, которая является максимальным числом частей ги-

перграфа H , через которые проходят маршруты от начальных ребер гиперграфа к конечным при данном разрезании.

Задача компоновки состоит в отыскании такого разрезания f из множества возможных F гиперграфа H , при котором минимизируется значение α_f при заданных ограничениях

$$\Phi = \min_{\alpha_f}, \text{ при } \begin{cases} |P_i| \leq B \\ C_i \leq C \end{cases}, \quad (2)$$

где B — число элементов в блоке; C — число внешних выводов в блоке.

Формулировка задачи компоновки с учетом критерия задержки распространения сигнала. Пусть каждой вершине и ребру ориентированного гиперграфа ставится в соответствие значение условной функциональной задержки (УФЗ). Если отсутствуют данные о результатах функционально-логического моделирования схемы, то значение УФЗ ставит в соответствие вершине или ребру определенный шаг волны, распространяющейся в гиперграфе от его начальных ребер (для которых УФЗ = 1) к конечным. Обозначим УФЗ вершины X_i через r_{xi} , а ребра e_j через r_{ej} . УФЗ ребра e_j , для которого вершина X_i интерпретирует элемент схемы, являющийся источником сигнала в цепи j , определяется как $r_{ej} = r_{xi} + 1$. УФЗ вершины X_i определяется как максимальное значение УФЗ инцидентных этой вершине ребер $E' = \{e_s, e_p, \dots, e_f\}$, для интерпретируемых которыми цепей схемы элемент, взаимно-однозначно соответствующий вершине X_i , не является источником сигнала в цепях $s, t, \dots, f \in I$ схемы, т. е. $r_{xi} = \max_{e_j \in E} r_{ej}$.

Если известны результаты функционально-логического моделирования схемы, то начальным ребрам графа присваиваются значения, равные нулю. УФЗ ребра e_j определяется как $r_{ej} = r_{xi} + \Delta r_{xi}$, где Δr_{xi} — оценка задержки элемента, соответствующего вершине X_i , полученная в результате моделирования схемы.

Каждая часть гиперграфа P_γ обладает такой характеристикой, как интервал значений УФЗ принадлежащих ей вершин:

$$\Delta R_\gamma = r_{\gamma\max} - r_{\gamma\min}, \quad (3)$$

где ΔR_γ — интервал значений УФЗ части P_γ гиперграфа H ; $r_{\gamma\max}$ — максимальное значение УФЗ вершин, принадлежащих P_γ ; $r_{\gamma\min}$ — минимальное значение УФЗ вершин, принадлежащих P_γ .

Произвольное фиксированное разбиение f гиперграфа H характеризуется максимальным значением интервала при данном разбиении:

$$\Delta R_f = \Delta R_{\gamma\max}. \quad (4)$$

При компоновке схемы ставится задача разрезать гиперграф так, чтобы получить минимальное

из возможных значений ΔR_f при следующих заданных ограничениях:

$$\Phi = \min \Delta R_{f_{\max}} \text{ при } \left\{ \begin{array}{l} |P_i| \leq B \\ C_i \leq C \end{array} \right\}, \quad (5)$$

где B — число элементов в блоке; C — число внешних выводов в блоке.

При учете обоих рассмотренных критериев задача компоновки формулируется как задача многокритериальной оптимизации и описывается целевой вектор-функцией, определенной в области D_X со значениями в целевом пространстве $\{F\} = R^{|F|}$:

$$F(X) = (f_1(X), f_2(X)), \quad (6)$$

где $f_1(X), f_2(X)$ — частные целевые функции; $\{X\}$ — ограниченное и замкнутое множество допустимых значений вектора варьируемых параметров $D_X = \{X | G(X) \geq 0\} \subset \{X\} = R^{|X|}$.

Положим, что

$$f_1(X) = \mu_1 K_1, f_2(X) = \mu_2 K_2, \quad (7)$$

где K_1 — критерий межблочных связей; K_2 — критерий задержки распространения сигнала; μ_1 и μ_2 — априори назначаемые ЛПР весовые коэффициенты при оценивании критериев K_1 и K_2 соответственно.

Тогда оптимизационная функция имеет вид:

$$F(X) = \mu_1 K_1 + \mu_2 K_2, \quad (8)$$

$$K_1, K_2 \rightarrow \min.$$

Поскольку ЛПР стремится минимизировать в области D_X каждую из частных целевых функций $f_1(X), f_2(X)$, т. е.:

$$\min_{X \in D_X} F(X) = F(X^*) = F^*, \quad (9)$$

где векторы X^*, F^* — искомое решение задачи многокритериальной оптимизации [5, 6].

2. Синтез структуры пространства признаков задач компоновки для дальнейшей процедуры классификации

Согласно концепции работы СППР ВМ САПР, после того, как на вход системы поступает формальное описание произвольной задачи компоновки, необходимо ее классифицировать по какому-либо признаку или набору признаков, чтобы отнести к определенному классу в онтологии задач.

Рассмотрим процедуру синтеза признаков на примере задачи компоновки конструктивных блоков ЭВА. Пусть исходными данными для компоновки является математическая модель КС (M), которая может быть интерпретирована в виде графа, мультиграфа или гиперграфа [7].

Положим, что критериями оптимизационной функции являются:

- число межблочных соединений;
- значение задержки распространения сигнала.

Чаще всего при решении задач компоновки используют ограничения на максимальное число выводов каждого блока и максимальное число элементов, назначаемых в эти блоки: число элементов в подсхеме (B); число внешних выводов подсхемы (C).

Оба эти ограничения в совокупности с критерием межблочных связей и критерием задержки распространения сигнала позволят получить наиболее оптимальное решение поставленной задачи.

Таким образом, с учетом формулы (8), получаем пространство признаков классификации задач компоновки конструктивных блоков, состоящее из целевой вектор-функции, ограничений, наложенных на управляемые параметры, и исходных данных задачи:

$$W = (F(X), B, C, M). \quad (10)$$

Таким пространством признаков описывается любая задача, принадлежащая к классу задач компоновки конструктивных модулей, и, соответственно, относится к определенному классу в онтологии задач.

3. Синтез обобщенной архитектуры СППР ВМ САПР

Как правило, СППР состоит из трех компонентов: системы управления базой данных (СУБД); системы управления базой моделей (СУБМ) и системы управления интерфейсом между пользователем и компьютером (рис. 1). База данных выступает в роли хранилища данных для СППР. Она содержит информацию, релевантную к классу проблем, решаемых СППР, и обеспечивает логическую структуру данных, с которой взаимодействует пользователь. Роль базы моделей в СППР аналогична. Целью создания моделей являются описание и оптимизация некоторого объекта или процесса. Использование моделей обеспечивает проведение анализа в системах поддержки принятия решений. Иначе говоря, целью базы моделей (БМ) является преобразование информации, полученной из базы данных (БД), в информацию, полезную для процесса принятия решений [8].

В настоящей работе предлагается представить БД в виде онтологии задач компоновки, а БМ — в виде онтологии методов решения задач данного класса. Использование онтологического подхода как способа хранения информации в СППР представляется логичным ввиду возможности не только



Рис. 1. Архитектура СППР ВМ САПР

систематизации и формализации разнородных видов знаний и данных системы, но и для организации удобного доступа к ним [9].

Онтология строится как сеть, состоящая из концептов и связей между ними. Связи могут быть различного типа (например, "является", "состоит из", "является исполнителем" и т. п.). Из ряда определений онтологии, приводимых в литературе [10—14], в данной работе будем использовать следующее ее формальное представление:

$$OHT = \langle C, P, R, F \rangle, \tag{11}$$

где *OHT* — онтология; *C* — конечное множество терминов, обозначающих объекты и понятия данной предметной области (ПрО); *P* — конечное множество свойств этих понятий; *R* — конечное множество отношений между понятиями ПрО; *F* — конечное множество функций интерпретации, заданных на концептах и/или отношениях онтологии [15].

В рамках данной работы разработаны две онтологии в среде Protégé 4.2: онтология задач компоновки КС (рис. 2) и онтология методов решения задач компоновки КС (рис. 3) [16].

Дадим пояснения к рис. 2. Класс "Компоновка" имеет три подкласса согласно известной классификации: "Компоновка конструкторских частей", "Задача покрытия", "Задача типизации". С помощью отношения "имеет" устанавливается связь подкласса "Компоновка конструкторских частей" с классами "Критерии_1", "Ограничения_1", "Исходные данные_1". В свою очередь, данные классы имеют соответственно экземпляры, перечисленные выше. Аналогично подклассы "Задача покрытия" и "Задача типизации" также имеют свои критерии, ограничения и исходные данные.

Проанализировав существующие классификации методов компоновки КС [17], авторы предлагают онтологию методов, представленную на рис. 3.

Из рис. 3 видно, что класс "Методы компоновки" подразделяется на четыре подкласса: "Точные", "Последовательные", "Итерационные", "Композиционные". Иерархия методов компоновки приведена на рис. 4.

В онтологии методов компоновки также приведены экземпляры перечисленных выше подклассов. Например, подкласс "Итерационные" имеет экземпляры: "Алгоритм_Кернигана_Лина", "Алгоритм_Федучио_Маттеуса". У подкласса "Случайные", относящегося к "Последовательные" экземпляры: "Метод_Монте_Карло", "Метод_случайных_назначений". Также в онтологии отражена связь через отношение "имеет" методов с диапазонами вычислительной сложности алгоритмов (ВСА) (рис. 5) [18].

Предполагается, что группа экспертов проводит соотнесение онтологии задач компоновки и онтологии методов решения, т.е. каждому классифицированному типу задачи ставят в соответствие метод

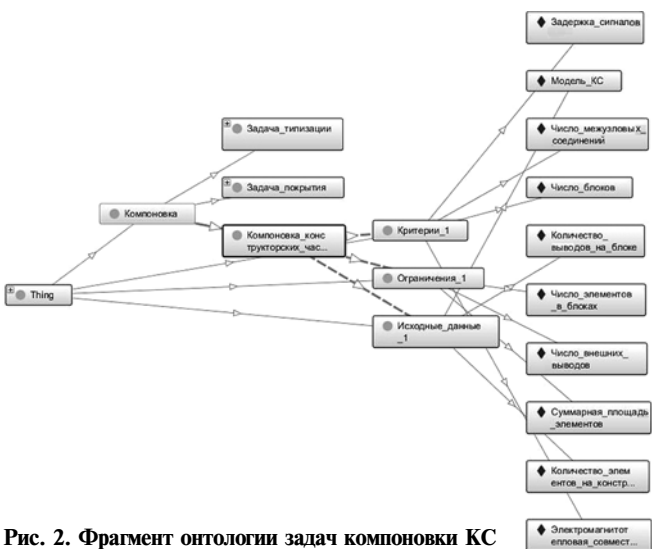


Рис. 2. Фрагмент онтологии задач компоновки КС

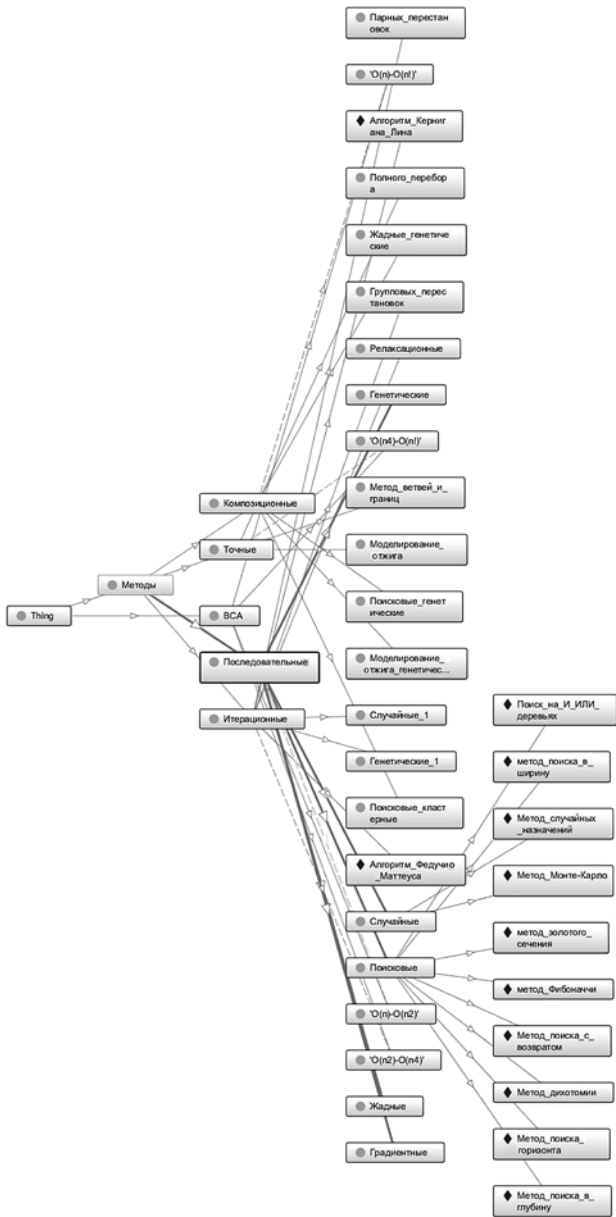


Рис. 3. Онтология методов решения задач компоновки КС

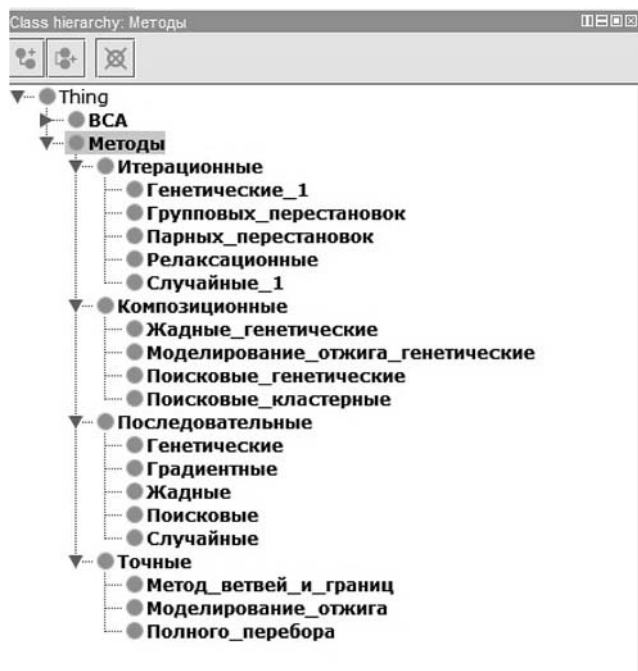


Рис. 4. Иерархия методов компоновки ЭВА в среде Protégé 4.2

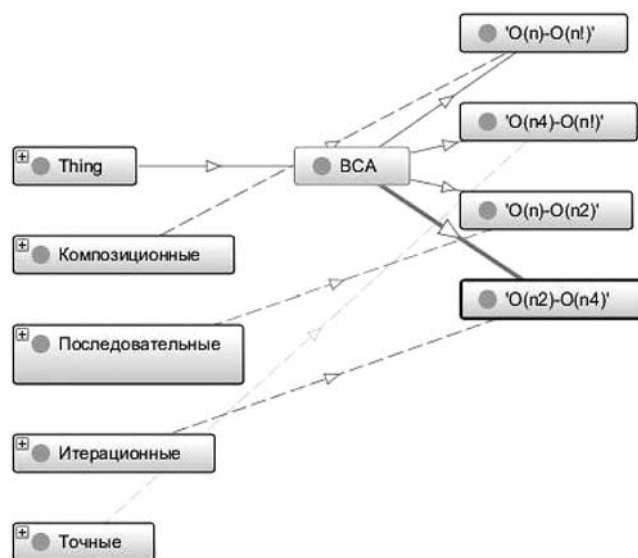


Рис. 5. Фрагмент онтологии методов решения задач компоновки КС

ее решения, предполагающий наилучший вариант. Таким образом, ЛПР получает в качестве результата работы СППР ВМ САПР рекомендацию по выбору метода решения произвольной задачи компоновки КС.

Заключение

В работе предложена архитектура СППР ВМ САПР, направленная на рекомендацию оптимального выбора метода решения задачи компоновки КС на этапе проектирования ЭВА. Решение задачи компоновки как задачи разрезания ориентирован-

ного гиперграфа позволило учесть два критерия и сформулировать комплексный. Разработаны две онтологии предметной области в среде Protégé 4.2, которые используются в модели СППР в качестве хранилища данных и знаний. С практической точки зрения реализация данных онтологий позволяет инженеру избежать решения слабоформализуемых задач, а значит сократить время и повысить качество всего процесса проектирования. Новизна данного исследования заключается в описании концепции работы такой СППР, что на основе ее рекомендаций предполагается выбор существующей или разработка новой системы автоматизированного проектирования (САПР) для решения задачи компоновки. В дальнейшем планируется рассмотреть случай неоднозначности работы классификатора. Для выработки оптимальной рекомендации будет предложена модифицированная архитектура СППР путем введения дополнительного элемента, позволяющего компенсировать неоднозначность в зависимости от поставленных требований.

Работа выполнена за счет частичного финансирования по ГЗ № 2.5537.2017 ВУ в ЮФУ.

Список литературы

1. **Евгеньев Е. Б.** Интеллектуальные системы проектирования: учеб. пособие. М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 334 с.
2. **Трахтенгерц Э. А.** Компьютерная поддержка принятия решений: Научно-практическое издание. Серия "Информатизация России на пороге XXI века". М.: СИНТЕГ, 1998. 376 с.
3. **Тихонов А. Н., Цветков В. Я.** Методы и системы поддержки принятия решений. М.: МАКС Пресс, 2001. 312 с.
4. **Курейчик В. В., Курейчик В. М.** Эволюционные методы компоновки блоков ЭВА. Таганрог: Изд-во ТРТУ, 2006. 120 с.
5. **Таберкин П. А.** Компоновка и размещение элементов схем ЭВА с минимизацией длительности задержки сигналов: Автореф. дис. канд. тех. наук. Таганрог, 1988. 16 с.
6. **Карпенко А. П.** Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. М.: Издательство МГТУ им. Н. Э. Баумана, 2014. 446 с.
7. **Курейчик В. М.** Математическое обеспечение конструкторского и технологического проектирования с применением САПР: учебник для вузов. М.: Радио и связь, 1990. 352 с.
8. **Decision Support Systems.** URL: <http://www.pitt.edu/~druzdzel/psfiles/dss.pdf>.
9. **Загоруйко Г. Б.** Разработка онтологии для интернет-ресурса поддержки принятия решений в слабоформализованных областях // Онтология проектирования. 2016. Т. 6, № 4 (22). С. 485—500.
10. **Курейчик В. М., Сафроненкова И. Б.** Создание онтологической модели систем автоматизированного проектирования в среде Protege 4.2 // Сборник трудов VII Всероссийской научно-технической конференции "Проблемы разработки перспективных микро- и наноэлектронных систем — 2016", Россия, Москва. М.: ИППМ РАН, 2016.
11. **Ной Н. Ф., Макгинесс Д. Л.** Разработка онтологий 01:руководство по созданию вашей первой онтологии. URL: http://ifets.ieee.org/russian/depositary/ontology101_rus.doc
12. **Добров Б. В., Иванов В. В., Лукашевич Н. В., Соловьев В. Д.** Онтологии и тезаурусы: модели, инструменты, приложения: учеб. пособие. М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2013. 173 с.
13. **Цуканова Н. И.** Онтологическая модель представления и организации знаний: учеб. пособие для вузов. М.: Горячая линия — Телеком, 2014. 272 с.

