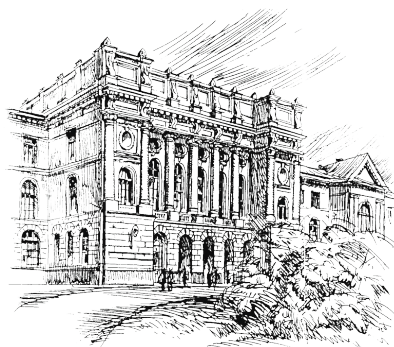


3(101)/2010



Научно-технические ведомости СПбГПУ

ИНФОРМАТИКА.
ТЕЛЕКОММУНИКАЦИИ.
УПРАВЛЕНИЕ

Санкт-Петербург. Издательство Политехнического университета

Министерство образования и науки

Санкт-Петербургский государственный политехнический университет

НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ СПбГПУ

РЕДАКЦИОННЫЙ СОВЕТ ЖУРНАЛА

Алферов Ж.И., академик РАН; *Васильев Ю.С.*, академик РАН (председатель);
Костюк В.В., академик РАН; *Лопота В.А.*, чл.-кор. РАН;
Окрепилов В.В., чл.-кор. РАН; *Рудской А.И.*, чл.-кор. РАН;
Патон Б.Е., академик РАН и НАН Украины; *Федоров М.П.*, чл.-кор. РАН;
Фортов В.Е., академик РАН.

РЕДАКЦИОННАЯ КОЛЛЕГИЯ ЖУРНАЛА

Васильев Ю.С., академик РАН (главный редактор); *Арсеньев Д.Г.*, д-р техн. наук, профессор;
Бабкин А.В., д-р экон. наук, профессор (зам. гл. редактора);
Боронин В.Н., д-р техн. наук, профессор; *Глухов В.В.*, д-р экон. наук, профессор;
Дегтярева Р.В., д-р ист. наук, профессор; *Иванов А.В.*, д-р техн. наук, профессор;
Иванов В.К., д-р физ.-мат. наук, профессор; *Козловский В.В.*, д-р физ.-мат. наук, профессор;
Рудской А.И., чл.-кор. РАН (зам. гл. редактора); *Юсупов Р.М.*, чл.-кор. РАН.

СЕРИЯ “ИНФОРМАТИКА. ТЕЛЕКОММУНИКАЦИИ. УПРАВЛЕНИЕ”

РЕДАКЦИОННЫЙ СОВЕТ СЕРИИ

Юсупов Р.М., чл.-кор. РАН – председатель;
Абрамов С.М., чл.-кор. РАН;
Воеводин В.В., чл.-кор. РАН;
Заборовский В.С., д-р техн. наук, профессор;
Козлов В.Н., д-р техн. наук, профессор;
Фотиади А.Э., д-р физ.-мат. наук, профессор;
Черноруцкий И.Г., д-р техн. наук, профессор.

РЕДАКЦИОННАЯ КОЛЛЕГИЯ СЕРИИ

Юсупов Р.М., чл.-кор. РАН – председатель;
Арсеньев Д.Г., д-р техн. наук, профессор – зам. председателя;
Бабкин А.В., д-р экон. наук, профессор – зам. председателя;
Болдырев Ю.Я., д-р техн. наук, профессор;
Давыденко М.О., технический секретарь;
Карпов Ю.Г., д-р техн. наук, профессор;
Коротков А.С., д-р техн. наук, профессор;
Макаров С.Б., д-р техн. наук, профессор;
Устинов С.М., д-р техн. наук, профессор;
Цикин И.А., д-р техн. наук, профессор;
Шабров Н.Н., д-р техн. наук, профессор.

Журнал выходит под научно-методическим руководством Российской академии наук с 1995 года.

Журнал включен в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

Журнал издается в пяти сериях:

Наука и образование;
Физико-математические науки;
Экономические науки;
Информатика, телекоммуникации, управление;
Гуманитарные и общественные науки.

Журнал зарегистрирован в Госкомпечати РФ. Свидетельство № 013165 от 23.12.94.

Подписной индекс **18390** в каталоге “Газеты. Журналы” Агентства “Роспечать”.

Журнал включен в базу данных “Российский индекс научного цитирования” (РИНЦ), размещенную на платформе Национальной электронной библиотеки на сайте <http://www.elibraru.ru>.

При перепечатке материалов ссылка на журнал обязательна.

Точка зрения редакции может не совпадать с мнением авторов статей.

Адрес редакции и издательства: Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29.
Тел. редакции серии (812) 552-62-16.

© Санкт-Петербургский государственный политехнический университет, 2010



Содержание

Проблемы передачи и обработки информации

Васильев А.Е., Литвинчук А.В., Петров Д.Д. Проектирование средств и систем обработки информации на основе сетей элементарных вычислителей нечетких логических функций.....	7
Когновицкий О.С. Анализ прямых и инверсных рекуррентных последовательностей.....	15
Журавлева Л.М. Повышение пропускной способности оптических линий связи.....	20

Радиотехника, антенны, СВЧ-устройства

Еремеев И.Ю., Замарин А.И., Семенюк С.С. Технология структурного анализа сигналов с псевдослучайной перестройкой несущей частоты в задачах радиомониторинга.....	27
Макаров С.Б., Рашич А.В., Атик С. Алгоритм приема спектрально-эффективных сигналов с OFDM.....	32

Системный анализ и управление

Ростов Н.В. Параметрическая оптимизация цифровых модальных регуляторов.....	39
Зубков А.Ф., Щербань А.Б., Семенов И.А. Структурно-синтаксический подход к поиску альтернатив управления сложными системами.....	45
Посов И.А., Рукшин С.Е. Архитектура системы проведения удаленных соревнований и организации работы с математическими задачами.....	49
Моледу М.Ф. Распределенное многокритериальное управление для ветропарков.....	54

Вычислительные машины и программное обеспечение

Ицыксон В.М., Глухих М.И. Язык спецификаций поведения программных компонентов.....	63
Васильев П.Н., Артамонов А.В., Маховенко Е.Б. Классификационная схема групповых подписей для построения распределенных приложений.....	71
Моисеев М.Ю. Автоматическое обнаружение дефектов в многопоточных программах методами статического анализа.....	77
Иванков А.А., Елисеев Д.С. Динамическое отслеживание модулем информационно-поисковой системы изменений в структуре или тексте интернет-ресурса.....	86
Сергеев Е.С., Гетманский В.В., Горобцов А.С. Перенос системы многотельной динамики на вычислительный кластер.....	93
Ицыксон В.М., Тимофеев Д.А. Технология модификации программного кода, основанная на параметризуемых шаблонах.....	99

Математическое моделирование: методы, алгоритмы, технологии

Беляев А.Б. Верификация алгоритма поддержки транзакционной памяти.....	107
Блеканов И.С., Бондаренко Д.С. Тематический краулинг на основе алгоритма HITS.....	111
Матвеев В.А. Исследование равновесных ситуаций в игровой задаче двух лиц с векторными выигрышами.....	119
Бабкова Е.В., Шерстюк С.С. Модель и алгоритм календарного распределения взаимозаменяемых ресурсов с учетом фронта выполнения работ.....	126

Месягутов М.А. Модификация метода ветвей и границ для поиска оптимального решения задачи одномерной упаковки прямоугольно-ориентированной структуры	130
Кочкаров А.А. , Хапаева Л.Х. Алгоритм распознавания предфрактального графа с чередованием затравок.....	134
Пашковский А.В. Повышение точности расчетов магнитного поля и силовых характеристик электромеханических устройств комбинированным методом стандартных и конечных элементов	140
Узденов А.А., Кочкаров Р.А. Алгоритм поиска внешне-внутреннего центра предфрактального графа с сохранением смежности старых ребер	145
Владимиров С.С. Моделирование процессов мажоритарного декодирования комбинации эквидистантного кода по k линейно-независимым элементам.....	149

Приборы, информационно-измерительные системы

Медведев А.В., Чапало И.Е. Применение цифровой обработки сигналов в системе охраны на основе межмодовой интерференции в волоконном световоде	157
---	-----

Международный конкурс-семинар «Таймырские чтения»

Мальчевский С.А. Использование мягких вычислений для оптимизации параметров девиации признаков при распознавании образов.....	164
Морозов А.И., Писарев А.И. Метод создания лингвистических переменных нечеткой модели	168
Фомичева С.Г., Елина Т.Н. Нейро-нечеткое моделирование стратегического взаимодействия участников рынка труда моноиндустриального региона.....	174
Попкова А.А., Фомичева С.Г. Методы априорной оценки степени агрегации и производительности запросов в хранилищах данных.....	182

Международный научно-практический семинар «Проблемы современной аналоговой микросхемотехники»

Крутчинский С.Г., Цыбин М.С. Аналого-цифровые преобразователи для смешанных систем на кристалле.....	192
Прокопенко Н.Н., Серебряков А.И., Манжула В.Г. Компенсация напряжения смещения нуля операционных усилителей с несимметричным включением активной нагрузки	196
Крутчинский С.Г., Титов А.Е. Мультидифференциальный операционный усилитель в режиме инструментального усилителя	200
Прокопенко Н.Н., Манжула В.Г., Белич С.С. Мультидифференциальный операционный усилитель с малым напряжением смещения нуля в условиях температурных и радиационных воздействий	204
Будяков А.С., Прокопенко Н.Н., Манжула В.Г. Анализ частотных характеристик входных каскадов СВЧ операционных усилителей для сложных функциональных блоков радиационно-стойкой аппаратуры	207

Научный семинар СПбГПУ

Черкесов Г.Н., Чуркин В.В. О проблеме расчета надежности восстанавливаемых систем при наличии запасных элементов.....	212
Сведения об авторах	213
Аннотации.....	217

Contents

Problems of transfer and information processing

Vassiliev A.E., Litvinchuk A.V., Petrov D.D. <i>Design of resources and systems of fuzzy-oriented data processing on the basis of networks of elementary fuzzy-logic solvers</i>	7
Kognovitzkiy O.S. <i>Analysis of direct and inverse recurrent sequences</i>	15
Zhuravleva L.M. <i>Capacity growth of the optical links</i>	20

Radio engineering, aeriels, SHF devices

Eremeev I.Y., Zamarin A.I., Semenuk S.S. <i>Technology of the structural analysis of frequency hopping signals in the tasks radiomonitoring</i>	27
Makarov S.B., Rashich A.V., Atik S. <i>Algorithm for spectrally efficient OFDM-signals reception</i>	32

The system analysis and management

Rostov N.V. <i>Parametrical optimization of digital modal controllers.....</i>	39
Zubkov A.F., Scherban A.B., Semenov I.A. <i>The structural-syntactic approach to search of alternatives of management by difficult systems.....</i>	45
Posov I.A., Rukshin S.E. <i>Architecture of a system for distance contests execution and organization of working with mathematical problems</i>	49
Mauledoux M.F. <i>Distributed control by multi-objective optimization for wind parks.....</i>	54

Computers and the software

Itsykson V.M., Glukhikh M.I. <i>A program component behavior specification language</i>	63
Vasilyev P.N., Artamonov A.V., Makhovenko E.B. <i>Classification scheme of group signatures for distributed applications development.....</i>	71
Ivankov A.A., Eliseev D.S. <i>WEB-page structure and text monitoring with a robot of search-engine</i>	77
Moiseev M.J. <i>Static analysis methods for automatic defect detection in multithreaded programs</i>	86
Sergeev E.S., Getmanski V.V., Gorobtsov A.S. <i>Transfer the system of multibodies dynamics to computing cluster</i>	93
Itsykson V.M., Timofeyev D.A. <i>Source code modification technology based on parametrized code patterns.....</i>	99

Mathematical modelling: methods, algorithms, technologies

Belyaev A.B. <i>Verification the algorithm of transactional memory.....</i>	107
Blekanov I.S., Bondarenko D.S. <i>Topical crawler the hits algorithm</i>	111
Matveev V.A. <i>The research of equilibria in two person game with vector payoffs.....</i>	119
Babkova E.V., Sherstyuk S.S. <i>Calendar allocation overtime resources model and algorithm with front of works</i>	126
Mesyagutov M.A. <i>A modification of branch-and-bound method for optimal solution of the 1 D packing problem of rectangular-oriented structure.....</i>	130

Kochkarov A.A., Hapaeva L.H. Algorithm of recognizing of prefractal graph with zapravka alternations	134
Pashkovsky A.V. Accuracy increase of calculations the magnetic field and power characteristics of electromechanical devices the combined method of standard and final elements	140
Uzdenov A.A., Kochkarov R.A. Algorithm of search of the outwardly-internal centre of pre-fractal graphs with preservation of contiguity of old edges	145
Vladimirov S.S. Modelling of major decoding processes of equidistant code combination with k linear-independent elements	149

Devices, information-measuring systems

Medvedev A.V., Chapalo I.E. Alarm system on basis of multimode interference in optical fiber with digital signal processing	157
--	-----

The international competition-seminar «Tajmyrsky readings»

Malchevskiy S.A. Soft computing for optimization deviation's parameters of features in pattern recognition	164
Morozov A.I., Pisarev A.I. Method of automatically creating of a fuzzy model industrial object	168
Fomicheva S.G., Yelina T.N. Neuro-fuzzy modeling of strategic interaction of participants of the labour market in monoindustrial region	174
Popkova A.A., Fomitcheva S.G. Aprioristic methods of the estimation for aggregation degree and productivity of calculations in data warehouse	182

The international scientifically-practical seminar «Problems of modern analogue micro circuitry»

Krutchinsky S.G., Tsybin M.S. Analog to digital converters multimodal SNK	192
Prokopenko N.N., Serebrjakov A.I., Manzhula V.G. Way of indemnification of the regular component of pressure of displacement of zero of operational amplifiers with asymmetrical inclusion of active loading.	196
Krutchinsky S.G., Titov A.E. Multidifferentsialny operational amplifier in treatment instrumentation amplifiers.	200
Prokopenko N.N., Manzhula V.G., Belich S.S. The multidifferential operational amplifier with small pressure of displacement of zero in the conditions of temperature and radiating influences	204
Budjakov A.S., Prokopenko N.N., Manzhula V.G. The analysis of frequency characteristics of subcircuits of entrance cascades for GHZ range of operational amplifiers for difficult functional blocks radiatsionnos-tojkoj of equipment of new generation	207

Scientific seminar SPbSPU

Cherkesov G.N., Tchurkin V.V. About the problem of calculation of reliability of restored systems in the presence of spare elements	212
About the authors	213
Abstracts	217

УДК 004.383.8

А.Е. Васильев, А.В. Литвинчук, Д.Д. Петров

ПРОЕКТИРОВАНИЕ СРЕДСТВ И СИСТЕМ НЕЧЕТКОЙ ОБРАБОТКИ ИНФОРМАЦИИ НА ОСНОВЕ СЕТЕЙ ЭЛЕМЕНТАРНЫХ ВЫЧИСЛИТЕЛЕЙ НЕЧЕТКИХ ЛОГИЧЕСКИХ ФУНКЦИЙ

Современные системы управления, как правило, функционируют в условиях действия различного рода недетерминированных факторов, в том числе связанных с изменением структуры или параметров объекта управления, с непостоянством внешней среды, с изменением требований метасистемы верхнего уровня к режимам управления. Учет воздействия этих факторов приводит к усложнению описания и реализации системы, в связи с чем при проектировании широко используются методы теории интеллектуальных систем. Так, в частности, для систем управления, в которых применение классических регуляторов оказывается малоэффективным, целесообразно использование методов и средств нечеткой обработки информации, что позволяет существенно снизить трудозатраты на разработку, при этом сохранить, а в ряде случаев – улучшить качество функционирования системы управления [1, 2].

Создание базы нечетких знаний (БНЗ) – это ключевой этап разработки системы нечеткой обработки информации, определяющий результативность применения аппарата нечетких вычислений и удовлетворение требований к показателям качества управления.

Источниками информации для построения БНЗ с учетом специфики области применения системы управления могут являться аналитически заданное соотношение «вход–выход», передаточная функция, представленная графиком, таблицей, набором данных об экспертном управлении, совокупность критериев качества, предъявляемых к системе управления и др. [4]. Обобщенный алгоритм построения БНЗ с учетом различий источников информации показан на рис 1.

Эвристический синтез нечетких систем проводится с применением как непосредственного задания БНЗ, так и на основе анализа серии экспертных решений. При использовании эвристического подхода базы значительных объемов могут оказаться неполны или противоречивы; для решения этой проблемы целесообразен переход к автоматизации процедуры генерации БНЗ.

В свою очередь, подходы, основанные на автоматизации синтеза, преобразуют базу данных «вход–выход» в БНЗ путем формальных преобразований (например, с помощью сочетания регрессионного и кластерного анализа) или итерационно, с применением так называемого «самообучения», например, на основе нейронных сетей. Так, в частности, сочетание нечетких и нейровычислителей получило название «ANFIS» (adaptive neuro-fuzzy inference system – адаптивные системы нейро-нечеткого вывода; другое название – «гибридные сети»).

Несмотря на очевидные достоинства, указанные подходы к автоматизации обладают рядом недостатков, наиболее значительными из которых являются:

чувствительность к начальным условиям – в зависимости от исходных данных и параметров работы алгоритма могут быть сгенерированы системы, близкие, с точки зрения функционирования, но существенно различные по своей структуре;

затруднительность анализа, в частности, выполнения оценки сложности и эквивалентности сгенерированных структур;

итерационность процедуры – в общем случае предполагается возврат на начальные стадии

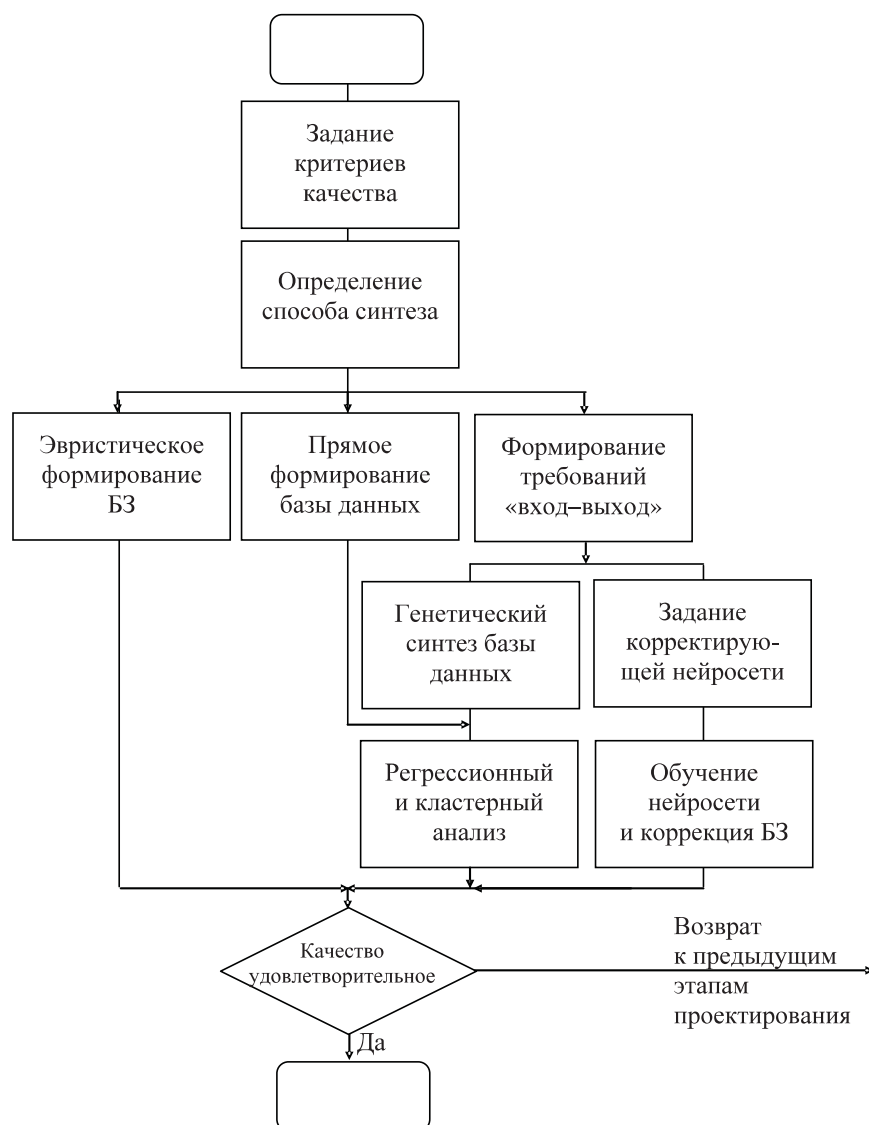


Рис. 1. Обобщенный алгоритм проектирования баз нечетких знаний

алгоритма и вмешательство в процесс синтеза разработчика.

Таким образом, значимость заявленной проблематики обусловлена необходимостью автоматизации анализа и синтеза подсистем принятия решений с нечеткой обработкой информации для устройств управления при одновременной минимизации трудоемкости анализа и синтеза: актуальным является развитие теории синтеза нечетких систем для предметной области встраиваемых приложений, направленной на улучшение качества и снижение трудоемкости проектирования устройств управления, повышение качества функционирования встраиваемых приложений [3, 5].

Авторами предлагается графовая модель нечеткого преобразователя – «нечеткая сеть», в узлах которой расположены элементарные нечеткие вычислители. Такое представление позволит свести проблему синтеза нечеткого преобразователя к построению системы связей между элементами («коммуникационной подсистемы») и поможет создать на относительно простой и недорогой технической базе достаточно сложные структуры с нечеткой обработкой информации.

Подобный подход применяется, в частности, в задачах синтеза цифровых комбинационных схем на функционально полном базисе (например, на базисе элементов с логической функцией «2И-НЕ»), а также в задачах композиции

конечных автоматов при построении сложных устройств. Предлагаемый метод синтеза отличается областью определения функций, характерной для нечетких множеств.

Наиболее близкой к рассматриваемой является идея организации нечетких элементов в сети, высказанная ранее Кофманом [2]. Однако сети нечетких элементов Кофмана включают произвольные элементы и ориентированы на анализ функций нечетких переменных.

В данной работе предлагается использовать только элементарные нечеткие преобразователи и обеспечить автоматизацию формального синтеза графовых моделей, вычисляющих заданные передаточные функции.

Утверждение о принципиальной реализуемости подобного подхода базируется на следующих фактах. Принципиальная возможность реализации:

любой аналитически заданной зависимости с наперед заданной точностью на основе нечетких вычислителей:

любой аналитически заданной зависимости с наперед заданной точностью на однотипных элементах двоичной логики функционально полного базиса;

любой системы нечетких преобразований с наперед заданной точностью на цифровых вычислителях.

Для решения поставленных задач требуется разработать математический аппарат декомпозиции нечетких соотношений, совокупность ме-

тодов оценивания сложности и эквивалентности нечетких сетей, совокупность методик синтеза нечетких сетей.

Предлагаемый способ задания нечеткой сети подразумевает использование следующих компонентов: входные контакты; линии связи (в т. ч. с задержками распространения сигнала); линейные (нелинейные) нечеткие преобразователи; усилительные элементы; выходные контакты.

Структура инструментального средства для проектирования и анализа нечетких сетей показана на рис. 2; пример задания структуры нечеткой сети приведен на рис. 3.

Выполнение элементарных нечетких вычислений в соответствии с маршрутами распространения информации по заданному ориентированному графу нечеткой сети может быть обеспечено при помощи универсальной программы-эмулятора, выполняющей однотипные вычисления по заданной схеме связей. Кроме того, такие вычисления могут быть реализованы на аппаратном вычислителе, построенном на основе схем программируемой логики.

Комментируя выбор программного или аппаратного варианта реализации, следует отметить, что применение программных эмуляторов нечетких вычислений в некоторых случаях сопряжено с проблемой времени выполнения эмуляции: быстроедействие системы принятия решений, реализованной на программном нечетком вычислителе, может оказаться неудовлетворительным с точки зрения требований к скорости вычислений. Таким образом, экономически выгодные программ-

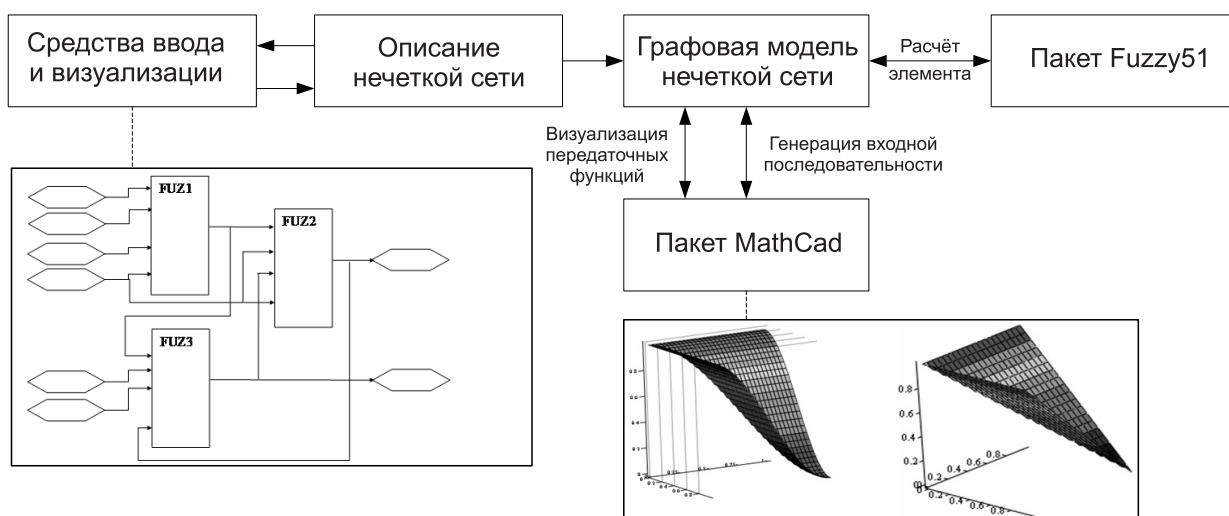


Рис. 2. Структура инструментального программного обеспечения

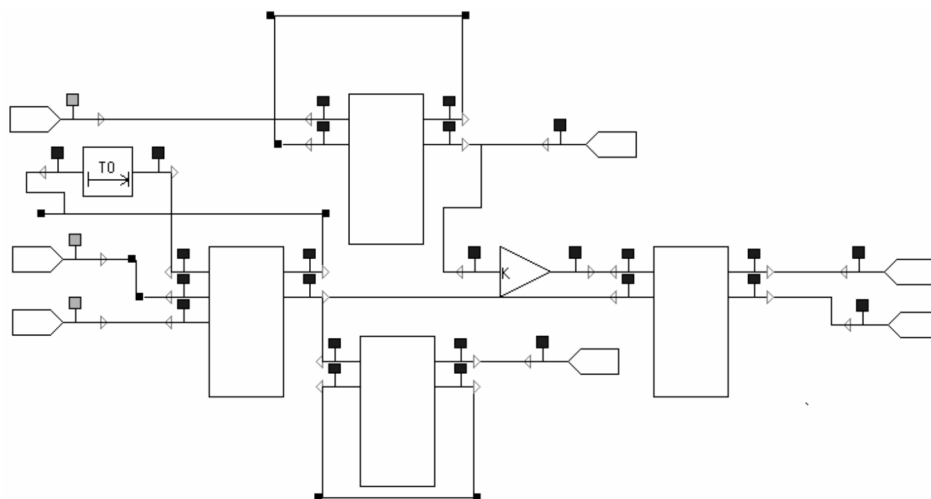


Рис. 3. Пример задания нечеткой сети

ные эмуляторы нечетких вычислений для микроконтроллеров общего назначения могут оказаться технически непригодными в ряде высокоскоростных приложений.

Так, например, специализированные вычислители – процессоры видеосигналов – реализуют сложные передаточные функции, работают в реальном масштабе времени и имеют нетиповую, уникальную структуру аппаратных средств, в связи с чем их разработка характеризуется значи-

тельной трудоемкостью. Передаточная функция таких вычислителей может быть описана нечеткой сетью, формирование которой может быть автоматизировано с помощью предлагаемых методов и средств, а выполнение нечетких вычислений на аппаратном уровне позволит обеспечить требуемые характеристики быстродействия.

На рис. 4 представлена структура системы управления, построенной с применением аппаратного вычислителя нечетких логических функций.

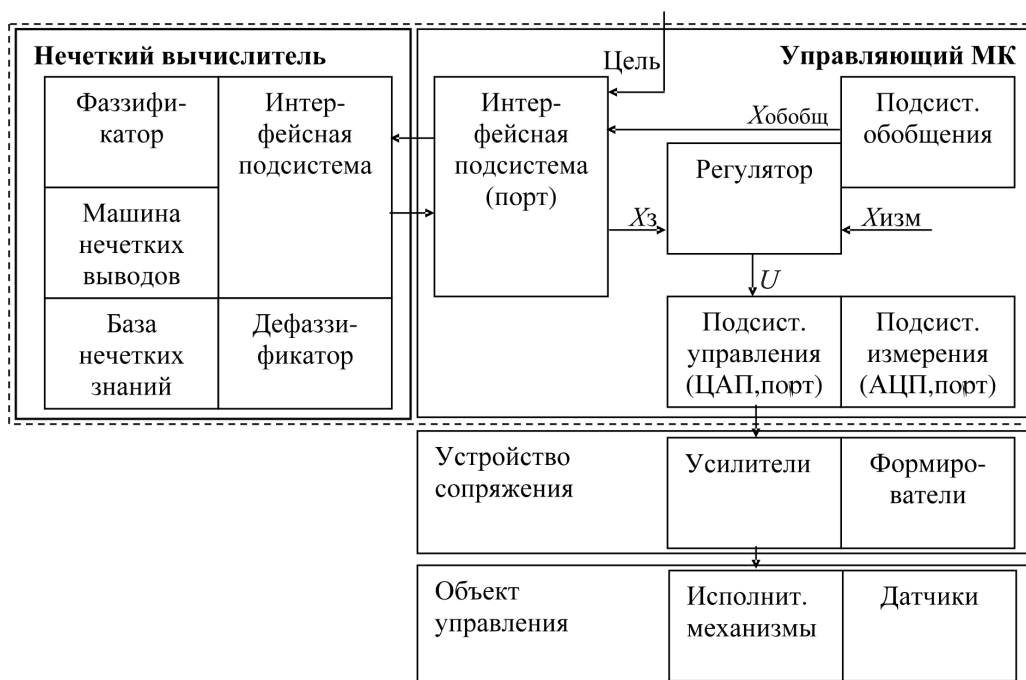


Рис. 4. Обобщенная структура системы управления с аппаратным вычислителем нечетких логических функций

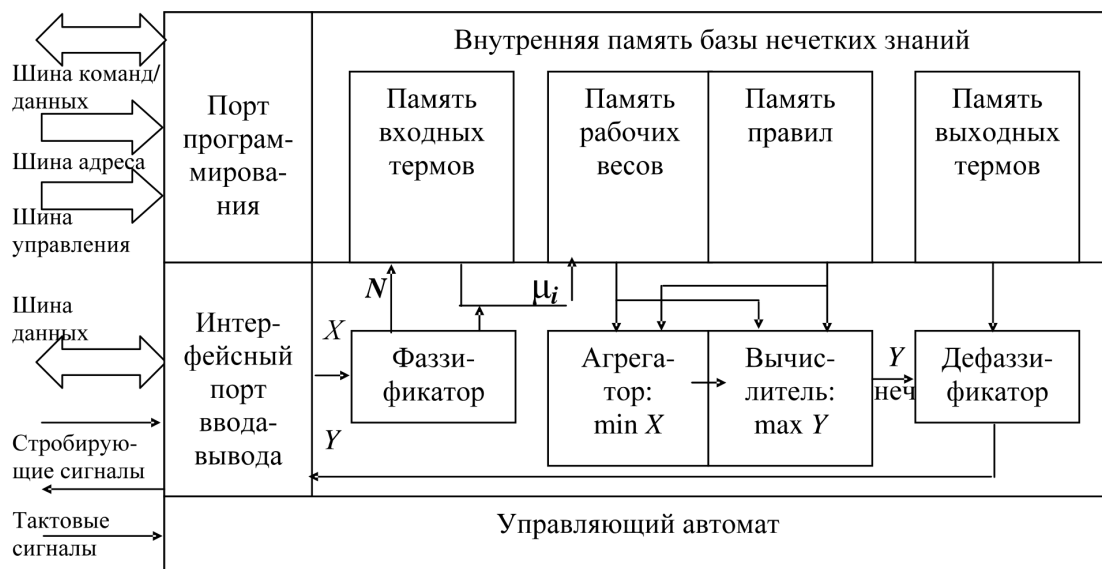


Рис. 5. Обобщенная структура автономного нечеткого вычислителя

Такой способ организации предполагает, что на нечетком вычислителе реализуется тактический уровень управления (с задаваемой извне целью), а на управляющем МК – исполнительный.

Одними из первых вычислители со встроенной аппаратной поддержкой нечетких вычислений были созданы в AT&T Bell Labs и Togai InfraLogic в 1986–1989 гг. В настоящее время сектор средств аппаратного обеспечения нечетких вычислений, несмотря на специфичность изделий, неуклонно расширяется.

Применяемые в настоящее время нечеткие вычислители можно разделить на две категории:

нечеткие решатели, выполненные в виде отдельного модуля или микросхемы, предназначенные для работы в кооперации с основным контроллером (в этом случае в англоязычной литературе для их обозначения применяется термин *Fuzzy Coprocessor* – нечеткий сопроцессор). К представителям этой группы относятся SAE81C99 (Siemens), WARP1.x (STMicroelectronics) и др.;

микроконтроллеры, содержащие особое периферийное устройство, связанное с АЛУ – блок нечеткой обработки, представляющие собой законченную аппаратную платформу для системы управления (для их обозначения применяется термин *ICU: Intelligent Controller Unit* – интеллектуальный контроллер). К таким fuzzy-контроллерам относятся ST52 (STMicroelectronics), 68HC12 (Motorola) и др.

Достоинство нечетких вычислителей перво-

го типа (рис. 5) – пригодность для расширения вычислительных возможностей и повышения производительности систем, выполненных на базе микроконтроллеров практически любого типа, за счет использования достаточно простого и универсального интерфейса обмена; недостаток – сравнительно высокая стоимость.

Нечеткие вычислители второго типа имеют структуру, объединенную на рис. 4 пунктирной линией. Их достоинством является функциональная завершенность, недостатком – специфичность и слабая распространенность системы команд.

Выбор типа вычислителя неоднозначен и определяется спецификой его конкретного применения, т. е. задача создания нечетких вычислителей обоих типов представляется по-прежнему актуальной. При этом в обоих случаях нечеткий решатель (внешний или внутренний) полностью обеспечивает расчет графовой модели нечеткой сети и предоставляет результаты системе управления.

Разработанный авторами аппаратный вычислитель-сoproцессор (первый тип) реализован на базе микросхем программируемой логики. Функционирование вычислителя демонстрируется на примере встраиваемой интеллектуальной системы управления (ВИСУ), решающей задачу поддержания заданной амплитуды колебаний маятника (рис. 6).

На рис. 7 показан вариант подключения модуля нечетких вычислений к управляющему микроконтроллеру.

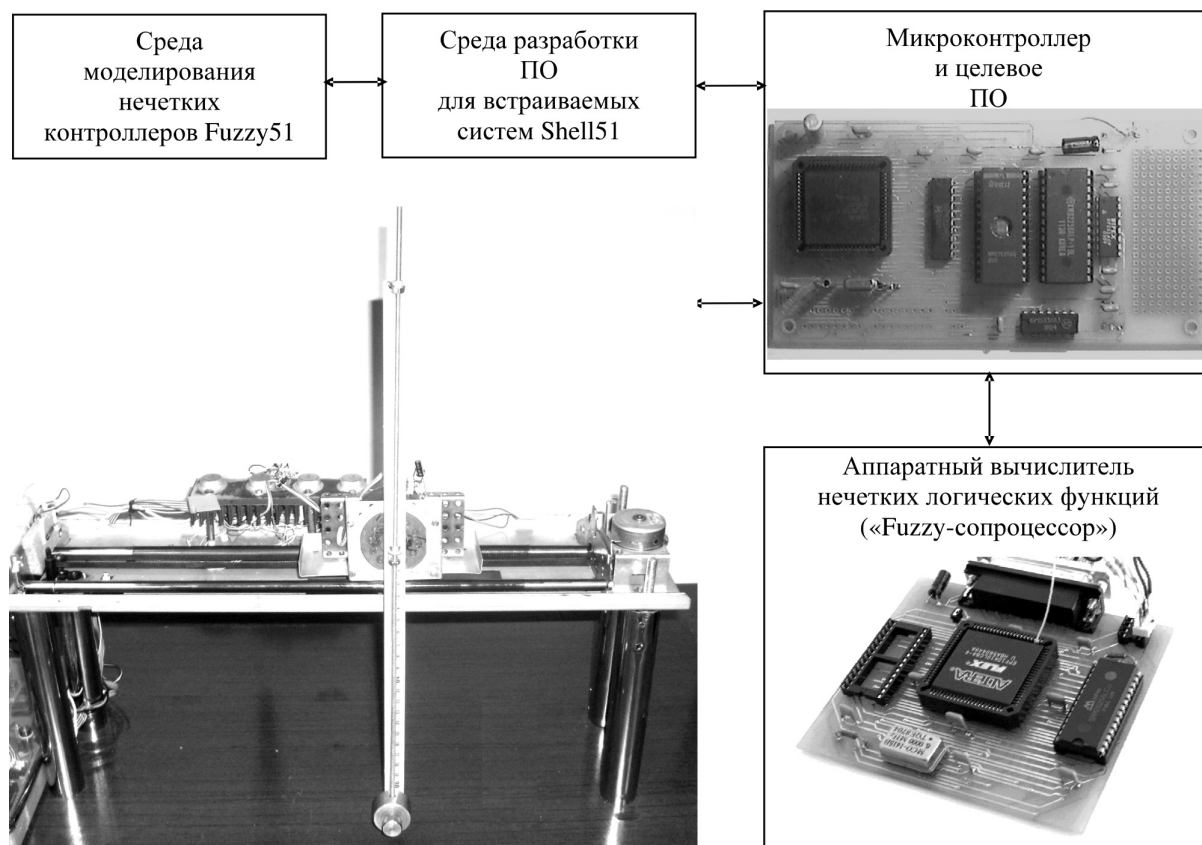


Рис. 6. ВИСУ с нечетким сопроцессором

По окончании инициализации сопроцессора схемой сброса, происходит загрузка в ПЛИС данных из конфигурирующей памяти прошивки, хранящей настройку макроячеек ПЛИС, обеспечивающую аппаратное выполнение алгоритмов нечетких вычислений. По окончании прошивки микросхема ПЛИС становится нечетким сопроцессором с внешней памятью баз нечетких знаний (БНЗ), размещаемой (в зависимости от позиции переключателя, установленного пользователем) в ОЗУ (что удобно для отладки) или в заранее запрограммированном ПЗУ (для эксплуатации отлаженного модуля с неизменяемой БНЗ).

Интеллектуальный контроллер (второй тип нечетких вычислителей) и средства его автоматизированного проектирования (рис. 8), разрабатываемые авторами в научной группе ВИСУ, имеют следующие принципиальные отличия от существующих:

- возможность конфигурирования микроконтроллера целевым пользователем;

- совместимость по системе команд с ядром MCS-51;

вычислительная автономность нечеткого периферийного модуля.

Последнее отличие предполагает, что процесс нечетких вычислений полностью обеспечивается этим модулем, при этом процессорное ядро микроконтроллера имеет возможность продолжать исполнение команд основной программы, не переключаясь на вспомогательные вычисления.

Актуальность дальнейшего развития данной проблематики для теории состоит в создании совокупности методов, обеспечивающих оценивание сложности и степени подобию нечетких преобразователей информации для встраиваемых интеллектуальных систем управления, а также обеспечивающих возможность синтеза нечетких преобразователей информации для аппаратных реализаций во встраиваемых системах.

Ценность дальнейшего развития данной проблематики для практики заключается в снижении ресурсоемкости встраиваемых реализаций нечетких преобразователей информации, синтезированных с применением предложенных методов,

и в разработке инструментальных комплексов автоматизированной генерации внутрикристальных нечетких преобразователей информации, использующих предложенные методы анализа и синтеза.

Рассмотренные подходы позволят повысить скорость внедрения и показатели производительности встраиваемых интеллектуальных систем управления при одновременной минимизации трудозатрат на их проектирование.

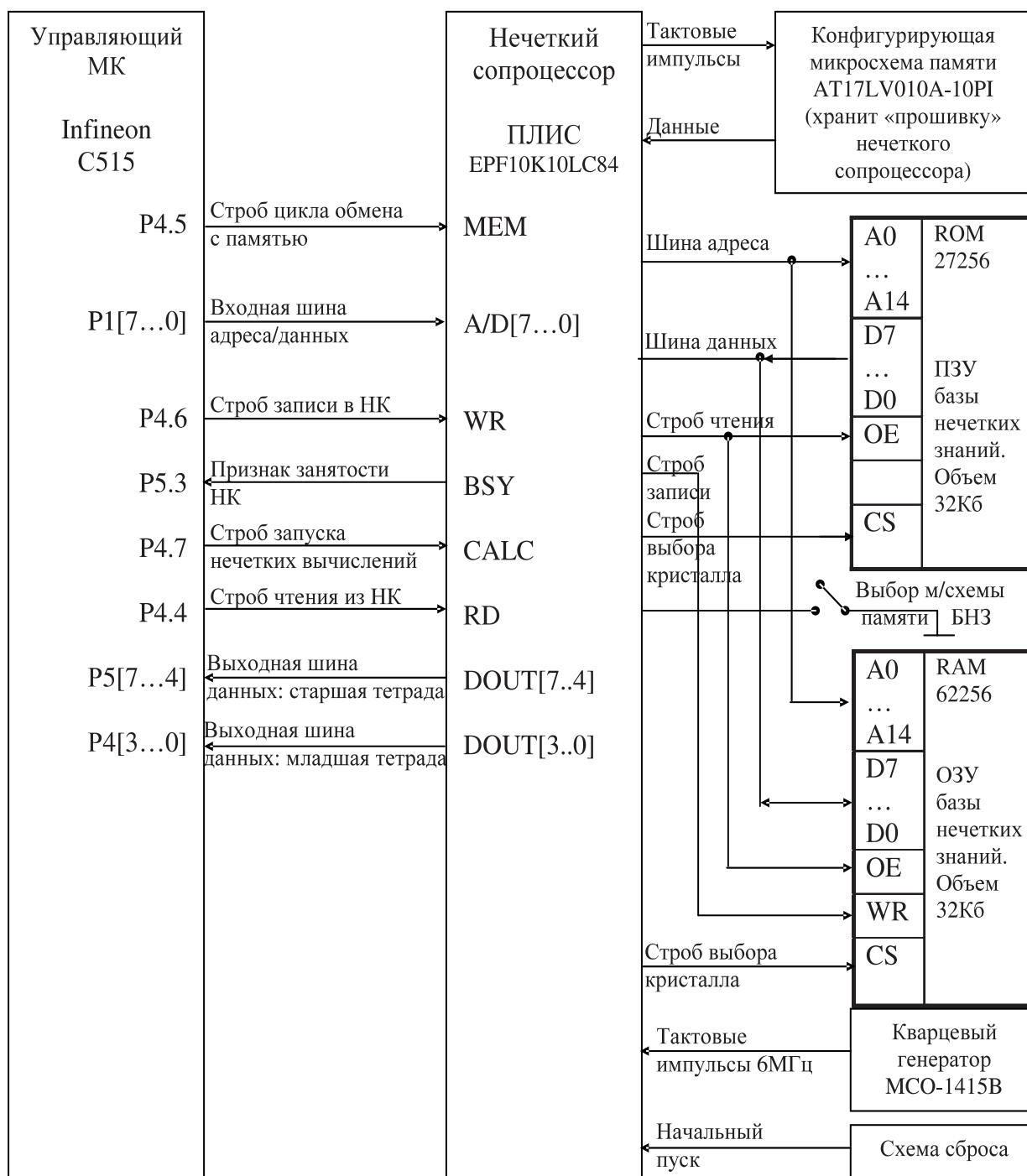


Рис. 7. Схема подключения модуля нечетких вычислений

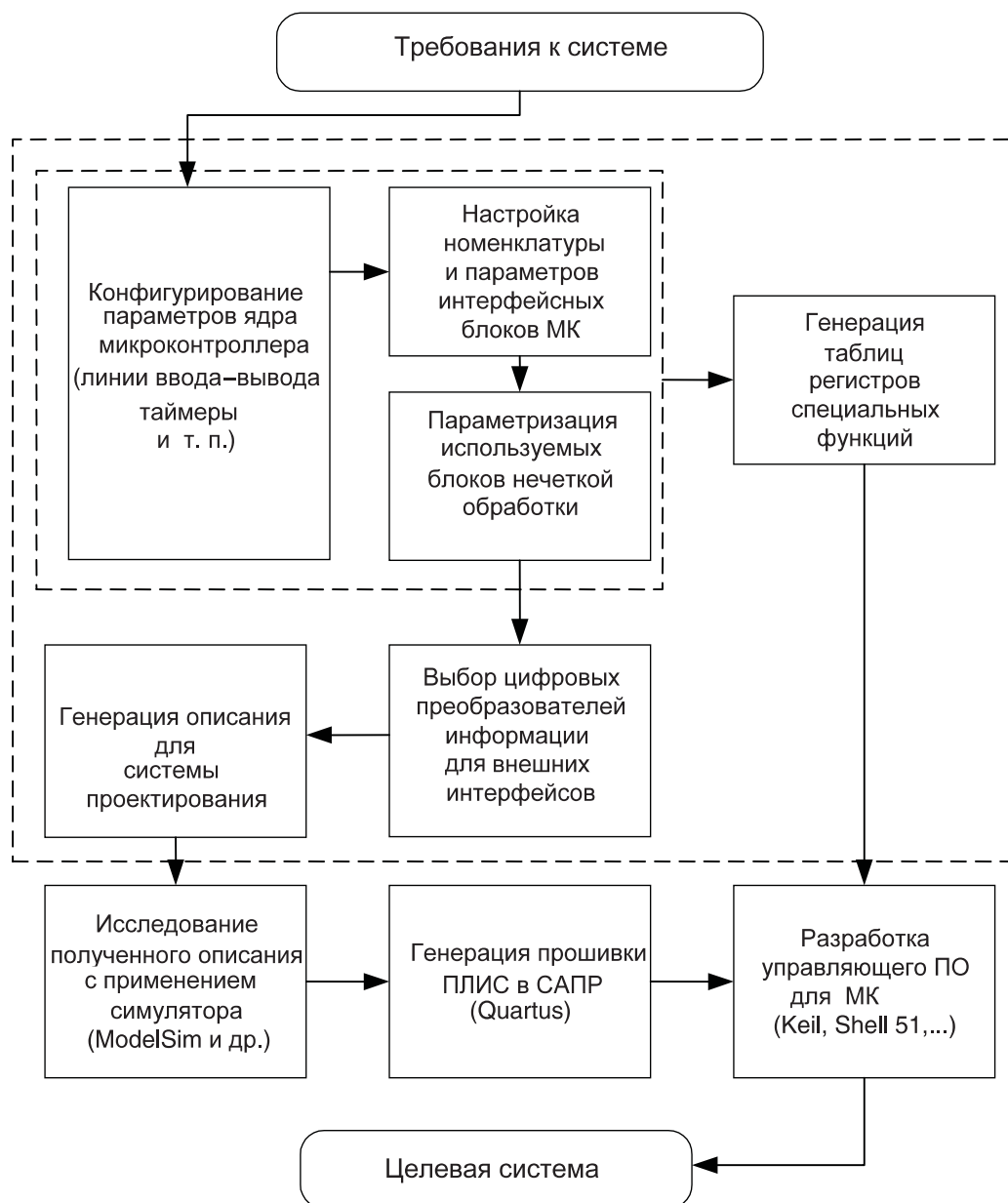


Рис. 8. Структура средств САПР микроконтроллеров с нечеткими решателями

СПИСОК ЛИТЕРАТУРЫ

1. Рассел, С. Искусственный интеллект: Современный подход [Текст]/С. Рассел, П. Норвиг; пер. с англ. –М.: Вильямс, 2006. – 1408 с.
2. Кофман, А. Введение в теорию нечетких множеств [Текст]/А. Кофман. –М.: Радио и связь, 1982. – 432 с.
3. Джонс, М.Т. Программирование искусственного интеллекта в приложениях [Текст]/М.Т. Джонс; пер. с англ. – М.: ДМК Пресс, 2006. – 312 с.
4. Борисов, А.Н. Обработка нечеткой информации в системах принятия решений [Текст]/А.Н. Борисов, А.В. Алексеев, Г.В. Меркурьева [и др.]. –М.: Радио и связь, 1989. – 304 с.
5. Васильев, А.Е. Микроконтроллеры. Разработка встраиваемых приложений [Текст]/А.Е. Васильев. – СПб.: БХВ-Петербург, 2008. – 304 с.



АНАЛИЗ ПРЯМЫХ И ИНВЕРСНЫХ РЕКУРРЕНТНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

С целью расширения спектра сигнала и повышения помехоустойчивости в широкополосных системах часто применяют специальные последовательности, например, М-последовательности, период которых по времени соответствует длительности элементарного сигнала [1]. При использовании сигналов с прямым расширением спектра (DSSS) единичный элементарный сигнал передается одним периодом прямой М-последовательности, а нулевой – ее инверсией. Таким образом, для передачи двоичных 1 и 0 используются противоположные последовательности $\{s\}$ и $\{\bar{s}\}$. Это может служить также средством криптозащиты передаваемых данных.

Кроме этого, М-последовательности и последовательности Гоулда применяют в асинхронно-адресных системах и в системах связи с кодовым разделением.

Восстановление на приеме исходной информационной последовательности осуществляется чаще всего с помощью корреляторов или согласованных фильтров. В современных широкополосных и сверхширокополосных системах периоды таких последовательностей могут быть достаточно большими, что может послужить препятствием для использования корреляторов или согласованных фильтров для их выделения.

Учитывая рекуррентный характер М-последовательностей или других аналогов, применяемых в большинстве случаев для расширения спектра, более эффективным по времени поиска такого сигнала и по объему материальных затрат может быть цифровой (дискретный) коррелятор. Известные корреляторы такого типа используют свойство рекуррентности М-последовательности (например, метод Уорда). Однако такой рекуррентный коррелятор ранее не применялся для выделения инверсной последовательности, судя по литературным источникам.

Вместе с тем, как показано ниже, и прямая рекуррентная последовательность, и ее инверсная последовательность могут рассматриваться как последовательности, порождаемые одним и

тем же характеристическим многочленом, и удовлетворять одному и тому же рекуррентному соотношению. Это позволяет применять для их выделения единый метод на основе двойственного базиса. Кроме того, количество используемых комбинаций увеличивается в два раза.

Пусть имеется некоторая линейная рекуррентная последовательность $\{s\}$ с периодом L , порождаемая характеристическим многочленом $g(x)$.

Тогда прямая последовательность $\{s\}$ и ее инверсия $\{\bar{s}\}$ представляют собой последовательности, удовлетворяющие одному и тому же рекуррентному уравнению, соответствующему характеристическому многочлену:

$$P(x) = (x + 1) \cdot g(x). \quad (1)$$

По аналогии с кодами БЧХЭ или ЛРД-последовательностями, последовательность $\{P\}$, соответствующая многочлену $P(x)$, будет равна сумме по mod2 последовательности $\{s\}$ и постоянного элемента $t_0 = 0$ или $t_0 = 1$. Если начальный элемент для двучлена $(x + 1)$ $t_0 = 0$, то $\{P\} = \{s\}$, т. е. формируется прямая последовательность $\{s\}$. Если $t_0 = 1$, то $\{P\} = \{s\} + \{1\} = \{\bar{s}\}$, т. е. будет сформирована инверсная последовательность.

Так как обе последовательности – и прямая, и обратная, – порождаются одним и тем же характеристическим многочленом $P(x)$ (1), то обе они будут удовлетворять одному и тому же рекуррентному соотношению.

К такого рода последовательностям можно отнести М-последовательности, последовательности Гоулда, коды БЧХЭ, ЛРД-последовательности.

Рассмотрим примеры этих последовательностей.

М-последовательности. Пусть у нас имеется М-последовательность $\{s\}$, порождаемая характеристическим многочленом $P_1(x) = 1 + x + x^4$ с первообразным корнем ε :

$$\begin{aligned} \{s\} &= \{T(\varepsilon^5)T(\varepsilon^6)T(\varepsilon^7)T(\varepsilon^8) \dots \\ &\dots T(\varepsilon^{14})T(1)T(\varepsilon)T(\varepsilon^2)T(\varepsilon^3)T(\varepsilon^4)\} = \end{aligned}$$

$$= (011010111100010) = \\ = (s_0 s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10} s_{11} s_{12} s_{13} s_{14}).$$

Такая М-последовательность в соответствии с характеристическим многочленом $P_1(x)$ удовлетворяет рекуррентному уравнению:

$$s_i = s_{i-4} + s_{i-3}; \quad i \geq 4. \quad (2)$$

Очевидно, что в зависимости от начального элемента ε^i в модулярном регистре для многочлена $P_1(x)$ может быть сформировано $(2^4 - 1)$ М-последовательностей, отличающихся только начальной фазой ε^i . Множество таких комбинаций представляет собой эквидистантный циклический код $(n, k) = (15, 4)$ с постоянным весом, равным 8.

Теперь рассмотрим циклический несистематический код БЧХЭ $(15, 5)$, комбинации которого представляют собой последовательности с характеристическим многочленом:

$$P(x) = P_1(x) \cdot P_2(x) = (1+x+x^4)(1+x) = 1+x^2+x^4+x^5.$$

Схема кодирующего устройства представлена на рис.1.

Как видно из рисунка, выходная последовательность $\{v\}$ будет равна М-последовательности $\{s\}$, каждый элемент которой складывается по mod2 с элементом b_0 . Если элемент $b_0 = 0$, то $\{v\} = \{s\}$, т. е. на выходе будет М-последовательность. Если же $b_0 = 1$, то $\{v\} = \{\bar{s}\}$, т. е. будет сформирована инвертированная М-последовательность.

При этом последовательность $\{v\} = \{v_0 v_1 v_2 v_3 v_4 v_5 \dots v_{12} v_{13} v_{14}\}$ будет удовлетворять рекуррентному уравнению:

$$v_i = v_{i-1} + v_{i-3} + v_{i-5}; \quad i \geq 5. \quad (3)$$

Легко проверить, что М-последовательность $\{s\}$ удовлетворяет уравнению (2) и (3), а инверсия последовательности – только уравнению (3).

Таким образом, если для передачи данных используются как прямые М-последовательности, так и инверсные, то их выделение можно осуществить как декодирование комбинаций циклического кода БЧХЭ (n, k) , в данном примере – кода $(15, 5)$, с использованием двойственного базиса [2].

Для выбранного многочлена $P(x)$ коэффициенты двойственного базиса будут иметь значения:

относительно $P_1(x) = 1 + x + x^4$:

α_1	α_2	α_3	α_4	α_5
ε^{10}	ε^9	ε	1	ε^{11}

относительно $P_2(x) = 1 + x$:

β_1	β_2	β_3	β_4	β_5
1	1	0	0	1

Пусть принятая последовательность имеет вид:

$$(v_0 \ v_1 \ v_2 \ v_3 \ v_4 \ v_5 \ v_6 \ v_7 \ v_8 \ v_9 \ v_{10} \ v_{11} \ v_{12} \ v_{13} \ v_{14}) \\ (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1)$$

Теперь по m -элементным участкам (m – степень многочлена $P_1(x)$) найдем начальные элементы C для многочлена $P_1(x)$ и D – для $P_2(x)$ в соответствии с выражениями [2]:

$$A = \varepsilon^{-n} (\alpha_1 v_n + \alpha_2 v_{n+1} + \dots + \alpha_m v_{n+m-1}), \\ D = (\beta_1 v_n + \beta_2 v_{n+1} + \dots + \beta_m v_{n+m-1}). \quad (2)$$

Пусть выделен безошибочный участок

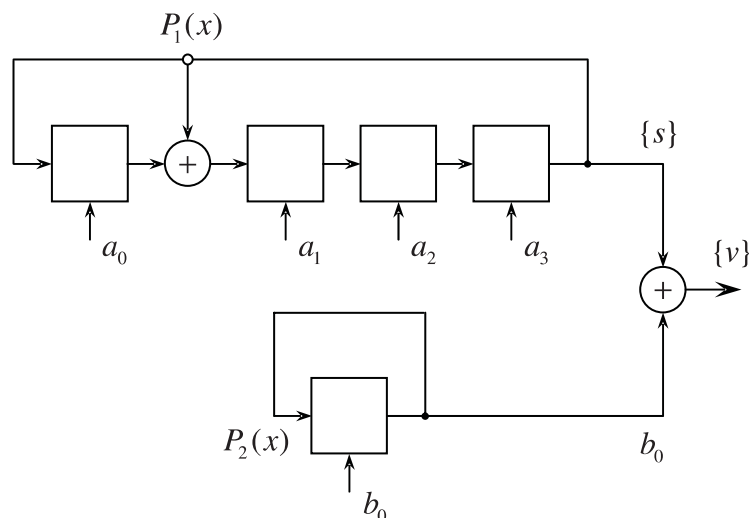


Рис. 1. Кодирующее устройство кода БЧХЭ $(15, 5)$



$(v_3v_4v_5v_6v_7) = (10100)$. Тогда $C = \varepsilon^{-3}(\alpha_1 + \alpha_3) = \varepsilon^{-3}(\varepsilon^{10} + \varepsilon) = \varepsilon^{-3} \cdot \varepsilon^8 = \varepsilon^5 \pmod{P_1(x)}$, аналогично $D = (\beta_1 + \beta_3) = 1 + 0 = 1$. Следовательно, $(a_0a_1a_2a_3) = (0110)$, а $b_0 = 1$.

То есть, из канала поступила инвертированная М-последовательность $\{s\}$ с ее начальным фазовым элементом $C = \varepsilon^5$.

Рассмотренный код БЧХЭ (15,5) содержит всего 32 кодовых комбинации, из них 15 прямых М-последовательностей ($b_0 = 0$), 15 инвертированных М-последовательностей ($b_0 = 1$) и еще две тривиальных последовательности – нулевая ($b_0 = 0$) и единичная ($b_0 = 1$) при $(a_0a_1a_2a_3) = (0000)$.

Прямые и инверсные последовательности Гоулда. Прямые последовательности Гоулда с периодом $(2^k - 1)$ представляют собой рекуррентные последовательности $\{s\}$, соответствующие характеристическому многочлену $P_1(x) = f_1(x) \cdot f_2(x)$, где $f_1(x)$ и $f_2(x)$ – примитивные многочлены степени k . Множество этих последовательностей с различными начальными фазами составляет $(2^k - 1)^2$ комбинаций $\{s\}$.

Если теперь в качестве характеристического многочлена выбрать расширенный многочлен $P(x) = (1 + x)P_1(x) = f_1(x) \cdot f_2(x) \cdot f_3(x)$, то мы получим циклический код БЧХЭ, состоящий из $(2^k - 1)^2$ прямых последовательностей Гоулда (при $b_0 = 0$), $(2^k - 1)^2$ инверсных последовательностей Гоулда (при $b_0 = 1$), по $2(2^k - 1)$ прямых и инверсных М-последовательностей с длиной периода $(2^k - 1)$ и двух тривиальных последовательностей – нулевой и единичной. Таким образом, будет всего 2^{2k+1} комбинаций.

Все эти последовательности будут удовлетворять одному и тому же рекуррентному соотношению. Следовательно, «прямые» и «инверсные» последовательности Гоулда представляют собой рекуррентные последовательности одного вида и могут декодироваться мажоритарно с использованием теории двойственного базиса. Одновременно с этим, «прямые» последовательности Гоулда могут декодироваться как рекуррентные последовательности на основе рекурсии, соответствующей многочлену $P_1(x)$.

Пусть, например, $P_1(x) = f_1(x) \cdot f_2(x) = (1 + x + x^4) \times$

$$\times (1 + x^3 + x^4) = 1 + x + x^3 + x^4 + x^5 + x^7 + x^8.$$

Рассмотрим в качестве примера сформированную по многочлену $P_1(x)$ «прямую» последовательность Гоулда:

$$\{s\} = (s_0s_1s_2s_3s_4s_5s_6s_7s_8s_9s_{10}s_{11}s_{12}s_{13}s_{14}),$$

$$1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1$$

которой соответствует рекуррентное уравнение:

$$s_i = s_{i-1} + s_{i-3} + s_{i-4} + s_{i-5} + s_{i-7} + s_{i-8}, \quad i \geq 8 \quad (5)$$

Если же формировать последовательность на основе расширенного многочлена:

$$P(x) = (1 + x)(1 + x + x^4)(1 + x^3 + x^4) =$$

$$= 1 + x^2 + x^3 + x^6 + x^7 + x^9,$$

то последовательность $\{s\}$ будет также соответствовать рекуррентному уравнению:

$$s_i = s_{i-2} + s_{i-3} + s_{i-6} + s_{i-7} + s_{i-9}, \quad i \geq 9. \quad (6)$$

Таким образом, приведенная выше «прямая» последовательность Гоулда удовлетворяет обоим рекуррентным уравнениям (5) и (6).

Инвертировав последовательность $\{s\}$, легко убедиться, что полученная инверсная последовательность Гоулда $\{v\} = \{\bar{s}\}$ будет так же удовлетворять только рекуррентному уравнению (6).

Произведем обработку «прямой» и «инверсной» последовательностей Гоулда $\{s\}$ и $\{v\}$ по m -элементным участкам с использованием двойственного базиса. Для этого сначала вычислим постоянные коэффициенты $\alpha_r, \beta_r, \gamma_i$ двойственного базиса относительно корней ε, μ, ω для сомножителей многочлена $P(x)$:

$f_1(\varepsilon) = 1 + \varepsilon + \varepsilon^4 = 0$; $f_2(\mu) = 1 + \mu^3 + \mu^4 = 0$; $f_3(\omega) = 1 + \omega = 0$ и найдем соответствующие им начальные элементы А, В и С (табл. 1).

Выберем, в качестве примера, m -элементный участок «прямой» последовательности Гоулда: $(s_4s_5s_6s_7s_8s_9s_{10}s_{11}s_{12}s_{13}s_{14}) = (101010111)$ и найдем начальные элементы А, В и С в виде:

$$A = a_0 + a_1\varepsilon + a_2\varepsilon^2 + a_3\varepsilon^3; \pmod{f_1(\varepsilon)},$$

$$B = b_0 + b_1\mu + b_2\mu^2 + b_3\mu^3; \pmod{f_2(\mu)},$$

$$C = c_0; \pmod{f_3(\omega)}.$$

Применив формулы (4) для выбранного участка, получим:

$$A = \varepsilon^{-4}(\alpha_1 + \alpha_3 + \alpha_5 + \alpha_7 + \alpha_8 + \alpha_9) = \varepsilon^4; \pmod{f_1(\varepsilon)},$$

$$B = \mu^{-4}(\beta_1 + \beta_3 + \beta_5 + \beta_7 + \beta_8 + \beta_9) = \mu^{12}; \pmod{f_2(\mu)},$$

$$C = (\gamma_1 + \gamma_3 + \gamma_5 + \gamma_7 + \gamma_8 + \gamma_9) = 0; \pmod{f_3(\omega)}.$$

Таким образом, выбранная последовательность Гоулда представляет собой сумму по mod2 двух М-последовательностей с начальными элементами $A = \varepsilon^4$ для $f_1(x)$ и $B = \mu^{12}$ для $f_2(x)$, т. е. является «прямой».

Возьмем теперь m -элементный участок инверсной последовательности Гоулда $\{v\}$:

Таблица 1

Значения вычисленных коэффициентов

$f_1(x)$	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9
	ε	1	ε^7	ε^{11}	ε^{10}	ε^9	ε^{10}	ε^3	ε^2
$f_2(x)$	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9
	μ^5	μ^4	μ^{12}	μ^{13}	μ^{12}	μ^{11}	1	μ^7	μ^6
$f_3(x)$	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7	γ_8	γ_9
	1	1	0	1	1	1	0	1	1

$$(v_6 v_7 v_8 v_9 v_{10} v_{11} v_{12} v_{13} v_{14}) = (\bar{s}_6 \bar{s}_7 \bar{s}_8 \bar{s}_9 \bar{s}_{10} \bar{s}_{11} \bar{s}_{12} \bar{s}_{13} \bar{s}_{14}) = (010100010).$$

Вычисляя аналогично, получим:

$$A = \varepsilon^{-6}(\alpha_2 + \alpha_4 + \alpha_8) = \varepsilon^4; \text{mod } f_1(\varepsilon);$$

$$B = \mu^{-6}(\beta_2 + \beta_4 + \beta_8) = \mu^{12}; \text{mod } f_2(\mu);$$

$$C = (\gamma_2 + \gamma_4 + \gamma_8) = 1; \text{mod } f_3(\omega).$$

Таким образом, информационными (начальными) элементами для выбранной «прямой» последовательности Гоулда будут $(a_0 a_1 a_2 a_3 b_0 b_1 b_2 b_3 c_0) = (110011000)$, а для инверсной $(a_0 a_1 a_2 a_3 b_0 b_1 b_2 b_3 c_0) = (110011001)$.

Они отличаются только элементом c_0 : при $c_0 = 0$ имеем «прямую», а при $c_0 = 1$ – инверсную последовательность Гоулда. Генератор таких последовательностей показан на рис. 2.

Изображенный генератор является кодирующим устройством кода БЧХЭ (15,9), на выходе которого могут быть сформированы 225 «пря-

мых» последовательностей Гоулда ($c_0 = 0$), 225 – инверсных ($c_0 = 1$), 15 «прямых» и 15 инверсных М-последовательностей, соответственно для $f_1(x)$ и $f_2(x)$, а также нулевая и единичная последовательность длиной 15.

В заключение можно сформулировать еще одно свойство последовательностей Гоулда: *инверсные последовательности Гоулда, так же, как и «прямые», обладают свойством зеркальности, при этом в каждой инверсной последовательности «зеркало» располагается в той же точке, что и в соответствующей ей «прямой» последовательности.*

Прямые и инверсные ЛРД-последовательности. Аналогично предыдущим примерам, покажем, что «прямые» и соответствующие им инверсные последовательности ЛРД (Лаборатории Реактивного Движения) являются рекуррентны-

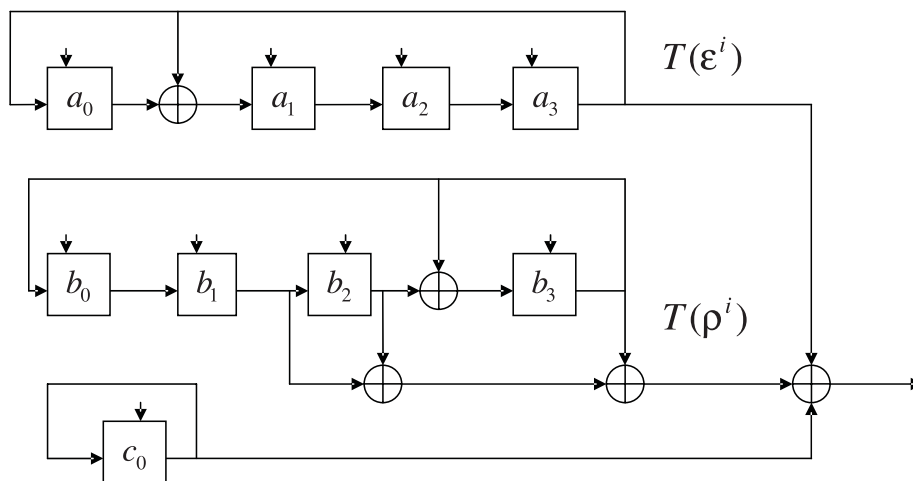


Рис. 2. Генератор «прямых» и инверсных последовательностей Гоулда с периодом 15
 c_0 – «прямая» последовательность Гоулда; $c_0 = 1$ – инверсная последовательность



ми и удовлетворяют одному и тому же рекуррентному уравнению.