14. Kureychik V. M., Semenova A. V. Domain ontology development for linguistic purposes. In 9th International Conference on Application of Information and Communication Technologies, pp. 83–87. IEEE Press, Rostov-on-Don, Russia, 2015. P. 14.
15. Гаврилова Т. А., Кудрявцев Д. В., Муромцев Д. И. Инженерия знаний. Модели и методы: учебник. СПб.: Лань, 2016. 324 с.

16. Антониоу Г., ван Хармелен Ф., Хоекстра Р. Семантический веб / Пер. с англ. Т. Шульга. М.: ДМК Пресс, 2016. 240 с.
17. Емельянов В. В., Курейчик В. В., Курейчик В. М. Теория и практика эволюционного моделирования. М.: ФИЗМАТЛИТ, 2003. 423 с.
18. Курейчик В. В., Сороколетов П. В. Композитные методы разбиения графовых моделей. Таганрог: Изд-во ТРТУ, 2006. 140 с.

V. M. Kureichik, D. Sc., Prof., Chair of Discrete Maths and Methods of Optimization, kur@tgn.sfedu.ru,
I. B. Safronenkova, Postgraduate Student, Assistant of Discrete Maths and Methods of optimization,
 safronenkova050788@yandex.ru

Autonomous Federal State Institution of Higher Education Southern Federal University, Taganrog

Decision Support System Development for the Choice of Problem-Solving Procedure of Commutation Circuit Partitioning

This paper concerned with architectural design of ontology-based decision support system intended to optimize the choice of problem-solving procedure of commutation circuit partitioning during the design phase of electronic computer. Analysis of formal commutation circuit partitioning problems definition against the criteria was performed. Mathematical model of present problem was posed. The fundamental difference of this model is consideration the criteria of inter-bay wiring and signal delay as local cost functions in multicriterion optimization problem. In this paper a space of classification attributes of commutation circuit partitioning problems was designed: local cost functions, constrains and initial data are defined. Data store and databank of ontology-based decision support system are presented as a CPP problems ontology and a problem-solving technique ontology. These domain ontologies were developed in Protégé 4.2.

Keywords: ontology, decision support system, commutation circuit partitioning problem, class, attribute, instances, hypergraph, optimization task

References

1. Evgen'ev E. B. *Intelligence CAD — system*, MGТУ im. N. Je. Bauman, Moscow, 2009, 334 p. (in Russian).
2. Trahtengerc Je. A. *Computer support of decision making*. Moscow, SINTEG, 1998, 376 p. (in Russian).
3. Tihonov A. N., Cvetkov V. Ja. *Methods and systems of decision making*. Moscow, MAKSPress, 2001, 312 p. (in Russian).
4. Kureichik V. V., Kureichic V. M. *Evolutionary methods of blocking*. Taganrog, TRТУ, 2006, 120 p. (in Russian).
5. Taberkin P. A. *Commutation circuit partitioning and feature placement with minimization signal delay. Author's abstract*. Taganrog, TRТИ, 1988, 16 p. (in Russian).
6. Karpenko A. P. *Modern search optimization algorithms. Bio-inspired algorithms*, Moscow, MGТУ im. N. Je. Bauman, 2014.
7. Kureichik V. M. *Mathematical support of engineering and production process with the help of CAD-systems*, Moscow, Radio i svjaz', 1990.
8. **Decision Support Systems**. URL: <http://www.pitt.edu/~druzdzel/psfiles/dss.pdf>.
9. Zagorul'ko G. B. Development of ontology for intelligent scientific internet resource decision-making support in weakly formalized domains, *Ontology of designing*, 2016, vol. 6, no. 4 (22), pp. 485–500.
10. Kureychik V. M., Safronenkova I. B. Creation of CAD-systems ontology using Protege 4.2, *In All-Russia Science&Technology Conference "Problems of Advanced Micro- and Nanoelectronic Systems Development"*, 2016, vol. 3, pp. 240–245, IPPM RAN.
11. Noy N., McGuinness D. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, (2001).
12. Dobrov B. V., Ivanov V. V., Lukashevich N. V., Solov'ev V. D. *Ontology and thesaurus: models, tools, applications*. Moscow, BINOM, 2013.
13. Cukanova N. I. *Ontology-based model of knowledge representation and knowledge management*. Moscow, Telekom, 2014.
14. Kureychik V. M., Semenova A. V. Domain ontology development for linguistic purposes. In 9th International Conference on Application of Information and Communication Technologies, IEEE Press, Rostov-on-Don, Russia, 2015, pp. 83–87.
15. Gavrilova T. A., Kudrjavcev D. V., Muromcev D. I. *Knowledge engineering. Models and methods*. St. Petersburg, Lan', 2016.
16. Antoniou G., Harmelen van F., Hoekstra R. *Semantic Web*. Moscow, DMK Press, 2016.
17. Emel'janov V. V., Kureichik V. V., Kureichik V. M. *Theory and practice of evolutionary modeling*, Moscow, FIZMATLIT, 2003.
18. Kureichik V. V., Sorokoletov P. V. *Composed methods of graph partition*. Taganrog, TRТУ, 2006.

В. В. Бурлов, канд. техн. наук, проф., проф. каф., e-mail: vladimir-burlov@yandex.ru,
Л. В. Ремонтова, доц., доц. каф., e-mail: remontova@mail.ru,
 Пензенский государственный технологический университет,
В. В. Косолапов, канд. техн. наук, доц. каф., e-mail: vladimir.kosolapov@mail.ru,
Е. В. Косолапова, ст. преподаватель, e-mail: K-art-inka@yandex.ru,
 Нижегородский государственный инженерно-экономический университет

Моделирование кривой и поверхностей второго порядка в системе КОМПАС-3D

Рассматриваются инструментальные средства и функциональные возможности использования программного продукта КОМПАС-3D. Демонстрируются различные подходы к построению кривой второго порядка — параболы и созданию производных от нее поверхностей в системе трехмерного моделирования. Статья направлена на более глубокое изучение геометро-графических дисциплин и продвижение отечественного IT-продукта в сфере образования.

Ключевые слова: ассоциативный чертеж, библиотека, график функциональной зависимости, гиперболический параболоид, конус, коническое сечение, канонические и параметрические уравнения, кривая, парабола, поверхность, система КОМПАС-3D

Введение

Решение задач по конструированию поверхностей является одной из основных задач инженерной геометрии. Современные достижения компьютерных технологий сделали компьютерную графику мощным инструментом, который используется в различных областях — архитектуре, технике, дизайне, в том числе в процессах проектирования и моделирования с составлением конструкторской документации. В настоящее время существующие программные комплексы позволяют создавать модели и выполнять расчеты конструкций практически любой формы, но любая конфигурация поверхности состоит из различного рода простых кривых линий и поверхностей. К ним можно отнести кривые и поверхности второго порядка, применение которых в нашей жизни достаточно велико. Например, оптическое свойство параболы широко используются сегодня в самых различных сферах жизни (карманный фонарик, автомобильные фары, прожекторы и т. д.).

В России благодаря мощным функциональным возможностям твердотельного и поверхностного моделирования стандартом для тысяч предприятий стала система трехмерного моделирования КОМПАС-3D. Однако в графическом редакторе КОМПАС-3D есть команда **Эллипс**, но нет команды **Парабола**, без которой невозможно создать 3D-модели соответствующих поверхностей второго порядка: гиперболического и эллиптического параболоидов, параболического цилиндра.

Цель работы — продемонстрировать алгоритмы решения пространственных задач по созданию кривых и поверхностей второго порядка в программе КОМПАС-3D на основе законов начертательной

геометрии с применением конических сечений кругового конуса, канонических и параметрических уравнений с построением 3D-моделей гиперболического и эллиптического параболоидов. Основы построения кривых и поверхностей второго порядка с помощью различных инструментов затрагиваются в многочисленных работах [1—5].

Использование конических сечений

Система КОМПАС-3D позволяет выполнить ассоциативный чертеж детали любой сложности с необходимыми разрезами и сечениями. Если в качестве 3D-модели использовать конус вращения, то на ассоциативном чертеже можно получить все кривые второго порядка в виде конических сечений. Сечения

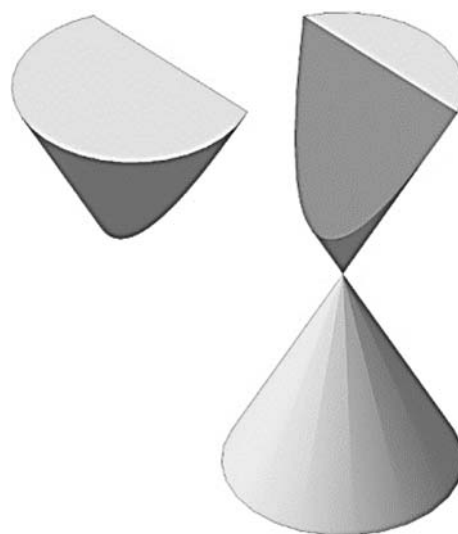


Рис. 1. Образование сечений конической поверхности

в виде парабол можно использовать для 3D-моделирования таких поверхностей, как, например, гиперболический и эллиптический параболоиды.

Напомним, при каких условиях можно получить необходимую кривую второго порядка — параболу [3, 4].

Если плоскость параллельна только одной образующей конуса, то коническое сечение есть парабола (рис. 1).

Для получения в КОМПАС-3D нужной для 3D-моделирования кривой создается двуполостный конус вращения с помощью операции **Вращения**.

Порядок создания конуса и конических сечений. Создаем новый документ *Деталь*, в дереве модели присваиваем детали имя **Конус** и сохраняем файл под этим именем.

1. За плоскость эскиза выбираем фронтальную плоскость проекций (XY).

2. Выбираем команду **Эскиз** (для создания эскиза).

3. Создаем эскиз конуса, для этого (рис. 2):

— проводим ось вращения вертикально через начало координат командой **Отрезок** со стилем линии *Осевая* (рис. 2, а);

— строим прямую через начало координат под произвольным углом (в представленном примере — 60°) к оси X, используя команду **Вспомогательная прямая** (рис. 2, а);

— проводим границы верхней и нижней части конуса прямыми, проведенными на произвольно заданном расстоянии друг от друга (в представленном примере задано расстояние 100 мм от центра координат вверх и вниз), используя инструмент **Горизонтальная прямая** со стилем линии *Тонкая*;

— проводим образующую конуса по вспомогательной прямой от верхней до нижней границы командой **Отрезок** со стилем линии *Основная* (рис. 2, б);

— у зоны начала координат делаем разрыв образующей конуса командой **Усечь кривую двумя точками** таким образом, чтобы как минимум один отрезок не касался оси вращения и ни один из них не пересекал ее (в представленном примере отрезок пересекался на 0,2 мм (рис. 2, б). В противном случае 3D-модель двуполостного конуса создаваться не будет.

4. Выходим из режима **Эскиз**.

5. Выбираем команду **Операция вращения** 3D-моделирования и командой **Создать объект** получаем результат (рис. 3).

Чтобы получить изображение параболы, создадим ассоциативный чертеж конуса. Командами **Создать чертеж** и **Менеджер документа** заготавливаем формат А2 чертежа и сохраняем его под именем **Конус**. В панели инструментальных средств **Виды** выбираем команду **Стандартные виды**. В опциях этой команды выбираем изображения **Главного вида** и **Вида сверху**. Размещаем данные проекции в левой части формата чертежа (рис. 4). Теперь на ассоциативном чертеже можно построить любые сечения.

Построение параболы. Для этого след плоскости сечения должен быть параллелен левой или правой образующей конуса на главном виде. На рис. 4 следы А—А и Б—Б секущих плоскостей параллельны правой образующей $S''-I''$ конуса. Сравнивая

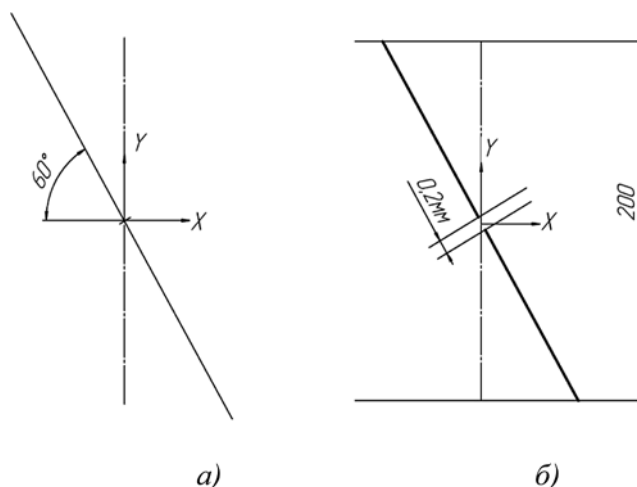


Рис. 2. Построение эскиза конуса в программе КОМПАС-3D

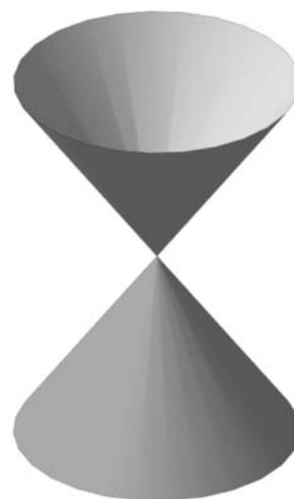


Рис. 3. Поверхность кругового конуса

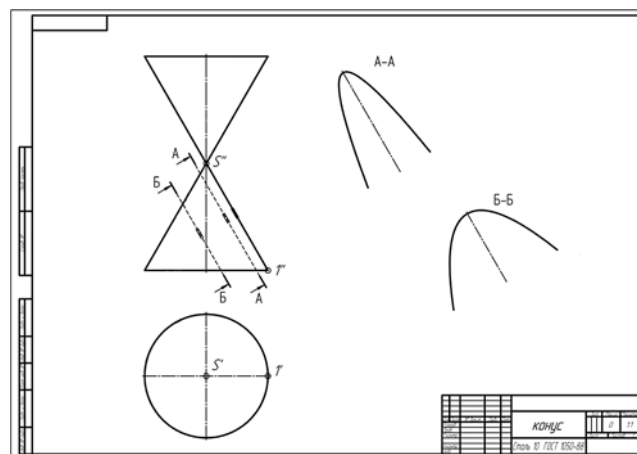


Рис. 4. Сечения поверхности конуса на ассоциативном чертеже

эти параболические сечения, видим, что чем ближе секущая плоскость к $S''-I''$, тем парабола уже.

В графическую программу Компас-график более поздних версий, начиная с v. 15.1, введен новый инструмент "Коническая кривая", позволяющий упростить процесс построения параболических кривых. Однако применение данного инструмента в частных случаях затрудняется тем, что необходимо знать ряд параметров, таких как положение точки пересечения касательных (т. 3 в программе Компас-график), определяемой углами наклона касательных к конической кривой, формой кривой, определяемой положением точки т. 4 в программе Компас-график или параметрами высота/коэффициент, что также способствует сложности построения. Кроме того, малейшая ошибка симметрии параболы, вызванная случайным сдвигом центра пересечения касательных, делает невозможным построение заданного геометрически верного параболоида, что исключено при использовании вышеуказанного метода получения параболы путем рассечения прямого кругового конуса секущей плоскостью параллельной одной из образующих.

Рассмотрим примеры использования параболы при 3D-моделировании.

Алгоритм создания гиперболического параболоида. Гиперболический параболоид можно представить как поверхность, полученную при перемещении первой параболы так, чтобы ее вершина скользила по второй параболе. Плоскости парабол взаимно перпендикулярны. В КОМПАС-3D этот процесс выполняется с помощью команды **Кинематическая операция**.

Создаем новый документ *Деталь*, в *Дереве модели* присваиваем детали имя **Гиперболический параболоид** и сохраняем файл под этим именем.

Шаг 1. За плоскость эскиза выбираем фронтальную плоскость проекций XU .

Шаг 2. Выбираем команду **Эскиз** и создаем эскиз параболы-траектории:

- открываем чертеж конуса с сечениями и командой **Выделить объект** выделяем параболу Б—Б и ее ось (рис. 4);
- командой **Копировать** из меню **Редактор** копируем выделенные объекты в буфер обмена с привязкой к вершине кривой; возвращаемся к документу **Гиперболический параболоид** и командой **Вставить** вставляем эскиз параболы с привязкой к началу координат;
- выделяем ось и параболу и командой **Поворот** поворачиваем фигуру вокруг начала координат так, чтобы ось параболы совпала с осью Y , а ветви параболы были направлены вверх. Затем выделяем параболу с ее осью и копируем ее в буфер обмена с привязкой к началу координат. После этого командой **Масштаб** увеличиваем размеры параболы в два раза;
- выходим из режима **Эскиз**.

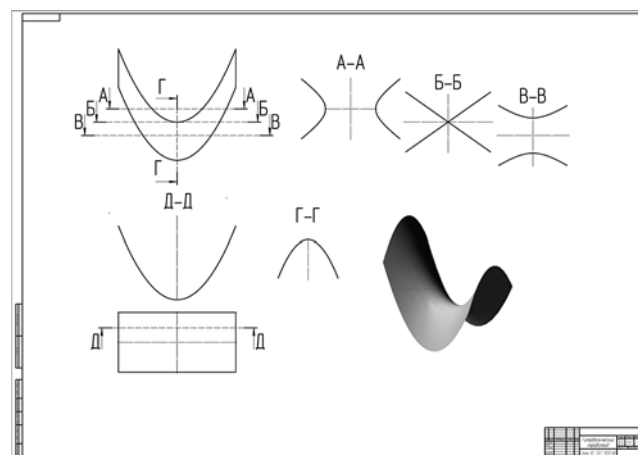


Рис. 5. Создание гиперболического параболоида

Шаг 3. Командой **Плоскость через вершину параллельно другой плоскости** из панели инструментов **Вспомогательная геометрия** создаем плоскость, параллельную плоскости ZY (профильная плоскость проекций) и проходящую через конечную точку параболы первого эскиза.