Как известно, ЛРД-последовательности являются составными. Период определенной ЛРД-последовательности равен произведению взаимно простых периодов составных М-последовательностей различной длины (в отличие от последовательностей Гоулда). Пусть, например, некоторой ЛРД-последовательности $\{s\}$ соответствует характеристический многочлен

$$P_1(x) = f_1(x) \cdot f_2(x) = (1 + x + x^2)(1 + x + x^3).$$

Так как многочлену $f_1(x)$ соответствует М-последовательность с периодом 3, а многочлену $f_2(x)$ – другая М-последовательность с периодом 7, то такая ЛРД-последовательность будет иметь период 21 и удовлетворять рекуррентному соотношению $s_i = s_{i-1} + s_{i-5}$, $i \geq 5$.

Пусть «прямая» ЛРД-последовательность имеет следующий вид:

$$\{s\} = (s_0 s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10} s_{11} s_{12} s_{13} s_{14} s_{15} \dots s_{20}) = (0100110001000011 \dots 1).$$

Тогда инверсная (по отношению к «прямой») последовательность будет следующая:

$$\{v\} = \{s\} = (s_0 s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10} s_{11} s_{12} s_{13} s_{14} s_{15} \dots s_{20}) = (1011001110111100 \dots 0).$$

Обе эти последовательности являются рекуррентными, удовлетворяющими расширенному характеристическому многочлену $P(x) = f_1(x)f_2(x)f_3(x)$, где $f_3(x) = 1 + x$, т. е. $P(x) = 1 + x + x^4 + x^6$. В соответствии с $P(x)$ рекуррентное уравнение будет иметь вид $s_i = s_{i-2} + s_{i-5} + s_{i-6}$ или $v_i = v_{i-2} + v_{i-5} + v_{i-6}$, $i \geq 6$.

Следовательно, и «прямые», и инверсные последовательности, порожденные многочленом $P(x)$, могут обрабатываться одним и тем же устройством, с целью их распознавания. Для обработки по m -элементным участкам необходимо вычислить постоянные коэффициенты α_p , β_p и γ_i двойственного базиса соответственно для корней ρ (для $f_1(x)$), μ (для $f_2(x)$) и $\omega = 1$ (для $f_3(x)$) (табл. 2.).

По аналогии с (4), вычислим для рассматриваемых «прямой» и инверсной ЛРД-последовательностей начальные элементы C , D и E , соответственно для $f_1(x)$, $f_2(x)$ и $f_3(x)$.

Пусть в «прямой» ЛРД-последовательности $\{s\}$ выделен m -элементный ($m = 6$) участок $(s_6 s_7 s_8 s_9 s_{10} s_{11}) = (000100)$. Обработав его, получим:

$$\begin{aligned} C &= \rho^{-6} \alpha_4 = \rho^{-6} \cdot \rho = \rho, \text{ mod } f_1(x); \\ D &= \mu^{-6} \beta_4 = \mu^{-6} \cdot \mu^6 = 1, \text{ mod } f_2(x); \\ E &= \gamma_4 = 0, \text{ mod } f_3(x). \end{aligned}$$

Выделим теперь произвольный m -элементный участок инверсной ЛРД-последовательности: $\{v\} : (v_4 v_5 v_6 v_7 v_8 v_9) = (001110)$.

Вычисляя по тем же формулам, получим:

$$\begin{aligned} C &= \rho^{-4} (\alpha_3 + \alpha_4 + \alpha_5) = \rho^{-4} (\rho^2 + \rho + \rho) = \rho^2 = \rho, \text{ mod } f_1(x); \\ D &= \mu^{-6} \beta_4 = \mu^{-6} \cdot \mu^6 = 1, \text{ mod } f_2(x); \\ E &= (\gamma_3 + \gamma_4 + \gamma_5) = 1, \text{ mod } f_3(x). \end{aligned}$$

Начальные элементы $C = a_0 + a_1 \rho$; $D = b_0 + b_1 \mu + b_2 \mu^2$ и $E = \ell_0$, записанные в векторной форме, определяют исходные информационные двоичные элементы $[a_0 a_1 b_0 b_1 b_2 \ell_0] = [011000]$ – для «прямой» ЛРД-последовательности ($\ell_0 = 0$) и $[011001]$ – для инверсной ($\ell_0 = 1$).

Генератор, формирующий «прямые» и инверсные ЛРД-последовательности в соответствии с $P(x)$, показан на рис. 3.

Таблица 2

Значения вычисленных коэффициентов

$f_1(x)$	α_1	α_2	α_3	α_4	α_5	α_6
	ρ^2	1	ρ^2	ρ	ρ	1
$f_2(x)$	β_1	β_2	β_3	β_4	β_5	β_6
	μ^6	μ	1	μ^6	μ	1
$f_3(x)$	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6
	1	0	0	0	1	1

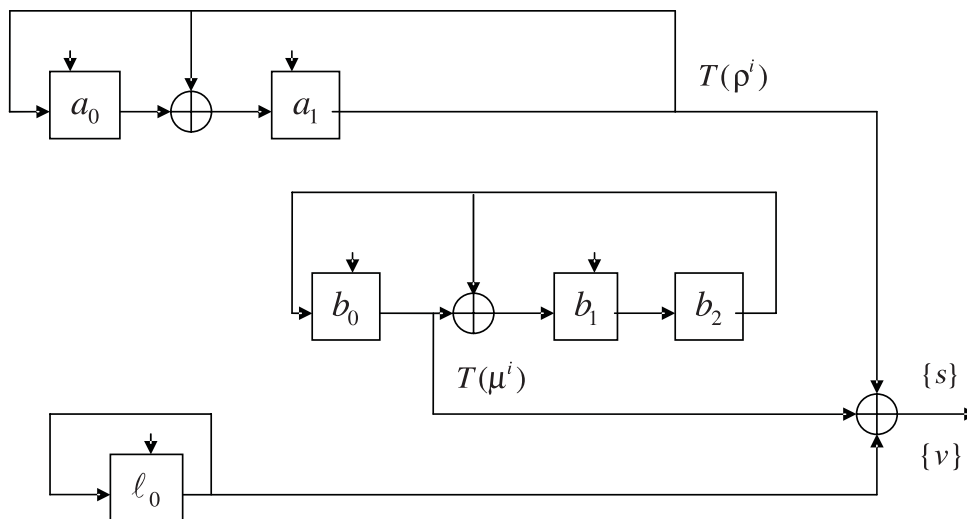


Рис. 3. Генератор «прямых» и инверсных ЛРД-последовательностей с периодом 21

$\ell_0 = 0$ – «прямая» ЛРД-последовательность; $\ell_0 = 1$ – инверсная ЛРД-последовательность

Таким образом, «прямые» и инверсные рекуррентные последовательности (М-последовательности, последовательности Голла, ЛРД-последовательности и др.) могут быть

выделены на приеме с помощью одного устройства, работающего по мажоритарному принципу на основе двойственного базиса [2].

СПИСОК ЛИТЕРАТУРЫ

1. Вишнеvский, В.М. Широкополосные беспроводные сети передачи информации [Текст]/В.М. Вишнеvский, А.И. Ляхов, С.Л. Портной [и др.]. –М.: Техносфера, 2005.

2. Когновицкий, О.С. Двойственный базис и его применение в телекоммуникациях [Текст]/О.С. Когновицкий. –СПб.:Линк, 2009.

УДК 621.391

Л.М. Журавлева

ПОВЫШЕНИЕ ПРОПУСКНОЙ СПОСОБНОСТИ ОПТИЧЕСКИХ ЛИНИЙ СВЯЗИ

Огромные потенциальные возможности оптических линий связи по широкополосности передаваемых сигналов предопределили их ведущую роль в организации магистральных сетей связи. Известно, что ресурсы оптического волокна (ОВ) по частотной полосе и скорости передачи цифровых сигналов достигают, в зависимости от длины оптической несущей, нескольких десятков терагерц [1]. Освоение этих возможностей позволит значительно увеличить пропускную способность волоконно-оптических линий связи (ВОЛС). Не-

обходимым условием для этого является переход на полностью оптические сети без промежуточных оптоэлектронных преобразований. Такие системы способны реализовать ресурсы оптических линий с помощью современных информационных и нанотехнологий.

Цели и задачи развития информационных технологий и способов производства элементной базы волоконно-оптических систем передачи (ВОСП) взаимосвязаны. Так, научные достижения в области технологий изготовления полупро-



водниковых лазеров, фотодетекторов, оптических волокон позволили построить высокоскоростные магистральные ВОСП протяженностью более 1000 км. Это привело, в свою очередь, к резкому росту потребителей сетей связи и необходимости дальнейшего повышения пропускной способности ВОСП с помощью информационных технологий SDH (синхронной передачи), АТМ (асинхронной передачи), WDM (волнового уплотнения) и т. д.

Новое наращивание информационной емкости оптического волокна V , пропускной способности C и скорости передачи оптических сигналов v стало возможным благодаря прорыву в области нанотехнологий и нанoeлектроники.

Представляет определенный интерес анализ предельных значений параметров V , C , v оптических линий связи и возможностей их повышения с помощью нанотехнологий.

Оценку проведем на примере двух основных информационных технологий передачи цифровых сигналов по оптическому волокну, а именно: временного TDM и волнового WDM уплотнений.

Согласно формуле Шеннона, информационная емкость непрерывного канала определяется следующим образом [1]:

$$V = T \cdot F \cdot D, \quad (1)$$

где T – время передачи информации (период опроса, определяемый по теореме Котельникова); F – широкополосность канала (величина, обратная длительности выборки сигнала в момент опроса); $D = \log_2(1 + P_c/P_{\text{ш}})$ – динамический диапазон канала; $(P_c/P_{\text{ш}}) = \rho$ – соотношение мощностей сигнала (квадрата амплитуды выборки) и шума (чувствительности фотодетектора или мощности темнового тока) на входе приемника.

Применительно к оптическому каналу время T можно принять равным периоду тактовой синхронизации (125 мс). Ширина F зависит от амплитудно-частотной (волновой) характеристики (ABX) оптического усилителя (ОУ) и выбранного «окна прозрачности» ОВ и может иметь порядок 1 ТГц [2]. Динамический диапазон D – это количество бит информации, приходящееся на одну выборку сигнала. После квантования амплитудно-импульсного сигнала на 256 уровней и двухпозиционного кодирования, получим, например, восьмиразрядную комбинацию, в которой каждый оптический импульс будет нести следующее количество бит информации:

$$(\rho \gg 1) : D = (\log_2 \rho)/n, \quad (2)$$

где n – число разрядов в кодовом слове.

Подробная детализация формулы Шеннона для ОВ с учетом критерия качества позволит сравнить технологии TDM и WDM по количеству передаваемой информации, скорости передачи сигналов и определить, какая из технологий наилучшим образом может увеличить:

число каналов и информационную емкость ВОЛС;

длину усилительного участка (УУ), в конце которого восстанавливается мощность оптического импульса, и регенерационного участка (РУ), в конце которого восстанавливается мощность и длительность оптического цифрового сигнала;

пропускную способность ВОСП.

Кроме того, с помощью формулы Шеннона можно увидеть наиболее перспективные пути совершенствования ВОСП.

Для этого необходимо преобразовать выражение для информационной емкости V так, чтобы оно включало показатель качества связи ρ , длину регенерационного участка (РУ состоит из нескольких УУ), количество каналов K , которое в общем случае равно произведению числа временных (тайм-слотов) N и числа частотных (волновых) каналов K_f .

Величина N рассчитывается как отношение периода тактовой синхронизации T_{mc} на длительность временного канала $\tau_k = \tau_s \cdot n$, где τ_s – длительность элементарного импульса. Число частотных каналов K_f вычисляется как отношение ширины «окна прозрачности» $F_{\text{ов}}$ на ширину одного частотного канала $\Delta f_k = b/\tau_s$, где коэффициент b определяет величину частотного разнеса между оптическими каналами и зависит от используемой аппаратуры ВОСП.

Значение динамического диапазона для оптических линий связи лучше преобразовать в дБ и представить в виде уровней по мощности:

$$\begin{aligned} \log_2(P_c/P_{\text{ш}}) &= \lg 2(\lg P_c - \lg P_{\text{ш}}) = \\ &= 0,301(a_c - a_{\text{ш}}) = 0,301 a_3, \end{aligned} \quad (3)$$

где a_c , $a_{\text{ш}}$, a_3 – значения уровней оптического сигнала, шума фотоприемника и величина защищенности передачи.

На основании расчетов диаграммы уровней ВОСП можно записать следующие соотношения (рис. 1) [3]:

$$a_{\text{вх}} - \alpha L = a_c, \quad (4)$$

где $a_{\text{вх}}$, a_c – входной и выходной уровни ВОСП;

α – километрический коэффициент затухания ОВ, зависящий от типа ОВ и величины оптической несущей; L – длина усилительного участка.

Входные уровни всех частотных каналов технологии WDM не должны в сумме превышать максимально допустимый уровень $A_{\text{вх}}$ для оптического волокна, который отделяет линейный и нелинейный режимы работы волокна. Величина входного уровня для каждого частотного канала находится из выражения [2]:

$$A_{\text{вх}} = a_{\text{вх}} + 10 \lg K_f, \quad (5)$$

которое легко получить из соотношения $(P_{\text{ов}}/P_1) = K_f$, означающего отношение суммарной пиковой мощности $P_{\text{ов}}$ на входе ОВ, разделенной между частотными каналами с мощностью P_1 . Для технологии TDM $K_f = 1$, поэтому значение $A_{\text{вх}}$ в формуле (5) может быть равным $a_{\text{вх}}$.

Таким образом, информационную емкость оптической линии в общем виде можно выразить так:

$$V^1 = V/0,301 = \left(\sum \Delta f_k \sum \tau_k \right) (a_{\text{вх}} + 10 \lg K_f), \quad (6)$$

где суммирование происходит по всем каналам K_f и N .

Дальнейшая детализация формулы Шеннона для оптического волокна требует учета особенностей распространения света по ОВ. Они заключаются не только в затухании сигнала по энергии, но и в расширении импульса по длительности из-за хроматической дисперсии. Дисперсию можно оценить с помощью коэффициента W . Величина $W = L/\tau_s$ [МГц · км] называется удельной полосой пропускания и зависит от типа ОВ, расположения «окна прозрачности» (значения оптической несущей), ширины спектра излучения лазерного диода $\Delta\lambda$ [1].

Отсюда, выражение Шеннона для оптического волокна с одинаковыми для всех каналов значениями τ_k и Δf_k будет иметь вид:

$$\begin{aligned} V_{\text{ов}}^1 &= K_f \left(\frac{b}{\tau_s} \right) N \left(\frac{L}{W} \right) n (a_{\text{вх}} + 10 \lg K_f) = \\ &= K_f N \cdot b \cdot n (a_{\text{вх}} + 10 \lg K_f). \end{aligned} \quad (7)$$

Как видно из этой формулы, общее число каналов по технологии WDM, равное $K = K_f \cdot N$, ограничено широкополосностью $F_{\text{ов}}$ и временем $T_{\text{мс}}$.

Для технологии TDM ($K_f = 1$) весь частотный ресурс $F_{\text{ов}}$ может расходоваться на число вре-

менных каналов N , а следовательно, на скорость передачи информации v , которая для двоичного кодирования по величине не отличается от скорости передачи сигналов v .

Для технологии WDM увеличение количества частотных каналов K_f уменьшает число временных N и, наоборот, уменьшение длительности элементарного импульса τ_s и соответствующее расширение полосы Δf_k снижает значение K_f . Это свидетельствует о том, что общее число каналов ВОЛС, определяемое на основании соотношения Хартли $K = K_f \cdot N$, одинаково для TDM и WDM.

Линейная зависимость между значениями K, K_f, N говорит о том, что не существует оптимальной длительности τ_s , позволяющей обеспечить максимальное значение пропускной способности ОВ.

Таким образом, при наличии соответствующей элементной базы обе технологии могут предоставить одинаковое число каналов ВОЛС. Эти каналы можно организовать с помощью волнового уплотнения ОВ (разделив общую полосу между отдельными каналами). Затем передавать оптические сигналы одновременно в нескольких частотных каналах (WDM). Другой способ – с помощью временного уплотнения осуществлять передачу сигналов в строго отведенные временные промежутки, которые занимают всю полосу «окна прозрачности» ОВ (TDM).

Отсюда следует, что обе технологии, обладая одинаковым энергетическим запасом $P_{\text{ов}}$, имеют одинаковую информационную емкость.

Проанализируем возможности технологий TDM и WDM по дальности связи.

Для технологии TDM величина $A_{\text{вх}}$ полностью расходуется на энергию оптического сигнала. Поэтому длина усилительного участка L зависит от энергии оптического импульса (произведения квадрата амплитуды на длительность импульса), километрического затухания ОВ α и уровня сигнала $a_c = a_{\text{ш}} + a_s$. Согласно энергетическому балансу ВОЛС (рис. 1) максимально возможная дальность усилительного участка равна [3]:

$$L_{\text{yy}} = \frac{A_{\text{вх}} - a_c}{\alpha}. \quad (8)$$

Однако в это выражение входят уровни сигналов, не учитывающие уменьшения дальности связи при увеличении скорости передачи сигналов. Два импульса одинаковой амплитуды, но разной длительности, будут иметь разные энергии (рис. 2).

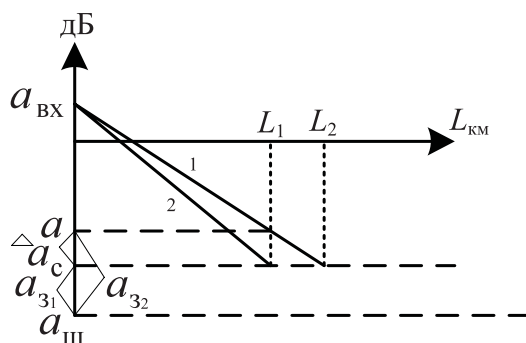


Рис. 1. Диаграмма уровней

Для простоты рассуждений в качестве примера выбраны импульсы прямоугольной формы. В линии связи, как известно, затухает энергия сигнала, величина которой и определяет длину УУ. Чем меньше энергия сигнала, тем быстрее он затухнет в линии связи, тем короче длина УУ. При одном и том же значении входного уровня $a_{вх}$, но разной длительности τ , скорость затухания (потери энергии импульса) в оптической линии будет пропорциональна величине τ (см. графики 1 и 2 на рис.1).

Допустим, график 1 соответствует эталонной скорости, например, STM-1 с уровнем защищенности a_{31} , а график 2 – скорости STM-4. Системы ВОСП с такими параметрами и одинаковой величиной защищенности обеспечат дальность связи, равную соответственно L_1 и $L_2 < L_1$. Чтобы эти расстояния были равны, необходимо повысить входной уровень второй системы. В рассматриваемом случае анализируются потенциальные возможности технологий TDM и WDM. Поэтому в качестве входного уровня принимается максимально возможное значение для выбранного оптического волокна $A_{вх}$, и входные уровни у разных систем не могут быть повышены. Такие же графики 1 и 2 получаются, если использовать разные оптические линии связи с различными коэффициентами километрического затухания $\alpha_1 < \alpha_2$. Выражение (8) справедливо для энергетических расчетов ВОЛС, организованных

с помощью разных оптических кабелей. Поэтому в формуле (8) требуются дополнительные уточнения для сравнения ВОСП с разными скоростями передачи информации. Это можно осуществить следующим образом: взять за основу расчетов в качестве эталонной системы, например, график 1; найти дополнительную величину Δa так, как это показано на рис. 1; получить новое значение защищенности $a_{32} = a_{31} + \Delta a$ для системы с другой скоростью передачи сигналов; согласно рассчитанной величине a_{32} определить дальность УУ для новой системы L_2 по графику 1.

Таким образом, уменьшение энергии элементарного импульса за счет сокращения его длительности учтем в (8) увеличением значения защищенности a_3 для сохранения качества связи.

На основании рис. 2 можно сделать вывод, что величина Δa зависит от отношения энергий исходных импульсов и при одинаковой их амплитуде (входном уровне $A_{вх}$) определяется отношением длительностей элементарных импульсов. Введем в числитель формулы (8) дополнительное слагаемое, равное $\Delta a = 10 \lg B/\tau$. Это дополнение учитывает необходимые изменения величины защищенности эталонной системы в зависимости от скорости передачи информации ВОСП, для которой рассчитывается $L_{эп}$. Тогда, величина максимальной дальности по энергетическому балансу (бюджету) $L_{эп}$ для технологии TDM определится как

$$L_{эп} = [A_{вх} - (a_c + 10 \lg B/\tau)], \quad (9)$$

где B – коэффициент, равный длительности импульса эталонной ВОСП; $a_c = (a_{ш} + a_{31})$ – значение уровня входного сигнала эталонной ВОСП; τ – длительность элементарного импульса ВОСП, причем $\tau < B$.

Следовательно, возможности ОВ по дальности для технологии TDM ограничены величиной $A_{вх}$, скоростью передачи информации и выбранным значением качества a_3 .

Длина регенерационного участка ВОСП выбирается с учетом временного баланса. Так, влия-

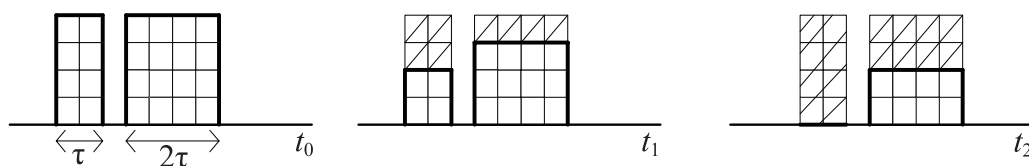


Рис. 2. Затухание оптических импульсов разной длительности

ние хроматической дисперсии допускает следующее значение длины регенерационного участка, которое рассчитывается как отношение коэффициента W на конечную скорость v [1]:

$$L_{\text{вр}} = T_{\text{мс}} \cdot W/N \cdot n \cdot m^1, \quad (10)$$

где коэффициент m^1 учитывает возможное повышение скорости передачи за счет линейного кода.

В этом случае дальность связи ограничивается скоростью передачи v (количеством временных каналов N), параметром оптического волокна W , широкополосностью лазерного излучения $\Delta\lambda$. В зависимости от скорости передачи сигналов, линейного кода и параметров ОБ (α , W) расстояние $L_{\text{вр}}$ может быть больше или меньше расстояния $L_{\text{эн}}$. Обычно длина усилительного участка меньше длины регенерационного участка.

Если приравнять значения $L_{\text{эн}}$ и $L_{\text{вр}}$ по формулам (9) и (10) друг другу, то получим выражение, с помощью которого можно подбирать параметры ВОСП (α , W , v , m^1 , $\Delta\lambda$, K), обеспечивающие максимально возможную дальность связи волоконно-оптических линий при использовании всех доступных ресурсов (по широкополосности $F_{\text{ов}}$ и входному уровню $A_{\text{вх}}$) оптического волокна.

Для технологии WDM уменьшение длительности τ_s влечет за собой расширение спектров сигнала и снижение числа частотных каналов K_f , а также перераспределение допустимого входного уровня $A_{\text{вх}}$ между волновыми каналами в сторону увеличения входных уровней $a_{\text{вх}}$ согласно формуле (5). При этом энергия оптического импульса может остаться неизменной за счет одновременного уменьшения τ_s и возможного увеличения амплитуды импульса.

Так, длина усилительного участка по энергетическому балансу $L_{\text{эн}}$ для технологии WDM с учетом выражений (5) и (9) равна:

$$L_{\text{эн}} = [A_{\text{вх}} - 10 \lg K_f - (a_c + 10 \lg B/\tau_s)] / \alpha. \quad (11)$$

Из этого выражения видно, что уменьшении τ_s поднимает уровень защищенности за счет слагаемого $10 \lg B/\tau_s$, что уменьшает дальность. Происходящее при этом сокращение числа волновых каналов K_f и повышение каждого канала увеличивает дальность связи за счет уменьшения величины $10 \lg K_f$ в формуле (11). Если в итоге получится, что величина K_f сравняется с величиной

B/τ_s , то значения $10 \lg K_f$ и $10 \lg B/\tau_s$ уравниваются друг друга, а дальность связи не изменится. Этот вывод справедлив при условии полного использования технологией WDM ресурсов ОБ по полосе $F_{\text{ов}}$ и входному уровню $A_{\text{вх}}$.

Если учесть влияние хроматической дисперсии на уширение длительности импульса, то можно рассчитать дальность связи для технологии WDM (длину регенерационного участка) по временному балансу. Так, при делении полосы $F_{\text{ов}}$ на отдельные частотные каналы можно записать:

$$K_f = F_{\text{ов}} \cdot \tau_s / b \quad (12)$$

или

$$K_f = F_{\text{ов}} \cdot L/b \cdot W.$$

Отсюда, длина регенерационного участка равна:

$$L_{\text{вр}} = K_f \cdot W \cdot b / F_{\text{ов}}. \quad (13)$$

Приравняв выражения дальности связи по энергетическому (12) и временному (13) балансам для технологии WDM, по аналогии с TDM можно получить параметры ВОСП, обеспечивающие максимально возможную дальность связи при условии использования ресурсов ОБ по широкополосности и входному уровню.

Сравнение дальности связи ВОСП, организованных с помощью технологий WDM и TDM, позволяет получить интересную закономерность. Так, при одной и той же защищенности передачи a_3 и скорости передачи сигналов величина $L_{\text{эн}}$ для технологии WDM (11) меньше, чем для TDM (9) в K_L раз:

$$\begin{aligned} K_L &= (A_{\text{вх}} - a_c) / (A_{\text{вх}} - a_c - 10 \lg K_f) = \\ &= \lg (P_{\text{ов}} / \rho) / \lg (P_{\text{ов}} / \rho \cdot K_f) = \\ &= \log_{(P_{\text{ов}} / \rho \cdot K_f)} (P_{\text{ов}} / \rho). \end{aligned} \quad (14)$$

Следовательно, увеличение количества частотных каналов в системе передачи с WDM уменьшает $L_{\text{эн}}$ не пропорционально числу K_f , а согласно логарифмической зависимости (14), которая снижает длину УУ с повышением K_f медленнее, чем в случае линейной зависимости.

Несмотря на то, что технология WDM имеет определенные ограничения по дальности связи по сравнению с TDM, технология волнового уплотнения является наиболее перспективной для полностью оптических сетей связи.

Интересно сравнить технологии TDM и WDM по пропускной способности.



Разделив формулу (1) для расчета информационной емкости ОВ на время тактовой синхронизации, получим в общем виде выражение для пропускной способности C оптического волокна. Так, $C = F_{\text{ов}} \log_2 \rho / n$ или $C = (K_f \cdot b / \tau_s) \cdot (\log_2 \rho) / n$.

Существующая тенденция к сокращению частотного разброса между оптическими несущими в технологии WDM, которая уже близка к ширине спектра оптического сигнала ($1/\tau_s$) [2], позволяет записать:

$$C \approx K_f \cdot v (\log_2 \rho) / n. \quad (15)$$

Несмотря на то, что для TDM $K_f = 1$, технология временного уплотнения не уступает по пропускной способности, т. к. может компенсировать отсутствие частотного уплотнения повышением скорости передачи сигналов.

Ограничением для увеличения пропускной способности и скорости передачи информации для обеих технологий являются технические возможности использования всего диапазона частот ОВ.

Поэтому главное направление исследований в области ВОСП – расширение частотных границ оптического волокна $F_{\text{ов}}$, а также повышение допустимых значений $P_{\text{ов}}$ (или уменьшение интенсивности излучения на единицу площади сечения сердцевины ОВ, например, за счет увеличения этого сечения). Этого можно добиться путем совершенствования конструкций и технологий изготовления ОВ. Примером нового типа ОВ с расширенными возможностями является фотонно-кристаллическое волокно (ФКВ).

Кроме оптических волноводов, следует улучшать АВХ оптических усилителей и снижать шумы, вносимые ОУ. Здесь также главную роль играют технологии производства оптических устройств (чистота исходных материалов, точность изготовления геометрических размеров оптоэлектронных устройств).

Чтобы оценить влияние технологий производства элементной базы ВОСП, введем в формулу пропускной способности соответствующие коэффициенты. Допустим, что на базе современных нанотехнологий и новых принципов работы электронных и оптоэлектронных устройств (транзисторов, лазеров, фотодетекторов), а также возможностей ОВ можно расширить параметр $F_{\text{ов}}$ в β_1 раз, снизить порог чувствительности фотопри-

емника $P_{\text{ш}}$ и увеличить соответственно параметр ρ в β_2 раз.

Для сравнения влияния нанотехнологий и информационных технологий на пропускную способность ОВ допустим, что совершенствование способов сжатия сигнала уменьшит число разрядов в кодовом слове в γ_1 раз, совершенствование способов приема повысит параметр ρ в γ_2 раз.

Перепишав формулу для пропускной способности (15) с учетом введенных коэффициентов, получим выражение, позволяющее оценить степень влияния информационных и нанотехнологий на величину C :

$$C = F_{\text{ов}} \cdot \beta_1 (\gamma_1 / n) \cdot \log_2 (\rho \cdot \beta_2 \cdot \gamma_2). \quad (16)$$

Отсюда видно, что коэффициенты β_1 и γ_1 непосредственно влияют на значение C , а коэффициенты β_2 и γ_2 – через функцию логарифма, которая растет медленнее, чем линейная зависимость. Это свидетельствует о том, что на пропускную способность в большей степени влияют коэффициенты β_1, γ_1 . Поэтому наиболее перспективными направлениями научных поисков является расширение диапазона $F_{\text{ов}}$ и информационное сжатие сигналов.

С другой стороны, информационное сжатие сигналов может повысить величину C только в несколько раз, в отличие от нанотехнологий, которые могут увеличить значение C в десятки раз [4].

Таким образом, соотношение между коэффициентами $\beta_1 \gg \gamma_1$ в формуле (16) позволяет сделать вывод об исключительном значении нанотехнологий для повышения пропускной способности ВОЛС.

Сравнительный анализ оптических технологий TDM и WDM показал, что каждая из них способна при наличии элементной базы целиком освоить ресурсы ОВ. Однако применительно к полностью оптическим сетям более перспективной является технология волнового уплотнения ВОЛС. Эта технология позволит значительно повысить гибкость оптических сетей за счет различных вариантов логических топологий, реализованных в одном оптическом волокне с использованием разных оптических несущих и алгоритмов волновой маршрутизации [1].

СПИСОК ЛИТЕРАТУРЫ

1. **Убайдуллаев, Р.Р.** Волоконно-оптические сети [Текст] / Р.Р. Убайдуллаев. – М.: Эко-Трендз. – 1998. – 267 с.
2. **Скляров, О.К.** Волоконно-оптические сети и системы связи [Текст] / О.К. Скляров. – М.: Солон-Пресс, 2004. – 261 с.
3. **Стерлинг, Дональд Дж.** Волоконная оптика [Текст] / Дональд Дж. Стерлинг. – М.: Лори, 1998. – 288 с.
4. **Мартинес-Дуарт, Дж. М.** Нанотехнологии для микро- и оптоэлектроники [Текст] / Дж.М. Мартинес-Дуарт, Р.Дж. Мартин-Палма [и др.]. – М.: Техносфера, 2007. – 368 с.

ТЕХНОЛОГИЯ СТРУКТУРНОГО АНАЛИЗА СИГНАЛОВ С ПСЕВДОСЛУЧАЙНОЙ ПЕРЕСТРОЙКОЙ НЕСУЩЕЙ ЧАСТОТЫ В ЗАДАЧАХ РАДИОМОНИТОРИНГА

В настоящее время для радиомониторинга систем связи имеется множество методов и технических средств [1]. Однако важной составной частью радиомониторинга систем связи, в которых применяются сигналы с псевдослучайной перестройкой несущей рабочей частоты (ППРЧ), является структурный анализ (СА) указанных радиосигналов. СА сигналов с ППРЧ в первую очередь связан с вопросами адаптации к структуре наблюдаемого радиосигнала в условиях полной или частичной априорной неопределенности относительно параметров и правила его формирования. Результаты СА могут быть использованы как в интересах радиоэлектронного подавления систем связи, так и при разработке алгоритмов вхождения в синхронизм по закону перестройки несущей частоты для станций, входящих в уже развернутую радиосеть, а также для минимизации взаимного влияния радиоизлучений со сложной частотно-временной структурой.

Основными этапами СА наблюдаемого радиосигнала с ППРЧ с целью решения конкретной прикладной технической задачи в условиях априорной неопределенности правила его формирования являются:

- подготовительный;
- представления исходных данных;
- идентификации;
- непосредственного решения прикладной технической задачи.

Иерархия параметрического описания сигналов с ППРЧ

Радиосигнал с ППРЧ является сложным объектом, который характеризуется комплексом параметров P , описывающих как его элементы,

так и их взаимосвязь, что позволяет, опираясь на оценки параметров обнаруженных частотных посылок (ЧП), полученных в ходе радиомониторинга заданного частотного диапазона, а также на закономерности процедур сигналообразования, выполнять их объединение в единый сигнал.

Анализ комплекса параметров P позволил осуществить его декомпозицию на множество первичных параметров P_1 , проявляющихся на каждой ЧП, и множество вторичных P_2 параметров, проявляющихся на сигнале в целом и характеризующих его как общность отдельных элементов, объединенных по некоторому правилу:

$$P = P_1 \cup P_2.$$

Множество первичных параметров P_1 включает длительность частотной посылки, несущую частоту, направление на источник радиоизлучения, уровень сигнала или отношение сигнал/шум, наличие и вид внутриимпульсной модуляции (манипуляции) и её параметры.

В свою очередь, множество параметров P_2 разнородно и должно быть детализировано:

$$P_2 = P_{2,1} \cup P_{2,2} \cup P_{2,3},$$

где $P_{2,1}$ – подмножество радиотехнических параметров (наличие и вид поляризации, период следования частотных посылок, факт прекращения излучения при перестройке на новую частоту, наличие и вид переходных процессов при перестройке несущей частоты, мощность излучения на конкретной частоте, шаг сетки частот); $P_{2,2}$ – подмножество структурных параметров (объем и состав адресной группы частот, закономерности частотно-временной структуры, тип и формат информационной нагрузки);

$P_{2,3}$ – подмножество пространственных параметров источника радиоизлучения (направление на источник радиоизлучения, местоположение источника радиоизлучения).

Первичные параметры позволяют производить первичную селекцию элементов сигнала, т. к. сигнал с ППРЧ состоит из элементов, однородных по первичным параметрам. Подмножество радиотехнических параметров обеспечивает возможность разделения сигналов идентичных средств радиосвязи на основе закономерностей, характеризующих процессы формирования и излучения радиочастотных колебаний различными радиоэлектронными средствами. Подмножество структурных параметров обеспечивает как возможность оптимизации процесса приема сигналов в целях обеспечения информационного доступа, так и возможность разделения сигналов средств радиосвязи, идентичных по комплексу радиотехнических параметров, но различимых по

комплексу структурных параметров. Подмножество пространственных параметров, с одной стороны, может быть использовано для оптимизации процесса приема и обеспечения информационного доступа, а, с другой, – для формирования пространственной картины источников радиоизлучений, дополняющей общую картину радиоэлектронной обстановки.

Основы первичного представления частотно-временной структуры сигнала

Сформированный комплекс параметров P можно рассматривать как шаблон, по которому осуществляется «сборка» ЧП сигнала с ППРЧ, а сам подготовительный этап – как этап обнаружения и многопараметрической селекции ЧП.

Подготовительный этап включает:

обнаружение элементов сигнала как отдельных радиоимпульсов;

Методологические основы подготовительного этапа

Основные операции	Метод обработки	Технологические процедуры
Обнаружение ЧП сигнала с ППРЧ на фоне шумов	Спектральный анализ	Быстрое преобразование Фурье в скользящем временном окне
Адаптация порога обнаружения по частоте и времени	Параметрическая и непараметрическая адаптация к уровню шума	Нелинейная фильтрация, в т. ч. ранговая
Обнаружение ЧП на фоне мешающих излучений и их частотно-временная селекция	Частотно-временной статистический анализ	Ранговая фильтрация Синтез цифрового фильтра Цифровая фильтрация
Формирование огибающей ЧП, оценивание временных параметров	Шумоподавление	Нелинейная фильтрация, в т. ч. с эффектом гистерезиса
Оценивание частотных параметров ЧП	Спектральный анализ, временной анализ полной фазы комплексного колебания	Быстрое преобразование Фурье, итеративная процедура оценивания частоты во временной области
Определение наличия и вида манипуляции на ЧП	Корреляционный, статистический, спектральный анализ, временной анализ квадратурных компонентов сигнала	Выполнение индикаторных преобразований в частотной и временной областях над отфильтрованными квадратурными компонентами сигнала
Селекция сигнальных объектов одного источника	Кластерный анализ	Селекция по параметрам с применением эталонных описаний



оценивание радиотехнических параметров обнаруженных отдельных радиоимпульсов;

объединение обнаруженных отдельных радиоимпульсов в единый сигнал с ППРЧ и описание частотно-временной структуры сигнала в виде множества пар «время–частота» в дискретные моменты времени на сетке частот.

Поиск элементов сигнала в условиях априорной неопределенности обеспечивается применением широкополосных радиоприемных устройств совместно с аналого-цифровыми преобразователями и аппаратно-программными комплексами, реализующими методы цифровой обработки сигналов. Основные операции и технологические процедуры подготовительного этапа представлены в таблице.

В результате выполнения данного этапа в целях структурного анализа сигнала с ППРЧ формируется множество пар «время–частота», представляющее собой первичное описание частотно-временной структуры данного сигнала:

$$S_0 = \{(t_0, f_0), (t_1, f_1), \dots, (t_{I-1}, f_{I-1})\},$$

$$t_{i+1} = t_i + T_{\text{прч}}, f_i = f_j + k \times \Delta f, k - \text{целое число},$$

где (t_i, f_i) – время и частота, соответствующие i -й ЧП; $T_{\text{прч}}$ – период следования ЧП; Δf – шаг сетки частот; I – количество ЧП, составляющих сигнал.

Очевидно, что чем больше априорной информации о параметрах процедуры сигналообразования в анализируемом сигнале, тем подробнее признаковый портрет отдельных ЧП и сигнала с ППРЧ в целом, и тем выше достоверность получаемой частотно-временной структуры сигнала.

Варианты представления частотно-временной структуры сигнала

На этапе представления исходных данных для СА выполняется преобразование множества пар значений «время–частота» в одну из форм описания частотно-временной структуры сигнала с применением недвоичных ПСП (НПСП). Можно выделить три формы представления, которые различаются по характеру взаимосвязи НПСП с частотно-временной структурой сигнала:

1) НПСП с линейным отображением элементов на сетку частот;

2) ранжированная НПСП;

3) индексная НПСП.

Элементы НПСП $Y_{\langle l \rangle} = \langle y_0, y_1, y_2, \dots, y_{I-1} \rangle$ с ли-

нейным отображением элементов на сетку частот удовлетворяют следующему соотношению:

$$y_i = \frac{f_i - f^{(0)}}{\Delta f},$$

где $f^{(0)}$ – опорная частота.

В качестве опорной может быть выбрана частота любой ЧП сигнала. Однако из практических соображений удобно выбирать минимальную частоту из встречающихся в сигнале частот:

$$f^{(0)} = \min(\{f_i\}_I).$$

Частотно-временная структура сигнала в этом случае однозначно определяется следующим комплексом параметров:

$$S_1 = \langle Y_{\langle l \rangle}, \Delta f, f^{(0)}, T_{\text{прч}}, t_0 \rangle.$$

Недвоичная ПСП с линейным отображением элементов на сетку частот является наиболее часто используемой формой описания взаимного расположения ЧП.

Применение ранжированной НПСП для описания взаимного расположения ЧП сигнала с ППРЧ оправдано в тех случаях, когда излучение сигнала осуществляется с пропусками частотных позиций в сетке частот, т. е. когда радиосредство работает на адресной группе частот (АГЧ) с запрещенными по тем или иным причинам частотами. Для получения ранжированной НПСП $Z_{\langle l \rangle} = \langle z_0, z_1, z_2, \dots, z_{I-1} \rangle$ осуществляется замена элементов НПСП $Y_{\langle l \rangle}$ их индексами в упорядоченном по возрастанию множестве значений элементов $Y_{\langle l \rangle}$:

$$z_i = \text{pos}(y_i, A_{\{M\}}^{(\text{ранж})}),$$

$$A_{\{M\}}^{(\text{ранж})} = \{y^{(0)}, y^{(1)}, \dots, y^{(M-1)}\}, y^{(l)} < y^{(k)} \text{ при } l < k,$$

где $\text{pos}(b, B_{\{l\}})$ – номер элемента b в упорядоченном множестве $B_{\{l\}}$; $A_{\{M\}}^{(\text{ранж})}$ – упорядоченное по возрастанию множество значений элементов $Y_{\langle l \rangle}$; M – объем адресной группы частот.

Частотно-временная структура сигнала в этом случае однозначно определяется следующим комплексом параметров:

$$S_2 = \langle Z_{\langle l \rangle}, A_{\{M\}}^{(\text{ранж})}, \Delta f, f^{(0)}, T_{\text{прч}}, t_0 \rangle.$$

Индексная НПСП применяется для анализа комбинаторной структуры НПСП. Для получения индексной НПСП $Z_{\langle l \rangle} = \langle z_0, z_1, z_2, \dots, z_{I-1} \rangle$ осуществляется замена элементов НПСП $Y_{\langle l \rangle}$ их индексами в множестве элементов, встречающихся в $Y_{\langle l \rangle}$,

упорядоченном в порядке их поступления:

$$z_i = \text{pos}(y_i, A_{\{M\}}^{(\text{инд})}),$$

$$A_{\{M\}}^{(\text{инд})} = \{y^{(0)}, y^{(1)}, \dots, y^{(M-1)}\}_M,$$

$$y^{(k)} = y_i \text{ если } k = \min(\text{pos}(y_i, Y_{<D>})),$$

где $\text{pos}(b, B_{<D>})$ – множество номеров элементов, равных b в последовательности $B_{<D>}$.

Сокращенно данное преобразование описывается следующей формулой:

$$z_i = \min(\text{pos}(y_i, Y_{<D>})).$$

Частотно-временная структура сигнала в этом случае однозначно определяется следующим комплексом параметров:

$$S_3 = \langle Z_{(I)}, A_{\{M_r\}}^{(\text{инд})}, \Delta f, f^{(0)}, T_{\text{прч}}, t_0 \rangle.$$

Основы идентификации НПСП

Идентификация НПСП включает проверку комплекса информативных признаков [2–4] и устранение неопределенности относительно правила и параметров сигналообразования. В качестве такого комплекса признаков удобно использовать параметры рекуррентного генератора, порождающего анализируемую НПСП, а задача идентификации НПСП в этом случае сводится к задаче отыскания параметров порождающего генератора [4].

Анализ принципов формирования НПСП, используемых в средствах радиосвязи с ППРЧ, показал, что широкое распространение получили принципы формирования на основе комбинационно-числового преобразования исходных двоичных ПСП. Введенное в [3] понятие обратного интервального преобразования элементов НПСП $z_i \in \{0, 1, 2, \dots, 2^s - 1\}$, где s – положительное целое число, позволяет получить гипотетическое состояние выхода эквивалентного многоканального генератора двоичных ПСП $X_{<s>,i}$:

$$X_{<s>,i} = F(s, z_i), X_{<s>,i} = \langle x_i^{(0)}, x_i^{(1)}, \dots, x_i^{(s-1)} \rangle,$$

$$x_i^{(k)} = \left\lfloor \frac{z_i - \delta_i^{(k)}}{2^{s-1-k}} \right\rfloor, \quad \delta_i^{(k)} = \begin{cases} 0, & \text{при } k = 0, \\ \sum_{l=0}^{k-1} 2^{(s-1-l)} x_i^{(l)}, & \text{при } k > 0. \end{cases} \quad (1)$$

В работах [5, 6] показано, что для отыскания коэффициентов функции обратной связи с порядком перемножения r и степенным порядком s ,

рекуррентно порождающей ПСП $\langle x_i \rangle_N$, может быть использована система линейных уравнений следующего вида:

$$\begin{cases} x_i = \sum_{\substack{[j_1=0(1)s] \\ [j_2=0(1)s] \\ [j_m=0(1)s]}} a_{j_1, j_2, \dots, j_m} x_{i-1}^{j_1} x_{i-2}^{j_2} \dots x_{i-m+1}^{j_{m-1}} x_{i-m}^{j_m} \bmod 2, \\ x_{i+1} = \sum_{\substack{[j_1=0(1)s] \\ [j_2=0(1)s] \\ [j_m=0(1)s]}} a_{j_1, j_2, \dots, j_m} x_i^{j_1} x_{i-1}^{j_2} \dots x_{i-m+2}^{j_{m-1}} x_{i-m+1}^{j_m} \bmod 2, \\ \dots \\ x_{i+L_A-1} = \sum_{\substack{[j_1=0(1)s] \\ [j_2=0(1)s] \\ [j_m=0(1)s]}} a_{j_1, j_2, \dots, j_m} x_{i+L_A-2}^{j_1} x_{i+L_A-3}^{j_2} \dots x_{i-m+L_A}^{j_{m-1}} x_{i-m+L_A-1}^{j_m} \bmod 2, \end{cases} \quad (2)$$

$$wt(\langle j_1, j_2, j_3, \dots, j_{m-1}, j_m \rangle) \leq r,$$

где L_A – количество неизвестных коэффициентов обратных связей, $L_A = \sum_{k=0}^r s^k C_m^k$; $wt(J_{<m>})$ – число ненулевых компонент вектора $J_{<m>}$; m – глубина обратной связи (память ПСП).

Для определения порождающей функции заданного порядка нелинейности необходимо из двоичных компонентов анализируемой НПСП, полученных в результате обратного интервального преобразования, составить и решить систему линейных уравнений (1) относительно неизвестных коэффициентов обратной связи порождающей функции. Систему линейных уравнений (1) удобно представлять в матричном виде и решать методом Гаусса.

В рамках СА радиосигналов следует выделить два важных фактора, которые следует учитывать. Первый фактор состоит в том, что исходные данные для анализа обычно представлены не полным периодом ПСП, а лишь ее сегментом. Второй фактор заключается в том, что анализируемые данные, как правило, получены из физических каналов связи и могут содержать искаженные элементы. В этих условиях результат идентификации носит вероятностный характер.

Решение прикладной технической задачи

На данном этапе результаты вскрытия правила и параметров сигналообразования используются для решения прикладных технических задач.

Одной из часто решаемых прикладных задач является синхронизация по фазе ППРЧ.



В случае, когда неопределенность относительно правила и параметров сигналообразования для синхронизации устранена, могут применяться методы, рассмотренные в [7]. Кроме того, для оценивания фазы НПСР могут использоваться специфические структурные свойства НПСР различных классов [2, 8]. Так, например, для не двоичных линейных рекуррентных последовательностей при известном формирующем полиноме m -й степени фаза НПСР определяется по m элементам.

В интересах радиоэлектронного подавления радиолиний, работающих в режиме с ППРЧ, результаты оценивания правила и параметров сигналообразования могут использоваться для формирования помехового радиоизлучения, максимально близкого по частотно-временной структуре к радиосигналу подавляемой радиолинии. С этой целью периодически выполняется уточнение параметров сигналообразования частотно-временной структуры радиоизлучения подавляемой радиолинии и последующая адаптация структуры помехового радиоизлучения.

Другой важной технической задачей является обнаружение изменения режимов работы радиосредств, работающих с перестройкой несущей частоты. При этом изменение режима работы РЭС может проявляться всего лишь в переходе с повседневного режима ППРЧ, обеспечивающего, например, режим частотного разнесения символов передаваемых сообщений в целях защиты от частотно-селективных помех, на режим ППРЧ с повышенной структурной скрытностью.

Результаты вскрытия правила и параметров сигналообразования в определенных условиях могут быть применены для обнаружения работы одного средства радиосвязи в различных радиосетях. Данное обстоятельство обусловлено различием в параметрах сигналообразования. В этом случае, при работе радиосредства в пределах одной радиосети параметры частотно-временной структуры остаются неизменными, а по числу выявленных сочетаний параметров можно оценить число радиосетей.

Кроме того, важной задачей является оценивание числа радиосетей с ППРЧ, развернутых в зоне радиовидимости или на объекте радиоэлектронного контроля, в условиях, когда разделение источников радиоизлучений по пеленгационным и радиотехническим признакам оказывается невозможным. В основе этого также лежит объективно необходимое различие в параметрах частотно-временной структуры радиоизлучений различных радиосетей.

Таким образом, методологические основы структурного анализа сигналов с перестройкой несущей частоты при радиомониторинге систем передачи информации включают элементы теории обнаружения и оценивания параметров радиосигналов, положения теории распознавания образов, теории идентификации псевдослучайных последовательностей, методы цифровой обработки сигналов, а также широкий класс методов решения задач практической направленности.

СПИСОК ЛИТЕРАТУРЫ

1. Рембовский, А.М. Радиомониторинг: задачи, методы, средства [Текст]/А.М. Рембовский, А.В. Ашихтин, В.А. Козьмин; под ред. А.М. Рембовского. – М.: Горячая линия – Телеком, 2006. – 492 с.
2. Еремеев, И. Ю. Структурный анализ сигналов со скачками частоты [Текст]/И.Ю. Еремеев, А.И. Замарин // Ч. 1: Теоретические основы. – Архангельск: АГМА – КИРА, 1999. – 140 с.
3. Еремеев, И.Ю. Применение обобщенного обратного интервального преобразования для идентификации не двоичных ПСП, используемых в системах радиосвязи с ППРЧ [Текст]/И.Ю. Еремеев // Научные технологии. – 2009. – Т. 10. – № 1. – С. 18–24.
4. Еремеев, И.Ю. Информативные структурные признаки не двоичных ПСП, используемых для псевдослучайной перестройки рабочей частоты в помехозащищенных системах передачи информации [Текст]/И.Ю. Еремеев//Проблемы развития системы специ-
- альной связи и специального информационного обеспечения государственного управления России: Матер. 4-й Всерос. науч. конф. 10-11 февраля 2005. – Орел: АФСО, 2005. – Ч. 1. – С. 112–114.
5. Ковалевский, В.Э. Основы теории нелинейных функций над конечными полями и её приложения [Текст]/В.Э. Ковалевский. – СПб.: ВИККА, 1997. – 179 с.
6. Тараненко, П.Г. Псевдослучайные и кодовые последовательности. Методы синтеза и анализа [Текст]/П.Г. Тараненко. – СПб.: ВКУ им. А.Ф.Можайского, 1999. – 110 с.
7. Горшков, В.В. Схемы слежения за задержкой сигналов с псевдослучайной перестройкой рабочей частоты [Текст]/В.В. Горшков, А.В. Варламов // Зарубежная радиоэлектроника. – 1990. – № 3. – С. 3–13.
8. Еремеев, И.Ю. Структурные свойства ансамблей частотно-временных матриц [Текст]/И.Ю. Еремеев, А.И. Замарин // Изв. вузов. Сер. Приборостроение. – 1998. – Т. 41. – № 8. – С. 45–52.

АЛГОРИТМ ПРИЕМА СПЕКТРАЛЬНО-ЭФФЕКТИВНЫХ СИГНАЛОВ С OFDM

Применение гладких огибающих на поднесущих сигналов с OFDM позволяет решать такие задачи повышения эффективности систем передачи данных, как снижение уровня внеполосных излучений, снижение пик-фактора колебаний, повышение скорости передачи информации.

В [1] предлагается метод формирования спектрально-эффективных сигналов с OFDM, основанный на линейном преобразовании комплексных модуляционных символов, поступающих на вход блока обратного быстрого преобразования Фурье (ОБПФ). Для приема таких сигналов следует осуществлять принятие решения о переданном комплексном символе после анализа сигналов на всех поднесущих частотах. Благодаря этому в процессе демодуляции в памяти устрой-

ства приема сохраняется набор возможных комплексных символов для каждой поднесущей, а окончательное принятие решения о символах осуществляется после учета всех связей между всеми сигналами на поднесущих частотах.

При использовании гладких огибающих сигналов каждая квадратурная составляющая для n -й поднесущей распадается на сумму $(2m+1)$ составляющих, сосредоточенных рядом с n -й поднесущей, где $(m+1)$ – количество учитываемых членов в разложении огибающей $a(t)$ в ряд Фурье. На рис. 1. проиллюстрировано явление интерференции между четырьмя поднесущими спектрально-эффективного сигнала с OFDM в частотной области при $m = 3$. Показаны составляющие группового сигнала на каждой из четырех соседних

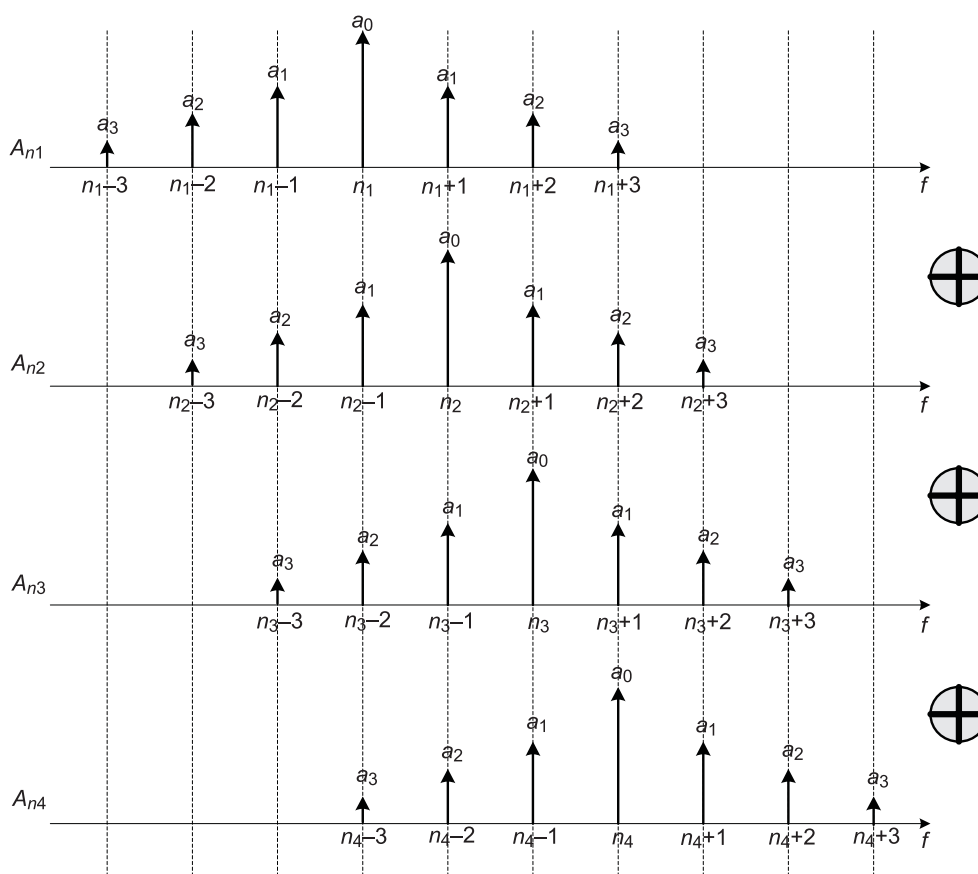


Рис. 1. Интерференция между поднесущими спектрально-эффективного сигнала с OFDM в частотной области



поднесущих с номерами $n_1, n_2, n_3, n_4, n_2 = n_1 + 1, n_3 = n_2 + 1, n_4 = n_3 + 1$. На указанных поднесущих частотах передаются символы $A_{n_1}, A_{n_2}, A_{n_3}$ и A_{n_4} соответственно. Видно, что в отличие от классических сигналов с OFDM, сигнал на каждой поднесущей распадается на семь составляющих (в общем случае на $2m+1$), расположенных рядом с исходной. При сложении сигналов всех поднесущих амплитуда и фаза каждого сигнала будет зависеть не только от символа, передаваемого на данной поднесущей, но и от символов, передаваемых на соседних поднесущих, а также значений коэффициентов $a_k, k = 0, 1, \dots, m$ в разложении огибающей в ряд Фурье.

В [1] предложен метод формирования спектрально-эффективных сигналов с OFDM на основе линейного преобразования вида

$$C_n = \sum_{k=-m}^m A_{n+k} a_k, \quad (1)$$

где $n = -N/2, (-N/2 + 1), \dots, (N/2 - 1)$; $a_k = a_{-k}, k = 0, \pm 1, \pm 2, \dots, \pm m$; C_n – новые модуляционные символы, поступающие на вход ОБПФ; A_{n+k} – исходные (информационные) модуляционные символы, $A_{n+k} = 0, (n+k) > N/2 - 1, (n+k) < -N/2$.

Рассмотрим решетчатую структуру для формирования спектрально-эффективных сигналов с OFDM (рис. 2). В узлах решетки находятся точки, соответствующие комплексным модуляционным символам алфавита до преобразования (1). Всякая входная последовательность символов канального

алфавита соответствует некоторому пути на решетке. Например, пути 1 соответствует последовательность битов 0, 0, 1, 0, 1, 0, 0, 0, пути 2 – 0, 1, 0, 0, 1, 1, 0, 1 и пути 3 – 1, 0, 1, 0, 1, 0, 0, 1, 1.

Каждому ребру соответствует определенный комплексный символ на выходе преобразования (1), являющийся отсчетом спектрально-эффективного сигнала с OFDM в частотной области. Количество ребер, выходящих из одного узла, определяется количеством членов m в разложении огибающей в ряд Фурье и в общем случае может принимать достаточно большое значение. Тем не менее, в практически важных случаях модуль всех коэффициентов в разложении огибающей в ряд Фурье, кроме первых двух–трех, относительно невелик и может не учитываться при анализе.

Учет только первых двух коэффициентов при анализе решетки спектрально-эффективного сигнала с OFDM приводит к тому, что количество ребер, выходящее из данного узла, равно четырем (рис. 3).

Задачу демодуляции спектрально-эффективного сигнала с OFDM можно рассматривать как задачу нахождения пути по решетке с помощью определенных правил демодуляции, так же как в алгоритме Витерби декодирования сверточных кодов. Заметим, что обработка данных осуществляется в частотной области, а не во временной, как в классическом декодере Витерби.

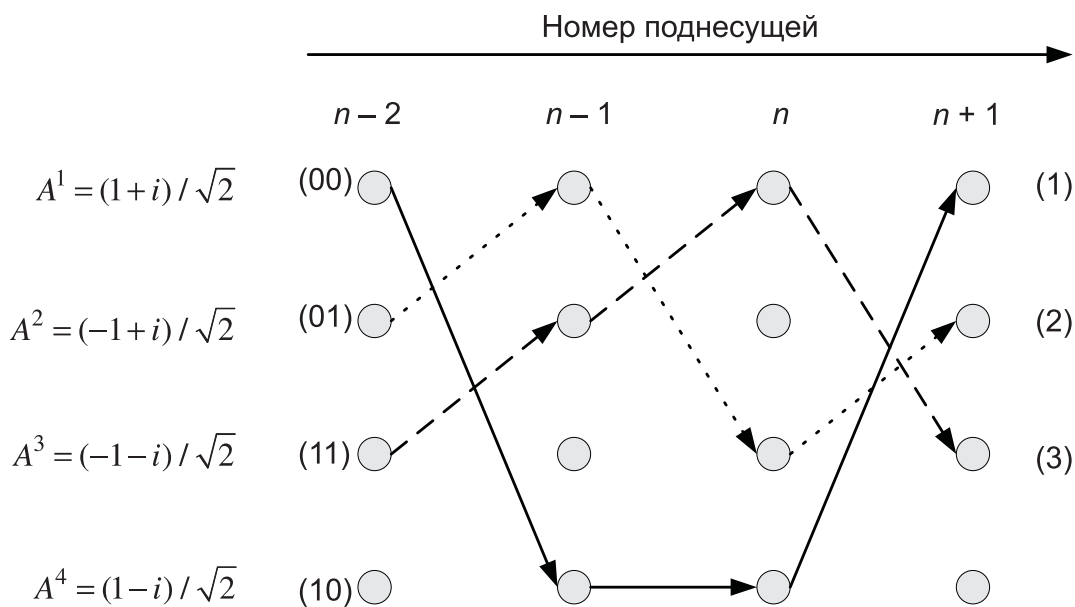


Рис. 2. Решетка для формирователя спектрально-эффективных сигналов с OFDM при модуляции ФМ-4

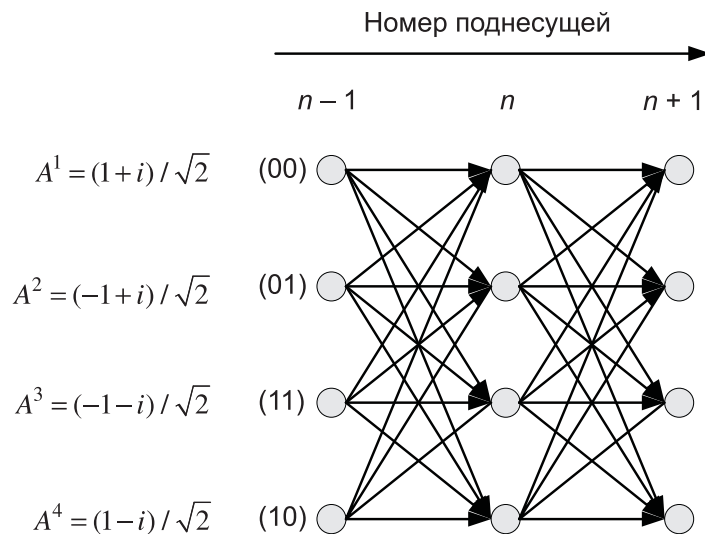


Рис. 3. Фрагмент решетки спектрально-эффективного сигнала с OFDM со всеми возможными переходами при учете первых двух символов в разложении огибающей в ряд Фурье и модуляции ФМ-4

Подобно алгоритму Витерби при решетчатой демодуляции спектрально-эффективного сигнала с OFDM на каждом переходе между соседними поднесущими определяется метрика для каждого пути на решетке. Вначале демодуляции количество путей экспоненциально растет, затем появляется возможность на каждом шаге исключить такое количество «старых» путей, которое соответствует количеству вновь появившихся. То есть на шаге алгоритма для каждого узла рассчитываются метрики всех четырех путей, которые входят в данный узел, и из них оставляется путь с наименьшей метрикой, а остальные пути удаляются. При любых принятых впоследствии данных метрика этих путей не станет лучше метрики «выжившего» пути. Таким образом, в памяти демодулятора всегда содержатся четыре пути, один из которых наиболее правдоподобный.

В отличие от декодера Витерби в демодуляторе спектрально-эффективных сигналов с OFDM метрика является не хемминговой, а рассчитывается на основании следующего выражения:

$$d_{n+1}^k = \left| \hat{C}_n - (A_{n-1}a_1 + A_n a_0 + A^k a_1) \right|^2. \quad (2)$$

Алгоритм демодуляции спектрально-эффективных сигналов с OFDM по решетке выглядит следующим образом:

1. *Инициализация демодулятора.* В начале демодуляции можно считать, что все состояния равновероятны, либо осуществлять демодуляцию из заранее известного состояния.

2. *Прямой проход по решетке.* Осуществляется последовательный анализ принятой последовательности с выхода БПФ, рассчитываются метрики путей, в памяти демодулятора сохраняются и обновляются «выжившие» пути.

3. *Обратный проход по решетке.* Из «выживших» путей выбирается путь с наилучшей метрикой и по нему восстанавливается переданная последовательность символов канального алфавита.

Важным достоинством решетчатого алгоритма демодуляции является то, что решение о переданных символах принимается после анализа всех поднесущих, и, если в некотором узле был выбран неверный путь, то позднее он может слиться с верным. Заметим, что для дополнительного увеличения помехоустойчивости возможно применение гладких огибающих только на поднесущих, расположенных на краях используемого диапазона частот.

Зависимости средней вероятности ошибки на бит от отношения сигнал-шум, полученные с помощью имитационного моделирования, для различного количества поднесущих в сигнале с OFDM представлены на рис. 4 и 5. По оси ординат отложены значения средней вероятности ошибки, по оси абсцисс – отношение сигнал-шум (отношение энергии бита к средней спектральной плотности мощности аддитивного белого гауссовского шума). Построены кривые помехоустойчивости при использовании 5, 10, 20 и 100 % поднесущих с гладкими огибающими со скоростью спада спектра ω_2 , ω_4 и ω_6 .