Шаг 4. Выделяем созданную параллельную плоскость.

Шаг 5. Выбираем команду **Эскиз** и создаем эскиз параболы-сечения:

- командой **Вставить** вставляем параболу с привязкой к началу координат;
- выходим из режима **Эскиз**.

Шаг 6. Командой **Кинематическая операция** создаем гиперболический параболоид.

На рис. 5 показан ассоциативный чертеж гиперболического параболоида с сечениями разного уровня и с 3D-изображением этой поверхности. Из полученных сечений видно, что если секущие плоскости параллельны оси гиперболического параболоида, т. е. параллельны профильной ($G-G$) или фронтальной ($D-D$) плоскости проекций, то в сечении будут параболы. Если секущие плоскости перпендикулярны к оси поверхности, то когда плоскость находится над вершиной параболоида ($A-A$) или под вершиной параболоида ($B-B$), то в сечениях будут гиперболы, действительные оси которых взаимно перпендикулярны. Если же плоскость проходит через вершину поверхности, то гипербола распадается на две прямые.

Использование канонических уравнений параболы

В КОМПАС-3D среди множества библиотек имеется библиотека **FTDraw**, входящая в меню библиотек **Прочие**. С помощью этой библиотеки по каноническому уравнению кривой можно построить график функции, т. е. изображение кривой параболы по заданным условиям. Затем полученное изображение можно использовать для 3D-моделирования поверхностей второго порядка.

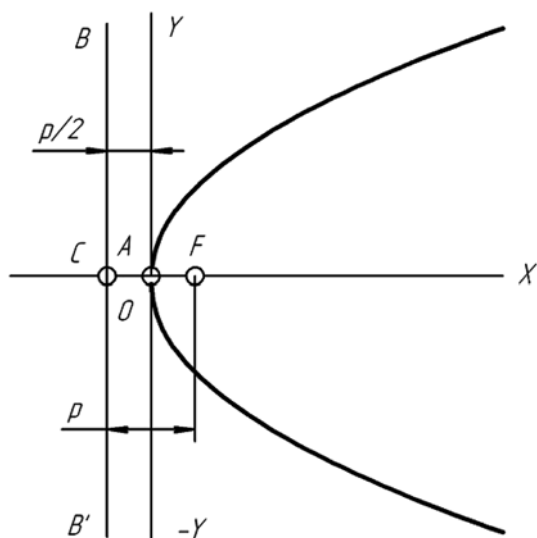


Рис. 6. Парабола

Геометрия и каноническое уравнение параболы. Элементы параболы (рис. 6): кривая параболы; A — вершина параболы, она совпадает с началом O координат; BB' — директриса параболы; F — фокус параболы.

Параметры параболы: $FC = p$ — фокальный параметр, т. е. расстояние от фокуса F до директрисы BB' .

Каноническое уравнение параболы:

$$y^2 = 2px. \quad (1)$$

Чем больше величина p , т. е. чем дальше фокус от вершины, тем шире раствор ветви параболы.

Сначала перепишем канонические уравнения (1) в другой форме, т. е. выразим для параболы зависимость координаты y от x :

$$y = \sqrt{2px}. \quad (2)$$

Теперь, задав значение фокального параметра p кривой и подставив его в формулу (2), можно получить ее изображение на экране монитора с помощью команд из библиотеки **FTDraw**.

Алгоритм создания параболы. Открываем документ **Фрагмент** и присваиваем ему имя **Парабола**. Назначим для параболы параметр p (в примере его параметр p равен 10 мм). В этом случае вершина параболы будет совпадать с началом координат, а по оси X на расстоянии 5 мм вправо будет расположен фокус параболы (рис. 6). Тогда по формуле (2) $y = \sqrt{2px}$.

Приступаем к работе с библиотекой **FTDraw** (рис. 7). Обращаемся к командам: **Менеджер библиотек** (📁) → **Про-**

чие (📁 Прочие). В списке библиотек **Прочие** щелкаем курсором на **Библиотека FTDraw**.

В диалоговом окне (ДО) **Библиотека FTDraw1.1** (рис. 7, а) выполняем двойной щелчок мышью на

кнопке  (**Построение графиков функциональных зависимостей по уравнению в декартовых координатах**). Появляется новое ДО **Построение графика функциональных зависимостей** (рис. 7, б).

Используя форму записи формул, принятую в системе КОМПАС-3D, записываем в этом ДО уравнения параболы $\text{Sqrt}(2 \cdot 10x)$ (рис. 7, б).

Заказываем пределы изменения x (в примере установлены пределы от 10 до 40 мм) и число точек кривой (для обеспечения более высокой точности графика в примере выбрано 30 точек). В ДО (рис. 7, в) **Построение графиков функциональных зависимостей** щелкаем на кнопке **Просмотр расчета** (🖨️). Если расчет выполнен по заданной формуле и при заданных границах, то просто закрываем ДО **Просмотр результатов расчета**.

В ДО **Построение графиков функциональных зависимостей** (рис. 7, в) щелкаем на кнопке **Указать положение базовой точки графика** (📍).

На экране появляется контекстное меню

 Ввод базовой точки графика. Курсором привязываемся к началу координат документа **Парабола**. Затем в ДО

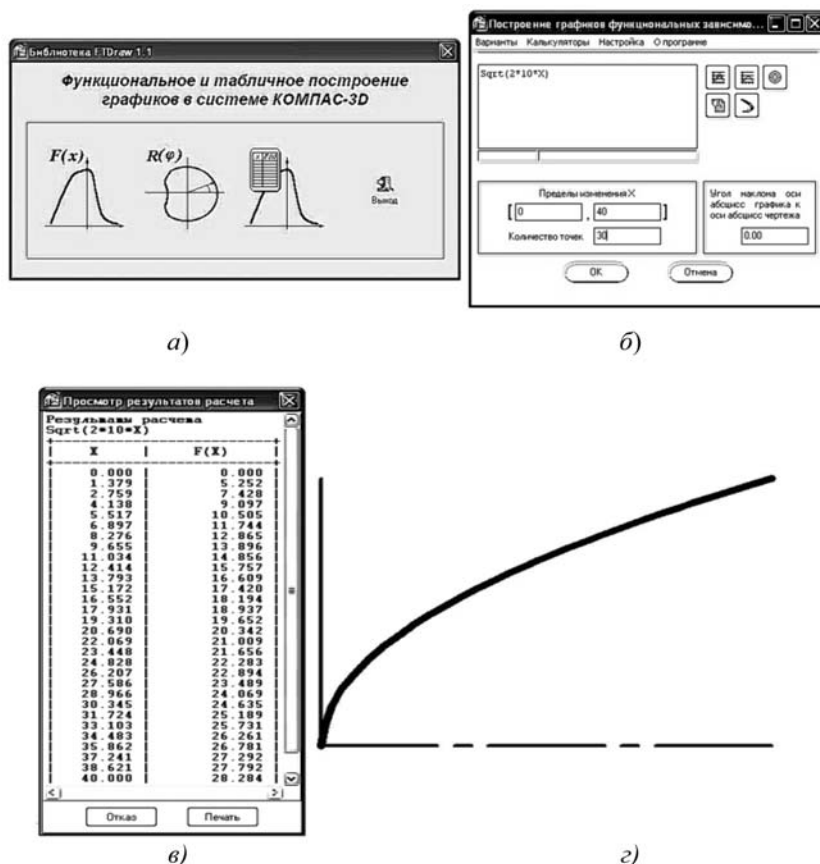


Рис. 7. Построение параболы в программе КОМПАС-3D

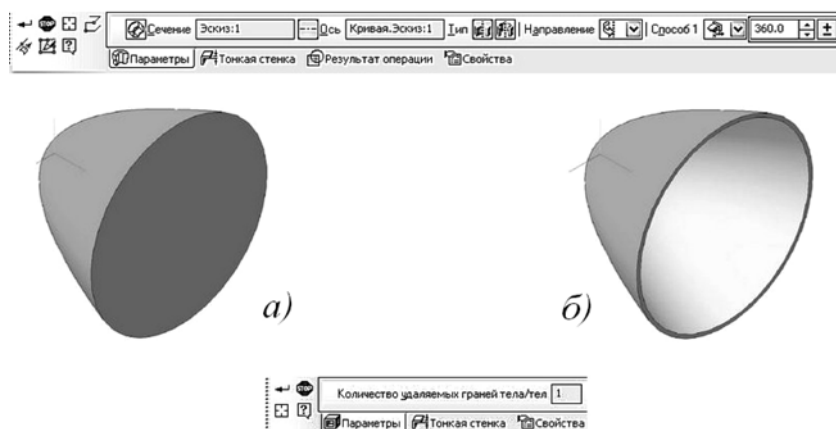


Рис. 8. Построение эллиптического параболоида

(рис. 7, в) щелкаем по кнопке **Построить график** () и по кнопке **ОК**. В результате получаем изображение верхней половины ветви параболы (рис. 7, г). Закрываем диалоговые окна.

Это изображение части параболы создано системой в виде **Макроэлемента**. Чтобы его использовать для 3D-моделирования, выделяем кривую и командой **Разрушить** из меню **Редактор** преобразуем ее в **кривую Безье**, а через вершину кривой проводим ось симметрии кривой командой **Отрезок** с типом линии **Осевая**. Выделяем фигуру командой **Выделить** и копируем ее в буфер обмена с помощью команды **Копировать все** с привязкой к вершине кривой.

Алгоритм создания эллиптического параболоида. Создаем новый документ **Деталь**, в **Дереве модели** присваиваем детали имя **Параболоид** и под этим именем сохраняем файл.

Шаг 1. За плоскость эскиза выбираем фронтальную плоскость проекций XU .

Шаг 2. Выбираем команду **Эскиз** (для создания эскиза параболы) и командой **Вставить** вставляем эскиз параболы с привязкой к началу координат.

Шаг 3. Выходим из режима **Эскиз**.

Шаг 4. Обращаемся к команде **Операция вращения**. На рис. 8 показана **Панель свойств** команды с назначенными параметрами. Опцией команды **Создать объект** () создаем 3D-модель параболоида (рис. 8, а).

Можно, например, создать из параболоида оболочку с толщиной стенки 1 мм (рис. 8, б). Для этого на **Панели свойств** выбираем команду **Оболочка** ().

Использование параметрических уравнений параболы

В графическом редакторе КОМПАС-3D при работе с документом **Деталь** в меню **Пространственные кривые** () имеется команда **Кривая по закону** (). Эта команда позволяет получить изображение кривой второго порядка, заданной в парамет-

рической форме в пространственной прямоугольной системе (x, y, z) координат. Так как мы имеем дело с плоской кривой, то при задании координат X и Y в параметрической форме координате Z необходимо задать значение, равное нулю.

Чтобы использовать эту кривую для моделирования поверхности второго порядка, необходимо спроецировать ее на плоскость XU эскиза. Созданные в эскизе проекции **Кривой по закону** ассоциативно связаны со своими исходными объектами. Например, при редактировании параметров кривой будет изменяться и эскиз. Это проекционная связь не дает возможность редактировать эскиз на месте.

Удаление этой проекционной связи возможно двумя путями: *вырезать созданный эскиз в буфер и обратно вернуть плоскость эскиза, либо после создания эскиза удалить **Кривую по закону***.

Параметрическое уравнение параболы. Используя формулу (2) канонического уравнения параболы, введем следующие параметрические зависимости:

$$x = t^2; y = t\sqrt{2p}; z = 0. \quad (3)$$

Это и есть параметрическое уравнение параболы, расположенной в плоскости XU , при этом параметр t определяет фактически длину параболы по оси X , параметр p — ширину раствора параболы.

Алгоритм создания гиперболического параболоида. Для создания гиперболического параболоида нужны две параболы: одна в роли направляющей, а вторая — в роли образующей. Их плоскости должны располагаться взаимно перпендикулярно.

Открываем документ **Деталь**. В **Дереве модели** с помощью команды **Свойства модели** присваиваем файлу имя **Гиперболический параболоид** и сохраняем файл под этим именем.

Параметрическое уравнение зададим в прямоугольной (x, y, z) системе координат. Для создания параболы введем, например,

- интервал изменения параметра — $t = [0; 7]$;
- расстояние от фокуса параболы до директрисы — $p = 7$.

Для этого активизируем инструментальную панель **Пространственные кривые** (). В этой панели выбираем команду **Кривая по закону** () и в **Панели свойств** команды, представленной в виде двух строк (рис. 9), записываем $x = t^2$; $y = t\sqrt{2p}$; $z = 0$. Завершаем работу команды опцией **Создать объект** ().

Шаг 1. За плоскость эскиза выбираем фронтальную плоскость проекций XU .

Шаг 2. Выбираем команду **Эскиз** (для создания эскиза параболы) и:

- командой **Спроецировать объект** (из меню **Геометрия**) создаем проекцию параболы с **Кривой по закону**;
- командой **Выделить** выделяем эскиз параболы;
- командой **Вырезать** убираем эскиз параболы в буфер обмена;
- командой **Вставить** вставляем эскиз параболы с привязкой к началу координат (рис. 10, а). В результате связь эскиза с кривой по закону удалена;
- проводим ось симметрии (рис. 10, б) и выделяем половину параболы, командой **Симметрия** создаем ее вторую половину;
- командой **Поворот** поворачиваем параболу на угол 90° , получаем эскиз первой параболы (рис. 10, в).

Шаг 3. Выходим из режима **Эскиз**.

Выполняем создание эскиза второй параболы по аналогичному алгоритму, только с другими данными (рис. 11, а—в):

- интервал изменения параметра $t = [0; 7]$,
- расстояние от фокуса параболы до директрисы $p = 3$.

Если при заполнении **Строки свойств** команды **Кривая по закону** невозможно изменить величину p , то можно изменить имя p , например, на имя k и после этого в **Строке свойств** задать значение k (в примере значение $k = 3$) (рис. 12).

Эскиз первой параболы будет служить направляющей гиперboloида в команде **Кинематическая операция**. Для эскиза второй параболы необходима плоскость, перпендикулярная к плоскости XU и проходящая через конечную точку первой параболы, созданную в эскизе 1.

Командой (**Плоскость через вершину параллельно другой плоскости**) из меню **Вспомогательная геометрия** создаем плоскость (**Параллельная плоскость 1**), проходящую через конец параболы и параллельную профильной плоскости проекций ZU .

Командой **Редактировать** эскиз 2 открываем изображение второй параболы, командой **Выделить** выделяем изображение, командой **Вырезать** переносим эскиз с привязкой к началу координат в буфер обмена. Закрываем пустой эскиз 2.

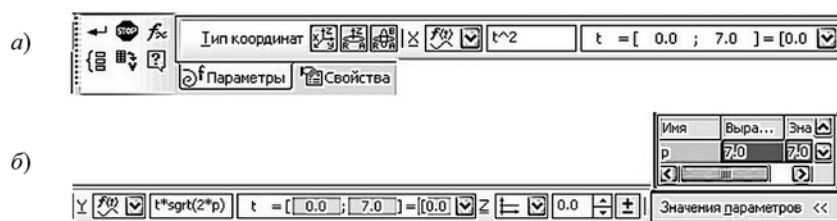


Рис. 9. Фиксация параметрического уровня параболы в программе КОМПАС-3D:

а — верхняя панель свойств команды **Кривая по закону**; б — нижняя панель свойств команды **Кривая по закону**

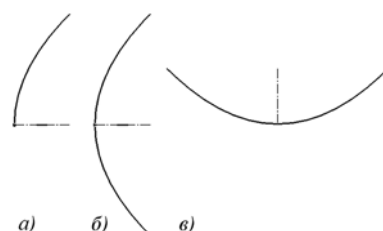


Рис. 10. Построение эскиза первой параболы

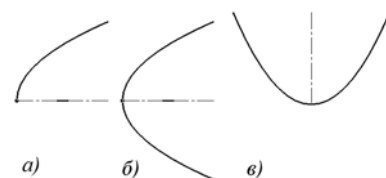


Рис. 11. Построение эскиза второй параболы



Рис. 12. Введение данных для построения эскиза второй параболы

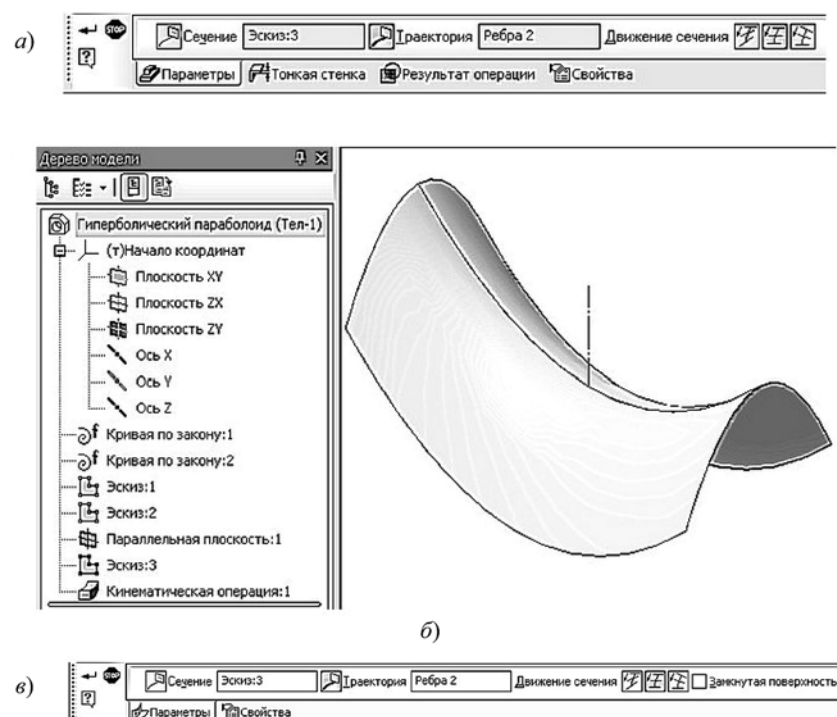


Рис. 13. Создание гиперболического параболоида

Выделяем созданную параллельную плоскость 1 и создаем в ней эскиз 3, т. е. из буфера обмена вставляем эскиз 2 с привязкой к началу координат. Закрываем эскиз 3.

Выбираем команду () **Кинематическая операция** (рис. 13, а). В качестве эскиза-сечения выделяем эскиз 3, а в качестве траектории выделяем эскиз 1, завершаем работу опцией  **Создать объект** (рис. 13, б).

На созданном изображении скрыты изображения **Кривой по закону 1** и **Кривой по закону 2**, а также конструктивные плоскости. Толщина гиперболического параболоида для примера назначена в размере 0,01 мм. Для получения изображения не тела, а поверхности можно воспользоваться командой **Кинематическая поверхность** (рис. 13, в) из инструментальной панели **Поверхности** ().

Заключение

В статье подробно изложены алгоритмы построения плоской кривой — параболы с использованием сечения кругового конуса, ее канонических и параметрических уравнений, а также моделирования производных 3D-поверхностей второго порядка в системе трехмерного моделирования. Раскрыты функциональные возможности программного продукта КОМПАС-3D при решении

задач по созданию таких поверхностей второго порядка, как эллиптический и гиперболический параболоиды.

Представленные методы позволяют создавать наглядные модели, которые можно показать на экране с разных сторон путем вращения или выбором ориентации модели относительно плоскостей проекций, что способствует развитию пространственного воображения.

Данная работа направлена на более глубокое изучение цикла геометро-графических дисциплин и продвижение отечественного IT-продукта в сфере образования. Она будет интересна студентам, преподавателям и специалистам в области геометрического моделирования.

Список литературы

1. **Дорохов А. С.** Компьютерное проектирование в системе AutoCAD. М.: Российский государственный аграрный университет — МСХА им. К. А. Тимирязева, 2016. 80 с.
2. **Дорохов А. С.** Выполнение чертежей с использованием системы "Компас-3D". М.: Российский государственный аграрный университет — МСХА им. К. А. Тимирязева, 2016. 76 с.
3. **КОМПАС-3D V.13:** руководство пользователя. В 3-х т. М.: Изд-во ИТАР-ТАСС, 2012. Т. 3. 424 с.
4. **Иванов Г. С.** Теоретические основы начертательной геометрии: учеб. пособие. М.: Машиностроение, 1998. 160 с.
5. **Фролов С. А.** Начертательная геометрия: учеб. для вузов. 3-е изд., перераб. и доп. М.: Машиностроение, 2008. 240 с.

V. V. Burlov, Professor, e-mail: vladimir-burlov@yandex.ru,

L. V. Remontova, Assistant Professor, e-mail: remontova@mail.ru,
Penza State Technological University, Penza,

V. V. Kosolapov, Assistant Professor, e-mail: vladimir.kosolapov@mail.ru,

E. V. Kosolapova, Senior Lecturer, e-mail: K-art-inka@ya.ru,
Nizhny Novgorod State University of Engineering and Economics, Knyaginino

Simulation of the Curve and Surfaces of the Second Order in the KOMPAS-3D System

The article focuses on algorithms for constructing the curve of a parabola and derivatives of second-order surfaces from it using the functional capabilities of the KOMPAS-3D software product, which allows solving various educational and professional engineering-graphic tasks. However, in the graphical editor of this program there is no Parabola command, without which it is impossible to construct surfaces such as elliptic and hyperbolic paraboloids that occur both in the design of simple and complex models.

In connection with the goal in the paper, the methods of solving spatial problems related to the modeling of second-order surfaces — elliptic and hyperbolic paraboloids in the system of three-dimensional modeling KOMPAS-3D — are described in detail. The methods are based on the use of: the section of a circular cone, the canonical and parametric equations of parabolas. For each of the presented variants, examples of creating 3D models of surfaces of the second order are considered. On the basis of the laws of descriptive geometry, applicable to surface problems, the feature of their solution and display on a computer screen is clearly shown.

The article is aimed at a deeper study of geometer-graphic disciplines and promotion of the domestic IT-product in the field of education. The work will be interesting for students, postgraduates, teachers and specialists in the field of geometric modeling.

Keywords: associative drawing, library, functional dependency graph, Hyperbolic paraboloid, cone, tapered section, canonical and parametric equations, curve, parabola, surface, KOMPAS-3D system

References

1. **Dorohov A. S.** *Komp'yuternoe proektirovanie v sisteme AutoCAD* (Computer designing in AutoCAD system). Moscow: Rossijskij gosudarstvennyj agrarnyj universitet — MSHA im. K. A. Timirjazeva, 2016. 80 p. (in Russian).
2. **Dorohov A. S.** *Vypolnenie chertezhej s ispol'zovaniem sistemy "Kompas-3D"* (Execution of drawings using the system "Kompas-3D"). Moscow: Rossijskij gosudarstvennyj agrarnyj universitet — MSHA im. K. A. Timirjazeva, 2016. 76 p. (in Russian).

3. **KOMPAS-3D V13: rukovodstvo pol'zovatelya v 3-kh t** (KOMPAS-3D V13: User's manual in 3 tons), Moscow, ITAR-TASS, 2012, vol. 3, 424 p. (in Russian).

4. **Ivanov G. S.** *Teoreticheskie osnovy nachertatel'noi geometrii* (Theoretical Foundations of Descriptive Geometry), Moscow, Mashinostroenie, 1998. 160 p. (in Russian).

5. **Frolov S. A.** *Nachertatel'naya geometriya* (Descriptive geometry), Moscow, Mashinostroenie, 2008. 240 p. (in Russian).

УДК 004.94

В. И. Анисимов, д-р техн. наук, проф., гл. науч. сотр., e-mail: vianisimov@inbox.ru,
Центр информационных технологий в проектировании РАН, г. Одинцово,
С. А. Васильев, аспирант, e-mail: venom-gt@list.ru,
Санкт-Петербургский государственный электротехнический университет,
Ю. И. Журавлев, академик, зам. директора по научной работе,
Вычислительный центр им. А. А. Дородницына ФИЦ ИУ РАН

Методы построения клиентских приложений схемотехнических САПР на основе классов наследования

Рассматриваются проблемы обеспечения быстродействия пользовательского интерфейса схемотехнических САПР. Описываются преимущества использования JavaScript стандарта ECMAScript 2016 г. для построения веб-интерфейса пользователя. Указывается необходимый функционал интерфейса веб-сервиса. Проводится сравнительный анализ существующих востребованных технологий построения веб-интерфейса для схемотехнической САПР. Приводится архитектура веб-приложения, построенная с применением технологии JavaScript на основе вложенных структур. Описываются принципы наследования от эталонного класса для создания структур вложенного типа.

Ключевые слова: схемотехническая САПР, JavaScript, ECMAScript, iframe, пользовательский интерфейс, обмен данными, сложноструктурированные данные, вложенные структуры, эталонный класс

Введение

Построение клиентских приложений схемотехнических САПР сопряжено с рядом проблем, связанных с обеспечением: возможности ввода большого количества информации об электронной схеме, быстродействия и интерактивности, предоставляемой пользователю, поддержки высокоскоростных протоколов передачи данных для обмена с сервером, вложенности функциональных компонентов расчетной системы, а также возможности их модернизации [1]. Стоит отметить высокую актуальность решения указанных проблем в разрезе специфики предметной области схемотехнических САПР, характеризующихся значительным количеством сложноструктурированных данных и интегрированием функциональных компонентов в более сложные структуры.

Решение данных проблем возможно путем применения в разработке клиентского приложения JavaScript-классов, формирующих и модифицирующих компоненты объектной модели документа

HTML и полнодуплексного протокола обмена данными WebSocket [2]. Язык JavaScript, бесспорно, стал незаменимым инструментом для разработчиков клиентской составляющей, при этом интерфейс пользователя может быть частично либо полностью организован за счет JavaScript. Основой для многократного использования компонентов клиентского приложения, легкой модернизации и уменьшения объема передаваемых интерфейсных данных является модульная структура, обусловленная использованием JavaScript-классов для создания компонентов объектной модели. Особенно удобной такая реализация стала с выходом стандарта ECMAScript 7 в 2016 г., изменения которой были внесены в стандарт и уже поддерживались основными браузерами на основании черновых спецификаций. Шестой и более полно седьмой стандарты ECMAScript имеют поддержку классов и делегирующее наследование на основе прототипов, что является особенно важным для создания вложенных структур клиентского приложения схемотехнической САПР с применением свойств кон-

структура класса функционального элемента, с возможностью добавления собственных свойств для каждого класса наследника [3].

При создании проекта клиентского приложения можно проследить, как соединяется цепь прототипов объекта, созданная классом. Когда создается класс, появляется и новая функция, соответствующая конструктору в дефиниции (определении) класса, содержащая все статические методы. Также создается объект, который будет свойством *property* у созданной функции и будет содержать все методы экземпляра класса. Для создания нового класса, который наследует все статические свойства наследования объект-функции от объект-функции суперкласса, заключительным этапом является построение объекта *prototype* новой функции и наследование суперкласса для обеспечения наследования методов экземпляра класса.

Описание функционала веб-интерфейса

Для многопользовательского удаленного проектирования электронных схем в специализированной распределенной среде требуется обеспечить клиентское приложение средствами аутентификации и регистрации пользователей. Также нужно обеспечить индикацию подключения к серверу, идентификацию и права пользователя (см. рисунок).

Схемотехническое проектирование подразумевает ввод большого количества данных, таких как различные простейшие компоненты и интегрированные системы электронной цепи, директивы расчета. Также следует предусмотреть загрузку параметров из базы данных клиент-серверного приложения и ввод данных из файла пользователя для всех типов компонентов схемы, для чего требуется предусмотреть отдельные поля на сайте, так как в противном случае восприятие интерфейса будет затруднительно. Есть несколько вариантов представления для пользователя таких полей, например,

на различных страницах клиентского приложения либо на вложенных *iframe*-структурах, либо на основе использования JavaScript-аппарата для формирования интерфейса. Использование отдельных страниц для каждого поля приведет либо к отсутствию полной информации о выполняемом проекте, либо к большому числу обращений к серверу [4]. Применение структуры *iframe* очень удобно для разработки и модернизации проекта, но имеет ряд недостатков:

- отсутствие поддержки кросс-доменной адресации на уровне технологии, что весьма актуально для распределенного приложения;
- затруднение доступа окна, загруженного во фрейм к родительскому окну в большинстве современных браузеров;
- сложность навигации пользователя по сайту;
- отсутствие возможности индексации поисковыми системами, поскольку на страницах, которые содержат контент, как правило, отсутствуют ссылки на другие документы;
- невозможность добавления внутренних страниц, так как фреймы скрывают адрес страницы, на которой находится посетитель, и всегда показывают только адрес сайта;
- повторная загрузка страницы может привести к потере введенных пользователем данных.

В свою очередь разработка клиентского приложения, в основе предполагающая формирование интерфейса пользователя полностью из объектно-ориентированного JavaScript-кода, наиболее полно подходит для организации схемотехнической системы автоматизированного проектирования по нескольким причинам. Во-первых, это позволяет экономить на объеме передаваемой служебной информации (интерфейс-образующей), во-вторых, позволяет повысить интерактивность системы.

Уменьшение объема передаваемой информации достигается путем модернизации единственной страницы, загруженной пользователем, при этом удается избежать дублирования кода и описания свойств объектов за счет их преемственного наследования от эталонного класса, которым может являться родительский класс. Повышение интерактивности достигается тем, что шаблон страницы формируется при загрузке за счет построения классовой иерархии, поэтому любая предоставленная пользователем информация может быть доступна для чтения и изменения в любом поле (псевдоокне) клиентского приложения. Таким образом, получается, что приложение имеет большое число окон, к которым имеет доступ пользователь, при этом одновременно в браузере отображаются только одно окно и корневые уз-

Веб-интерфейс клиентского приложения

лы древовидной иерархии навигации по интерфейсу. Также стоит отметить, что использование классов для описания объектов позволяет разработчику удобно администрировать и модернизировать данное клиентское приложение.

Архитектура клиентского веб-приложения

При загрузке страницы приложения пользователь получает пустую HTML-страницу с подключенным JavaScript кодом, формирующим интерфейс. Код получает доступ к тэгу *body* объектной модели документа, и в интерпретируемом коде формируется шаблон страницы на основе классов, отражающих свойства и поведение будущих объектов:

```
engine = new _engine();
window.onload = function () {
    var place = document.getElementsByTagName("body");
    place = place[0];
    place.appendChild(engine.dom);
};
```

Для создания всех видимых и внутренних элементов объектной модели HTML-документа создается специальный класс, определяющий иерархию вложенности путем получения ссылки на родительский элемент в качестве параметра. Этот класс можно считать основоположником клиентского приложения, так как он в прямом смысле создает новые объекты путем частичного самокопирования [5]. Также все классы наследуют конструктор данного эталонного класса, оставляя за собой возможность добавления в конструктор дополнительных параметров:

```
class _elem {
    constructor(parent, type) {
        this.dom = document.createElement(type);
        if (parent) parent.dom.appendChild(this.dom);
    }
}
```

Ярким примером наследования конструктора эталонного класса с привнесением собственных уникальных свойств (тэг
) является класс, определяющий в проекте перевод строки:

```
class _br extends _elem {
    constructor(parent) {
        super(parent, "br");
    }
}
```

Вызов экземпляра определяется классом, указывающим на родительский элемент с помощью команды: "this.Br = new _br(this)". Таким образом, эталонный класс упрощает задачу наследования и позволяет вкладывать любую структуру в родительский элемент.

Класс, отвечающий за псевдоокно авторизации пользователя, имеет несколько вложенных элементов:

```
class _authorisation extends _elem {
    constructor(parent) {
        super(parent, "div");
        this.count = 0;
    }
    create_div(){
        if(this.count == 0){
            this.dom.setAttribute("class", "div2");
            this.dom.setAttribute("id", "Authorisation");
            this.Login = new _input(this, "Login", "Логин");
            this.Br = new _br(this);
            this.Password = new _input(this, "Password", "Пароль");
            this.Br = new _br(this);
            this.Enter = new _button(this, "Войти");
            this.count ++;
        }
        else{
            this.dom.setAttribute("style", "display: block");
            this.count ++;
        }
    }
}
```

На основе данного принципа формируется весь пользовательский интерфейс разрабатываемого программного обеспечения. Шаблон формируется конструкторами классов, модификацию элемента определяют методы класса, оформление осуществляется путем применения каскадных таблиц стилей. Предлагаемая архитектура формирования клиентского приложения на основе JavaScript-классов предоставляет возможности для формирования интерфейса пользователя из большого числа вложенных структур.

Выводы

Применение подхода формирования пользовательского интерфейса посредством JavaScript-кода позволяет разрешить ряд актуальных проблем, определяемых спецификой предметной области разработки клиентских приложений для распределенных схемотехнических систем автоматизированного проектирования. Такой спецификой являются:

- возможность ввода большого количества информации об электронной схеме;
- быстродействие и интерактивность, предоставляемая пользователю;
- вложенность функциональных компонентов расчетной системы;
- возможность модернизации [6].

Следует, однако, отметить, что данная методология имеет ряд недостатков, например, отсутствие возможности индексации страниц поисковыми системами, так как по сути приложение имеет всего одну страницу. Однако необходимость иметь производительный компьютер для интерпретации

"толстого" клиентского приложения разгружает ответные серверы для обеспечения большей пропускной способности системы в целом.

При использовании предлагаемого подхода достигается существенное уменьшение объема служебной информации, максимальная асинхронность и интерактивность интерфейса. При этом разработанные классы, определяющие элементы интерфейса, легко модернизируются, не требуют копирования кода, а также обеспечивают возможность их многократного использования, в том числе и в других проектах.

Работа выполнена по теме (проекту) "Исследование и разработка математического, информационного и программного обеспечения веб-сервисов распределенных систем автоматизации схемотехнического проектирования для моделирования больших интегральных схем в частотной и временной областях с учетом вариации параметров и внешних воз-

действий в условиях функционирования в гетерогенных средах" (грант 0071-2014-0007).

Список литературы

1. **Исаев И.** Актуальность использования веб-интерфейсов в современной информационной среде // Вестник Московского государственного университета печати. 2012. № 9. С. 81—85.
2. **Анисимов В. И., Гридин В. Н., Васильев С. А.** Методы построения схемотехнических систем автоматизированного проектирования с использованием сервис-ориентированного подхода на базе протокола WebSocket // Системы и средства информатики. 2016. № 2 (26). С. 136—146.
3. **Жуйков Р., Шарыгин Е.** Методы предварительной оптимизации программ на языке JavaScript // Труды Института системного программирования РАН. 2015. № 6. С. 67—86.
4. **Зотов В. А.** Реализация языка JavaScript ajax и node.js // Вестник Московского государственного университета печати. 2013. № 9. С. 46—47.
5. **Артюхова А. С.** Проектирование web-интерфейса средствами генетического программирования // Научное периодическое издание "IN SITU". 2016. № 11. С. 17—20.
6. **Воронцов Ю. А., Козинцев А. В.** Стандарты веб-сервисов для создания распределенных информационных систем // Век качества. 2015. № 3. С. 55—72.

V. I. Anisimov, D. Sc., Professor, Chief Researcher, e-mail: vianisimov@inbox.ru,
Design information technologies Center Russian Academy of Sciences, Moscow region, Odintsovo,
S. A. Vasilev, Graduate student, e-mail: venom-gt@list.ru,
Saint Petersburg State Electrotechnical University, St. Petersburg,
Yu. I. Zhuravlev, Academic, Deputy Director on Scientific Work,
Institution of Russian Academy of Sciences Dorodnicyn Computing Centre of RAS

Methods for Constructing Client Applications, Circuit CAD Class-Based Inheritance

The problems of ensuring the speed of the user interface of circuit design CAD are considered. The advantages of using the JavaScript standard ECMAScript 2016 for building a web user interface are described. The necessary functionality of the web service interface is indicated. Perform a comparative analysis of the existing demanded technologies for building a web-based interface for circuit design CAD. The architecture of the web application, built using JavaScript technology, is based on nested structures. The principles of inheritance from the reference class for creating nested structures are described.

Keywords: circuit design CAD, JavaScript, ECMAScript, iframe, user interface, data exchange, complex structured data, nested structures, reference class

References

1. **Isaev I.** Aktual'nost' ispol'zovaniya veb-interfejsov v sovremennoj informacionnoj srede (Relevance of use of web interfaces in the modern information environment), *Vestnik Moskovskogo gosudarstvennogo universiteta pechati (Bulletin of Moscow State University of the printing)*, 2012, no. 9, pp. 81—85 (in Russian).
2. **Anisimov V. I., Gridin V. N., Vasil'ev S. A.** Metody postroeniya shemotekhnicheskikh sistem avtomatizirovannogo proektirovaniya s ispol'zovaniem servis-orientirovannogo podhoda na baze protokola WebSocket (Methods of creation of circuitry automated design engineering systems with use service — the oriented approach on the basis of the WebSocket protocol), *Sistemy i sredstva informatiki (Systems and means of informatics)*, 2016, no. 2 (26), pp. 136—146 (in Russian).
3. **Zhujkov R., Sharygin E.** Metody predvaritel'noj optimizacii programm na jazyke JavaScript (Methods of preliminary optimization of programs in the JavaScript language), *Trudy Instituta sistemnogo programmirovaniya RAN (Works of Institute of system programming of RAS)*, 2015, no. 6, pp. 67—86 (in Russian).
4. **Zotov V. A.** Realizacija jazyka JavaScript ajah i node.js (Implementation of the JavaScript ajax and node.js language), *Vestnik Moskovskogo gosudarstvennogo universiteta pechati (Bulletin of Moscow State University of the printing)*, 2013, no. 9, pp. 46—47 (in Russian).
5. **Artjuhova A. S.** Proektirovanie veb-interfejsa sredstvami geneticheskogo programmirovaniya (Design of the web interface means of genetic programming), *Nauchnoe periodicheskoe izdanie "IN SITU" (Scientific periodical "IN SITU")*, 2016, no. 11, pp. 17—20 (in Russian).
6. **Voroncov Ju. A., Kozincev A. V.** Standarty veb-servisov dlja sozdaniya raspredelennykh informacionnykh sistem (Standards of web services for creation of the distributed information systems), *Vek kachestva (Century of quality)*, 2015, no. 3, pp. 55—72 (in Russian).