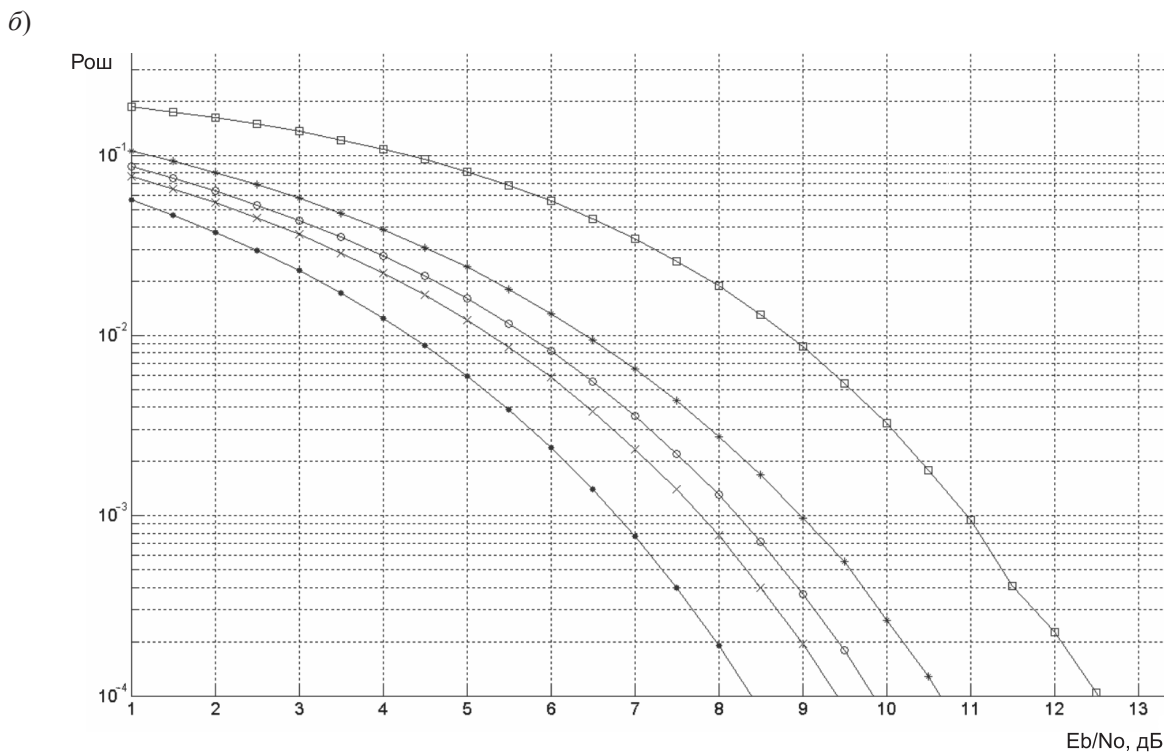
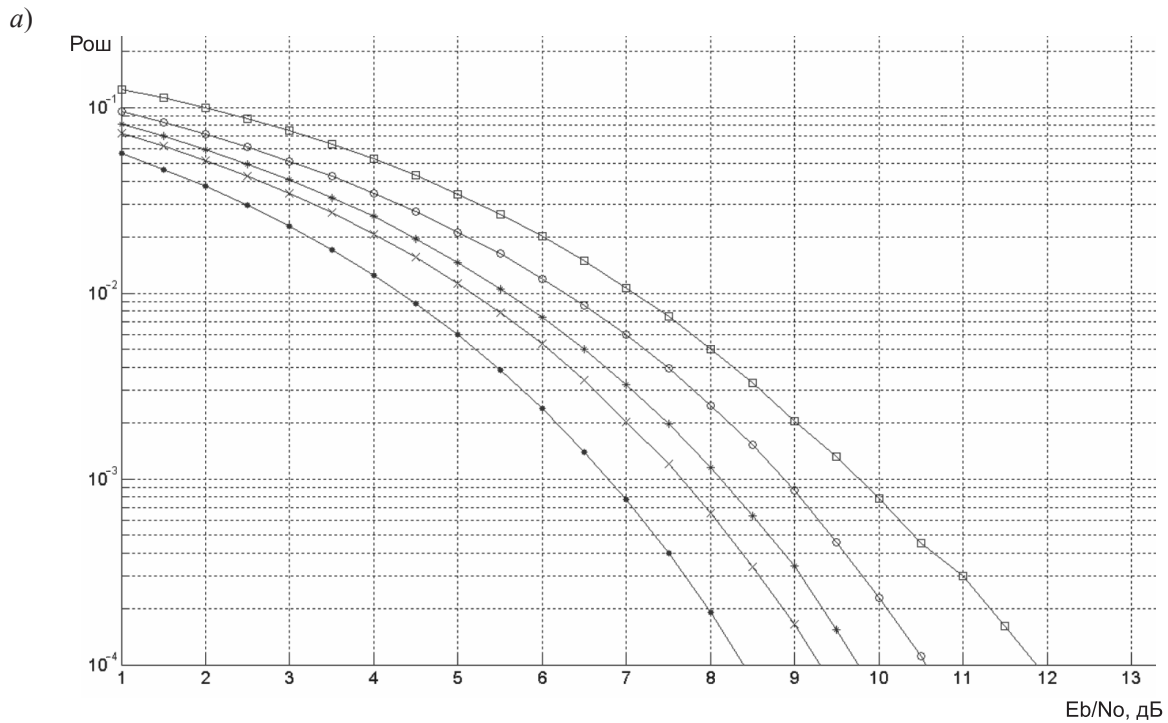
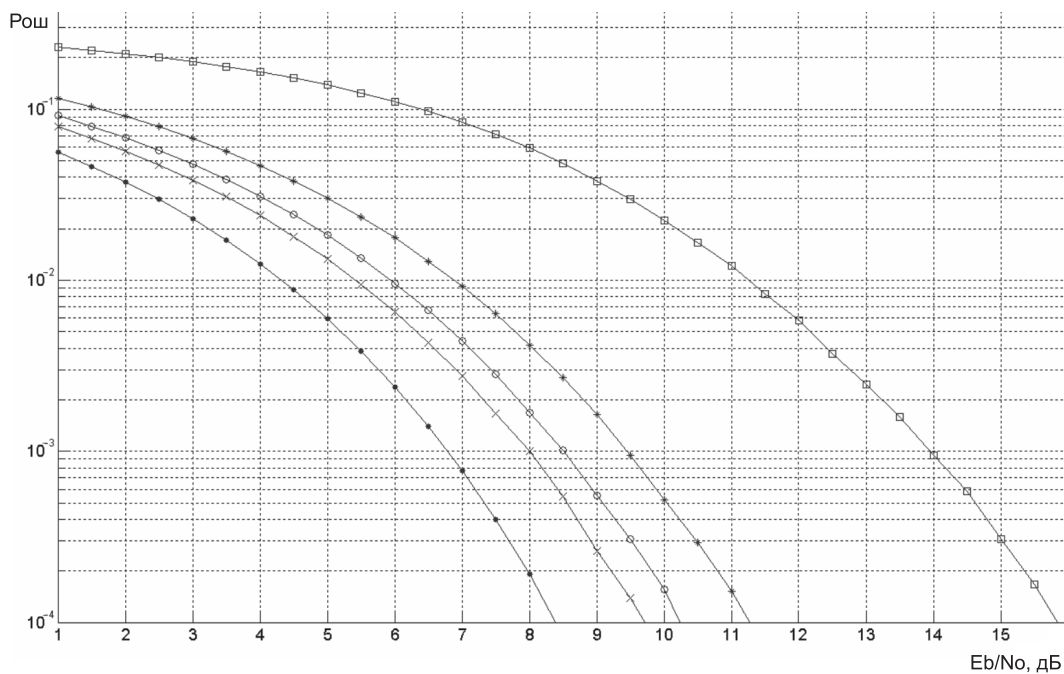


Рис. 4. Зависимости вероятности ошибок при приеме спектрально-эффективных сигналов с OFDM по алгоритму демодуляции по решетке для случаев 5, 10, 20 и 100 % гладких огибающих поднесущих в сигнале с OFDM, $N = 256$, $N_{\text{исп}} = 200$: $a - \omega^2$, $b - \omega^4$, $c - \omega^6$
(—×—) СПЭ OFDM 5%; (—○—) 10%; (—◆—) 20%; (—□—) 100%; (—●—) СПЭ OFDM ФМ-4

Продолжение рис. на с. 36

б)



а)

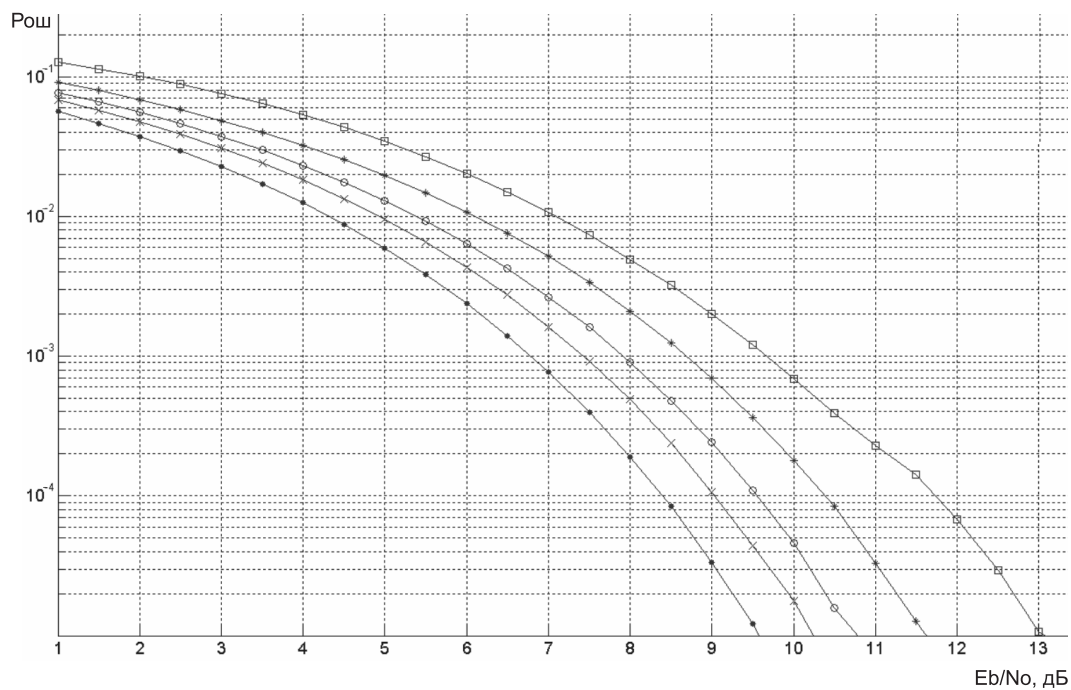
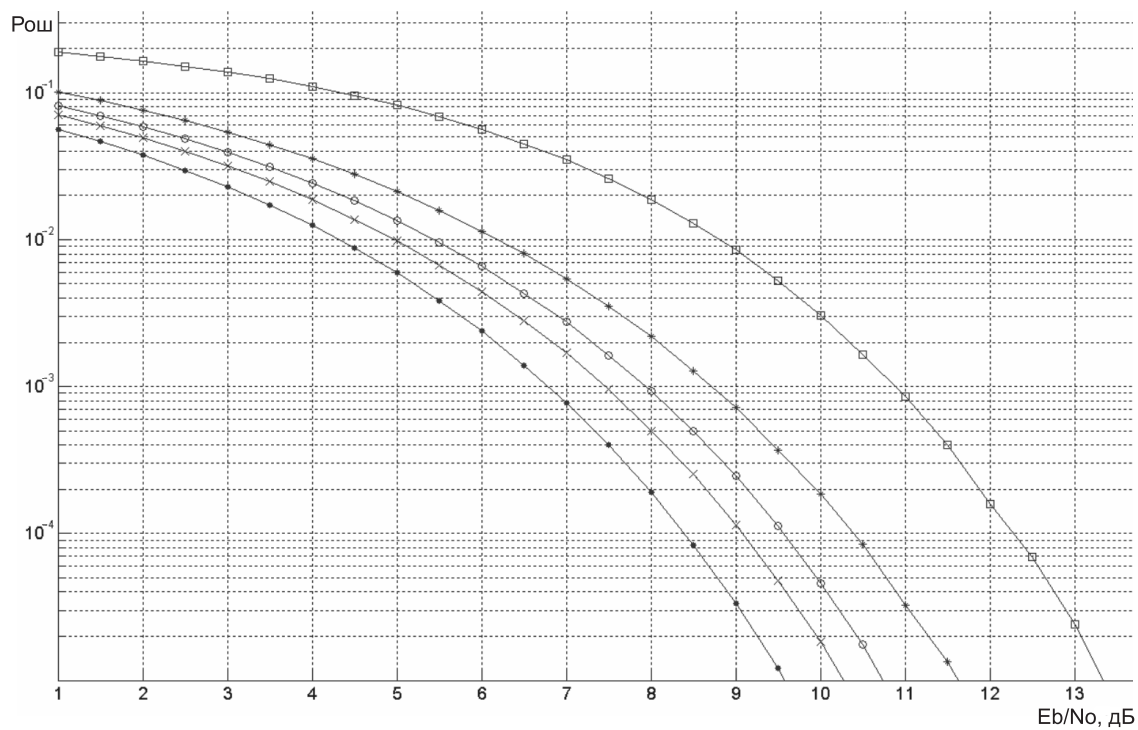


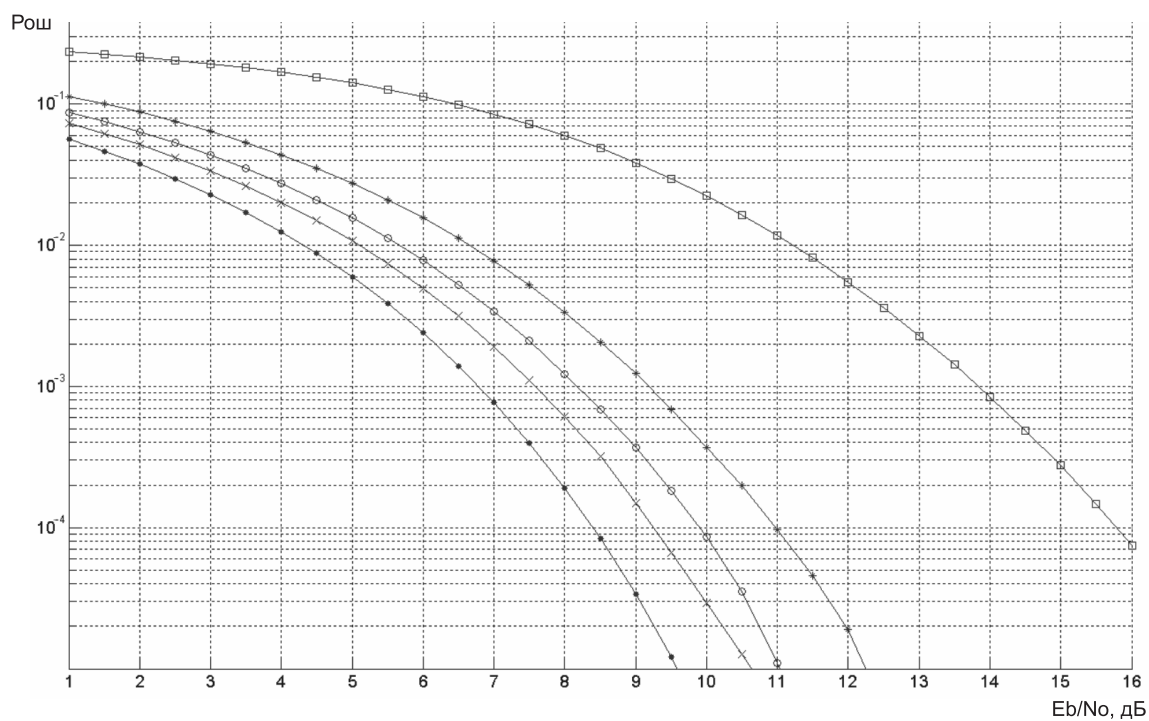
Рис. 5. Зависимости вероятности ошибок при приеме спектрально-эффективных сигналов с OFDM по алгоритму демодуляции по решетке для случаев 5, 10, 20 и 100 % гладких огибающих поднесущих в сигнале с OFDM, $N = 1024$, $N_{\text{исп}} = 840$: $a - \omega^2$, $b - \omega^4$, $c - \omega^6$
(—×—) СПЭ OFDM 5 %; (—○—) 10 %; (—◆—) 20 %; (—□—) 100 %; (—●—) СПЭ OFDM ФМ-4

Продолжение рис. на с. 37

б)



в)



Из анализа зависимостей, представленных на рис. 4, 5, следует, что энергетический проигрыш при приеме по алгоритму Витерби спектрально-эффективных сигналов с OFDM по сравнению с когерентным приемом классических сигналов с OFDM с ФМ-4 для 5, 10, 20 и 100 % поднесущих с гладкими огибающими в сигнале, составляет:

при $N = 256$, $N_{\text{исп}} = 200$

– 0,7, 1,3, 2,1 и 3,4 дБ, для $n = 1$;

– 1,0, 1,5, 2,3 и 4,1 дБ, для $n = 2$;

– 1,3, 1,7, 2,7 и 7,3 дБ, для $n = 3$;

при $N = 1024$, $N_{\text{исп}} = 840$

– 0,6, 1,1, 1,8 и 3,2 дБ, для $n = 1$;

– 0,7, 1,1, 2,0 и 3,8 дБ, для $n = 2$;

– 0,8, 1,5, 2,5 и 7,3 дБ, для $n = 3$.

При фиксированном значении параметра n помехоустойчивость приема спектрально-эффективных сигналов с OFDM ухудшается с увеличением в сигнале количества поднесущих с гладкими огибающими. Это обусловлено происходящим увеличением помех неортогональности между поднесущими. Худшей помехоустойчивостью обладают сигналы с OFDM с 100 % гладких огибающих.

Как и ожидалось, при увеличении скорости спада спектра сигнала (увеличение параметра n) достоверность приема спектрально-эффективных сигналов с OFDM ухудшается. Это

объясняется тем, что при больших значениях параметра n , сигнальные точки преобразованного по (1) созвездия находятся ближе друг к другу и к осям ординат. Минимальное евклидово расстояние между точками для больших n меньше, что и приводит к уменьшению средней вероятности ошибки на бит.

Заметим, что при приеме спектрально-эффективных сигналов с OFDM с 1024 поднесущими наблюдается энергетический выигрыш в 0,1–0,5 дБ по сравнению с приемом спектрально-эффективных сигналов с OFDM с 256 поднесущими.

Предложена методика приема спектрально-эффективных сигналов с OFDM на основе анализа решетки подобно алгоритму декодирования Витерби. Энергетический проигрыш при приеме по данному алгоритму составляет от 0,7 до 7,3 дБ для вероятностей ошибки 10^{-4} – 10^{-5} в зависимости от параметров спектрально-эффективных сигналов с OFDM.

Возможным результатом применения разработанной методики является сигнал с OFDM с уменьшенным значением пик-фактора при сохранении полосы занимаемых частот, уменьшении уровня ВПИ и энергетическом проигрыше 0,7–2,7 дБ при демодуляции по решетке.

СПИСОК ЛИТЕРАТУРЫ

1. Рашич, А.В. Метод формирования спектрально-эффективных OFDM-сигналов на основе неортогональных базисных функций [Текст]/А.В. Рашич, С.Б. Макаров // Научно-технические ведомости СПбГПУ. – 2009. – № 2 (76). – С. 94–98.

2. Школьный, Л.А. Оптимизация формы огибающей радиоимпульса по минимуму внеполосных излучений [Текст]/Л.А. Школьный // Радиотехника. – 1975. – Т. 30. – № 6. – С. 12–15.

ПАРАМЕТРИЧЕСКАЯ ОПТИМИЗАЦИЯ ЦИФРОВЫХ МОДАЛЬНЫХ РЕГУЛЯТОРОВ

В микропроцессорных системах автоматического управления (САУ) объектами высокого порядка, содержащими, например, упругие связи, широкое применение находят модальные регуляторы (МР) с наблюдателями (Н). Синтез таких регуляторов осуществляют по линеаризованным моделям САУ методом размещения полюсов или путем решения уравнений Риккати [1, 2, 4, 8–10]. Однако на практике результаты синтеза часто требуют оптимизации численными методами с учетом ограничений и нелинейностей, присущих реальным системам.

Закон управления в таких системах (рис. 1) выглядит следующим образом:

$$u[n] = K_g g[n] - K^T X_o[n],$$

где $g[n]$, $u[n]$ – дискретные входное и управляющее воздействия; $X(t)$, $X_o[n]$ – m -вектор состояния дискретного наблюдателя (*observer*); K^T – m -вектор-строка параметров МР; K_g – коэффициент передачи по входу.

Оптимизируемыми параметрами МР являются элементы K^T и параметр $K_g = 1/K_{sys}$, где $K_{sys} = C_d^T (E - A_d + B_d K^T)^{-1} B_d$ – коэффициент передачи замкнутой дискретной системы.

Синтез цифровых МР

Синтез САУ рассматриваемого класса методом размещения полюсов проводится в три этапа (рис. 1):

- 1) синтезируют дискретный наблюдатель m -го порядка, где m – порядок модели ОУ;
- 2) синтезируют замкнутую дискретную систему m -го порядка с МР, но без наблюдателя;
- 3) моделируют непрерывно-дискретную систему $2m$ -го порядка с наблюдателем для проверки полученных результатов синтеза.

Могут использоваться наблюдатели с различными структурами (базисами переменных состояния), имеющими разное количество параметров. Если структура наблюдателя соответствует структуре дискретной модели ОУ с матрицей A_d , все элементы которой могут быть ненулевыми даже при сильно разреженной матрице ОУ, то количество параметров наблюдателя (число выполняемых операций умножения) может достигать значения $(m \times m + 3m)$. При использовании наблюдателя с канонической структурой, соответствующей дискретной модели ОУ в форме наблюдаемости с матрицей A_o и векторами B_o и L_o , общее количество параметров наблюдателя равно $3m$. Кроме того, при синтезе канонического наблюдателя методом размещения полюсов его параметры рассчитываются значительно проще, без составления и решения алгебраической системы.

Рассмотрим этапы компьютерного синтеза цифровых САУ рассматриваемого класса (рис. 1).

На *этапе 1.1* осуществляется дискретизация модели ОУ и каноническое преобразование полученной дискретной модели с помощью соответствующей матрицы P_o :

$$A_o = P_o^{-1} A_d P_o; B_o = P_o^{-1} B_d; C_o^T = C_d^T P_o.$$

На *этапе 1.2* выбирается эталонный характеристический полином $Q_o^*(z)$ m -й степени с желаемым размещением корней (полюсов замкнутого дискретного наблюдателя) внутри круга единичного радиуса на комплексной z -плоскости.

На *этапе 1.3* вычисляется вектор $L_o = (L_{1o}, L_{2o}, \dots, L_{mo})^T$, от значений элементов которого зависят собственные числа матрицы $A_o - L_o C_o^T$ наблюдателя и его динамические свойства. Поскольку матрица A_o имеет каноническую форму, то характеристический полином наблюдателя выглядит

следующим образом:

$$Q_o(z) = \det(zE - A_o + L_o C_o^T) = z^m + (a_m + L_{mo})z^{m-1} + \dots + (a_2 + L_{2o})z + a_1 + L_{1o}.$$

Из условия равенства коэффициентов характеристических полиномов

$$Q_o(z) = Q^*(z) \quad (1)$$

получаем выражения для определения параметров дискретного наблюдателя:

$$L_{io} = a_{io}^* - a_i; i = 1, \dots, m,$$

где a_{io}^* – коэффициенты его эталонного полинома.

При синтезе дискретного наблюдателя с неканонической структурой для определения его параметров требуется решить систему алгебраических уравнений, соответствующую условию (1). Также можно воспользоваться формулой Аккермана [9, 10], принимающей в рассматриваемом случае следующий вид:

$$L_X = \varphi(A_d) \begin{bmatrix} C_d^T \\ C_d^T A_d \\ C_d^T A_d^2 \\ \dots \\ C_d^T A_d^{m-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{bmatrix},$$

где матрица наблюдаемости формируется из матрицы A_d и вектора-строки C_d^T , а матрица $\varphi(A_d)$ соответствует эталонному характеристическому полиному наблюдателя:

$$\varphi(A_d) = A_d^m + a_{mo}^* A_d^{m-1} + \dots + a_{2o}^* A_d + a_{1o}^* E.$$

На *этапе* 2.1 в целях упрощения синтеза МР строится каноническая дискретная модель ОУ в форме управляемости. Для этого с помощью соответствующей матрицы T_c можно преобразовать дискретную модель ОУ, либо преобразовать дискретную каноническую модель, используя матрицу T_{oc} :

$$A_c = T_{oc}^{-1} A_o T_{oc}; B_c = T_{oc}^{-1} B_o; C_c^T = C_o^T T_{oc}.$$

На *этапе* 2.2 для дискретной САУ, замкнутой через МР, выбирается эталонный характеристический полином $Q_c^*(z)$ m -й степени с желаемым размещением корней (полюсов системы) внутри круга единичного радиуса на комплексной Z -плоскости.

На *этапе* 2.3 для дискретной системы с канонической моделью ОУ в форме управляемости вычисляются элементы вектора-строки

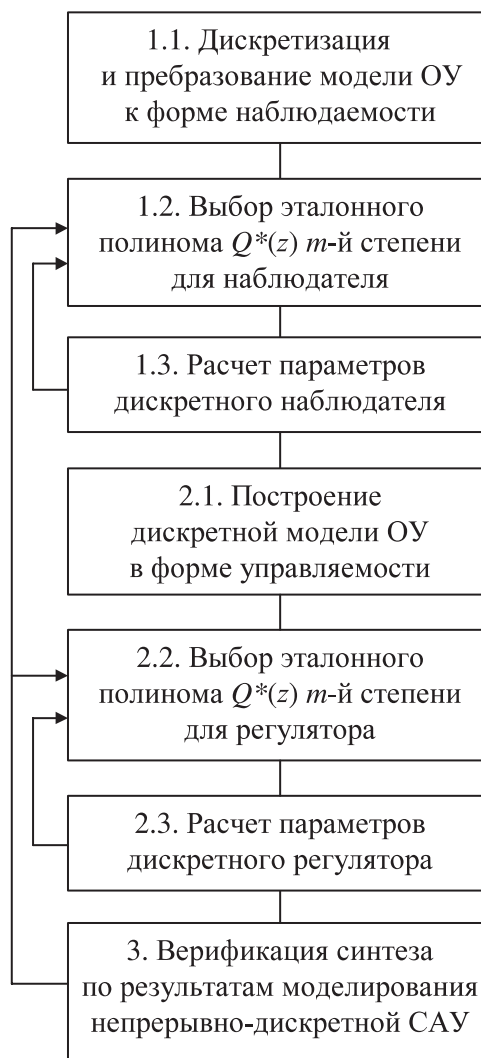


Рис. 1. Этапы синтеза цифровых МР

$K_c^T = (K_{1c}, K_{2c}, \dots, K_{mc})$ обратных связей МР, от значений которых зависят собственные числа матрицы $A_c - B_c K_c^T$ замкнутой дискретной системы и ее динамические свойства. Поскольку матрица A_c имеет каноническую форму, то характеристический полином системы выглядит следующим образом:

$$Q_o(z) = \det(zE - A_c + B_c K_o^T) = z^m + (a_m + K_{mc})z^{m-1} + \dots + (a_2 + k_{2c})z + a_{1c} + K_{1c}.$$

Из условия равенства коэффициентов характеристических полиномов

$$Q_c(z) = Q^*(z) \quad (2)$$

получаем выражение для определения параметров дискретного МР



$$K_{ic} = a_{ic}^* - a_i; i = 1, \dots, m,$$

где a_{ic}^* – коэффициенты эталонного полинома дискретной системы.

Но, поскольку в обратной связи реальной системы используется вектор состояния наблюдателя X_o , а не модели объекта управления X_c , параметры регулятора необходимо пересчитать по формуле:

$$K^T = K_c^T T_{oc}^{-1},$$

где T_{oc} – матрица прямого преобразования дискретной модели из формы наблюдаемости в форму управляемости, вычисленная на этапе 2.1.

Если синтез проводится с использованием неканонической модели ОУ, то для определения параметров МР необходимо составить и решить систему алгебраических уравнений, соответствующую условию (2). Также можно воспользоваться формулой Аккермана, принимающей в данном случае следующий вид:

$$K^T = [0 \ 0 \ \dots \ 0 \ 1] [B_d \ A_d B_d \ A_d^2 B_d \ \dots \ A_d^{m-1} B_d]^{-1} \varphi(A_d),$$

где $[B_d \ A_d B_d \ A_d^2 B_d \ \dots \ A_d^{m-1} B_d]$ – матрица управляемости дискретной модели ОУ; $\varphi(A_d)$ – матрица, удовлетворяющая эталонному характеристическому полиному дискретной системы

$$\varphi(A_d) = A_d^m + a_{mc}^* A_d^{m-1} + \dots + a_{2c}^* A_d + a_{1c}^* E.$$

На 3-м этапе проводится верификация результатов синтеза наблюдателя и МР путем компьютерного моделирования вначале дискретной, а затем непрерывно-дискретной САУ с проверкой ее показателей качества. При этом анализируется влияние собственной динамики дискретного наблюдателя на динамику САУ в целом, а также ее грубость и робастность. При неудовлетворительных результатах осуществляется возврат к предыдущим этапам, на которых уточняется размещение корней эталонных характеристических полиномов наблюдателя и МР внутри круга единичного радиуса на комплексной Z -плоскости.

Изложенная методика предлагалась в [7] для синтеза цифровых электроприводов мобильных роботов, содержащих упругие связи в механических передачах.

Рассмотрим постановки и решения задач оптимизации цифровых САУ с МР на основе минимизации интегральных критериев, косвенно оценивающих динамические показатели системы.

Оптимизация цифровых МР

На практике могут выбираться критерии следующих видов [3, 6, 11]:

$$J_1(e, X) = \frac{1}{N} \sum_{n=0}^{N-1} (e^2[n] + c(\nabla e[n]/T_0)^2 + ru^2[n]); \quad (3)$$

$$J_2(X[n], X) = \frac{1}{N} \sum_{n=0}^{N-1} (X^T[n] Q X[n] + ru^2[n]), \quad (4)$$

где $e[n] = g[n] - y[n]$ – ошибка между ступенчатым входным воздействием и выходом системы; $\nabla e[n] = e[n] - e[n-1]$ – конечная разность ошибки; $u[n]$ – управляющее воздействие; (c, r) – весовые коэффициенты; $X[n]$ – вектор состояния объекта управления; Q – симметрическая положительно полуопределенная весовая матрица.

Интегральные критерии (3) и (4) можно оценивать (вычислять) при любом характере переходных процессов в САУ. Однако такая оптимизация сопряжена с субъективностью выбора весовых коэффициентов критерия. При неудачном задании их получаемые настройки параметров регуляторов могут оказаться не самыми оптимальными, с технической точки зрения.

Минимумам J_1 и J_2 в общем случае соответствуют экстремали (переходные характеристики $e_{ext}[n]$) с перерегулированием. Для его снижения или полного исключения следует увеличивать значения весовых коэффициентов (c, r) или элементов матрицы Q . Критерий J_2 используется также при алгебраическом синтезе МР на основе решения уравнения Риккати [2, 5].

Оба критерия можно применять для оптимизации САУ с нелинейными моделями ОУ, модальными и любыми другими типами регуляторов. При оптимизации МР для вычисления критерия J_2 в качестве вектора состояния $X[n]$ нелинейного ОУ, который не измеряется, можно использовать вектор состояния линейного наблюдателя $X_o[n]$.

Оптимизацию итерационными численными методами можно более эффективно проводить не в пространстве параметров регулятора, а в пространстве параметров используемого метода синтеза.

На схеме такой оптимизации (рис. 2) процедуры синтеза МР и анализа динамики САУ вложены в общий итерационный процесс оптимизации по выбранному критерию. При этом компьютерный синтез МР осуществляется целенаправленным образом, а не методом проб и ошибок. При анализе динамики САУ в ее модели может учитываться ограничение на амплитудные значения выхода регулятора $|u[n]| \leq U_{max}$.

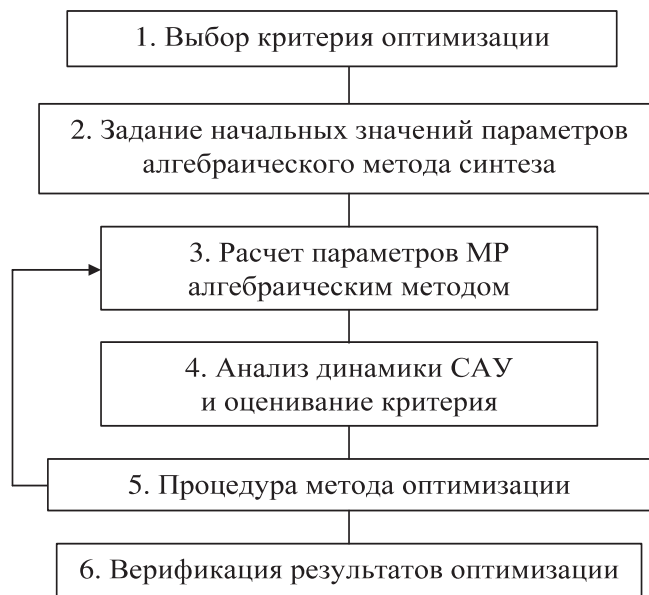


Рис. 2. Схема оптимизации МР, совмещенной с синтезом

Чувствительность динамики САУ и, следовательно, критериев к непосредственным вариациям параметров МР процедурой применяемого итерационного метода оптимизации обычно оказывается высокой. При этом устойчивость САУ не гарантируется. Чувствительность же критериев к вариациям параметров метода синтеза может быть ниже, а процедура синтеза может regularизировать процесс оптимизации. Кроме того, на варьируемые параметры метода синтеза проще накладывать прямые ограничения по устойчивости САУ и другим условиям.

В методе размещения полюсов число варьируемых параметров m равно количеству параметров МР. При этом итерационную оптимизацию, совмещенную с синтезом, можно проводить в пространствах комплексных или только вещественных полюсов. При оптимизации МР на основе решения уравнения Риккати параметрами метода синтеза являются коэффициент r и элементы матрицы Q в квадратичном критерии (4). При задании матрицы Q диагонального вида размерность пространства параметров метода синтеза можно снизить до $(m + 1)$. Но при вариациях весовых коэффициентов в процессе оптимизации устойчивость САУ будет обеспечиваться алгебраическим методом синтеза лишь при условии сохранения выпуклости квадратичного критерия.

Примеры оптимизации

Для ОУ 3-го порядка с передаточной функцией

$$W(s) = \frac{K}{(T_1 s + 1)(T^2 s^2 + 2\xi Ts + 1)}$$

векторно-матричная модель имеет следующую структуру:

$$A_x = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{1}{T^2} & -\frac{2\xi}{T} & -\frac{1}{T^2} \\ 0 & 0 & -\frac{1}{T_1} \end{pmatrix}; B_u = \begin{pmatrix} 0 \\ 0 \\ \frac{K}{T_1} \end{pmatrix};$$

$$B_f = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; C_x^T = (1 \quad 0 \quad 0).$$

Задав значения $K = 100$, $T_1 = 0,005$ с, $T = 0,05$ с, $\xi = 0,5$ с и приняв период дискретности $T_0 = 0,01$ с, осуществим дискретизацию модели и преобразуем ее к канонической форме наблюдаемости:

$$A_o = \begin{pmatrix} 0 & 0 & -a_1 \\ 1 & 0 & -a_2 \\ 0 & 1 & -a_3 \end{pmatrix}; B_o = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}; C_o^T = (0 \quad 0 \quad 1),$$

где

$$a_1 = -0,1108; a_2 = 1,06; a_3 = -1,9179;$$

$$b_1 = 0,0276; b_2 = 0,203; b_3 = 0,0818.$$

При синтезе наблюдателя с канонической структурой зададим для его эталонного характеристического полинома вектор трех полюсов



$$P = [0,05; 0,1 + 0,5i; 0,1 - 0,5i]$$

и определим значения вектора обратных связей и коэффициент передачи наблюдателя:

$$L_o^T = [0,0978; -0,7900; 1,6679];$$

$$K_o = C_o^T (E - A_o + L_o C_o^T)^{-1} B_o = 0,3103.$$

Моделированием можно проверить, что при подключении синтезированного наблюдателя к ОУ ошибка наблюдения имеет малые значения.

При предварительном синтезе МР зададим для эталонного характеристического полинома замкнутой дискретной системы такой же вектор полюсов P , как и при синтезе наблюдателя, и найдем значения векторов обратных связей регулятора и коэффициент передачи системы:

$$K_c^T = [0,0978 \quad -0,7900 \quad 1,6679];$$

$$K^T = K_c^T T_{oc}^{-1} = [2,0953 \quad 4,9294 \quad 7,4503];$$

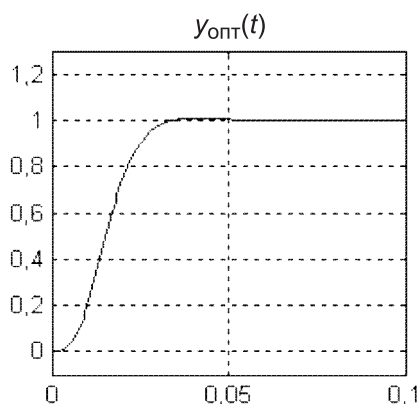
$$K_{sys} = C_c^T (E - A_c + B_c K_c^T)^{-1} B_c = 0,3103;$$

$$K_g = 1/K_{sys} = 3,2231.$$

Однако моделирование синтезированной цифровой САУ показывает, что ее переходная характеристика имеет достаточно большое перерегулирование. Поэтому проведем оптимизацию МР по критерию J_1 , задав весовые коэффициенты ($c = 0,5E - 3$; $r = 2,0E - 3$), при которых переходный процесс в системе не будет иметь перерегулирования.