А. Н. Архангельский, инженер, e-mail: aleksandr.arhangel'skiy@bk.ru,
АО "Российские Космические Системы" Москва

Методы обработки топологических компонентов при проектировании и изготовлении электронной аппаратуры

Предлагаются методы обработки топологических компонентов (ТК), применяемые при проектировании и изготовлении печатных плат. Необходимость в подобных методах возникает в силу того, что объектами обработки и манипулирования в печатных платах являются ТК, а именно: контактные площадки, проводники, области металлизации и т. д. Под обработкой ТК понимается изменение его исходной геометрии в зависимости от функциональной необходимости. Предлагаемый метод распространяется на ТК, которые представляют собой односвязные области, охватываемые замкнутым контуром без самопересечений.

Ключевые слова: печатные платы, топология, контактная площадка, проводник, область металлизации, логическая операция, объединение, пересечение, вычитание

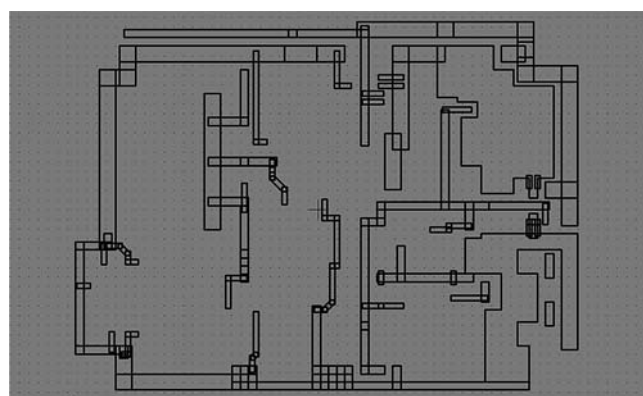
Введение

Для понимания того, что имеем в виду под обработкой топологических компонентов (ТК), обратимся к рис. 1 и 2.

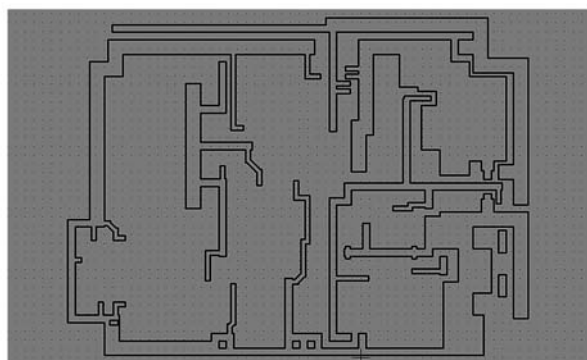
На рис. 1, а изображен фрагмент топологии печатной платы (ПП) после его создания в системе проектирования, например, в системе AltiumDesigner [1]. Изображенный фрагмент состоит из ТК,

а именно: прямоугольников, многоугольников (полигонов), линий и т. д., при этом один ТК может входить в состав другого, и наоборот. Такое изображение часто используется в ряде случаев. Однако некоторые задачи требуют дополнительной обработки данного изображения в силу ряда причин. Например, ЕСКД требует охвата областей металлизации единым полигоном. Так, для ТК, отображаемых в послыстных чертежах платы, желательно их представить в виде изображения, приведенного на рис. 1, б [2]. Как мы видим, исходная топология определенным образом переработана.

Достаточно большое число подобных задач имеет место при подготовке производства ПП, а также при разработке программ проведения их электро-



а)



б)

Рис. 1. Вид ТК:

а — после проектирования ПП; б — после их преобразования

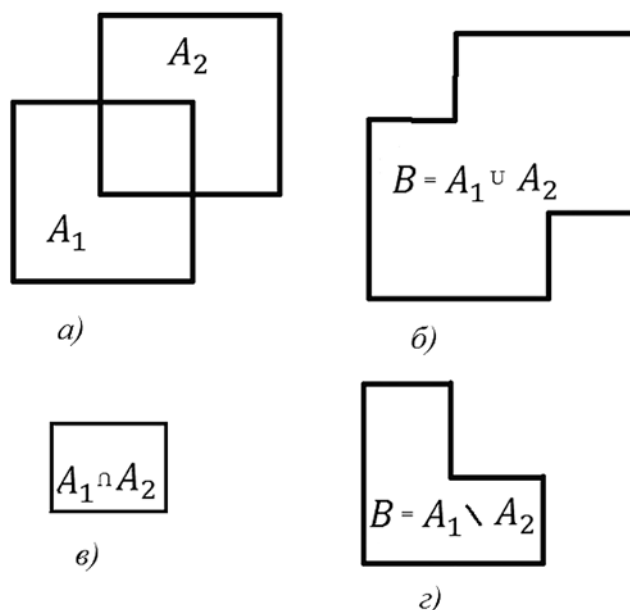


Рис. 2. Интерпретация семантики логических операций

Под ТК будем понимать односвязанную область, охватываемую замкнутым контуром без самопересечений и касаний. При этом контур состоит из векторов, расположенных на линиях, описываемых уравнениями первого порядка, т. е.

$$A = U_1^N [R_i(x_{i1}, y_{i1}), (x_{i2}, y_{i2})], \quad (1)$$

где R_i — i -й вектор; $(x_{i1}, y_{i1}), (x_{i2}, y_{i2})$ — координаты его начальной и конечной точек, при этом конечная координата R_i совпадает с начальной координатой $R_{(i+1) \bmod i}$.

Таким образом, под преобразованием ТК мы будем понимать построение описания ТК, зависящего от взаимного расположения компонентов и семантики преобразования.

Несмотря на то, что прикладные системы CAD/CAM имеют в своем составе достаточно большой набор преобразований ТК, методы их программной реализации не представлены в документации на системы. В настоящей статье вниманию читателей предлагается оригинальный метод выполнения преобразований над ТК.

Семантика преобразований, подлежащих рассмотрению в данной работе, представляет собой логические операции над множествами геометрических точек, входящих в ТК, а именно: "объединение", "вычитание", "пересечение". Классическая трактовка этих операций приведена на рис. 2. На рис. 2, а приведены два ТК A_1 и A_2 . Результаты логических операций "объединение", "пересечение", "вычитание" представлены на рис. 2, б–г.

Алгоритм преобразования ТК A_1 и A_2

Введем следующую систему обозначений, которую мы будем применять в дальнейшем. Для описания объектов из ТК A_m будем применять обозначение O_i^m , где m указывает на принадлежность к объекту A_m , i — номер объекта в A_m .

Алгоритм преобразования основан на исследовании взаимных расположений ТК A_1 и A_2 и построении на этой основе математической модели ТК, представляемой в виде

$$E = (G \cup V), \quad (2)$$

здесь G — геометрическая составляющая ТК A_1 и A_2 , имеющая вид $G = G^1 \cup G^2$, при этом $G^1 = U_1^N [R_i^1(x_{i1}, y_{i1}), (x_{i2}, y_{i2}) \cup L_i^1]$, где R_i^1 — описание векторов для ТК A_1 , L_i^1 — набор дополнительных параметров, описывающих вектор R_i^1 (полное описание L_i^1 будет дано ниже);

$$G^2 = U_1^M [R_j^2(x_{j1}, y_{j1}), (x_{j2}, y_{j2}) \cup L_j^2],$$

где R_j^2 — описание векторов для ТК A_2 ; L_j^2 — набор дополнительных параметров, описывающих вектор R_j^2 .

Составляющая V определяется взаимным расположением ТК и имеет вид $V = V^1 \cup V^2$, при этом $V^1 = U_1^N [(x, y)_i^1 \cup P_i^1]$, где $(x, y)_i^1$ — массив координат точек пересечения и касания вектора R_i^1 с ТК A_2 ; P_i^1 — набор дополнительных параметров, характеризующих взаимные расположения векторов (полное описание P_i^1 будет дано ниже).

Аналогичным образом определяем

$$V^2 = U_1^M [(x, y)_j^2 \cup P_j^2].$$

Число составляющих V_i^1 равно числу векторов R_i^1 , причем каждому вектору R_i^1 соответствует своя составляющая V_i^1 . Аналогичное соответствие имеет место и для A_2 .

В результате построения модели E векторы R_i^1 и R_j^2 будут разбиты на "однородные отрезки" (полное определение такого отрезка будет приведено ниже), координаты которых располагаются в массивах $(x, y)_i^1$ и $(x, y)_j^2$, при этом каждый отрезок имеет свой набор коэффициентов $P_i^1(L_j^2)$, отражающих его расположение относительно смежного ТК. Каждая семантика предполагает, что в результирующую фигуру должны входить отрезки с определенной принадлежностью к ТК с заданным набором сочетаний коэффициентов. Перечень этих сочетаний будем именовать как "шаблон". Его структура будет определена ниже.

Уравнение прямой, на которой располагается вектор [4] $R_i[(x_{i1}, y_{i1}), (x_{i2}, y_{i2})]$, проходящий через две точки $(x_{i1}, y_{i1}), (x_{i2}, y_{i2})$, имеет вид

$$\Omega_i(x, y) = -x(y_1 - y_2) - y(x_2 - x_1) + y_1x_2 - x_1y_2 = 0 \quad (3)$$

и называется *характеристическим уравнением прямой*.

Воспользуемся функцией, приведенной в работе [5], которая определяет положение точки (x, y) относительно вектора с характеристическим уравнением (3):

$$F(x, y) = \begin{cases} 2, & \text{если } \Omega_i(x, y) < 0; \\ 1, & \text{если } \Omega_i(x, y) = 0; \\ 0, & \text{если } \Omega_i(x, y) > 0. \end{cases} \quad (4)$$

Эта функция показывает, с какой стороны от вектора размещена точка. Если в уравнение (3) подставить координаты точки, расположенной справа от вектора, то значение $\Omega_i(x, y)$ будет > 0 ,

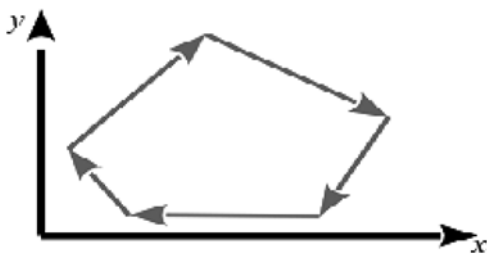


Рис. 3. Правосторонне ориентированный контур

при этом $F(x, y) = 0$; если точка расположена слева от вектора, то $\Omega_i(x, y)$ будет < 0 , и $F(x, y) = 2$; если точка располагается на векторе, то $\Omega_i(x, y) = 0$, а $F(x, y) = 1$. Нетрудно убедиться, что "внутренность" контура расположена справа от векторов, его описывающих (рис. 3). В терминах печатной платы "внутренность" есть не что иное, как область металлизации, охватываемая контуром.

Введем понятие "контактирования" векторов ТК. Два вектора будут "контактирующими" в следующих случаях: если они имеют точку пересечения и она принадлежит одновременно двум векторам; если они имеют точку пересечения и она является начальной/конечной для одного вектора и принадлежит другому вектору; если два вектора располагаются на одной прямой и имеют на ней общие точки. Если ТК имеют "контактирующие" векторы, то ТК также являются "контактирующими".

Основные признаки, которые будут определять взаимные расположения R_i^1 относительно ТК A_2 , выглядят следующим образом:

- вектор R_i^1 может полностью или частично находиться внутри ТК A_2 , при этом такое расположение будем именовать как "in" — R_i^1 внутри ТК A_2 ;
- вектор R_i^1 может располагаться на одной прямой с R_j^2 и иметь с ним общие точки, при этом такое расположение будем именовать как "dub" — R_i^1 касается ТК A_2 ;
- вектор R_i^1 может быть расположен вне ТК A_2 , такое расположение будем именовать "out" — R_i^1 вне ТК A_2 .

Приведенные определения справедливы и для расположения R_j^2 относительно ТК A_1 . Каждый из этих признаков для R_i^1 имеет свой набор данных в составляющей V_i^1 .

Полное множество S вариантов взаимных расположений векторов R_i^1 и R_j^2 , "контактирующих" друг с другом, приведено в таблице. Для отличия векторов R_i^1 и R_j^2 друг от друга в таблице вектор R_i^1 имеет толщину, в три раза большую, чем вектор R_j^2 . Множество S может быть представлено в следующем виде:

$$S = U_0^2[S^D].$$

где D — индекс, классифицирующий варианты взаимных расположений, основанный на вычислении функции (4) для конечных точек вектора R_i^1 по отношению к R_j^2 .

Пусть значение $F(x_i, y_i)$ для $R_i^1 [(x_{i1}, y_{i1}), (x_{i2}, y_{i2})]$

по отношению к прямой $\Omega_j^2(x, y)$ равны $F(x_{i1}, y_{i1}) = H$, $F(x_{i2}, y_{i2}) = T$.

Варианты взаимных расположений векторов

Номер варианта взаимного расположения	Расположение векторов: R_i^1 — толстый R_j^2 — тонкий	Расположение векторов: R_i^1 — толстый R_j^2 — тонкий	Сравнение расположения R_i^1 со следующими расположениями	Подмножество
1, 2			R_j^2	S^2
3, 4			R_j^2 и R_{j+1}^2	S^2
5, 6			R_j^2	S^1
7, 8			R_j^2	S^1
9, 10			R_j^2 и R_{j+1}^2	S^1
11, 12			R_j^2 и R_{j+1}^2	S^1
13, 14			R_j^2 и R_{j+1}^2	S^1
15, 16			R_j^2 и R_{j+1}^2	S^1
17, 18			R_j^2	S^0
19			R_j^2	S^0
20, 21			R_j^2	S^0
22			R_j^2	S^0
23, 24			R_j^2	S^0
25			R_j^2	S^0
26, 27			R_j^2	S^0
28			R_j^2	S^0
29, 30			R_j^2	S^0
31			R_j^2	S^0
32, 33			R_j^2	S^0
34			R_j^2	S^0

Тогда $D = [H - T]$ классифицирует расположение векторов следующим образом:

$D = 2$ — конечные точки вектора R_i^1 расположены с внешней и внутренней стороны ТК A_2 (варианты 1—4 в таблице);

$D = 1$ — одна из конечных точек вектора R_i^1 , располагается с внешней или внутренней стороны ТК A_2 , а другая точка R_i^1 принадлежит R_j^2 (варианты 5-16 в таблице);

$D = 0$ для векторов R_i^1 и R_j^2 , расположенных на одной прямой и имеющих общие точки (варианты 17—34 в таблице).

Алгоритм исследования взаимного расположения ТК A_1 относительно A_2 представлен на рис. 4.

Для исследования расположения A_2 относительно A_1 необходимо реализовать алгоритм, приведенный на рис. 4, еще раз поменяв A_1 и A_2 местами. После этого мы будем иметь взаимные расположения ТК A_1 и A_2 друг относительно друга, представленные в составляющих V_1 и V_2 .

Перед переходом к вопросу о структуре данных при обработке ТК оценим временные затраты на их проведение. Как правило, подобная обработка

проводится по принципу "каждый с каждым", т. е. первый элемент обрабатывается вначале со вторым, затем с третьим и т. д. до тех пор, пока не будет осуществлена обработка с последним элементом N . Затем, то же самое проводится со вторым элементом и т. д. до тех пор, пока не будет обработан последний элемент. Временные затраты на такую обработку пропорциональны $M!$.

Размерность ТК в печатных платах может достигать несколько десятков тысяч. В такой ситуации чрезвычайно важно сократить время обработки ТК либо по возможности исключить саму обработку. В нашем случае проблема решается следующим образом.

Для каждого ТК A_i определяются значения x_{i_min} , x_{i_max} , y_{i_min} , y_{i_max} , где x_{i_min} — минимальная координата x в пределах A_i ; y_{i_min} — минимальная координата y в пределах A_i ; x_{i_max} — максимальная координата x в пределах A_i ; y_{i_max} — максимальная координата y в пределах A_i . Аналогичным образом формируются координаты для ТК A_j . Перед тем как обрабатывать A_i и A_j , проверяется условие

$$x_{i_min} > x_{j_max} \cup y_{i_min} > y_{j_max} \cup x_{j_min} > x_{i_max} \cup y_{j_min} > y_{i_max}. \quad (5)$$

Если условие (5) выполняется, то данные для формирования $\Omega^i(x, y)$ и $\Omega^j(x, y)$ не готовятся, и расчет функции $F(x, y)$ для векторов не осуществляется.

Применение этой процедуры в программе, предназначенной для формирования таблицы цепей на основе топологии печатной платы, содержащей порядка 45 000 ТК, написанной на C++, позволило достичь времени обработки на компьютере "Intel® Core i5-4590 CPU@3.30GHz 3,30 GHz" порядка 30 с, что сопоставимо со временем аналогичной обработки в системе CAM-350 [6].

Структура данных для математической модели ТК

Рассмотрим более детально структуру данных для формирования модели (2).

В дальнейшем для более компактного изложения материала будем рассматривать уравнения (3) в следующем виде:

$$Ax + By + C = 0, \quad (6)$$

где A , B , C — коэффициенты уравнения прямой и $A = y_2 - y_1$; $B = x_1 - x_2$; $C = y_1x_2 - x_1y_2$.

Две прямые

$$A_1x + B_1y + C_1 = 0;$$

$$A_2x + B_2y + C_2 = 0$$

пересекаются в точке (x_p, y_p) , где

$$x_p = (B_1C_2 - B_2C_1)/(A_1B_2 - A_2B_1);$$

$$y_p = (A_1C_2 - A_2C_1)/(A_1B_2 - A_2B_1). \quad (7)$$

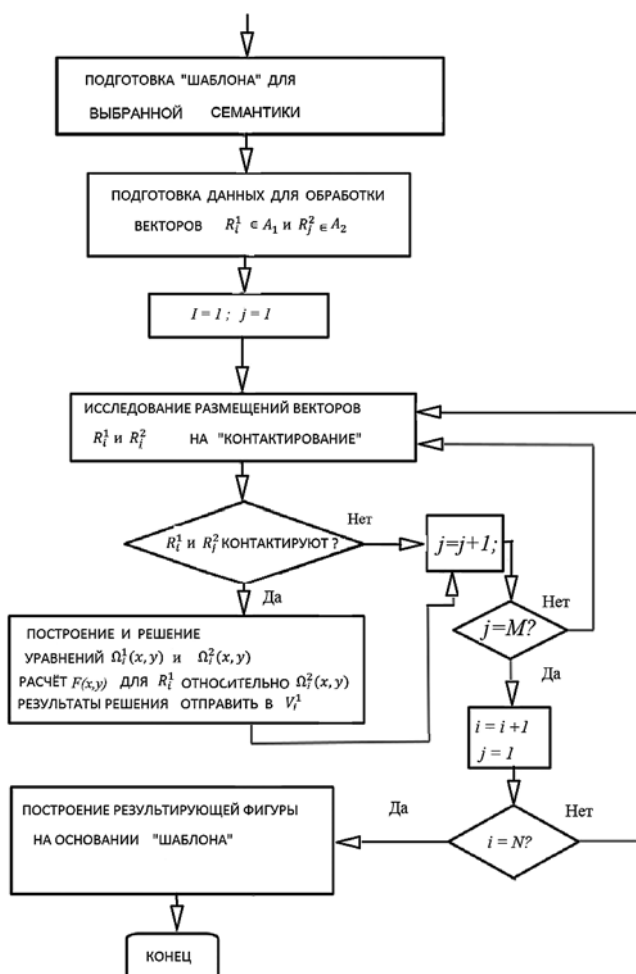


Рис. 4. Алгоритм исследования взаимного расположения ТК A_1 относительно A_2

Полная структура данных для модели E_i приведена на рис. 5.

В геометрическую составляющую G_i входят следующие параметры:

x_1, y_1, x_2, y_2 — начальные и конечные точки вектора;

$x_{\min}, x_{\max}, y_{\min}, y_{\max}$ — минимаксные значения координат для проверки условия (5);

A, B, C — коэффициенты уравнения прямой (6).

Составляющая V_i наполняется по мере обработки ТК и представляет собой набор массивов данных типа *uzel*, связанных между собой в единую цепь. Структура данных типа *uzel* приведена на рис. 6.

В состав *uzel* входят следующие составляющие:

(x_1, y_1) — начальная точка вектора, или (x_2, y_2) — конечная точка вектора, или (x_p, y_p) — точка пересечения векторов для подмножеств S^2 и S^1 , или

$(x_{d1}, y_{d1}), (x_{d2}, y_{d2})$ — точки, определяющие общую составляющую для векторов R_i^1 и R_j^2 , относящихся к подмножеству S^0 ;

Tip — тип взаимного расположения векторов в соответствии с таблицей;

Start_fin — идентификатор типа координаты, принимает следующие значения: 1 — для начальной точки вектора, 2 — для конечной точки, 3 — для точки пересечения векторов, 4 — для описания общих точек отрезков из подмножества S^0 ;

Prin — идентификатор принадлежности R_i^1 к A_2 , при этом если часть вектора (или вектор целиком) находится внутри A_2 , то *prin* принимает значение **In**; если вектор вне контура, то *prin* принимает значение **out**; если векторы из подмножества S^1 или S^2 , то *prin* принимает значение **dub/dub**, при этом **dub** указывает на то, что векторы имеют одинаковые направления, а **dub** указывает на встречные направления векторов; если принадлежность не определена, то *prin* принимает значение **nnn**, которое присваивается составляющей *prin* перед началом исследования взаимных расположений и не изменяется до тех пор, пока не появятся "контактирующие" векторы;

Napr указывает направление распространения принадлежности от (x_p, y_p) к (x_1, y_1) или к (x_2, y_2) . В первом случае *Napr* принимает значение 1, в противном случае — 0.

Построение модели

Математический аппарат для идентификации взаимных расположений векторов представляется тривиальным, но достаточно объемным, в силу чего не приводится в данной статье. Перед началом исследования расположений вектора $R_i^1 \in A_1$ относительно векторов $R_j^2 \in A_2$ сформируем цепочку, изображенную на рис. 7, состоящую из двух массивов типа *uzel*.

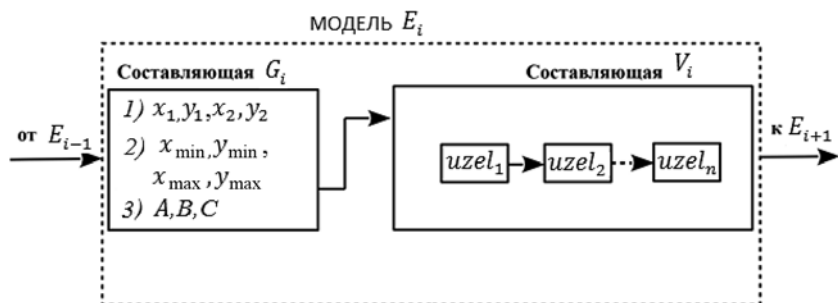


Рис. 5. Структура данных для модели E_i

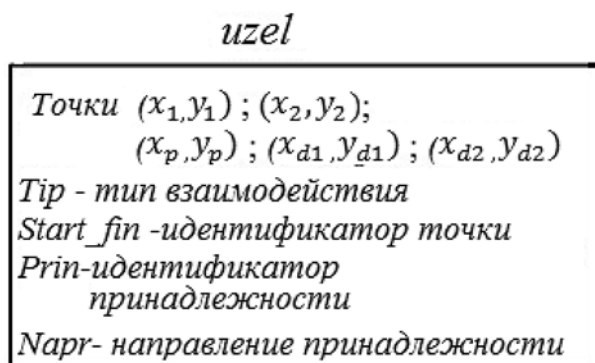


Рис. 6. Структура данных типа *uzel*



Рис. 7. Начало формирования структуры данных

Далее начинаем исследование расположений (сканирование) всех векторов R_j^2 из A_2 относительно R_i^1 . Каждое "контактирование" векторов порождает новый *uzel*, заносимый в цепь. В результате сканирования сформируется совокупность массивов *uzel*, изображенная на рис. 8. В дальнейшем эту совокупность будем называть "цепью".

При отсутствии "контактирования" между векторами новые *uzel* формироваться не будут. В этом случае модель останется в своем начальном состоянии.

"Цепь" после сканирования должна быть упорядочена по возрастанию или убыванию координат в *uzel*. Если $x_{i1} > x_{i2}$, то сортировка осуществляется по возрастанию координаты x , если $x_{i1} < x_{i2}$, то сортируем по убыванию координаты x . Если $x_{i1} = x_{i2}$, то сортируем по координате y по таким же правилам, как и для координаты x . После завершения указанной обработки "цепь" примет вид, приведенный на рис. 9.

Сформированная "цепь" представляет собой вектор с точками $(x_1, y_1), (x_2, y_2)$ и размещенными на нем точками "контактирования" с векторами

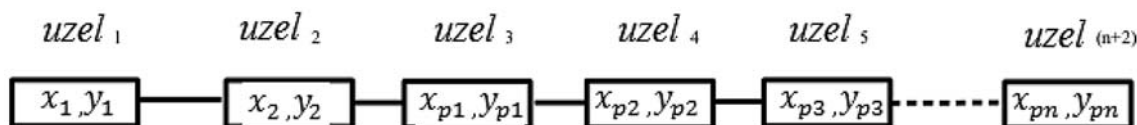


Рис. 8. "Цепь" после завершения сканирования

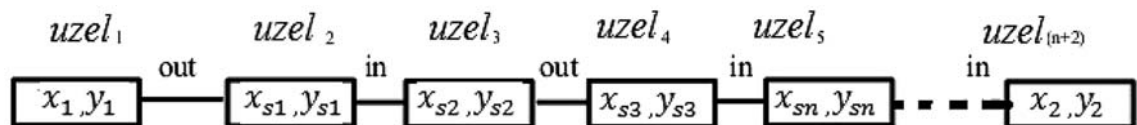


Рис. 9. "Цепь" после сортировки

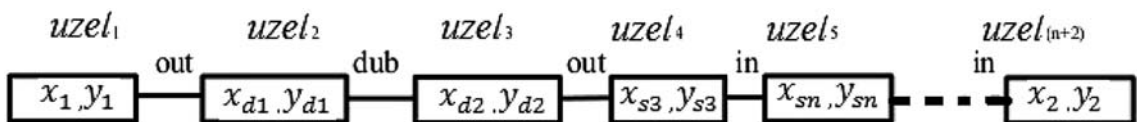


Рис. 10. Фрагмент модели ТК для векторов из подмножества S^0

ТК A_2 . В результате обработки наш вектор разбивается на набор составляющих $[(x_1, y_1), (x_{s1}, y_{s1})]; [(x_{s2}, y_{s2}), (x_{s3}, y_{s3})]; \dots [(x_{sn}, y_{sn}), (x_2, y_2)]$, каждая из которых имеет свою принадлежность к ТК A_2 .

Пример "цепи" для подмножеств S^0 , приведен на рис. 10.

Точки (x_{d1}, y_{d1}) , (x_{d2}, y_{d2}) являются координатами общего отрезка двух векторов из подмножества S^0 , т.е. два вектора ТК A_1 и A_2 расположены на одной прямой и соприкасаются.

Введем термин "однородный отрезок", под которым будем понимать вектор или его часть, имеющие одно значения параметра *prin* на протяжении всей его длины.

"Однородный отрезок" будем обозначать K .

Введем систему записи для "однородного отрезка" K . Справа от K в фигурных скобках $\{\}$ будем помещать следующую информацию:

- m — номер "однородного отрезка" в векторе, который может быть его порядковым номером, либо — 0 в том случае, если сам вектор является "однородным отрезком";
- *prin* — значение принадлежности.

В качестве примера приведем следующие записи для "однородного отрезка":

$$1) K_3^1 \{4, \text{out}\}, 2) K_2^2 \{0, \text{in}\} \text{ и т. д.}$$

В первом примере "однородный отрезок" занимает 4-ю позицию в 3-м векторе ТК A_1 и расположен вне ТК A_2 . Во 2-м примере 2-й вектор ТК A_2 является "однородным отрезком" и целиком находится внутри ТК A_1 .

Рассмотрим пример выполнения преобразования над ТК A_1 и A_2 , приведенными на рис. 11. ТК A_1 описывается следующим набором координат: $[10/10; 10/20; 30/20; 30/10; 10/10]$, а ТК A_2 — $[20/10; 20/30; 30/30; 30/10; 20/10]$.

Топологическое преобразование можно условно разбить на три этапа:

- 1) исследование взаимных расположений ТК;
- 2) формирование значений параметра *prin* векторов, не имеющих точек "контактирования" с другими векторами;
- 3) реализация семантики преобразования.

После выполнения первого этапа мы получим следующие результаты для ТК A_1 :

$$[10/10 - 10/20] = K_1^1 \{0, \text{nnn}\};$$

$$[10/20 - 20/20] = K_2^1 \{1, \text{out}\};$$

$$[20/20 - 30/20] = K_2^1 \{2, \text{in}\};$$

$$[30/20 - 30/10] = K_3^1 \{0, \text{dub}\};$$

$$[30/10 - 20/10] = K_4^1 \{1, \text{dub}\};$$

$$[20/10 - 10/10] = K_4^1 \{2, \text{out}\}$$

и для ТК A_2 :

$$[20/10 - 20/20] = K_1^2 \{1, \text{in}\};$$

$$[20/20 - 20/30] = K_1^2 \{2, \text{out}\};$$

$$[20/30 - 30/30] = K_2^2 \{0, \text{nnn}\};$$

$$[30/30 - 30/20] = K_3^2 \{1, \text{out}\};$$

$$[30/20 - 30/10] = K_3^2 \{2, \text{dub}\};$$

$$[30/10 - 20/10] = K_4^2 \{0, \text{dub}\}.$$

По результатам 1-го этапа надо отметить следующий момент. Первый вектор ТК A_1 и второй вектор ТК A_2 имеют *prin* **nnn**, т. е. принадлежность не определена. Прежде чем рассмотреть метод оп-

ределения принадлежности, отметим следующие очевидные моменты. Признак $prin = \text{nnn}$ имеет место только для тех векторов, которые не контактируют с другими векторами, при этом принадлежность $prin = \text{nnn}$ распространяется целиком на весь вектор. Изменение принадлежности вектора может происходить только в местах "контактирования". Очевидно, что векторы с неопределенной принадлежностью должны ее "наследовать" от соседних векторов, т. е. для вектора R_i^1 соседним вектором будет R_{i-1}^1

или R_{i+1}^1 . Алгоритм "Распространение принадлежности в рамках модели E_i " представлен на рис. 12. Действие алгоритма распространяется на ТК A_1 , который содержит N векторов и обязательно "контактирует" с ТК A_2 . Блоки, помеченные знаком *, содержат так называемые "циклические" операции (операции, взятые в кружок) сложения и вычитания. Эти операции выполняются следующим образом: для операции сложения выполняется $i = i + 1$, при этом если $i = N$, то $i = 0$; для операции вычитания выполняется $i = i - 1$, при этом если $i = 0$, то $i = N$.

После реализации действий алгоритма, представленного на рис. 12, результаты преобразований изменятся следующим образом. Описание "однородного отрезка" K_1^1

$$[10/10 - 10/20] = K_1^1 \{0, \text{nnn}\}$$

изменится на $[10/10 - 10/20] = K_1^1 \{0, \text{out}\}$,

$$\begin{aligned} \text{а } K_2^2 \text{ с } [20/30 - 30/30] &= K_2^2 \{0, \text{nnn}\} - \\ \text{на } [20/30 - 30/30] &= K_2^2 \{0, \text{out}\}. \end{aligned}$$

Последний этап заключается в построении "шаблона", позволяющего отобрать из всего набора "однородных отрезков" именно те отрезки, которые содержат значения $prin$, необходимые для заданной семантики. Для этого определим структуру "шаблона" в виде списка, в котором должна быть указана принадлежность "однородного отрезка" к ТК A_1 или A_2 , а также признак $prin$, соответствующий заданной семантике. Операции объединения, пересечения и вычитания будут выглядеть следующим образом:

$$A_1 \cup A_2 = \bigvee_1^N K_i^1 [\text{out}] \cup \bigvee_1^M K_j^2 [\text{out}] \cup \bigvee_1^N K_i^1 [\text{dub}];$$

$$A_1 \cap A_2 = \bigvee_1^N K_i^1 [\text{in}] \cup \bigvee_1^M K_j^2 [\text{in}] \cup \bigvee_1^N K_i^1 [\text{dub}];$$

$$A_1 \setminus A_2 = \bigvee_1^N K_i^1 [\text{out}] \cup \bigvee_1^M K_j^2 [\text{in}].$$

Для операции объединения в результирующую фигуру должны войти все "однородные отрезки" из

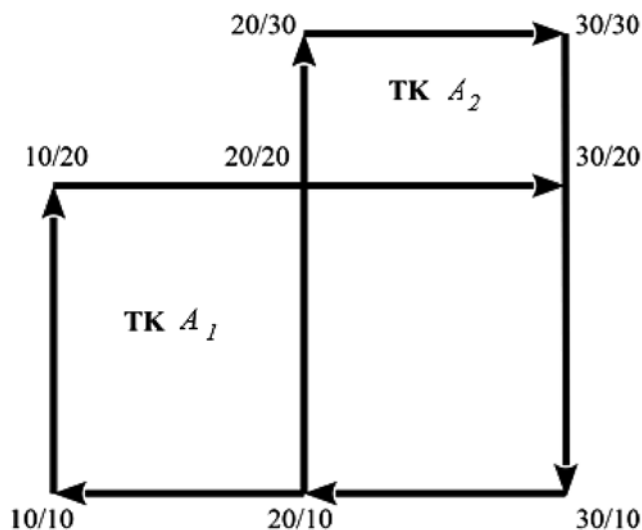


Рис. 11. Пример выполнения преобразований над ТК

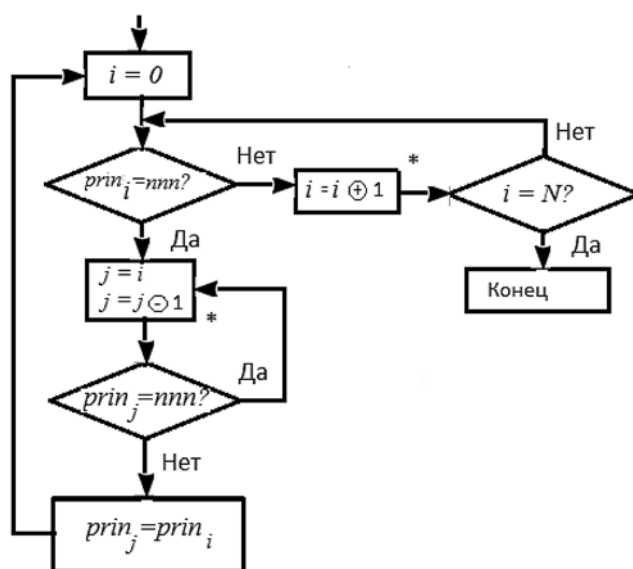


Рис. 12. Алгоритм "Распространение принадлежности в рамках модели E_i "

A_1 , расположенные вне A_2 , все "однородные отрезки" из A_2 , расположенные вне A_1 , и только отрезок из A_1 , если два отрезка принадлежат S^0 и т. д. Преобразование заканчивается после того, как будут выбраны все отрезки K , задаваемые формулами, определяющими семантику.

Возможность включать в результирующую ТК "однородные отрезки" с различными принадлежностями позволяет реализовывать самые экзотические семантики преобразований.

Основы данного метода обработки ТК были разработаны в период с 1985 по 1990 г. в рамках создания программного обеспечения для выпуска конструкторской документации на печатные платы (см. рис. 1).

Внедрение новых линий изготовления печатных плат потребовало возврата к забытым мето-

дам, их совершенствованию и применению при разработке программного обеспечения для выпуска групповых фотошаблонов на двусторонние и многослойные печатные платы.

Заключение

- Предложена методика выполнения логических операций над ТК, основанная на исследовании их взаимных расположений.
- Разработана структура данных, позволяющая формализовать и рассчитать параметры принадлежности в рамках предложенного математического аппарата.
- Проведена классификация вариантов взаимных размещений векторов, на базе которой строится математическая модель ТК и исследуются взаимные размещения ТК.
- Предложенные методы прошли штатную эксплуатацию на предприятии в рамках программного обеспечения для выпуска конструкторской документации на печатные платы, а также

при изготовлении групповых фотошаблонов. Эксплуатация программного обеспечения подтвердила справедливость и корректность предложенных методов и алгоритмов.