Пример 1. В качестве оптимизируемых параметров примем вещественный полюс $p1$, вещественную $p2$ и мнимую $p3$ части пары комплексно-сопряженных полюсов системы:

$$P = [p1; p2 + p3i; p2 - p3i].$$



Задав начальный вектор $P_{init} = [0,05; 0,1 + 0,5i; 0,1 - 0,5i]$ и используя для расчетов параметров МР функцию *acker* из библиотеки Control System пакета MATLAB, с помощью функции *fminsearch* через 30 итераций найдем оптимальный вектор полюсов

$$P_{opt} = [0,0570; 0,1362 + 0,2107i; 0,1362 - 0,2107i]$$

и оптимальные параметры регулятора

$$K_{opt}^T = [2,7336 \quad 4,8479 \quad 6,4651]; K_{g,opt} = 2,3859.$$

На рис. 3 представлены кривые переходной характеристики и управляющего воздействия, полученные по дискретно-непрерывной модели оптимизированной САУ.

Пример 2. Проведем оптимизацию МР по тому же критерию J_1 , но используя в качестве варьируемых параметров три вещественных полюса замкнутой системы:

$$P = [p1; p2; p3].$$

Задав начальный вектор из трех полюсов $P = [-0,2; -0,2; -0,2]$ и используя функцию *acker* для расчетов параметров МР, с помощью функции *fminunc* через 10 итераций получим оптимальный вектор вещественных полюсов

$$P_{opt} = [-0,2544 \quad 0,1535 \quad -0,0062]$$

и оптимальные параметры регулятора

$$K_{opt}^T = [2,6925 \quad 6,1135 \quad 8,6759]; K_{g,opt} = 3,4198.$$

Кривые переходной характеристики и управляющего воздействия, полученные по дискретно-непрерывной модели оптимизированной САУ, представлены на рис. 4.

Метод размещения полюсов позволяет алгоритмизировать этапы синтеза цифровых МР, при

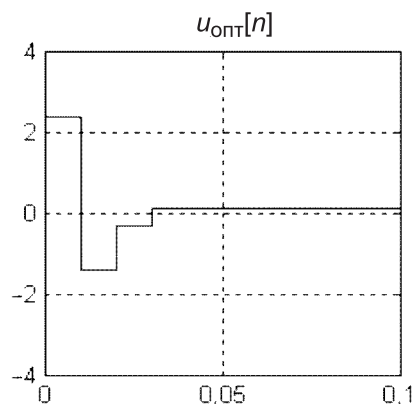


Рис. 3. Результаты оптимизации в пространстве комплексных полюсов

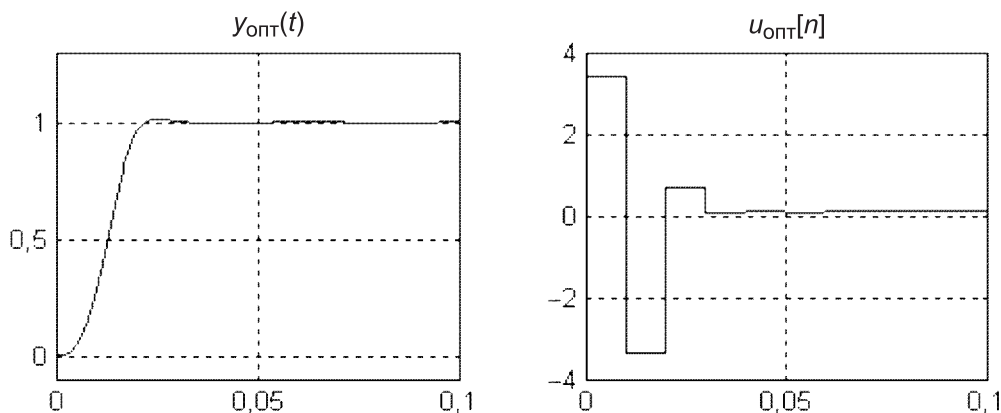


Рис. 4. Результаты оптимизации в пространстве вещественных полюсов

этом наиболее эффективно синтез проводится по каноническим векторно-матричным дискретным моделям ОУ.

Алгебраический синтез цифровых МР целесообразно осуществлять не традиционным способом проб и ошибок, а совмещая его с итерационной параметрической оптимизацией по критериям различного вида, при этом процедуры синтеза, анализа динамики и оценивания показателей цифровой САУ включаются в общий цикл оптимизации.

Итерационную оптимизацию цифровых САУ, совмещенную с синтезом регуляторов, целесообразно проводить в пространстве полюсов или весовых коэффициентов (параметров метода синтеза), размерность которого не превышает количества параметров регулятора.

Для исключения в САУ статических ошибок, синтез и оптимизацию цифровых МР следует проводить с обеспечением в них астатических свойств по методике, изложенной в [3, 6].

СПИСОК ЛИТЕРАТУРЫ

1. Изерман, Р. Цифровые системы управления [Текст]/Р. Изерман; пер. с англ. –М.: Мир, 1984.
2. Козлов, В.Н. Теория автоматического управления [Текст]/В.Н. Козлов, В.Е. Куприянов, В.А. Троицкий [и др.]. –М.: Высш. шк., 2006.
3. Козлов, В.Н. Теория автоматического управления. Компьютерные технологии: учеб. пособие [Текст]/В.Н. Козлов, Н.В. Ростов. –СПб.: Изд-во Политехн. ун-та, 2008.
4. Куо, Б. Теория и проектирование цифровых систем управления [Текст]/Б. Куо; пер. с англ. –М.: Машиностроение, 1986.
5. Методы классической и современной теории автоматического управления: учеб. в 3 т. [Текст]/Т. 2: Синтез регуляторов и теория оптимизации систем автоматического управления; под ред. Н.Д. Егупова. –М.: Изд-во МГТУ им. Н.Э. Баумана, 2000.
6. Ростов, Н.В. Компьютерные технологии в науке. Синтез и оптимизация: учеб. пособие [Текст]/Н.В. Ростов. –СПб.: Изд-во Политехн. ун-та, 2008.
7. Ростов, Н.В. Методика компьютерного синтеза цифровых электроприводов мобильных роботов с наблюдателями состояния [Текст]/Н.В. Ростов, А.М. Щепановский//Экстремальная робототехника: Матер. X науч.-техн. конф. –СПбГТУ, 1999.
8. Franklin, G.F. Feedback Control of Dynamic Systems: 4 ed [Текст]/G.F. Franklin, J.D. Powell, A. Emami-Naeini. –Upper Saddle River: Prentice Hall, 2002.
9. Ogata, K. Discrete-Time Control Systems: 2 ed [Текст]/K. Ogata. –Upper Saddle River: Prentice Hall, 1995.
10. Ogata, K. Modern Control Engineering: 4 ed [Текст]/K. Ogata. –Upper Saddle River: Prentice Hall, 2002.
11. Rostov, N.V. Computer-Aided Design of Digital Control Systems [Текст]/N.V. Rostov, S. Chae, Y.S. Oh. –St. Petersburg: SPbSTU Publishing Center, 2001.



СТРУКТУРНО-СИНТАКСИЧЕСКИЙ ПОДХОД К ПОИСКУ АЛЬТЕРНАТИВ УПРАВЛЕНИЯ СЛОЖНЫМИ СИСТЕМАМИ

Характерной тенденцией современного развития является решение комплексных проблем, требующих тесной взаимосвязи многообразных средств различных аспектов общественной жизни, приводящих к необходимости рассмотрения сложных, трудноописываемых объектов, явлений и процессов. Техника, экономика, социология и образование все больше превращаются в области оперирования сложными эргатическими системами (космические проекты, инновационные экономические и социальные программы и др.).

Изменение типа научных и практических задач, их фактическое превращение в задачи управления организацией и функционированием сложных систем, сопровождается необходимостью формирования общенаучных и специально-научных концепций, использующих в той или иной форме идеи системного подхода. В связи с этим важным представляется продолжение системных исследований в направлении получения новых возможностей управления сложными системами, в т. ч., путем конструирования непривычных представлений и неожиданных стратегий, основанных на известных парадигмах поискового мышления [1]. К таким возможностям относится разработка формализованного аппарата структурно-синтаксического подхода к управлению на основе структурной формализации характеристик сложных систем для постановки и решения задач поиска альтернатив управления.

Сущность структурно-синтаксического подхода

Специфика системного подхода к анализу сложных объектов находит отражение, прежде всего, в специфике первичного системного описания. Первичное системное описание требует формирования в том или ином виде системного объекта, который принято называть структурой системы. Поскольку системный подход ориентирует на раскрытие механизмов целостности, а также на выявление многообразных типов связности сложного объекта (сложной системы) и

сведение их в единую теоретическую картину, структурное описание представляется наиболее приспособленным для решения этих задач.

Структурное описание предлагается формализовать с использованием методов синтаксического описания, т. е. построения синтаксических моделей формальных структур [2], которые позволяют обобщенно отображать аддитивные и целостные свойства связности всех аспектов системного представления. Структурно-синтаксический подход рассматривается в качестве альтернативного подхода к описанию классов состояний сложных объектов в процессе реализации принципа ситуационного управления [3].

Предлагаемый подход к реализации ситуационного управления назван «идентификационно-структурным управлением», поскольку предполагает сочетание структурного подхода к описанию состояний объекта управления с необходимостью решения задач структурной идентификации. Принцип идентификационно-структурного управления (сокращенно – *S*-принцип), представляющий структурную модификацию принципа ситуационного управления, сформулирован как «возможность управления объектом, как сложной системой, на основе идентификации структур его состояния в пространстве структур ситуаций с выбранными альтернативами управления» [4]. Структурный подход в сочетании с *S*-принципом представляется одним из перспективных направлений методологии системных исследований [5] в силу его универсальности, с точки зрения принципа поискового мышления и инвариантности относительно предметной области и уровня сложности исследуемого объекта

Познавательные установки структурного подхода и *S*-принципа базируются на познавательных установках ситуационного управления. Для их применения необходимы методы идентификации структурной связности объекта управления в классе множеств описаний структурной связности его эталонных состояний. Следовательно, можно утверждать, что необходимым условием

реализации структурного подхода к управлению сложными системами на основе S -принципа является адекватное структурное описание целостных свойств состояний таких систем средствами, инвариантными относительно семантики предметной области.

К выбору аппарата структурного моделирования предлагается подходить с позиций его возможностей синтаксической формализации структурных описаний. Под синтаксическим структурным описанием везде в дальнейшем будем понимать синтаксические модели формальных структур, т. е. структур, отображающих свойства связности сложных систем без конкретизации специфики аспекта описания, определяемой сущностями элементов и видов отношений [4]. Обобщенную синтаксическую структурную модель (ОСМ) формализовано представим кортежем $S_q^k = \langle E, V_E^P, V_S^P \rangle$. Здесь $E = \langle E_S, E_I \rangle$ – носитель модели; E_S – подмножество основных элементов носителя, задающих элементы моделируемой системы на выбранном уровне глубины структурного описания; $E_I \subset I$ – подмножество вспомогательных элементов носителя, идентифицирующее подмножество E_S , его элементы и атрибуты элементов; I – универсальное множество элементов; V_E^P – подмножество предикатов, задающих виды отношений между элементами подмножеств E_S и E_I ; V_S^P – подмножество предикатов, задающих виды отношений элементов подмножества E_S .

Подмножество основных элементов E_S синтаксически идентифицируется во множестве E с помощью двухместного предиката v_π , который задается на подмножестве $E = E_S \cup \pi, E' \subset E$, где $\pi \in E_I$ – некоторый элемент номинальной шкалы, так что $\forall e_i \in E_S \forall e_j \in E_I \setminus \pi (v_\pi(e_i, \pi) = 1 \& v_\pi(e_j, \pi) = 0)$. Каждый основной элемент носителя синтаксически идентифицируется (нумеруется) в подмножестве E_S с помощью двухместного предиката v_π , который задается на подмножестве $E_S \cup N_o \subset E$, где $N_o \subset E_I$ – подмножество элементов номинальной шкалы, так что $\forall e_i \in E_S \forall e_j \in E \setminus N_o \exists n_k \in N_o \times (v_\pi(e_i, n_k) = 1 \& v_\pi(e_j, e_j) = 0)$.

Элемент подмножества E_S идентифицируется в подмножестве значений c -го атрибута $A_c = \{a_1^c, a_2^c, \dots, a_e^c, a_m^c\}$, $c = 1, 2, \dots, q$, $|E_S| \neq m$, с помощью двухместного предиката $v_{E_c}^P \in V_E^P$, заданного на $E_S \cup N_c \subset E$, содержащем подмножество элементов номинальной шкалы N_c , для которого существует единственное отображение $f_c: A_c \rightarrow N_c$.

Двухместный предикат $v_{E_c}^P \in V_E^P$ задается так, что $\forall e_i \in E_S \forall e_j \in E \setminus N_c \exists n_e^c \in N_c (v_{E_c}^P(e_i, n_e^c) = 1 \& v_{E_c}^P(e_j, e_j) = 0)$. Введенное синтаксическое правило позволяет в рамках ОСМ моделировать количественное или качественное значение a_e^c любого конкретного c -го атрибута, задаваемого множеством своих значений A_c , $c = 1, 2, \dots, q$, определяемым спецификой предметной области в процессе системного описания.

Для моделирования множества видов отношений между элементами подмножества E_S используется подмножество предикатов $V_S^P = \{v_{S_1}^P, v_{S_2}^P, \dots, v_{S_h}^P, v_{S_k}^P\}$. Связность h -го вида элементов E_S формализуется с помощью $v_{S_h}^P \in V_S^P$ как h -я синтаксически правильная структурная совокупность, удовлетворяющая условию:

$$\forall e_p, e_q \in E_S \forall e_k \in E_I [v_{S_h}^P(e_p, e_q) = 1 \vee v_{S_h}^P(e_p, e_q) = 0 \& v_{S_h}^P(e_q, e_k) = 0 \& v_{S_h}^P(e_p, e_k) = 0].$$

Таким образом, синтаксические правила формирования ОСМ включают правила синтаксического описания элементов структурной модели E_S и видов отношений на множестве элементов. Они формализуются с помощью сигнатуры ОСМ $V^P = \langle V_E^P, V_S^P \rangle$. Модель S_q^k предлагается в качестве обобщенной основы формирования, в терминах теоретико-множественного подхода, конкретных синтаксических структурных моделей сложных систем. Использование предлагаемой ОСМ фактически означает переход от содержательного описания структуры сложной системы к ее синтаксическому аналогу, формализованному средствами, инвариантными относительно предметного описания, однако обладающему структурной реальностью, необходимой для предметного анализа.

Формализация структурно-синтаксического подхода

Поскольку задачи идентификации систем, описываемых с применением наиболее общего математического аппарата, в частности, теоретико-множественных отношений, в настоящее время недостаточно проработаны [5], формализации их постановок, в первую очередь, требуют формализации выделенных множеств и отношений, с точки зрения систематизации процесса сравнения, приводящего к содержательным результатам. В соответствии с S -принципом идентификационно-



структурного управления, сущность стратегии структурного анализа состоит в выявлении «эквивалентных», «связноподобных» структур различного уровня «связноподобности». Для реализации структурно-синтаксического подхода необходимо четко определить, что такое «близкая система» [5], и в этой связи формализовать понятие «эквивалентные» структуры. В частности, предлагается интерпретировать понятие «эквивалентные структурные представления сложных систем» в терминах рассматриваемого структурно-синтаксического подхода с использованием формализованных определений различных типов изоморфных отображений ОСМ $S^k = \langle E, V_E^P, V_S^P \rangle$. При определении типов изоморфных отображений ОСМ вида $S^k = \langle E, V_E^P, V_S^P \rangle$ будем считать, что множества предикатов $V_E^P = \{v_{E_0}^P, v_{E_1}^P, \dots, v_{E_c}^P, \dots, v_{E_q}^P\}$, $V_S^P = \{v_{s_1}^P, v_{s_2}^P, \dots, v_{s_h}^P, v_{s_k}^P\}$ априорно упорядочены и $v_{E_0}^P = v_n^P$.

Определение 1. Две q -атрибутные ОСМ k -го порядка назовем изоморфными по $v_{s_h}^P$ -му виду структурной связности или h -изоморфными, где $h = 1, 2, \dots, k, J = \{1, 2, \dots, c, \dots, q\}$, если выполняется условие: $\forall e_i^1 \in E_{s_1} \forall c \in J \forall N_c^1 \subset E_1 \forall n_{c_j}^1 \in N_c^1 \exists e_i^2 \subset \subset E_2 \exists N_c^2 \subset E_2 \exists n_{c_j}^2 \in N_c^2 \forall e_i^1, e_r^1 \in E_{s_1} \exists e_i^2, e_r^2 \in E_{s_2} \exists v_{s_h}^P \in V_S^P [e_i^1 \leftrightarrow e_i^2 \& N_c^1 \leftrightarrow N_c^2 \& n_{c_j}^1 \leftrightarrow n_{c_j}^2 \& v_{E_c}^P \times (e_i^1, n_{c_j}^1) = v_{E_c}^P (e_i^2, n_{c_j}^2) = 1 \& n_{c_j}^1 = n_{c_j}^2 \& v_{s_h}^P \times \times \& v_{s_h}^P (e_i^1, e_r^1) = v_{s_h}^P (e_i^2, e_r^2)]$.

Поскольку $S_{q_1}^k$ и $S_{q_2}^k$ находятся в отношении изоморфизма, которое, в соответствии с определением (1), удовлетворяет свойствам рефлексивности, симметричности и транзитивности, будем считать, что $S_{q_2}^k$ является изоморфным отображением $S_{q_1}^k$ по $v_{s_h}^P$ -му виду структурной связности или h -изоморфным отображением $S_{q_1}^k$ и, наоборот. Обозначим такой тип изоморфного отображения $\phi_i(h) : S_{q_1}^k \cong S_{q_2}^k$ или $S_{q_1}^k \cong S_{q_2}^k$. По аналогии формализуем понятие изоморфизма (изоморфного отображения) по l -видам структурной связности.

Определение 2. Две q -атрибутные ОСМ k -го порядка $S_{q_1}^k = \langle E_1, V_E^P, V_S^P \rangle$ и $S_{q_2}^k = \langle E_2, V_E^P, V_S^P \rangle$ будем называть изоморфными в смысле изоморфизма l -го порядка, если при $V_{s_l}^P \subset V_S^P$, $|V_{s_l}^P| = l, l < k, J = \{1, 2, \dots, q\}$ выполняется условие: $\forall e_i^1 \in E_{s_1} \forall c \in J \forall N_c^1 \subset E_1 \forall n_{c_j}^1 \in N_c^1 \exists V_{s_l}^P \subset V_S^P \forall v_{s_h}^P \in V_{s_l}^P \exists e_i^2 \in E_{s_2} \exists N_c^2 \subset E_2 \exists n_{c_j}^2 \in N_c^2 \forall e_i^1, e_r^1 \in E_{s_1} \exists e_i^2, e_r^2 \in E_{s_2} [e_i^1 \leftrightarrow e_i^2 \& N_c^1 \leftrightarrow N_c^2 \& n_{c_j}^1 \leftrightarrow n_{c_j}^2 \& v_{E_c}^P (e_i^1, n_{c_j}^1) = v_{E_c}^P (e_i^2, n_{c_j}^2) = 1 \& n_{c_j}^1 = n_{c_j}^2 \& v_{s_h}^P (e_i^1, e_r^1) = v_{s_h}^P (e_i^2, e_r^2)]$.

Определение (2) изоморфизма $S_{q_1}^k$ и $S_{q_2}^k$ формирует понятие изоморфного отображения l -го порядка. В соответствии с определением (2) будем считать, что $S_{q_2}^k$ является изоморфным отображением l -го порядка $S_{q_1}^k$ и наоборот. Введенный тип отображения назовем изоморфным отображением l -го порядка q -атрибутных ОСМ k -го порядка для $e < k$, и обозначим $\phi_i^l : S_{q_1}^k \cong S_{q_2}^k$ или $S_{q_1}^k \cong S_{q_2}^k$. Очевидно, что случай $l = k$ является предельным в смысле возможного порядка изоморфного отображения q -атрибутных ОСМ произвольного k -го порядка. Изоморфные отображения k -го порядка q -атрибутных ОСМ k -го порядка будем называть предельно-изоморфными или просто изоморфными обозначать $\phi_i^l : S_{q_1}^k \cong S_{q_2}^k$ или $S_{q_1}^k \cong S_{q_2}^k$.

Определения (1), (2) позволяют сформулировать следующие утверждения относительно достаточных условий существования введенных типов изоморфных отображений q -атрибутных ОСМ k -го порядка.

Утверждение 1. Две q -атрибутные ОСМ k -го порядка $S_{q_1}^k = \langle E_1, V_E^P, V_S^P \rangle$ и $S_{q_2}^k = \langle E_2, V_E^P, V_S^P \rangle$ будут изоморфны в смысле изоморфизма l -го порядка, где $l = 1, 2, \dots, k-1$, если они изоморфны в смысле изоморфизма $l+i$ -го порядка, где $i = 1, 2, 3, \dots, k-1$.

Данное утверждение логически следует из (2) и определяет, в качестве достаточного условия изоморфного отображения l -го порядка q -атрибутных ОСМ k -го порядка, при $l < k$, существование изоморфного отображения высшего порядка.

Утверждение 2. Две q -атрибутные ОСМ k -го порядка $S_{q_1}^k \cong S_{q_2}^k$ будут изоморфны по $v_{s_h}^P$ -му виду структурной связности, т. е. h -изоморфными, где $h \leq k$, если они изоморфны в смысле изоморфизма l -го порядка, где $h \leq l$.

Справедливость утверждения (2) следует из определений (1), (2) при выполнении требования однозначного упорядочения множества V_S^P , которое, в данном случае, производится без каких-либо дополнительных ограничений.

В терминах формализованных типов изоморфных отображений ОСМ $S_{q_1}^k$ предлагается, с учетом условий существования таких отображений, формализовать постановки и методы решения задач реализации структурно-синтаксического подхода к поиску альтернатив управлений сложными системами. Если структуры исследуемых и эталонных по управлению состояний задавать синтаксически структурными моделями вида $S_{q_1}^k$, задачи

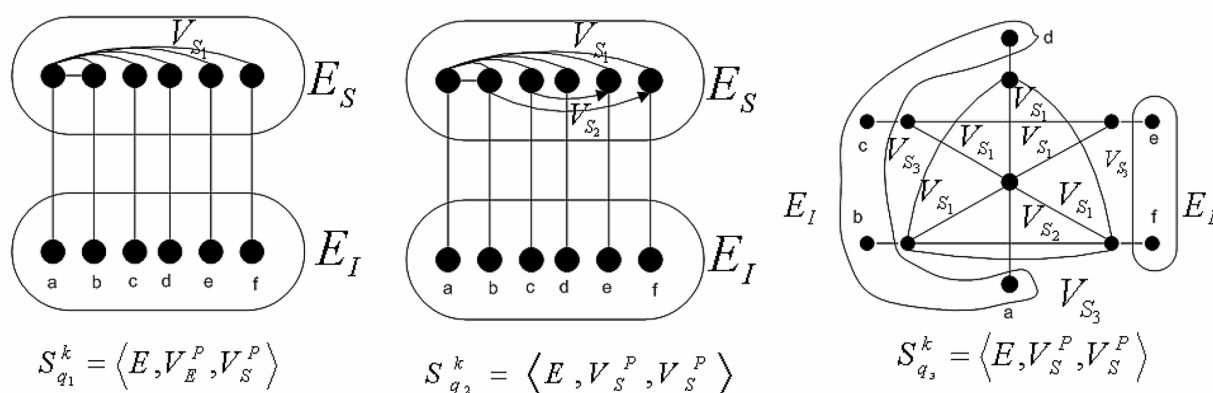


Рис. 1. Графическая интерпретация модели эталонной функциональной организации

поиска альтернатив управления сложными системами, в соответствии с S -принципом, формализуются и решаются как задачи поиска введенных типов изоморфных отображений ОСМ.

Проиллюстрируем возможности предлагаемого подхода на примере выбора оптимальной альтернативы функциональной организации промышленной компании. Пусть предложены две альтернативы. Первая представляет собой традиционную централизованную функциональную организацию, при которой каждая из функций (исследования и разработки, производство, маркетинг) передана самостоятельным организационным структурам, управляемым непосредственно руководством. Вторая альтернатива отличается от первой тем, что функциональная организация дополняется объединением однородных видов деятельности. Вводятся связи между функциональными единицами (отделами), объединенными в одну организационную структуру. В частности, формируются две подструктуры. Организационно в одну группу объединяются отдел исследования и разработок с отделом маркетинга и сбыта, а

в другую группу – администрация и отдел кадров. В качестве эталонной, выбрана организационная структура, в которой отдел сбыта, производственный отдел и отдел НИОКР объединены в «продуктовую группу», на которую возложены задачи координации деятельности с целью разработки стратегической политики. На рис. 1 приведены графические интерпретации ОСМ альтернатив и эталонной функциональной организации.

Первая альтернатива представлена ОСМ $S_{q_1}^k = \langle E, V_E^P, V_S^P \rangle$, в которой $k = 1$; $|V_S^P| = 1$; $q_1 = 1$; $|V_E^P| = 1$; $A_C = \{a, b, c, d, e, f\}$, где a – руководство; b – отдел сбыта; c – администрация; d – производственный отдел; e – отдел кадров; f – отдел НИОКР. Вторая альтернатива описана $S_{q_2}^k = \langle E, V_E^P, V_S^P \rangle$. Здесь $k = 2$; $|V_S^P| = 2$; $q_2 = 1$; $|V_E^P| = 1$; $A_C = \{a, b, c, d, e, f\}$. ОСМ Эталонная функциональная организация задана ОСМ $S_{q_3}^k = \langle E, V_E^P, V_S^P \rangle$, в которой $k = 3$; $|V_S^P| = 3$; $q_3 = 1$; $|V_E^P| = 1$; $A_C = \{a, b, c, d, e, f\}$.

Поиск изоморфных отображений типа $\varphi_I^e: S_{q_3}^k \cong S_{q_1}^k$ и $\varphi_I^e: S_{q_3}^k \cong S_{q_2}^k$ показал, что для первой

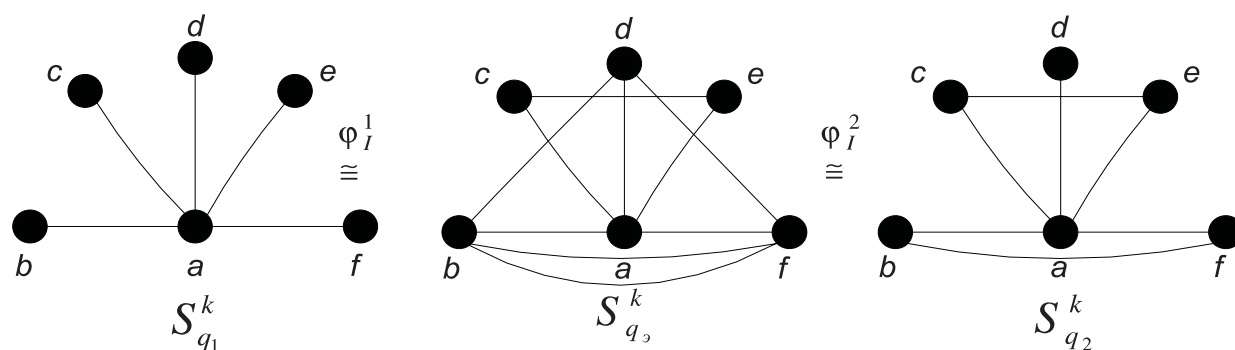


Рис. 2. Изоморфные отображения альтернатив



альтернативы может быть получено изоморфное отображение первого порядка $\varphi_I^1: S_{q_1}^k \cong S_{q_1}^k$, а для второй альтернативы – изоморфное отображение второго порядка $\varphi_I^2: S_{q_2}^k \cong S_{q_2}^k$ (рис. 2).

Следовательно, в соответствии с сущностью S -принципа, в терминах формализации изоморфных отображений ОСМ можно сделать вывод, что вторая альтернатива организационной структуры предпочтительнее, поскольку для ее ОСМ получено изоморфное отображение более высокого порядка в ОСМ эталонной структуры.

Формализация поиска альтернатив управлений сложными системами, как поиска различных типов изоморфных отображений ОСМ $S_{q_i}^k$, представляет собой альтернативный вариант формализации принятия решений по управлению в си-

туациях, когда невозможны или, по крайней мере, сильно затруднены, непосредственная причинно-следственная оценка взаимосвязи альтернативы управления с параметрами управляемого объекта и аналитическое описание такой взаимосвязи. Предлагаемый подход реализуется как на этапе подготовки множеств эквивалентных эталонных по управлению состояний, так и на этапе идентификаций исследуемого состояния в классы множеств эталонных состояний. Использование синтаксических структурных моделей позволяет применять инвариантные, относительно предметной области, формализованные постановки и методы поиска различных типов их изоморфных отображений для интерпретации задач и методов идентификационно-структурного управления.

СПИСОК ЛИТЕРАТУРЫ

1. Антонов, А.В. Системный анализ [Текст]/А.В. Антонов; 3-е изд., стер.–М.: Высш. шк., 2008.–454 с.
2. Клир, Дж. Системалогия. Автоматизация решения системных задач [Текст]/Дж. Клир; пер. с англ.–М.: Радио и связь, 1990.–544 с.
3. Поспелов, Д.А. Ситуационное управление: теория и практика [Текст]/Д.А. Поспелов. –М.: Наука, 1986.–288 с.
4. Щербань, А.Б. Обобщённые структурные модели информационных объектов. [Текст] /А.Б. Щербань, К.Е. Братцев, Т.В. Жашкова [и др.]/Изв. высш. учеб. заведений: Поволжский регион. Сер. Технические науки.–2009.–№ 1 (9).–С. 12–22.
5. Касти, Дж. Большие системы. Связность, сложность и катастрофы [Текст]/Дж. Касти; пер. с англ. –М.: Мир, 1982.–216 с.

УДК 004.031.42

И.А. Посов, С.Е. Рукшин

АРХИТЕКТУРА СИСТЕМЫ ПРОВЕДЕНИЯ УДАЛЕННЫХ СОРЕВНОВАНИЙ И ОРГАНИЗАЦИИ РАБОТЫ С МАТЕМАТИЧЕСКИМИ ЗАДАЧАМИ

Одной из серьезных проблем, с которой столкнулось педагогическое сообщество в конце XX в. при попытке ввести в практику средства автоматизации учебного процесса, стало упрощенное представление механизмов усвоения знаний обучаемым. Развитие программированного обучения (АСО) основывалось на предположении о том, что обеспечение обратной связи в форме тестов, с которыми чередуется изложение материала, будет достаточным для эффективного обучения любому

предмету. Такой упрощенный взгляд был подвергнут обоснованной и содержательной критике, АСО были названы «электронными переворачивателями страниц» [1], однако, с развитием сети Интернет мы снова видим возрождение этого подхода в системах дистанционного обучения (ДО).

Основной причиной послужила реализация технократического подхода к поддержке педагогических процессов. «Программистами» создавалась система, которую «педагогам» предлагалось использовать сообразно тем правилам

и методикам, которые определялись техническими параметрами системы. При этом реальный педагогический опыт преподавателя практически игнорировался: ему предлагалось приспособить свои знания к данной системе.

После появления в 80-х гг. работ Симура Паперта (в старой транскрипции – Сеймура Пейперта) [2] стало активно развиваться другое направление: создание микромиров. В микромирах научные знания представлялись моделями и инструментами, которыми ученик мог манипулировать, и именно управляемая деятельность и рефлексия становились основой для овладения фундаментальными идеями. Это направление дало толчок к развитию предметных сред обучения [3], которое в математике особенно ярко проявилось в развитии сред динамической геометрии [4].

В то же время, в системах поддержки ДО мы можем наблюдать, в основном, старый подход, когда для организации контроля знаний и проведения научных соревнований используются системы с узкоспециализированным набором форм организации деятельности и ограниченным набором форм представления решений и ответов на задания.

Крайнюю позицию занимают системы тестирования, в которых ответ выбирается из предложенных. Использование таких систем в организации ЕГЭ вызвало широкую критику, хотя имеются и удачные примеры использования этой технологии при проведении конкурса «Кенгуру» и осуществлению многопараметрического анализа знаний выпускников школ [5]. В любом случае, мы видим, что ограничения на формы деятельности преподавателя играют существенную ограничивающую роль для его творческой деятельности [6].

Эффективность систем динамической геометрии и их широкое распространение основано на разделении предметной и дидактической части в программных продуктах этого класса [3]: программисты работают над совершенствованием модели предметной среды (инструменты геометрических построений, преобразований, измерений и пр.), не вводя в систему никаких «дидактических» ограничений; преподаватели математики строят дидактическое окружение самостоятельно, сообразно их педагогическому опыту, методическим целям и особенностям контингента.

Целесообразно перенести этот подход на создание систем для проведения удаленных соревнований и организации работы с математическими задачами.

Постановка задачи

Для постановки задачи рассмотрим существующие подходы, используемые для проведения соревнований и организации работы с задачами. В первую очередь таким подходом является применение модулей тестирования, распространяющихся с системами ДО, самыми известными из которых являются Moodle, Sakai, .LRN. Системы позволяют создавать тестирования и проводить их среди учеников. Эти системы являются универсальными, т. е. подходят для широкого класса учебных предметов и поддерживают различные регламенты участия, включая настройки времени участия, ограничения на просмотр своих результатов и др. Недостатком подобных систем является ограниченность способов общения с учеником: в базовом случае участник может лишь выбирать варианты из нескольких предложенных или вводить ответ в строку ввода. Создавать тестирования можно вне рамок систем удаленного обучения. В Интернете в виде онлайн сервисов существует ряд систем для создания и проведения тестирований. Основное ограничение этих систем также состоит в ограниченности количества способов общения с участником.

Наибольшее число систем поддержки проведения олимпиад разработано для олимпиад по программированию [7]. Помимо проведения самих соревнований, они используются в кружках по программированию для проведения занятий и тренировок. Некоторые системы доступны в Интернете и могут использоваться для самостоятельных тренировок. Основной особенностью таких систем является широта спектра анализируемых решений: программа, написанная на любом (допустимом системой) языке, может автоматически анализироваться системой в качестве решения поставленной задачи. Решение проверяется на множестве тестов, которые пишутся авторами для каждой задачи. Последнюю особенность мы можем отождествить с одной из основных черт систем динамической геометрии – верификацией ответа на множестве примеров, которая позволяет при необходимости обеспечить обратную связь [8], и дает основу для анализа ошибок, допущенных при решении задачи. В то же время, регла-



мент, принятый в системах поддержки соревнований по программированию в существующих системах поддержки соревнований [7] встроен в систему, что не позволяет использовать ее для организации мероприятий по работе с этими задачами на основе другого регламента.