Список литературы

1. **Суходольский В. Ю.** Altium Designer проектирование функциональных узлов РЭС на печатных платах. Санкт-Петербург: "БХВ — Петербург", 2010. 345 с.
2. **Архангельский А. Н., Швецов Е. О.** Выпуск конструкторской документации на печатные платы, спроектированные в системе PCAD200X // Печатный монтаж. 2009. № 1. С. 26—29.
3. **Архангельский А. Н., Афанасьев Д. В., Зарубин А. Л., Кондратьев В. Н.** Автоматизированные методы контроля печатных плат // Контроль. Диагностика. 2012. № 1. С. 15—16.
4. **Корн Г., Корн Т.** Справочник по математике для научных работников и инженеров. М.: Наука, 1977. 580 с.
5. **Рвачев В. Л.** Теория R-функций и некоторые ее применения. Киев: Наукова думка, 1982. 551 с.
6. **CAM350-программа** технологической подготовки печатных плат // EDAEXPRESS. ЗАО "Родник". Апрель 2007. Вып. 15.
7. **Down Stream Technologies.** Solutions for Post Processing PCB Designs. [CША]. URL: www.downstreamtechnolo.com. (дата обращения — 09.2015).

A. N. Arkhangelskiy, Engineer, e-mail: aleksandr.arhangelskiy@bk.ru,
Joint Stock Company "Russian Space Systems", JSC "RSS", Moscow

Processing Methods of Topological Components in Design and Manufacturing of the Printed Circuit Boards

The article presents methods of processing for topological component (TC), used in design and manufacture of printed circuit boards. The need for such methods arises from the fact that the objects of the manipulation and processing in PP are TC, namely contact areas (CA), conductors, fields of metallization etc. TC processing means changing its original geometry, depending on the functional necessity, for example: "trimming the CA." The proposed method covers on TC which are enclosed loop without self-intersections and consists of segments located on straight lines described by equations of the first order.

Keywords: printed circuit boards, topological components, contact area, conductor, polygon, field of metallization, logical operation, combination, intersection, subtraction

References

1. **Suhodol'skiy V. Ju.** AltiumDesignerproektirovaniefunkcional'nyhuzlovRJeSnapechatnyhplatah. Sankt-Peterburg: "BHV — Peterburg", 2010, 345 p. (in Russian).
2. **Arhangel'skiy A. N., Shvecov E. O.** Vypusk konstruktorskoy dokumentatsii na pechatnye platy, sproektirovannye v sisteme PCAD200X, *Pечатnyjmontazh*, 2009, no. 1, pp. 26—29 (in Russian).
3. **Arhangel'skiy A. N., Afanas'ev D. V., Zarubin A. L., Kondrat'ev V. N.** Avtomatizirovannye metody kontrolja pechatnyh plat, *Kontrol' i diagnostika*, 2012, no. 1, pp. 15—16 (in Russian).

4. **Korn G., Korn T.** *Spravochnik po matematike dlja nauchnyh rabotnikov i inzhenerov*. Moscow: Nauka, 1977. 580 p. (in Russian).
5. **Rvachjov V. L.** *Teorija R-funkcij i nekotorye ego primeneniya*, Kiev: Nauka dumka, 1982. 551 p. (in Russian).
6. **CAM350-programma** tehnologicheskoy podgotovki pechatnyh plat. ZAO "Rodnik". EDA EXPRESS. Aprel' 2007. Vyp. 15 (in Russian).
7. **Down Stream Technologies.** Solutions for Post Processing PCB Designs. [SShA] URL: www.downstreamtechnolo.com (date of access 09.2015).

УДК 004.9:378.147

В. Н. Юрин, д-р техн. наук, проф., e-mail: ontimati@yandex.ru,
Московский авиационный институт (национальный исследовательский университет)

Мониторинг использования компьютерного инжиниринга в инженерном образовании

Приведены методика и результаты многолетнего мониторинга отечественного инженерного образования на основе прикладных информационных технологий. Предложены показатели использования компьютерного инжиниринга в инженерном образовании на разных этапах его развития с учетом готовности к этому инженерного образования и потребностей конкурентоспособной промышленности. Даны экспертные оценки каждого этапа по упомянутым показателям.

Ключевые слова: прикладные информационные технологии, инженерное образование, компьютерный инжиниринг, экспертное оценивание, конструкторско-технологическая подготовка производства, машиностроение, автоматизированное производство, кадровое обеспечение, принцип разделения труда

Введение

Современное машиностроение развивается в направлении полного перехода на использование прикладных информационных технологий (ПИТ), реализующих идеологию электронной информационной поддержки жизненного цикла продукции (ИПИ/CALS) от проектирования до эксплуатации и утилизации в виде автоматизированных систем PLM (Product Life Management) [1]. ПИТ привели к кардинальному изменению среды создания, обмена и формы представления технической информации. Коренным образом изменились организация, условия и средства инженерного труда, в связи с этим во многом изменился характер труда инженера, появилась новая терминология, трансформируется нормативно-техническая документация, регламентирующая инженерную деятельность. Естественно, это потребовало соответствующих изменений в инженерном образовании (ИО), планомерного решения связанных с этим проблем. Использование ПИТ — не простая замена традиционного кульмана и других инструментов инженеров на электронные. Это по существу смена парадигмы и производства, и образования, связанная с интеграцией производственных и информационных технологий, переходом от бумажных конструкторских и технологических документов к электронной документации и электронному документообороту на всех стадиях жизненного цикла изделий [2].

С появлением программных и информационных продуктов, созданных в качестве профессиональных инструментов для широкого круга специа-

листов, в первую очередь инженеров (а не исследователей, специалистов по САПР и информационным технологиям, программистов, интеграторов и т. п.), возникло и эффективно развивается новое направление инженерной деятельности при создании техники — компьютерный инжиниринг. Компьютерный инжиниринг — совокупность методов и средств практического решения повседневных инженерных задач специалистами производства с помощью средств вычислительной техники и ПИТ [2, 3].

При этом важно понимать, что многократно описанные преимущества использования ПИТ в полной мере (эффективно) могут быть реализованы только при системном использовании комплекта автоматизированных систем (с применением CAD/CAM/CAE/CAPP и т. п., PLM-решений), наличии критической массы рабочих мест и специалистов, способных решать свои профессиональные задачи с использованием ПИТ в едином информационном пространстве проектно-производственной кооперации, работающей над созданием изделия [4]. А для этого нужны квалифицированные инженерные кадры.

Инженерное образование должно соответствовать потребностям промышленности и требованиям научно-технического прогресса, что сегодня диктует необходимость подготовки специалистов автоматизированного производства как основной функции технической высшей школы [5]. Именно это обеспечит конкурентоспособность и ИО, и промышленности.

Постановка задачи исследования

При исследованиях в области использования компьютерного инжиниринга в ИО основное внимание уделяют конкретному опыту авторов в данном направлении, прежде всего, применению различных программных средств в учебном процессе преподавания инженерных дисциплин при наибольшем внимании к геометро-графической подготовке студентов.

Цель настоящего исследования — совершенствование отечественного ИО для более полного удовлетворения потребностей промышленности в современных инженерных кадрах, владеющих ПИТ, при решении каждодневных профессиональных задач. Для этого необходимо провести анализ продвижения использования компьютерного инжиниринга в учебный процесс ИО от знакомства с элементами ПИТ к подготовке специалистов автоматизированного производства с выявлением этапов этого движения, критериев оценки решаемых на них задач, с количественной оценкой степени решения этих задач, определения слабых мест ИО на основе компьютерного инжиниринга для выработки рациональных управляющих воздействий на ИО.

В настоящей работе предпринята попытка рассмотреть эволюцию применения компьютерного инжиниринга в отечественном ИО (в целом, без дифференциации по специальностям и регионам) с учетом потребностей промышленности в автоматизированной конструкторско-технологической подготовке производства, развивая идеи работ [2, 6, 7]. По сути это детализация выполнения государственных и региональных программ и мероприятий, направленных на приведение квалификации выпускников вузов в соответствие с современными требованиями промышленности [8—10].

Метод решения задачи исследования

В качестве метода решения задачи исследования использовался мониторинг — систематический сбор и анализ информации, необходимой для изучения объекта исследования — отечественного ИО и определения управляющих воздействий на него.

В качестве информационной базы проводимого мониторинга использованы основные доступные источники получения информации об объекте исследования:

- публикации главных участников применения ПИТ в инженерном образовании — преподавателей инженерных вузов (статьи, тезисы докладов, диссертации); наибольшую долю составили материалы конференций по тематике, связанной с использованием компьютерного инжиниринга в ИО, проведенных в период с 1993 г. в МАТИ¹, МИФИ, МЭИ (Москва), УГАТУ

¹ Ныне Авиационно-технологический комплекс (АТК) МАИ.

(Уфа), СИБСТРИН (Новосибирск), ПНИПУ (Пермь), БГТА (Брянск) и др. Это тысячи публикаций, отражающих опыт и тенденции применения ПИТ в учебном процессе ИО, авторов из всех регионов страны от Калининграда до Владивостока;

- личный опыт автора — вузовского преподавателя инженерных дисциплин с 1969 г., организатора и руководителя секций "Компьютерные технологии в учебном процессе ИО" на Всероссийской научно-технической конференции "Новые материалы и технологии" (МАТИ, 1997—2014 гг.) [11] и на Международной молодежной научной конференции "Гагаринские чтения" (МАТИ, 2007—2015 гг., МАИ, 2016 г.);
- материалы и личный опыт автора — инициатора и координатора Всероссийского (с международным участием) молодежного конкурса "Компьютерный инжиниринг" (в период 1999—2008 гг. в рамках программы "НИЦ АСК — вузам" [12] и Международной молодежной научной конференции "Гагаринские чтения" МАТИ) — конкурса курсовых и дипломных проектов студентов инженерных вузов, работ молодых специалистов вузов и промышленности, выполненных на персональных ЭВМ с использованием автоматизированных систем конструирования и технологической подготовки производства — CAD/CAM/CAE/CAPP-систем и технологий ИПИ/CALS, а также необходимого для этого учебно-методического и информационного обеспечения учебного процесса инженерного образования [13]. Конкурс привлек внимание студентов, аспирантов и преподавателей вузов всех экономических районов России [2]. В 10 конкурсах 1999—2008 гг. приняли участие 800 работ свыше 1200 авторов из более 100 вузов и предприятий более пятидесяти городов всех федеральных округов России — от Санкт-Петербурга до Владивостока, а также ряда городов Белоруссии, Казахстана, Киргизии и Украины. За 10 лет конкурс "Компьютерный инжиниринг" стал значимым для научно-педагогической и студенческой общественности российских вузов средством стимулирования применения ПИТ в инженерном образовании — и студентами, и преподавателями инженерных дисциплин.

Изучение материалов указанных выше конференций, диссертаций по рассматриваемой тематике, работ конкурса "Компьютерный инжиниринг" как актуальных, достоверных, регулярных, доступных источников информационной базы мониторинга позволяет на репрезентативной выборке провести сбор, обработку и анализ информации об использовании компьютерного инжиниринга в отечественном ИО при подготовке специалистов в области конструкторско-технологической подготовки производства. При этом по предложенным показате-

лям мониторинга можно получить интегральные оценки уровня развития ИО в целом, без дифференцирования по специальностям или регионам, учитывая распространенность достижений вузов по каждому такому показателю.

Показатели мониторинга — различные на разных этапах развития ИО, поскольку для каждого из этапов актуальны свои характеристики, определяющие развитие ИО на основе компьютерного инжиниринга.

Количественная оценка для выбранных показателей мониторинга проводится методом индивидуальных экспертных оценок [14, 15] путем попарного сравнения упомянутых показателей.

Результаты исследования

Проведенное исследование показало, что в применении компьютерного инжиниринга в ИО можно выделить три основных этапа.

На *первом* этапе в ряде вузов реализовывалась "островная" стратегия информатизации ИО с использованием ПИТ лишь некоторыми студентами и преподавателями, чьи интересы профессионально связаны с развитием этих технологий — САПР, ПИТ и т. п. Это период до 1990 года (здесь и для остальных этапов приведены усредненные сроки реализации, так как для разных вузов и специальностей они различны в силу кадровых, финансовых, региональных, отраслевых и других взаимосвязанных особенностей). Для этого этапа по целому ряду объективных и субъективных причин, рассмотренных в работе [2], в большинстве случаев характерны две крайности:

- либо традиционная конструкторско-технологическая подготовка производства (КТПП) с эпизодическим использованием ПИТ при обучении и, как следствие, недостаточной подготовкой в области информационных технологий;
- либо подготовка разработчиков САПР при недостаточной конструкторско-технологической подготовке.

На рис. 1 представлены микромоделли ИО на первом этапе применения компьютерного инжиниринга в виде векторных графов с экспертными оценками F_i по $i = 6$ критериям — показателям, характеризующим информатизированное ИО как сферу деятельности в сравнении с идеальным вариантом.

Подробный анализ показателей, приведенных на рис. 1, представлен в работах [2, 6], где показано, что на данном этапе успех информатизации ИО в значительной мере определяется показателем "Разрыв преподавания информационных технологий (ИТ) и инженерных дисциплин (ИнжД)" (показатель 1, рис. 1). Основная причина (помимо финансовых проблем) — слабое владение ИТ преподавателями ИнжД, тогда как основная часть необходимых современному инженеру знаний ИТ связана с практическим освоением ряда профессиональных инструментов, используемых при решении задач КТПП (в первую очередь, автоматизированных систем CAD/CAM/CAE).

На *втором* этапе был осуществлен переход к стратегии замещения традиционного ИО образованием на основе компьютерного инжиниринга с освоением основных инструментов современного инженера — CAD/CAM/CAE-систем с участием основной массы студентов и преподавателей широкого круга общетехнических и специальных инженерных дисциплин. Для инженерных специальностей в России это период до 2005—2006 гг.

На рис. 2 показан ряд микромоделей ИО с экспертными оценками по тем же показателям, что на рис. 1, но для 1999, 2004 и 2006 гг.

Как видно из сравнения рис. 1 и рис. 2, масштабы имевших место ранее недостатков развития ИО на основе компьютерного инжиниринга постепенно уменьшаются:

- практически ликвидировано неравенство доступа к ПИТ инженерных кафедр и подразделений, связанных с развитием САПР и ИТ (см. показатель 5 на рис. 2);

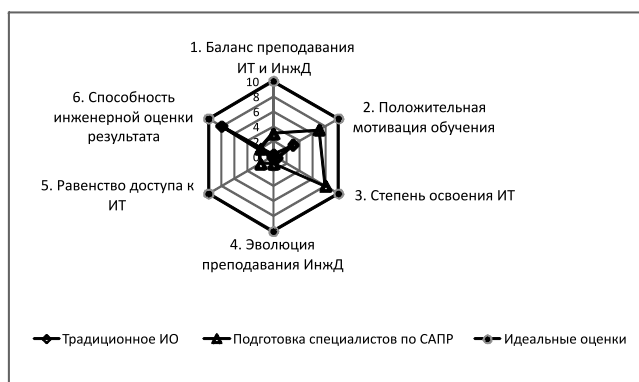


Рис. 1. Микромоделли инженерного образования на первом этапе применения компьютерного инжиниринга

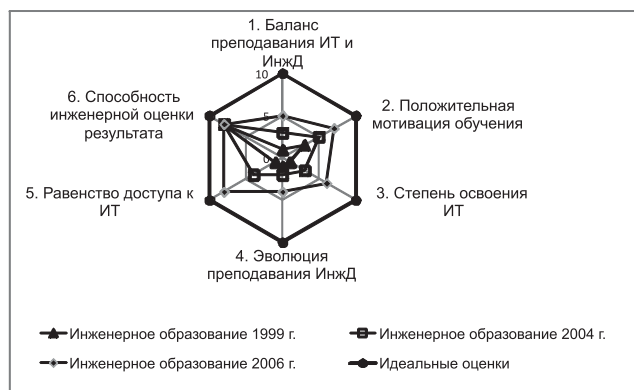


Рис. 2. Микромоделли инженерного образования на втором этапе применения компьютерного инжиниринга

- улучшился баланс преподавания ИнжД и ПИТ (показатель 2), произошла существенная эволюция преподавания ИнжД (показатель 4): все шире ПИТ используют постоянно на всем протяжении обучения в вузе, а ИнжД преподают на основе упомянутых технологий;
- повысилась степень освоения ПИТ основной массой студентов и их использование на всем периоде обучения в вузе (показатель 3), увеличилась доля преподавателей ИнжД, способных работать с CAD/CAM/CAE-системами при курсовом и дипломном проектировании;
- при освоении ПИТ удастся сохранять у студентов способности инженерной оценки результата, получаемого с их применением (показатель 6), при преподавании ИнжД на базе таких технологий.

Таким образом, на первых двух этапах ИО решало прежде всего основные вопросы перехода от традиционного ИО к ИО на основе компьютерного инжиниринга. Из актуальных задач промышленности здесь обращалось внимание лишь на освоение ПИТ, в первую очередь CAD/CAM/CAE-систем как основного инструмента современного инженера. Но главной задачей ИО на втором этапе становится не выпуск инженеров, знакомых с возможностями автоматизации работ, а переход к подготовке специалистов автоматизированного производства, для которых упомянутые средства автоматизации — каждодневный профессиональный инструмент.

На *третьем* этапе, который продолжается до настоящего времени, происходит расширение спектра применяемых в учебном процессе ИО средств автоматизации инженерного труда с обучением выполнению цепочек автоматизированных конструкторских и технологических работ на основе 3D-моделей изделия и переходом к использованию информационного взаимодействия в условиях коллективной и параллельной работы с применением PDM-систем PLM-решений. Таким образом, на данном этапе ИО приступило к постепенному решению задач эффективного использования воз-

можностей компьютерного инжиниринга для повышения производительности труда, что соответствует требованиям технического прогресса в конкурентоспособной промышленности.

В силу отличия первоочередных задач, решаемых на третьем этапе развития ИО, по сравнению с предыдущими этапами оценка развития применения компьютерного инжиниринга в ИО осуществляется с использованием других показателей. На рис. 3 показаны микромоделли ИО для 2006 и 2016 гг. Приведенные здесь экспертные оценки F_i показателей i свидетельствуют о том, что соответствующие работы в ИО проводятся в России относительно медленно.

Этому имеются веские объективные причины. К основным из них относятся:

- отечественная промышленность (в целом при наличии ряда лидеров — предприятий, прежде всего, оборонно-промышленного комплекса) сильно отстает в использовании информационного взаимодействия и параллельного инжиниринга при разработке и изготовлении изделий с использованием ПИТ [7];
- вопросы эффективного параллельного выполнения работ КТПП в недостаточной степени проработаны теоретически и методически [16];
- интеграция процессов КТПП существующими у инженерных кафедр и подразделений вузов средствами обеспечена большей частью лишь в условиях гомогенной среды используемых САПР (в рамках интегрированных продуктов компаний АСКОН (Россия), Топ Системы (Россия), Siemens PLM Software (Германия), Dassault Systemes (Франция), SolidWorks Corp. (США), Autodesk (США), PTC (США) и др.); при этом возможности выполнения цепочек автоматизированных работ КТПП на основе 3D-моделей изделий ограничены конфигурацией имеющихся в вузе (на кафедре) программных продуктов;
- для эффективного информационного взаимодействия при выполнении работ КТПП необходим переход к структурированным конструкторским и технологическим документам, управлению входящими в них данными с помощью PDM-систем, создания и эксплуатации единого информационного пространства, использования "облачных" технологий, что требует применения специальных технологий, дорогостоящих программно-технических средств и самое главное — нового уровня вспомогательного персонала, оплата труда которого по нынешним условиям рынка существенно превышает зарплату преподавателя кафедры, т. е. таких кадров на инженерных кафедрах просто нет;
- организация информационного взаимодействия участников работ КТПП — важнейший резерв повышения производительности труда [4], который находится вне сферы профессиональных

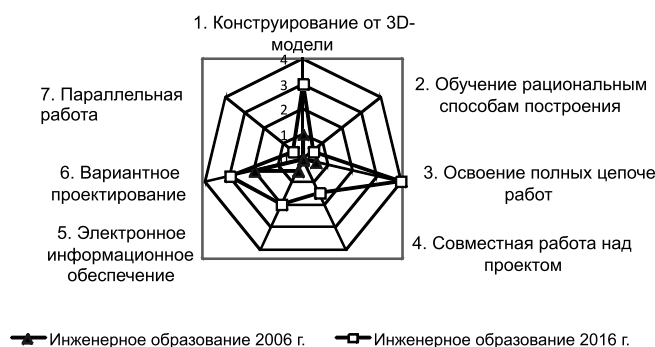


Рис. 3. Микромоделли инженерного образования на третьем этапе применения компьютерного инжиниринга

интересов выпускающих кафедр инженерных вузов и преподавателей ИнжД. При этом такие кафедры в большинстве случаев вынуждены опираться на свои силы и ограниченную поддержку дружественных предприятий промышленности, но не имеют поддержки специализированных ИТ-кафедр и подразделений САПР, информационных систем, вузы пока редко практикуют создание специальных ресурсных центров, центров коллективного пользования и т. п., доступных студентам машиностроительных специальностей и преподавателям ИнжД.

По существу на данном этапе использования компьютерного инжиниринга в ИО речь идет о формировании и применении в учебном процессе PLM-системы. Но это сложная человеко-машинная система, требующая участия специалистов ПИТ. Создание и внедрение ее — весьма трудоемкая специализированная работа [17]. Поэтому эффективное использование возможностей компьютерного инжиниринга при обучении КТПП возможно лишь при кооперации усилий:

- преподавателей — энтузиастов ПИТ, усилиями которых до сих пор во многом развивалось ИО с применением компьютерного инжиниринга;

- руководителей выпускающих кафедр, осознающих необходимость информационного взаимодействия и параллельного выполнения работ при обучении КТПП как важнейшего условия удовлетворения современных потребностей промышленности в кадрах, готовых эффективно использовать возможности компьютерного инжиниринга;

- руководителей вузов и их подразделений, готовых поддержать реинжиниринг учебного процесса на основе компьютерного инжиниринга как необходимое условие организации выпуска конкурентоспособных специалистов производства (поскольку активная поддержка верхнего менеджмента — одно из основных условий при любом реинжиниринге [18]);

- кафедр САПР и информационных систем, ресурсных центров вузов, где профессионально развивают ПИТ, создают инфраструктуру эффективного использования компьютерного инжиниринга;

- заинтересованных промышленных предприятий, в том числе разработчиков и интеграторов ПИТ, готовых содействовать использованию компьютерного инжиниринга в ИО (передачей опыта, финансами и т. п.) для организации выпуска современных специалистов производства.

Это позволит реализовать, наконец, принцип разделения труда, когда преподаватели ИнжД готовят специалистов КТПП, готовых использовать современные ПИТ, а специалисты ПИТ обеспечивают инфраструктуру для этой работы. Реализация упомянутого принципа разделения труда при подготовке специалистов КТПП является решающим условием повышения качества их обучения, по-

скольку объем инженерных (профильных) знаний и объем знаний, необходимых для формирования, эксплуатации и развития PLM-систем, в настоящее время соизмеримы.

Выпускники вузов, освоившие использование компьютерного инжиниринга на примере PLM-системы при решении задач КТПП в учебном процессе, могут стать помощниками в деле инициации применения PLM-систем на предприятиях, что актуально с учетом не всегда пока должного интереса отечественной промышленности к инновациям [19].

Заключение

Проведенный мониторинг использования компьютерного инжиниринга в отечественном ИО на основе анализа работ преподавателей и студентов машиностроительных вузов позволяет выявить показатели, оценка которых может быть использована при управлении развитием ИО для обеспечения его соответствия потребностям промышленности и требованиям научно-технического прогресса.

Предложенная методика количественного анализа результатов этого мониторинга актуальна для оценки качества ИО на кафедральном, вузовском, отраслевом, национальном уровнях.

Список литературы

1. **Международная** энциклопедия CALS-технологий. Авиационно-космическое машиностроение / Гл. ред. А. Г. Братухин. М.: НИЦАСК, 2015. 608 с.
2. **Юрин В. Н.** Компьютерный инжиниринг и инженерное образование. М.: Эдиториал УРСС, 2002. 152 с.
3. **Yurin V. N., Zlygarev V. A.** Computer engineering — the basis of training of specialists for machine — building industry // ICEE'95. International Conference of Engineering Education. UNESCO. Abstracts. Moscow: ICEE, 1995. P. 23.
4. **Злыгарев В. А., Юрин В. Н.** Анализ производительности инженерного труда для построения CALS-систем // V Международная конференция-форум "Применение ИПИ (CALS) — технологий для повышения качества и конкурентоспособности наукоемкой продукции". Материалы конференции. М.: MBM, 2003. С. 75—79.
5. **Юрин В. Н.** Системная организация технологического развития в машиностроении: проблемы и решения // Технодоктрина 2014. Сб. докладов 1-й Всерос. форума технологического лидерства России. М.: РУСТО, 2014. С. 370—372.
6. **Юрин В. Н.** Инженерное образование и информационные технологии: проблемы и опыт их решения // Вестник машиностроения. 1998. № 5. С. 44—51.
7. **Юрин В. Н.** Расширение использования возможностей прикладных информационных технологий при обучении конструкторско-технологической подготовке производства // Научные труды (Вестник МАТИ). 2010. Вып. 16 (88). С. 310—315.
8. **Совершенствование** системы подготовки кадров для предприятий оборонно-промышленного комплекса // Материалы VII Всерос. совещания, Ижевск, 21—22 октября 2014 г., ФГБОУ ВПО "ИжГТУ им. М. Т. Калашникова". Ижевск: ИННОВА, 2014. 136 с.
9. **Иванов А. В., Кузнецов О. В.** О совершенствовании подготовки профессиональных кадров для организаций ОПК // Высшее образование в России. 2015. № 8—9. С. 32—38.

10. **Кожина Т. Д., Волков С. А.** Совершенствование системы подготовки кадров для предприятий оборонно-промышленного комплекса // Вестник Рыбинской государственной авиационной технологической академии. 2016. № 1 (36). С. 123—129.

11. **Юрин В. Н.** Компьютерные технологии в учебном процессе инженерного образования // Информационные технологии. 1999. № 3. С. 45—46.

12. **Злыгарев В. А.** Основные положения программы "НИЦ АСК — вузам" / Новые материалы и технологии — НМТ-98. Тез. докл. Всерос. научно-техн. конф. 17—18 ноября 1998 г., МАТИ. М.: ЛАТМЭС, 1998. С. 390.

13. **Юрин В. Н.** К информатизации инженерного образования: конкурс "Компьютерный инжиниринг" // Информационные технологии. 1999. № 9. С. 54—55.

14. **Саати Т.** Принятие решений. Метод анализа иерархий. М.: Радио и связь, 1993. 320 с.

15. **Альбом** технологической оснастки для станков с ЧПУ в авиадвигателестроении: учеб. пособие. Ч. 1. Станочные приспособления для станков с ЧПУ в авиадвигателестроении / В. Ф. Безья-

зычный, В. Д. Корнеев, В. Н. Ливанов, Т. Д. Кожина, В. Н. Юрин, И. Н. Аверьянов / Под общ. ред. В. Ф. Безязычного. М.: Машиностроение, 2000. 147 с.

16. **Маликов С. Б.** Метод анализа технического риска при организации подготовки опытного производства деталей. Автореф. дисс. ... канд. техн. наук. М.: МАТИ, 2012. 19 с.

17. **Злыгарев В. А.** Технология внедрения PLM-решений на основе аутсорсинга // Информационные технологии на службе оборонно-промышленного комплекса России. Сб. трудов второй конференции. 10—12 апреля 2013 г. М.: Connect!, 2013. С. 78—79.

18. **Яблочников Е. И., Молочник В. И., Фомина Ю. Н.** Реинжиниринг бизнес-процессов проектирования и производства: учеб. пособие. СПб.: СПбГУИТМО, 2008. 152 с.

19. **Стратегия** инновационного развития Российской Федерации на период до 2020 года. Утверждена распоряжением Правительства РФ от 8 декабря 2011 года № 2227-р. URL: <http://gov.garant.ru/SESSION/PDA/linkProxy?subjectId=70106125&link-Type=65537> (дата обращения 15.04.2017).

V. N. Yurin, Professor, e-mail: ontimati@yandex.ru
Moscow Aviation Institute (National Research University)

Monitoring the Use of Computer Engineering in Engineering Education

The methodology and results of long-term monitoring of Russian engineering education on the basis of applied information technologies are given. The indicators of the use of computer engineering in engineering education at different stages of its development, taking into account the readiness for this engineering education and the needs of a competitive industry are proposed. Expert assessments of each stage for these indicators are given.

Keywords: applied information technology, engineering education, computer engineering, expert evaluation, design-technological preparation of production, engineering, automated production, staffing, principle of division of labor

References

1. **Mezhdunarodnaya** ehnciklopediya CALS-tehnologij. Aviacionno-kosmicheskoe mashinostroenie. Gl. red. A. G. Bratuhin, (International encyclopedia of CALS-technologies. Aerospace engineering). Moscow, NIC ASK, 2015. 608 p. (in Russian).

2. **Yurin V. N.** *Kompjuterniy inzhiniring i inzhenernoe pbrozovanie*. (Computer engineering and engineering education). Moscow: Editorial URSS, 2002, 152 p. (in Russian).

3. **Yurin V. N., Zlygarev V. A.** Computer engineering — the basis of training of specialists for machine — building industry, *ICEE'95. International Conference of Engineering Education. UNESCO. Abstracts, Moscow, ICEE, 1995*, p. 23.

7. **Yurin V. N.** Rasshirenie ispol'zovaniya vozmozhnostej prikladnyh informacionnyh tehnologij pri obuchenii konstruktorskotekhnologicheskoy podgotovke proizvodstva. (The increasing use of the possibilities for applied information technology in teaching design and technological preparation of production, *Nauchnye trudy (Vestnik MATI)*, 2010, no. 16 (88), pp. 310—315 (in Russian).

4. **Zlygarev V. A., Yurin V. N.** Analiz proizvoditel'nosti inzhernogo truda dlya postroeniya CALS-sistem. *V Mezhdunarodnaya konferenciya-forum "Primenenie IPI (CALS) — tehnologij dlya povysheniya kachestva i konkurentosposobnosti naukoemkoj produkcii". Materialy konferencii*. (Performance analysis of engineering work for

the construction CALS systems). Moscow, MVM, 2003, pp. 75—79 (in Russian).

5. **Yurin V. N.** Sistemnaya organizaciya tekhnologicheskogo razvitiya v mashinostroenii: problemy i resheniya. (System organization of technological development in mechanical engineering: problems and solutions), *Tekhnodoktrina 2014. Sbornik dokladov. 1-j Vseros. forum tekhnologicheskogo liderstva Rossii*, Moscow, RUSTO, 2014, pp. 370—372 (in Russian).

6. **Yurin V. N.** Inzhenernoe obrazovanie i informacionnye tekhnologii: problemy i opyt ih resheniya. (Engineering education and information technology: problems and solutions), *Vestnik mashinostroeniya*, 1998, no. 5, pp. 44—51 (in Russian).

8. **Sovershenstvovanie** sistemy podgotovki kadrov dlya predpriyatij oboronno-promyshlennogo kompleksa. (Improving the system of personnel training for enterprises of military-industrial complex), *Materialy VII Vseros. soveshchaniya, Izhevsk, 21—22 oktyabrya 2014 g., FGBOU VPO "IzhGTU im. M. T. Kalashnikova"*, Izhevsk; INNOVA, 2014, 136 p. (in Russian).

9. **Ivanov A. V., Kuznecov O. V.** O sovershenstvovanii podgotovki professional'nyh kadrov dlya organizacij OPK. (About improvement of professional personnel training of companies for military-industrial complex), *Vysshee obrazovanie v Rossii*, 2015, no. 8—9, pp. 32—38 (in Russian).

10. **Kozhina T. D., Volkov S. A.** Sovershenstvovanie sistemy podgotovki kadrov dlya predpriyatij oboronno-promyshlennogo komple-

ksa. (Improving the system of personnel training for enterprises of military-industrial complex), *Bulletin of Rybinsk state aviation technological Academy*, 2016, no. 1 (36), pp. 123–129 (in Russian).

11. **Yurin V. N.** Komp'yuternye tekhnologii v uchebnom processe inzhenerenogo obrazovaniya. (Computer technology in the learning process of engineering education), *Informacionnye tekhnologii*, 1999, no. 3, pp. 45–46 (in Russian).

12. **Zlygarev V. A.** Osnovnye polozheniya programmy "NIC ASK — vuzam". (The main provisions of the programme "NIC ASK — universities"). *Novye materialy i tekhnologii — NMT-98, Tez. dokl. Vseros. nauchno-tekhn. konf. 17–18 noyabrya 1998 g.*, MATI, Moscow, LATMES, 1998, p. 390. (in Russian).

13. **Yurin V. N.** K informatizatsii inzhenerenogo obrazovaniya: konkurs "Komp'yuternyj inzhiniring" (To informatization of engineering education: competition "Computer engineering"), *Informacionnye tekhnologii*, 1999, no. 9, pp. 54–55 (in Russian).

14. **Saati T.** Prinyatie reshenij. Metod analiza ierarhij. (Decision-making. Method of hierarchy analysis). Moscow, Radio i svyaz', 1993, 320 p. (in Russian).

15. **Al'bom tekhnologicheskoy osnastki dlya stankov s CHPU v aviadvigatelestroenii, Uchebnoe posobie.** CH.1. Stanochnye prispособleniya dlya stankov s CHPU v aviadvigatele-stroenii. (Album of tooling for CNC machines in air engines. Textbook. Part 1. Machine accessories for CNC machine tools in aircraft engine building). V. F. Bez'yazychnyj,

V. D. Korneev, V. N. Livanov, etc. Pod obshch. red. V. F. Bez'yazychnogo. Moscow, Mashinostroenie, 2000. 147 p. (in Russian).

16. **Malikov S. B.** Metod analiza tekhnicheskogo riska pri organizatsii podgotovki opytnogo proizvodstva detalej. (The method of analysis of technical risk in the provision of training of pilot production parts). *Avtoref. diss. ... kand. tekhn. nauk.* Moscow, MATI, 2012. 19 p. (in Russian).

17. **Zlygarev V. A.** Tekhnologiya vnedreniya PLM-reshenij na osnove outsorsinga. (Implementation methods of PLM solutions based on outsourcing), *Informacionnye tekhnologii na sluzhbe oboronno-promyshlennogo kompleksa Rossii. Sb. trudov vtoroj konferencii. 10–12 aprelya 2013*, Moscow, ID Connect!, 2013, pp. 78–79 (in Russian).

18. **Yablochnikov E. I., Molochnik V. I., Fomina Yu. N.** Reinzhiniring biznes-processov proektirovaniya i proizvodstva, (Reengineering of business processes of design and production). *Uchebnoe posobie.* Saint Petersburg: SPbGUITMO, 2008, 152 p. (in Russian).

19. **Strategiya innovacionnogo razvitiya Rossijskoj Federacii na period do 2020 g.** (The strategy of innovative development of the Russian Federation for the period till 2020). *Uverzhdena rasporyazheniem Pravitel'stva RF ot 8 dekabrya 2011 goda № 2227-r.* [Elektronnyj dokument] — URL: <http://gov/garant.ru/SESSION/PDA/linkProxy?subjectId=70106125&linkType=65537>. (Data of access 15.04.2017) (in Russian).



Поздравляем

директора издательства "Новые технологии",
члена редколлегии журнала
"Информационные технологии"

**Бориса Игоревича
АНТОНОВА**

с юбилеем!

Борис Игоревич, Ваш творческий путь от редактора издательства "Недра" до члена Главной редакции научно-технической литературы Госкомиздата СССР, главного редактора литературы по машиностроению и горному делу, главного редактора и первого заместителя директора издательства "Машиностроение", директора Издательства "Новые технологии" является ярким примером служения отечественной научно-технической литературе.

Благодаря профессионализму, неиссякаемой энергии, активной жизненной позиции и щедрости души Вы по праву снискали высокий авторитет среди коллег. Ваш огромный жизненный и профессиональный опыт является примером истинного, беззаветного служения Родине и преданности любимому делу. Вы являетесь организатором издательства "Новые технологии" — одного из ведущих отечественных издательств в области приоритетных направлений науки и технологий. Вы внесли неоценимый вклад в становление и развитие целого ряда научно-технических журналов, входящих в перечень ВАК РФ, включая журнал "Информационные технологии".

Уважаемый Борис Игоревич!

Редакционный совет, редакционная коллегия и редакция журнала поздравляют Вас и желают Вам крепкого здоровья, большого счастья, новых творческих успехов в Вашей многогранной деятельности!

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова.*

Корректор *Н. В. Яшина.*

Сдано в набор 10.08.2017. Подписано в печать 22.09.2017. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT1017. 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.