Последний класс систем – это системы, разработанные специально для проведения определенных олимпиад. Специализированные системы идеально справляются с олимпиадами, для которых были разработаны, но и набор возможностей по общению с учеником в них ограничен возможностями, необходимыми для конкретной олимпиады.

Таким образом, актуальной является задача разработки системы организации удаленной поддержки работы с задачами математического характера (включая задачи по информатике, физике, химии), которая:

- допускает широкий спектр видов заданий и способов проверки правильности их выполнения;

- позволяет легко комбинировать задачи с различными типами оформлений решений;

- позволяет создавать различные регламенты работы, включающие временные параметры работы ученика, особенности диалога с ним в процессе автоматической обработки решений и оглашение частичных или полных результатов работы.

Приведем примеры желательных представлений решения задачи участником олимпиады, отсутствующих в существующих системах проведения олимпиад. Так, в олимпиадах по математике естественно в качестве ответа просить участника ввести объект, более сложный, чем число. Например, ввести ответ в графической форме или представить его предикатом от параметров задачи (т. е. в качестве ответа представить логическое суждение). Для решения задач по геометрии целесообразно использовать в качестве подключаемых модулей современные средства динамической геометрии. Они позволяют производить построения на чертеже, используя виртуальные инструменты, а также проверять правильность алгоритмов построений. Это дает возможность расширить класс геометрических задач, доступных автоматической проверке, задачами на построение.

Архитектура системы

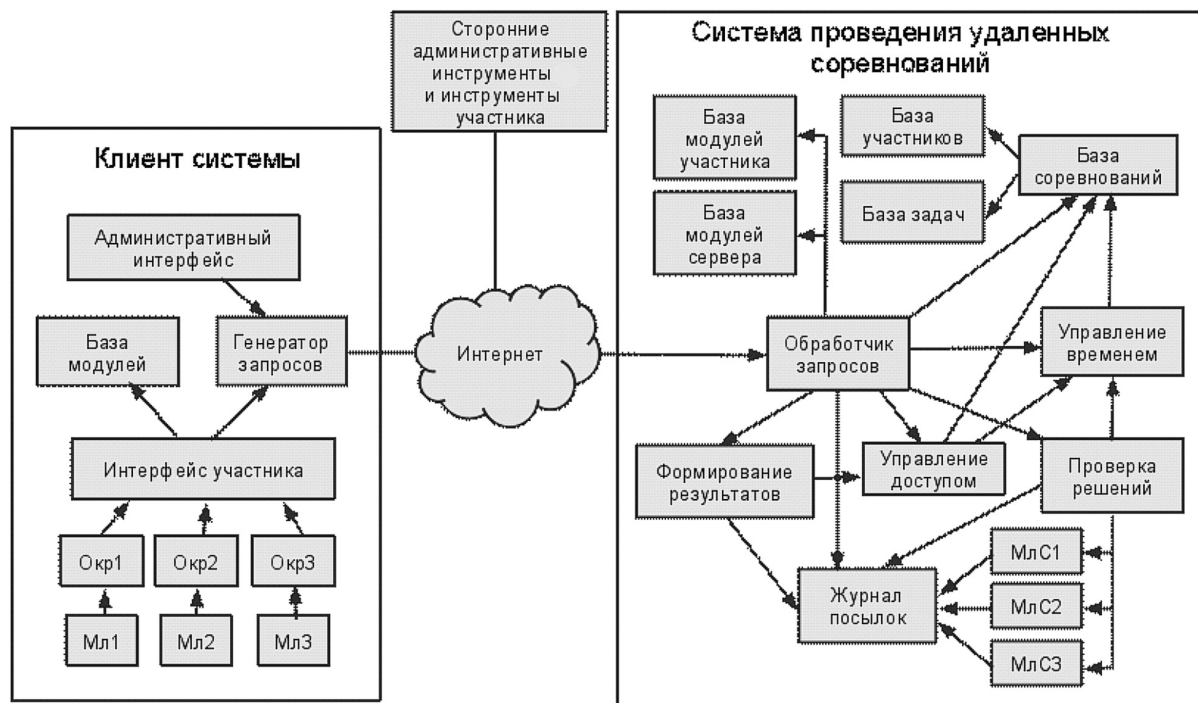
Потребность расширять способы ввода и проверки решений накладывает требования на

формат представления задач в системе. Кроме условия и ответа, они должны содержать другую информацию, необходимую для системы проведения соревнований. Информацию, которая относится к одной задаче, можно понимать как набор файлов, хранящихся в некоторой папке или архиве. Данные архива могут значительно различаться по структуре в зависимости от задачи, и должны храниться в базе данных системы как атомарные объекты, с которыми система не может работать напрямую, а работает только через модули отбраковки условия и проверки решения.

Внутри архива задача должна содержать текст условия на естественном языке и формальное условие. Например, в геометрических задачах формальным условием будет файл с геометрическим построением и предикат, который используется для проверки правильности построения. Аналогично условию задачи, ответ также может быть разделен на две части – формальный и неформальный. В отличие от формального условия, формальный ответ должен обязательно присутствовать в задаче. Это те данные, которые используются для автоматической проверки решения, и обойтись без них невозможно. Неформальный ответ – это отформатированный текст с разбором задачи.

Формальное условие и формальный ответ не всегда удается отделить друг от друга. Например, в геометрических задачах на построение формальный ответа совпадает с условием, т. к. является проверочным предикатом. При разработке формата конкретного класса задач, разработчик должен самостоятельно принять решение, как распределить по файлам формализованные условия и ответы.

Формальное условие и формальный ответ не могут существовать без того объекта, который способен их понять. Каждой задаче необходимо содержать информацию о тех программных модулях, которые отображают формальное условие и автоматически проверяют ответ. Так же, помимо разобранной выше информации, архив с задачей может хранить дополнительные данные. Например, целесообразно хранить в задаче исходные данные, использовавшиеся для ее создания. Задачу в таком сложном формате невозможно собирать вручную, особенно если ее составляют преподаватели и методисты, не имеющие опыта программирования. Поэтому для составления задач необходимо разрабатывать отдельное про-



Общий план архитектуры системы

граммное обеспечение, которое будет хранить в файле с задачей дополнительную необходимую информацию.

Перейдем к обзору архитектуры самой системы проведения соревнований; ее общий план изображен на рисунке. При разработке архитектуры учитывался следующий набор возможностей системы:

- управление правами на участие в соревнованиях;
- выдача участникам условий задач и модулей для решения задач;
- слежение за временем соревнования;
- перенаправление решений участников модулям проверки решений и составление таблицы результатов по получаемым ответам;
- сохранение истории отосланных решений;
- выдача результатов соревнования в соответствии с правами на просмотр;
- администрирование банков задач и соревнований;
- организация общения участников и жюри, выдача информация о соревнованиях и пр.

Система должна представлять собой сервис, принимающий запросы от участников соревнований и администраторов. Сервис обрабатывает

запросы, которые варьируются от запросов на изменение параметров соревнований, таких, как настройки регламента, времени и других, до запросов участников, содержащих сдаваемые для проверки решения. В левой части изображен клиент системы, а справа – веб-сервис, система проведения соревнований. Стрелки отображают зависимости модулей друг от друга по данным или по исполнению. Выделение системы проведения соревнований в отдельный сервис и документирование его интерфейса дает возможность разработчикам реализовывать сторонние инструменты для работы с системой проведения соревнований, не ограничиваясь разработкой инструментов в рамках платформы, на которой создан основной клиент системы.

Клиент системы состоит из интерфейса участника, использующегося участниками во время проведения олимпиады, административного интерфейса, применяющегося жюри для создания соревнований и наполнения их задачами и участниками. Клиент системы хранит базу модулей участника, необходимых для отображения разных типов интерфейса ввода решений. При подключении участника к соревнованию модули активизируются и подключаются к интерфейсу участника



через окружение модулей. Окружение хранит информацию о решаемой задаче и предоставляет модулю интерфейс для доступа к генератору запросов для отправки решения задач и получения ответа.

На стороне сервера хранятся модули ввода решений. Они необходимы только для того, чтобы раздавать их участникам, которые по той или иной причине еще не имеют их у себя. Модули выдаются до начала олимпиады, чтобы уменьшить нагрузку на сервер. На сервере находятся модули проверки решений, которые получают информацию от модулей ввода и сообщают информацию о результате проверки. Система передает им решения участников и ждет ответ. Ответ приходит позже, когда задача будет проверена; он содержит сообщение, верно ли решена задача, количество баллов или другую релевантную информацию о результатах проверки.

Все решения, посылаемые участниками, сохраняются в журнале посылок. Результаты обработки решений, полученные в модулях проверки, также записываются в журнале посылок. Журнал посылок необходим для хранения истории, проведения апелляций и решения проблем, связанных с указанием неправильного ответа к задаче. Пройдя по журналу посылок, можно перепроверить решения участников после смены ответа. Каждый участник имеет доступ к связанным с ним записям журнала посылок.

Система следит за регистрацией участников соревнований. В некоторых случаях ее может производить только жюри, например, если это второй тур, в который прошли не все участники первого. Система выдает участникам задачи. Как было описано выше, задача содержит внутри себя намного больше информации, чем требуется для участника, поэтому часть обязанностей системы состоит в том, чтобы выделить информацию, которая будет отправлена участнику. Выдача условий задач может происходить либо одновременно, либо (как бывает при самостоятельной работе учащихся) в момент, когда участник запустил свое соревнование.

Система проведения олимпиад следит за временем. В простом понимании это означает, что участники не должны иметь возможность послать решение или получить условия до того, как олимпиада началась или после того, как она

закончилась. В слежение за временем входят и другие задачи. Например, от времени зависит, доступны ли участнику результаты свои или других участников. При классическом тестировании после окончания доступны только свои результаты, а до окончания – ничьи. В олимпиадах во время соревнования доступны результаты других участников. Исключением является конкурс «Конструируй, исследуй, оптимизируй» (КИО) [9], в котором участники соревнуются в решении оптимизационных задач, при этом не зная о прогрессе конкурентов.

За формирование результатов отвечает отдельный модуль, в первом приближении он соответствует генератору турнирной таблицы. Таблица хранит только часть информации о решениях участников и выдает разную информацию для разных типов пользователей. Например, если в настройках соревнования указано, что участник во время олимпиады имеет доступ только к своим результатам, то при формировании результатов для отображения участнику будет создана только одна строка турнирной таблицы. Помимо уже обсужденных особенностей соревнований, связанных со временем доступа к результатам, сами результаты имеют сложную структуру. В классической ситуации участник получает за каждую задачу некоторое количество баллов. В некоторых олимпиадах, кроме баллов за задачу, участник получает штрафные баллы, например, за неверные попытки. В конкурсе КИО формат результата по задаче отличается в каждой задаче. Если в задаче требуется построить, например, как можно более короткий путь, результатом будет одно число – длина пути. В другой постановке это могут быть уже два числа, например, длина пути и количество требующихся поворотов. Поэтому модуль формирования результатов должен уметь правильно отобразить результаты участников, сравнить их и расположить по порядку, определив победителей.

Мы рассмотрели возможности, которые, в общем случае, должна иметь система проведения удаленных соревнований. Основные требования к системе – это поддержка большого количества методов ввода и проверки решений с возможностью расширения и поддержка различных регламентов проведения соревнования. Требование

расширяемости делает необходимым ввод таких сущностей, как модули создания задач, ввода и проверки решений, и документированный интерфейс общения модулей с системой. Набор

модулей не фиксирован, способен расширяться. Отсутствующие модули могут быть реализованы и добавлены в систему по инициативе преподавателей и методистов.

СПИСОК ЛИТЕРАТУРЫ

1. Nivergelt, J.A. Pragmatic Introduction to Courseware Design [Текст]/J.A Nivergelt//Computer, IEEE Computer Society.–Sept. 1980.–P. 7–20.
2. Пейперт, С. Переворот в сознании. Дети, компьютеры и плодотворные идеи; пер. с англ.–М.: Педагогика, 1989.
3. Башмаков, М.И. Информационная среда обучения [Текст]/М.И. Башмаков, С.Н. Поздняков, Н.А. Резник. –СПб.: Свет, 1997.
4. Самахова, А.А. Европейский образовательный проект по динамической геометрии ИнтерГео [Текст]/А.А. Самахова//Компьютерные инструменты в школе. –2009.–№ 4.–С. 10–19.
5. Российская страница международного математического конкурса «Кенгуру» [Электронный ресурс]/<http://www.kenguru.sp.ru/>
6. Pozdnyakov, S. Challenging mathematics beyond the classroom enhanced by technological environments [Текст]/S. Pozdnyakov, V. Freiman, D. Kadjevich [et al.]//Challenging Mathematics In and Beyond the Classroom.–Springer Science–Business Media, LLC, 2009.–Ch. 3.–P. 97–131.
7. Северо-Восточный Европейский полуфинальный регион Чемпионата мира по программированию среди студентов ACM ICPC [Электронный ресурс]/<http://neerc.ifmo.ru/information/index.html>
8. Манцеров, Д.И. Среда Verifier-KD: верификация решений задач по математике [Текст]/Д.И. Манцеров//Компьютерные инструменты в образовании.–2006.–№ 4.–С. 36–41.
9. Посов, И.А. Разбор задач «Задача отшельников» и «Математическое скалолазание» конкурса КИО-2009 [Текст] / И.А. Посов // Компьютерные инструменты в школе.–2009.–№ 4.–С. 20–27.

УДК 519.873

М.Ф. Моледу

РАСПРЕДЕЛЕННОЕ МНОГОКРИТЕРИАЛЬНОЕ УПРАВЛЕНИЕ ДЛЯ ВЕТРОПАРКОВ

Современные тенденции развития технологических производств, объектов промышленного назначения требуют создания новых технологий управления большими распределенными объектами и процессами. Например, задачи оперативного управления технологическими комплексами, распределенными энергосистемами и электросетями.

Традиционным подходом к управлению большими системами являлись системы с централизованной структурой управления, особенность которых – алгоритм жесткой централизации, предполагающий фиксированное закрепление алгоритмов управления за отдельными подсистемами, что приводит в некоторых ситуациях к возникновению проблем, связанных с неэффективностью управления. Это происходит по следующим причинам: сложность модели объекта и

высокая размерность, труднопредсказуемые ситуации, большие потоки информации, подлежащие обработке, задержки в оперативности обработки поступающей информации и принятии решения в распределенных объектах, многоуровневая иерархия целей. Анализ перечисленных проблем показал, что одной из перспективных стратегий в решении подобной задачи является децентрализованное управление на основе использования агентного подхода и принципа распределения подзадач между отдельными подсистемами [6].

Ветропарки – системы производства энергии, являющиеся автономными и управляемыми модулями производства. В текущем технологическом состоянии каждый конвертер-турбина-генератор следует рассматривать как модуль. Этот модуль в настоящее время работает как отдельная си-



стема. По этой причине каждый ветрогенератор может быть представлен как агент в распределенных интеллектуальных системах. Агент – интеллектуальный объект, который может осознавать окружающую среду, наблюдать за ее состоянием при помощи датчика и предпринимать некоторые действия, чтобы изменить состояние среды. Ветрогенераторы расположены в ветропарке. Таким образом, целью работы является повышение эффективности группового поведения в команде на основе оптимальных стратегий коллективного планирования действий для ветропарка в целом. Распределенная интеллектуальная система – вычислительная система, где несколько автономных или полуавтономных агентов взаимодействуют и кооперируются или конкурируют, чтобы выполнить некоторый набор задач или удовлетворить некоторый набор целей.

Технология распределенной интеллектуальной системы в настоящее время еще не применялась в управлении системами энергетики. Однако у этого подхода есть большой потенциал к управлению крупно- и среднемасштабными возобновляемыми источниками энергии, распределенными источниками энергии (DER) и гибкой гибридной интеграцией в будущих автоматических системах. По крайней мере, два главных европейских проекта R&D (МикроСеть [2] и CRISP [3]) исследовали такой потенциал. В данной статье рассматривается применение распределенных интеллектуальных систем в ветропарках.

Формулировка задачи коллективного поведения ветрогенераторов

При решении задачи коллективного управления динамическими объектами, заключающейся в планировании групповых действий, предполагалось, что динамические объекты группы могут общаться друг с другом с помощью канала обмена информацией. Именно благодаря наличию такого обмена динамические объекты могут оптимизировать свои действия.

В этом плане наиболее «живучими» системами группового управления являются системы, использующие стратегию так называемого «стайного управления» [7].

В системах управления на основе стайной стратегии управления динамическими объектами, входящими в группу, источником информации

о действиях других динамических объектов группы являются среда E , в которой они функционируют, и обмен информации между ними. Фиксируя с помощью своих сенсоров изменения состояния среды, динамические объекты, в принципе, могут согласовывать свои действия с целью оптимизации процесса решения задачи, поставленной перед группой динамических объектов.

Таким образом, стайное управление в группе динамических объектов может быть реализовано на основе следующих принципов. Каждый динамический объект располагает информацией о цели, поставленной перед группой, адаптивной моделью индивидуального влияния на среду, а также получает информацию о состоянии среды в результате измерений.

Задача стайного управления группой динамических объектов формируется аналогично общему случаю задачи управления группой динамических объектов. Состояние каждого динамического объекта R_j группы описывается вектором $R_j(t) = [r_{j,1}, r_{j,2}, \dots, r_{j,h}]^T, j = \overline{1, N}$, а состояние среды – вектором $E(t) = [e_1, e_2, \dots, e_w]^T$. Каждый динамический объект $R_j, j = \overline{1, N}$, может выполнять некоторые свои действия, описываемые вектором $A_j(t) = [a_{j,1}, a_{j,2}, \dots, a_{j,u}]^T, j = \overline{1, N}$, с помощью которых он может изменять как состояние среды, так и состояния других динамических объектов группы.

Как показано выше, эти изменения могут быть описаны системами уравнений вида:

$$\frac{dR_j}{dt} = F_j(R_1, A_1, \dots, R_N, A_N, E, t), \quad j = \overline{1, N},$$

$$\frac{dE}{dt} = F(E, R_1, A_1, \dots, R_N, A_N, E, t).$$

На состояния динамических объектов и среды, а также на действия динамических объектов в конкретных ситуациях могут налагаться некоторые ограничения, в общем случае определяемые системами неравенств:

$$\Phi(R_1, R_2, \dots, R_N, E) \leq 0, \quad \Gamma(R_1, A_1, \dots, R_N, A_N, E) \leq 0,$$

которым должны удовлетворять допустимые состояния динамических объектов группы и их действия. Целью действий группы динамических объектов является преобразование среды из исходного (текущего) состояния E^0 в некоторое целевое состояние E^k оптимальным образом.

Задачей синтеза стайного управления является разработка алгоритма выбора каждым динамическим объектом своих действий для достижения цели, поставленной перед группой, в условиях неопределенности. В то же время он располагает информацией об индивидуальных возможностях по трансформации состояния S_i и среды E за счет индивидуальных действий.

Действия задачи коллективного управления в группе динамических объектов должны быть такими, чтобы суммарные затраты группы динамических объектов на достижение цели минимизировались, либо достигли максимальной выгоды. Под действиями, в частности, могут пониматься некоторые подзадачи, на которые декомпозируется цель, общая для группы динамических объектов. Эти подзадачи могут формулироваться по-разному. Наиболее часто под действием понимается достижение некоторой промежуточной цели (подцели), заданной координатами. Поэтому под оптимизацией коллективных действий в большинстве случаев можно понимать оптимальное или близкое к оптимальному распределение целей или подцелей между динамическими объектами группы. Для этого предполагается, что «эффект» от достижения динамическим объектом R_j цели $X_i \in \{X\}$ определяется значением некоторой оценки эффективности:

$$d_{ji} = F(s_j, x_p, E).$$

Обычно значение оценки эффективности d_{ji} определяется некоторыми целевыми функциями:

$$d_{ji} = [f_1^{ji}(s_j, x_p, E), f_2^{ji}(s_j, x_p, E), \dots, f_m^{ji}(s_j, x_p, E)]^T,$$

где $[f_1^{ji}, f_2^{ji}, \dots, f_m^{ji}]^T$ – функции, зависящие от параметров достижения динамическим объектом R_j цели $X_i \in \{X\}$.

Таким образом, d_{ji} определяется Парето-оптимальным множеством. Традиционным подходом в решении задач многоцелевой оптимизации коллективного поведения роботов является оптимизация по доминирующему критерию, в то время как все остальные критерии, кроме выбранного главным, рассматриваются в качестве ограничения или с помощью коэффициентов. Однако такой подход к решению подобных задач значительно снижает эффективность принимаемых решений. Другим широко используемым в практике подходом в решении задач рассматриваемого класса являются генетические алгоритмы ГА, которые представляют собой адаптивные мето-

ды поиска и наиболее часто употребляются для решения задач функциональной оптимизации. Их характерная особенность, позволяющая находить эффективные решения в задачах многокритериальной оптимизации – применение эволюционного подхода к поиску оптимального решения на основе математических моделей кодирования и модификации наилучших вариантов, отбираемых в процессе решения [5].

Распределенная интеллектуальная система

Автором развивается концепция децентрализованного управления распределенными системами как сложная стратегическая задача управления, решаемая путем применения принципа декомпозиции в распределении функциональности подсистем. Решение основано на разбиении стратегической цели управления на подзадачи. Начальным этапом развиваемого решения является формализация стратегической цели, определяемой с учетом параметров окружающей среды и условий эксперимента.

Реализация подобной стратегии базируется на использовании принципа распределения функциональности, предполагающего разделение целевых функций управления с учетом системного подхода в единой стратегии управления большими системами.

Особенностью развиваемого подхода является определение условий наилучшего взаимодействия агентов для достижения стратегической цели на основе многоцелевой функции методом многокритериальной оптимизации. Научная проблема формулируется как разработка методов и алгоритмов оптимального командного поведения (взаимодействия) рациональных интеллектуальных агентов в многоагентной среде для достижения общей стратегической цели [6].

Распределенная интеллектуальная система представлена на рис. 1. Ее главная задача состоит в оптимизации мощности, полученной от ветра, и сбалансировании полученной мощности и мощности, которую используют для переориентации каждого ветрогенератора. Задача оптимизации представляется следующим образом.

Вектор решений $X = [\alpha_1, \alpha_2, \dots, \alpha_n]^T \quad X \in \Omega$.

Целевые функции:

$$F(x) = [f_1(x), f_2(x), \dots, f_k(x)]^T;$$

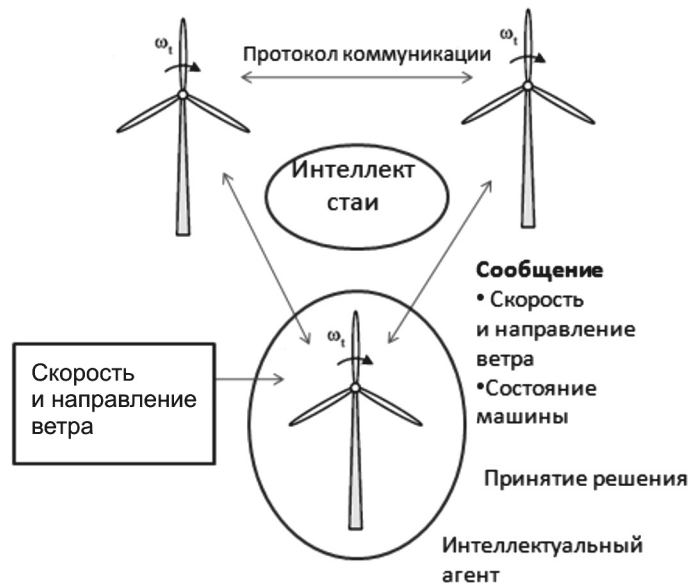


Рис. 1. Краткое описание распределенной интеллектуальной системы

$$f_1(x) = \sum_{i=0}^n P \max_i - \sum_{i=0}^n K \left(\tanh \left(\frac{v_i \cos(\alpha_i - \theta_i)}{6} - 2 \right) + 1 \right);$$

$$f_2(x) = \sum_{i=0}^n T \frac{(\alpha_i^{t-1} - \alpha_i)^2}{\Delta t^2}.$$

Ограничения

$$g(X) = \forall i \in \{1, \dots, n\}, |\alpha_{opti} - \alpha_i| - \pi/24 \leq 0. \quad (2)$$

Генеральная многоцелевая задача определяется как минимизация (или максимизация) $F(x) = (f_1(x), f_2(x), \dots, f_k(x))$ с учетом $g_i(x) \leq 0$, $i = \{1, \dots, m\}$ и $h_j(x) = 0$, $j = \{1, \dots, p\}$ $x \in \Omega$. Решение МЭ минимизирует (или максимизирует) компоненты вектора $F(x)$, где X представляет собой N -мерный вектор решения $X = (x_1, \dots, x_n)$ от некоторого множества решения Ω . Следует отметить, что $g_i(x) \leq 0$ представляет собой ограничения, которые должны быть выполнены при сведении к минимуму (или максимуму) $F(x)$, и Ω содержит все возможные решения x , которые могут быть использованы для удовлетворения оценки $F(x)$ [4].

Для описания целевых функций, необходимо понять цели оптимального поведения распределенной системы управления. Первая цель $f_1(x)$ – найти оптимальное равновесие между энергией, отправляемой в сеть ветропарком, и энергией, используемой для исправления ошибки переориентации. Другими словами, следует переориентировать корпус и лезвия турбины, чтобы получить максимум энергии (1) с учетом механических

ограничений $g(x)$ (2). Вторая цель $f_2(x)$ – добраться до правильного угла переориентации прежде, чем поток ветра поступит в соответствующую турбину с наименьшей затратой энергии с учетом механических ограничений $g(x)$. Для этого нужно иметь информацию об окружающей среде $E(t) = \langle e_1, e_2, \dots, e_w \rangle$ других агентов, чтобы получить правильный угол переориентации прежде, чем его достигнет поток ветра.

Другая важная цель оптимизации состоит в моделировании стратегии переориентации, которая позволяет держать гондолы в направлении ветра. С помощью распределенной системы можно избежать переориентации в случаях местной турбулентности, поскольку агенты (ветрогенераторы) имеют более широкое представление среды, т. к. каждый участник сообщает другим о состоянии окружающей среды $E(t) = \langle e_1, e_2, \dots, e_w \rangle$.

Чтобы избежать использования ненужной информации (информация о ветре, который уже прошел через текущую турбину) другими агентами, применяются критерии, описанные в следующем выражении:

$$\theta_j + \frac{\pi}{2} < \tan^{-1} \frac{y - y_j}{x - x_j} \geq \theta_j - \frac{\pi}{2},$$

где x, y – позиции текущего агента; x_j, y_j – позиции других агентов; θ_j – направление ветра, которое распознают другие агенты.

Используя эти данные, вычисляется средний угол ветра в ветропарке в зависимости от силы и

направления ветра в каждом верогенераторе. Так можно рассчитать оптимальную угловую скорость переориентирования каждого ветрогенератора. Для того чтобы найти оптимальное решение многокритериальной проблемы (1) и (2) был использован эволюционный алгоритм многоцелевой оптимизации [5]. Сущность предлагаемого алгоритма сводится к следующему. Приближение к множеству Парето происходит с помощью ранжирования по Парето, которое делит популяцию на недоминируемые подмножества: во-первых, все недоминируемые индивиды популяции помечены как первый ранг, а, во-вторых, они временно удалены от популяции, и процесс повторяется – остальные недоминируемые индивиды популяции получают второй ранг и т. д., пока вся популяция ранжируется. Методика сохранения разнообразия основана на расстоянии объединения одной из возможных оценок плотности пространства решений недоминируемых подмножеств. Отбор эволюционного алгоритма основан на турнирах, т. е. нужно выбирать индивидов для спаривания. T -индивиды случайно выбираются из популяции, и далее сравниваются друг с другом с помощью оператора сравнения, определенного

выше. Победителем становятся родители. Замена эволюционного алгоритма детерминированная, состоит в объединении всех родителей и потомства подмножеств и выборе лучших N -индивидов в этой глобальной популяции с помощью того же оператора сравнения. Селекция реализует отбор N наилучших особей из объединенной популяции размером $2N$ посредством сортировки всех особей в этой популяции по недоминированию и последующему выбору N особей с наивысшей приспособленностью [5].

Применение генетических операторов к хромосомам, отобранным с помощью селекции, приводит к формированию популяции потомков от созданного на предыдущем шаге родительского пула. Процесс повторяется до достижения критерия остановки [5].

Если многокритериальная задача (1) и (2) решается каждым ветрогенератором централизованным образом, то требуется огромное количество вычислительных операций; многие из них повторяются в ветропарке в целом. Используя данные Парето-фронт, представленные на рис. 2, можно ввести ПМЭ (параллельный многоцелевой эволюционный алгоритм) с архитектурой master-

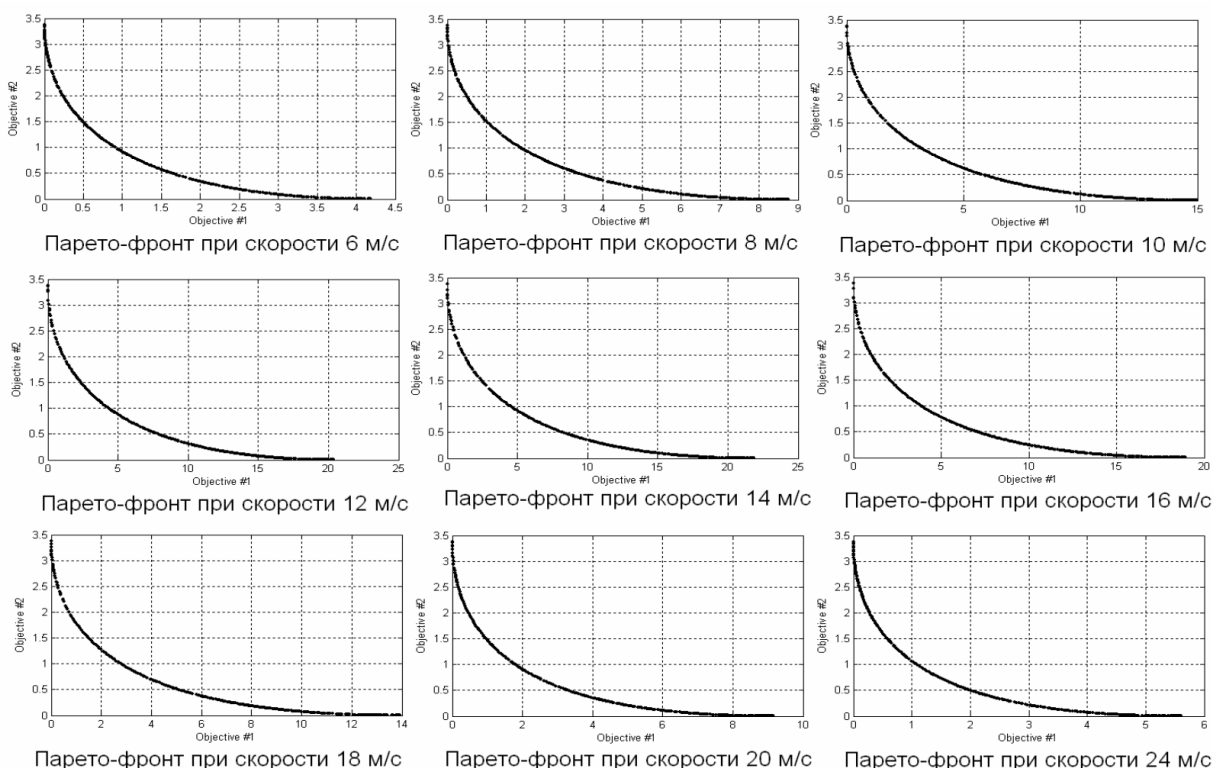


Рис. 2. Парето-фронты с ошибками переориентации 30° и масштаб в кВт



slave, где каждый динамический объект может найти Парето-фронт для выполнения своей задачи и отправить информацию в другие динамические объекты. С помощью этой информации каждый динамический объект может найти глобальный Парето-фронт для задачи с меньшими вычислительными усилиями, используя информацию из локальных Парето-фронт, которые уже вычислены другими агентами [5]. Локальная задача определена следующим образом.

Вектор решений $X = [\alpha_i]$ $X \in \Omega$.

Целевые функции:

$$F(x) = [f_1(x), f_2(x)]^T;$$

$$f_1(x) = P \max_i - K \left(\tanh \left(\frac{v_i \cos(\alpha_i - \theta_i)}{6} - 2 \right) + 1 \right);$$

$$f_2(x) = T \frac{(\alpha_i'^{-1} - \alpha_i)^2}{\Delta t^2}.$$

Ограничения $g(X) = |\alpha_{opti} - \alpha_i| - \pi/24 \leq 0$.

Для всех тестовых задач эволюционного алгоритма на различных скоростях ветра была использована популяция числом 100, вероятность кроссовера 0,8, вероятность мутации $1/N$ (где N – число переменных). Проводился эволюционный алгоритм для 150 поколений. Были использованы переменные реальные цифры, моделируемый бинарный кроссовер (SBX) и полиномиальный оператор мутации.

Моделирование ветропарка

Моделирование скорости ветра очень важно, т. к. диктует действия генераторов ветра и определяет особенности ветропарка для предсказания вывода энергии, анализа преобразования энергии и системной динамики.

Моделирование базируется в параметрах ветропарка [1] и развито в C++ Borland Builder.

На рис. 3 показано моделирование целого ветропарка. Белые точки – ветрогенераторы; черные стрелки – лезвия; другие стрелки показывают скорость потока ветра, для этого используется цветной масштаб.

Платформа изменяет направление ветра каждые 30° от 360° горизонта, скорость от 5 до 20 м/с и продолжительность ветра на определенном направлении и скорости.

Представлен интерфейс, состоящий из трех графиков сравнения моделей. Первый показывает, как ветропарк с централизованным управлением исправляет ошибку переориентации в то время, когда ветрогенераторы распознают изменение направления ветра. Распределенное интеллектуальное управление для оптимального равновесия энергии начинает процесс переориентации намного раньше. На первом графике представлена генерирующая мощность целого ветропарка; на втором – показана текущая мощность, используемая для переориентации каждого ветрогене-

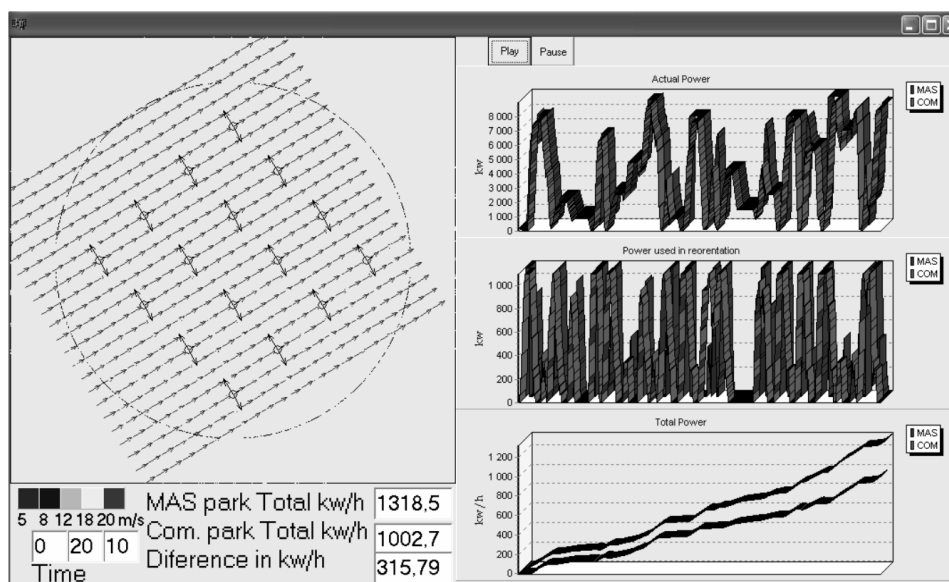


Рис. 3. Моделирование ветропарка и графики в C++ Borland

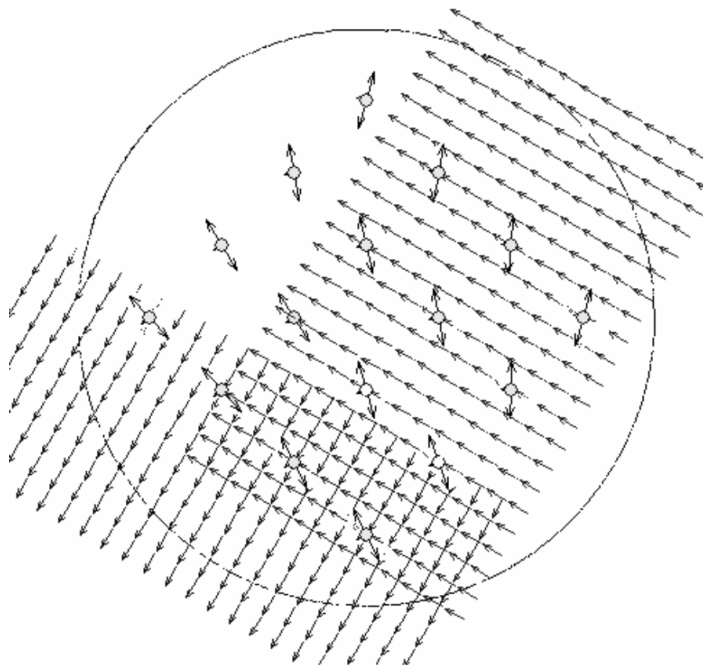


Рис. 4. Моделирование ветропарка и графики в C++ Borland

ратора к правильному направлению; на третьем графике видна глобальная сумма полученной мощности за минусом мощности, использованной для переориентации (см. рис. 3).

Экспериментальные результаты

Для анализа результатов необходимо рассматривать модель оптимального поведения распределенной системы. Одна из основных целей оптимизации заключается в достижении правильной ориентации, с использованием данных, полученных другими агентами. На рис. 4 показано, как после изменения направления ветра агент наконец начинает переориентацию еще до распознавания нового направления ветра.

Другая цель – получение максимальной энергии ветра и нахождение оптимального баланса между энергией, направленной в сетку от ветрового парка, и энергией, используемой для исправления ошибки ориентации. Для анализа этой цели нужно применять точку сравнения, поэтому графики также свидетельствуют о том, как работает централизованное управление переориентации. Чтобы получить эти графики, используются выражения (3) и (4):

$$Pp = \sum_{j=0}^n 300 \left(\tanh \left(\frac{v_j \cos(\alpha_j - \theta_j)}{6} - 2 \right) + 1 \right); \quad (3)$$

$$Pa = \sum_{j=0}^n T \frac{\Delta \beta_j^2}{\Delta t^2}. \quad (4)$$

На первом графике (рис. 5) видно, что распределенная система получает больше энергии. На втором графике показано, как распределенная система использует больше энергии в самом начале, чем централизованное управление. Это означает, что в данной ситуации необходимо было раньше исправить ошибку переориентации для получения максимальной энергии. На последнем графике представлены энергии в кВт/ч, выработанные системами.

На рисунке представлено сравнение поведения распределенной и централизованной систем управления. График показывает сумму энергии в кВт/ч в эквиваленте 1200 циклов, в высокопеременном окружении. Результатами являются 1334 кВт/ч парка распределенного управления и 1050 кВт/ч, полученные централизованным управлением. Разница между этими энергиями составляет 284 кВт/ч. Это означает, что распределенное управление на 20 % эффективнее централизованного. Фактически в этом ветропарке работают шестнадцать турбин; эффективность распределенного управления дает результаты, равные работе девятнадцати ветрогенераторов.

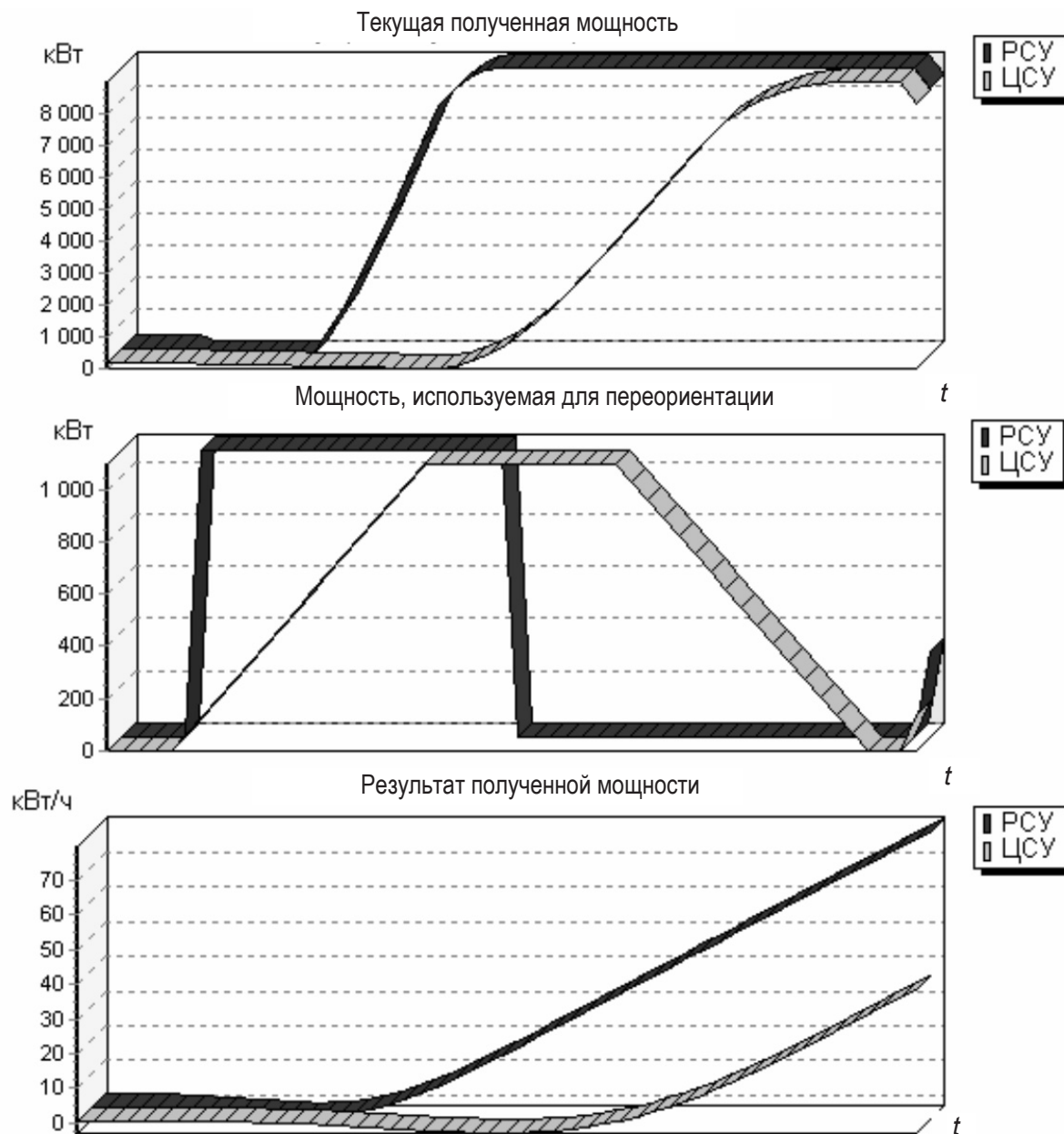


Рис. 5. Сравнение мощности

Предложенный параллельный эволюционный алгоритм многокритериальной оптимизации, использующий знания других динамических объектов о решаемой задаче, сокращает время поиска оптимального решения задачи целераспределения в коллективе динамических объектов. При этом даже при определенном увеличении размерности коллектива динамических объектов применение генетических алгоритмов позволяет по-прежнему

получать решения за интервал времени, приемлемый для данной модели ветропарка.

Следует отметить, что предложенный алгоритм многоцелевой оптимизации повышает эффективность коллективного поведения взаимодействующих роботов за счет более эффективного распределения целей путем организации итерационной процедуры оптимизации коллективного решения.

СПИСОК ЛИТЕРАТУРЫ

1. **Daniels, L.-K.** Harvest the Wind a Wind Energy Handbook for Illinois Handbook style [Текст]/ L.-K. Daniels, S.-E. Johnson, W. Slaymaker//Western Illinois University: Windindustry.–2004.
2. **Oyarzabal, J.** Agent based Micro Grid Management system [Текст]/J. Oyarzabal, J. Jimero, A. Engler [et al.]// International conf. in future power systems.–Amsterdam, Nov. 2005.
3. **Schaeffer, G.J.** Field Tests applying Multi-Agent Technology for distributed control: Virtual Power Plants and Wind Energy [Текст]/ J Schaeffer, C.J. Warmer, I.G. Kamphuis [et al.]//Presented at CRIS Workshop 2006, Influence of Distributed and Renewable Generation on Power System Security, Magdeburg, Germany. – Dec. 6-8. 2006.
4. **Coello, C.A.** Evolutionary Algorithms for Solving Multi-Objective Problems [Текст]/ C.A. Coello, G.B. Lamont, D.A. Van Veldhuizen. –NY: Springer-Verlag, 2007. –Ch. 1.
5. **Моледу, М.Ф.** Применение генетических алгоритмов в задачах многоцелевой оптимизации коллективного поведения роботов [Текст]/ М.Ф. Моледу, В.П. Шкодырев //Научно-технические ведомости СПбГПУ. –2009.–№ 4.–С. 157–164.
6. **Арсеньев, Д.Г.** Интеллектуальные компьютерные сети – применение в задачах децентрализованного управления распределенными объектами и технологическими комплексами [Текст]/Д.Г. Арсеньев, В.П. Шкодырев//В сб. докл. Межд. науч-техн. конф. «Актуальные проблемы информационно-компьютерных технологий, мехатроники и робототехники».–2009.
7. **Каляев, И.А.** Интеллектуальные роботы [Текст]/ И.А. Каляев, В.М. Лохин, И.М. Макаров [и др.].–М.: Машиностроение, 2007.



Вычислительные машины и программное обеспечение

УДК 004.432, 004.052.42

В.М. Ицыксон, М.И. Глухих

ЯЗЫК СПЕЦИФИКАЦИЙ ПОВЕДЕНИЯ ПРОГРАММНЫХ КОМПОНЕНТОВ

Вопросы проектирования программных систем становятся все более актуальными в современном мире. Так как сложность программного кода постоянно возрастает, качество программных систем должно соответствовать требованиям. Одной из ключевых характеристик качества программного обеспечения (ПО) является надежность.

Немаловажную роль в повышении надежности ПО играет своевременное обнаружение программных ошибок (дефектов). Особенно важна задача обнаружения дефектов в программах на языках С и С++ в связи с большим объемом существующего С/С++ кода и особенностями языковых средств: отсутствием контроля программных ошибок, наличием операций с указателями, макроопределений и т. п. При большом объеме исходного кода проекта поиск дефектов путем тестирования может быть трудоемкой задачей. Кроме того, тестирование не гарантирует обнаружение всех дефектов в программе. Поэтому все большее распространение получают автоматизированные методы поиска дефектов, основанные на статическом анализе исходного кода программы. Эти методы используют в качестве входных данных только исходные тексты анализируемого проекта, не требуют запуска программы и при этом гарантируют анализ всех возможных трасс выполнения.

Статический анализ предполагает проверку всего исходного кода программного проекта, который может состоять из множества модулей. При этом часто возникает ситуация, когда поведение исследуемой программы определяется не только ее исходным текстом, но и внешними компонентами. Такими компонентами могут быть стандартные библиотеки (например, `libc` или `pthread`) или используемые в составе проекта проприетарные модули. Сложность анализа в таком случае за-

ключается в отсутствии части исходных текстов проекта, что не позволяет полностью провести построение модели программы [1] и полноценно проанализировать программный проект.

Решением проблемы может служить использование механизмов задания семантики компонента с отсутствующим исходным кодом. Задача полностью специфицировать поведение внешнего компонента не ставится, достаточно описать аспекты поведения, влияющие на анализируемый проект. Такого рода описания будем называть «частичными спецификациями компонентов» или «аннотациями». Подобные спецификации позволяют задавать для системы анализа поведение компонента с требуемой степенью детализации.

С точки зрения программы внешний компонент (например, функция) выглядит как черный ящик, связанный с внешним миром при помощи аргументов и возвращаемого значения. Данные для функции передаются в качестве параметров, а видимое действие функции может заключаться в формировании возвращаемого значения и модификации объектов программы, адреса которых переданы в функцию через переменные-указатели. При подобной модификации в языке С часто используются копирование и заполнение участков памяти, а также строковые операции.

Кроме указанного влияния функция может иметь побочные эффекты: изменение значений глобальных переменных и создание объектов окружения (например, операционной системы), имеющих свой жизненный цикл. К таким объектам относятся:

- выделенные блоки в динамической памяти;
- файлы и файловые дескрипторы;
- сокеты (socket);
- потoki ввода-вывода (stream);
- семафоры;
- критические секции;

мьютексы
и т. п.

Будем в дальнейшем называть такие объекты «ресурсами». Каждый ресурс обладает своим жизненным циклом, специфичным для каждого типа ресурсов. За время жизни ресурс меняет свое состояние в соответствии с определенными правилами. Причиной смены состояния может служить вызов функции API или событие операционной системы. Ресурс может иметь один или несколько атрибутов. Примером атрибута может являться размер выделенного участка динамической памяти. Атрибуты также могут изменяться в течение жизни ресурса. Основные операции с ресурсами: создание ресурса; освобождение; изменение состояния и атрибутов ресурса.

Особенность статического анализа программ, оперирующих с ресурсами, – необходимость учета текущего состояния ресурса и значений его атрибутов при обнаружении дефектов в отдельных конструкциях программы.

Рассмотрим существующие подходы к описанию поведения компонентов с отсутствующим исходным кодом.

Подходы к заданию спецификаций поведения компонентов. В настоящее время используются три подхода к описанию поведения компонентов:

- расширение целевого языка (C или C++) с помощью специализированных аннотирующих комментариев;

- расширение целевого языка с использованием его внутренних средств;

- разработка внешних средств для описания поведения компонентов.

Расширение аннотирующими комментариями. Данный подход заключается во встраивании специальным образом оформленных комментариев в требуемые места программы. При этом такой комментарий обычно аннотирует именно ту конструкцию языка, рядом с которой он расположен, т. е. описание семантики привязано к контексту.

Яркими представителями этого подхода являются язык ACSL, используемый средством Frama-C, и язык аннотирующих комментариев, используемый средством Splint.

Язык ACSL (ANSI/ISO C Specification Language) [2] позволяет задать дополнительную информацию о поведении функций и конструкций, используемую при анализе программы. Ин-

формация представляется комментариями особого вида (`/*@ ... */` или `//@ ...`).

Язык ACSL позволяет программисту описывать контракты, которые должны быть выполнены функцией. Контракт задается описанием предусловий, постусловий и поведения функции. Предусловия (*requires*) должны выполняться непосредственно до вызова функции, постусловия (*ensures*) должны гарантированно выполняться после вызова функции. В языке имеются развитые средства задания сложного поведения и описания изменяемых функцией переменных (*behavior*, *assumes*, *assigns*). ACSL также позволяет задавать контракт для отдельных конструкций языка.

Специальная группа средств предназначена для описания инвариантов поведения, т. е. условий, которые должны всегда выполняться в определенной части программы. К ним относятся:

- инварианты условий (*assert*) – справедливы в конкретной точке программы;

- инварианты циклов (*invariant*) – справедливы внутри определенного цикла;

- глобальные инварианты (*global*) – справедливы на всем протяжении программы.

Язык аннотирующих комментариев, используемый средством Splint [3], является менее функциональным. Он также использует комментарии особого рода `/*@ ... @*/`, с помощью которых можно задавать следующие свойства:

- способность функции влиять на параметры и глобальные переменные;

- вид возвращаемого указателя (указатель на вновь выделенный объект, на статическое хранилище и т. п.);

- ограничения на значения параметра–указателя функции (например, может ли параметр–указатель функции равняться `null`);

- направление передачи информации через параметр–указатель функции (т. е. является ли параметр–указатель функции входным или выходным параметром).

Интеграция в целевой язык – это одновременно и достоинство, и недостаток данного подхода. С одной стороны, интеграция позволяет привязать аннотирующие комментарии к определенным конструкциям программы, что облегчает интерпретацию этих комментариев и позволяет в удобной форме задать свойства любых объектов и конструкций программы.

С другой стороны, интеграция требует моди-



фикации компилятора целевого языка, что может являться непростой задачей (крайне затрудняется использование для компиляции готовых средств). Кроме этого, при любом изменении аннотаций происходит модификация исходного кода и, как следствие, его перекомпиляция. Это требует дополнительных затрат времени, особенно, если анализируемая программа имеет большие размеры.

Существует проблема аннотирования сторонних проектов, развивающихся независимо. Внешние аннотации будут полностью потеряны при выпуске разработчиком новой версии, что потребует повторного внедрения аннотаций в исходный код.

Расширение внутренними средствами языка. Данный подход основывается на использовании специализированных средств и расширений языка, а также команд препроцессора. Обычно в качестве таких средств используются спецификаторы типов или атрибуты. Как правило, эти дополнения не влияют на работу программы, однако могут использоваться компилятором для проведения дополнительных оптимизаций, а средствами анализа – для получения дополнительной информации об объектах программы.

Одним из представителей этого подхода является пакет CLANG [4]. Данный пакет использует атрибуты компилятора GCC `__attribute__`, вызовы `assert`, а также настройки `#pragma` для расширения программы дополнительной семантикой. С помощью перечисленных средств можно, например, указать, является ли функция завершающей в программе, задать дополнительную информацию о значениях параметров функции или помечать вызовы `assert` гарантированно выполняющиеся условия.

Аналогичный подход используется в среде Microsoft Visual Studio (начиная с версии 2005). Язык аннотаций SAL [5] (Source Annotation Language), поддерживаемый средой, обладает следующими возможностями:

для параметра–буфера можно задать вариант использования (только чтение, только запись, чтение/запись), а также размер буфера;

для параметра–указателя или возвращаемого значения–указателя можно задать признак «указывает на динамическую память, созданную в функции»;

для возвращаемого значения можно задать признак «не может быть игнорировано»;

для указателя на строку можно задать признак

«является форматной строкой»;

для точек выхода из функции можно задать признак «завершилась с ошибкой».

Похожий подход используется в средстве PC Lint [6], где вызовами `assert` задаются гарантированно выполняющиеся условия.

Основное достоинство подхода – простота реализации. Все перечисленные атрибуты и настройки уже встроены в компилятор, и нет необходимости в его модификации. С другой стороны, возможности данного подхода сильно ограничены; с его помощью невозможно описывать поведение сложных функций.

Внешние средства для описания семантики. Основное отличие данного подхода заключается в независимом описании поведенческих свойств компонентов программы без внесения изменений в исходный код. Для этого обычно разрабатывается специализированный язык описания семантики программы, не связанный непосредственно с целевым языком и не ограниченный его описательными средствами.

Среди специализированных языков можно отметить язык Metal, используемый системой статического анализа хгсс, и язык CASL, используемый для проведения дедуктивной верификации.

Язык Metal [7] предназначен для создания расширений компиляции, так называемых «checker-ов». При их написании в программе выделяется один или несколько конечных автоматов, каждый из которых привязывается к одной или нескольким переменным, существующим в программе. Например, указателю может соответствовать автомат с состояниями «указывает на освобожденную память», «указывает на выделенную память». Дополнительно могут выделяться глобальные автоматы, состояние которых не связано с конкретными переменными.

Для каждого автомата определяются правила переходов из состояния в состояние, а также дефекты, связанные с действиями в данном состоянии. При этом используются ссылки на конкретные функции или операции. В листинге 1 приведен пример описания на языке Metal дефектов «разыменование освобожденного указателя» и «двойное освобождение указателя».

CASL (Common Algebraic Specification Language) [8] – язык алгебраических спецификаций, основанный на логике первого порядка. Основные его возможности связаны с описанием свойств отношений, типов и операций над ними,

```
state decl any_pointer v;
start: { kfree(v) } ==> v.freed;
v.freed: { *v } ==> v.stop,
{ err(«Использование %s после освобождения!», mc_identifier(v)); }
| { kfree(v) } ==> v.stop,
{ err(«Двойное освобождение %s!», mc_identifier(v)); };
```

Листинг 1: Пример описания на языке Metal

функций. Синтаксис языка использует форму Бэкуса–Наура, логические операции и операции над множествами, предикаты. С помощью спецификаций на этом языке можно задавать и проверять определенные утверждения, касающиеся анализируемой программы. Язык используется в некоторых средствах дедуктивной верификации.

К классу средств, использующих специализированные языки, относится также средство Fortify [9]. В нем применяются XML-описания, предназначенные для задания новых правил поиска дефектов.

Специализированные языки потенциально могут иметь широкий спектр возможностей. Аннотации на таких языках обычно не встраиваются в программу и поэтому не требуют модификации компилятора, а задача разбора аннотаций обычно решается достаточно просто, т. к. сложность этих языков значительно ниже сложности целевого языка. С другой стороны, привязка таких аннотаций к конкретным объектам программы часто оказывается затруднительной.

Резюмируя обзор существующих подходов к построению аннотаций, следует отметить, что специализированные языки обеспечивают наибольшую гибкость аннотаций, при этом не требуя сложных операций с компилятором целевого языка. Из рассмотренных языков для составления спецификаций функций наиболее подходит язык Metal. Однако действия функции часто не ограничиваются изменением состояния некоторого конечного автомата и формированием дефектов. Кроме этого, в языке Metal отсутствуют некоторые требуемые операции (например, заполнение и копирование участков памяти).

По указанным причинам было принято решение о создании собственного специализированного языка. Задача привязки аннотаций к конкретным объектам программы может быть решена, например, за счет сравнения сигнатур функций или прямого указания местонахождения объекта в исходном коде программы.

Требования к специализированному языку. Основной целью разрабатываемого язы-

ка является возможность описания поведения используемых в программе функций, исходный код которых отсутствует либо слишком сложен для анализа. Так как предполагается применение аннотированных функций в процессе анализа программ на языке C, то логичным решением является использование для аннотаций C-подобного императивного языка.

Определим требования, предъявляемые к языку аннотаций. Большинство требований – отражение функционального назначения языка. Структурно язык должен предоставлять возможность описания трех сущностей: ресурсов, функций и глобальных объектов.

Для ресурсов необходимо предусмотреть задание:

- имени ресурса;
- перечня возможных состояний;
- типа ресурса;
- атрибутов ресурса.

Для описания функции необходимо указать ее сигнатуру и представить ее поведение. Для описания поведения функции должны применяться конструкции управления ходом вычислительного процесса и операции, манипулирующие с объектами программы. При этом необходимо предусмотреть работу как с обычными, так и с интервальными значениями объектов. Должны быть реализованы следующие возможности:

- арифметические операции, операция присваивания;
- операции с участками памяти (заполнение, копирование);
- конструкция сигнализации о наличии дефекта;
- операции с работы ресурсами (создание, освобождение, смена состояния, изменение атрибутов).

Для описания глобальных переменных и ресурсов требуется механизм их декларирования и начальной инициализации.

Ограничения языка. Статический анализ является чрезвычайно ресурсоемкой операцией, требующей высокой вычислительной мощности и больших объемов памяти. Среди самых



затратных частей анализа следует отметить анализ вложенных циклов и рекурсивных функций (как известно, трудоемкость статического анализа растет экспоненциально с увеличением числа вложенных циклов). Одной из целей введения частичных семантических спецификаций компонентов является существенное снижение требуемой ресурсоемкости. Этого можно достигнуть с помощью следующих ограничений:

за счет запрета использования рекурсивных вызовов функций в теле поведенческих описаний;

за счет запрета использования любых конструкций, способных порождать в графе потока управления поведенческого описания циклы. К таким запрещенным операциям относятся конструкции циклов и переходов.

Указанные ограничения обеспечат простоту анализа поведенческих описаний компонентов, т.е. в графе потока управления всей программы не появится новых циклов и рекурсий. Несмотря на то, что данные ограничения снижают мощность языка, его изобразительных средств по-прежнему достаточно для решения поставленных задач.

Описание разработанного языка. В соот-

```
return [0:99];      // Возвращаемое значение функции может
                    // принимать любое значение от 0 до 99
errno = [-1:+inf]; // Переменная errno может принимать
                    // любое значение от -1 до максимального значения
```

Листинг 2: Пример использования интервальных значений

Язык поддерживает грамматику выражений. Помимо традиционных операций поддерживаются специфичные для ресурсного анализа операции (обращение к состоянию или атрибуту ресурса), а также операции для работы со строками.

Описание ресурсов. Секция описания ресурсов предназначена для декларирования типов ресурсов, использующихся в библиотеке. Для каждого типа ресурса задаются имя и перечень состояний, в которых ресурс может находиться во время выполнения программы. Переход ресурса из одного состояния в другое осуществляется либо путем вызова функций библиотеки, либо по событию операционной системы. Для любого типа ресурса можно задавать атрибуты (например, ресурс «динамическая память» характеризуется атрибутом «размер») и дополнительные свойства. К таким свойствам в первую очередь относится свойство `memory`, определяющее, что ресурс представляет собой какой-либо вид памяти (т.е. допускает работу с ним как с массивом).

ветствии с перечисленными выше требованиями и ограничениями был разработан язык аннотаций PanLang (program annotation language). В качестве основы синтаксиса используется синтаксис языка C. Для реализации специфичных для описания семантики возможностей язык был расширен дополнительными конструкциями.

Типы данных. Язык PanLang использует те же базовые типы данных, что и целевой язык C. Поддерживаются константы базовых типов. Основным отличием от стандартных языков программирования является возможность использования интервальных значений для переменных. Интервальные значения служат основой для применения алгоритмов интервального анализа при проведении статического анализа программы. При этом объектам программы могут присваиваться не только конкретные значения, но также и интервалы, которые описывают диапазон значений, который могут принимать объекты после вызова аннотируемой функции. Такими объектами могут быть возвращаемые значения функции, параметры функции, переданные через указатели, или глобальные переменные.

Пример описания ресурсов приведен в листинге 3. Следует отметить, что любой ресурс включает неявное состояние `Closed`; при переходе в данное состояние ресурс освобождается.

```
resource File
{
    Opened
}

resource Socket
{
    Created,
    Bound,
    Established,
    Listen
}

resource Heap(int size, Memory)
{
    Allocated
}
```

Листинг 3: Пример описания ресурсов

Следует отметить, что в некоторых операционных системах (например, в ОС семейства Linux) используется универсальный дескриптор ресурсов, который в зависимости от контекста может оперировать как файлами, так и сокетами. В таком случае ресурсы «файл» и «сокет» в секции описания могут быть объединены.

Описание поведения функций. Для каждой функции задается сигнатура, имя функции, при необходимости – дополнительные имена (псевдонимы) и описание поведения. Сигнатура включает в себя имена и типы аргументов, а также тип результата.

Секция поведения содержит последовательность операторов, описывающих поведение аннотируемой функции. Поддерживаются несколько групп операторов.

Управление вычислительным процессом. Для описания хода вычислительного процесса в языке предусмотрены следующие конструкции:

конструкция ветвления `if`, поддерживаются также недетерминированные условия;

оператор завершения программы `exit (status)`.

Модификация значений объектов. Используемый набор операторов аналогичен имеющемуся в языке C:

оператор определения переменной `type var = value` – создает новую переменную заданного типа, может задать также ее начальное значение;

оператор присваивания значения `var = value, *ptr = value`.

Работа с участками памяти. Так как, согласно требованиям, в языке PanLang отсутствуют циклы и рекурсии, в нем предусмотрены специальные групповые операции над участками памяти:

оператор копирования `copy(dest, source, length)` – осуществляет копирование одного участка памяти в другой;

оператор заполнения `set(dest, value, length)` – осуществляет заполнение участка памяти постоянными значениями.

Работа с ресурсами. Ресурсы являются специализированными объектами, действия с которыми выполняются через дескриптор. Дескрипторы ресурсов обладают всеми свойствами обычных переменных. Кроме этого, поддерживаются специальные операторы, работающие с дескрипторами ресурсов:

оператор выделения ресурса `descr = new`

`Resource(type, state, attr, value, canleak)`, где `descr` – дескриптор ресурса; `type` – тип ресурса; `state` – начальное состояние ресурса; `attr` – начальное значение атрибута (если атрибут отсутствует, этот аргумент опускается); `value` – ассоциированное с дескриптором целое значение; `canleak` – признак возможности утечки ресурса;

оператор освобождения ресурса `delete(type) descr` – уничтожает ресурс заданного типа, на который указывает переменная-дескриптор;

оператор установки атрибута ресурса `descr->attr = attrvalue` – задает новое значение атрибута ресурса, на который указывает переменная-дескриптор;

оператор изменения состояния ресурса `state(descr) = newstate` – задает новое состояние ресурса, на который указывает переменная-дескриптор.

Специальные операторы. Операторы данной группы ориентированы на конкретную задачу, решаемую системой статического анализа. При использовании языка PanLang для обнаружения дефектов в эту группу входит оператор сигнализации о дефекте `defect(type, description)`. Этот оператор указывает, что выполнение данного участка программы является программной ошибкой (дефектом), задает тип дефекта и его описание.

Пример описания поведения функции. В качестве примера использования языка приведена аннотация для функции `realloc` (листинг 4).

Описание глобальных объектов. Секция глобальных объектов предназначена для спецификации глобальных переменных, доступных из функций библиотеки, а также для описания создаваемых ресурсов при инициализации библиотеки. Такими ресурсами могут быть, например, стандартные потоки ввода-вывода (листинг 5).

Применение аннотирующих описаний. Наличие аннотирующих описаний функций позволяет повысить полноту и точность статического анализа. В частности, аннотации позволяют обнаруживать такие дефекты:

утечку динамической памяти и ресурсов;
двукратное освобождение динамической памяти и ресурсов;

нарушение протокола работы с ресурсами;
некорректные значения параметров функций.

В качестве простого примера можно рассмотреть фрагмент программы (листинг 6).

```
void* realloc(void* ptr, unsigned size)
{
    if (size <= 0) {
        delete(Heap) ptr;
        return 0;
    }
    else {
        void* res = new resource(Heap, Allocated, size, none, true);
        if (ptr->size <= size) {
            size = ptr->size;
        }
        copy(res, ptr, size);
        delete(Heap) ptr;
        return res;
    }
}
```

Листинг 4: Пример описания поведения функции

```
globals
{
    int stdin = new resource(File, Opened, 0, false);
    int stdout = new resource(File, Opened, 1, false);
    int stderr = new resource(File, Opened, 2, false);
    int errno = [0:+inf];
}
```

Листинг 5: Пример описания глобальных объектов

```
int s = socket(AF_INET, SOCK_STREAM, 0);
struct sockaddr_un addr;
memset(&addr, 0, sizeof(struct sockaddr_un));
addr.sun_family = AF_INET;
strncpy(addr.sun_path, «example.com», sizeof(addr.sun_path));
bind(s, (struct sockaddr*)&addr, sizeof(struct sockaddr_un));
listen(s, 256);
char buf[512];
// Дефект: получаем данные из несоединенного сокета
int res = recv(s, buf, 512, 0);
close(s);
```

Листинг 6: Пример аннотирующего описания

Для анализа этого фрагмента необходимо иметь аннотирующие описания функций `socket`, `memset`, `strncpy`, `bind`, `listen`, `recv` и `close`. Все они входят в стандартную библиотеку языка C. В листинге 7 приведен пример аннотации функции `recv`, внутри которой происходит обнаружение дефекта.

Вызов функции `socket` создает ресурс типа «сокеты», который последовательно проходит состояния `Created`, `Bound`, `Listen`. В аннотации функции `recv` указано, что если переданный

функции сокет находится в состоянии, отличном от `Established`, это приводит к дефекту протокола работы с ресурсом (получение данных из несоединенного сокета). Именно такая ситуация и диагностируется анализатором при использовании указанных аннотаций. Следует отметить, что данный дефект обнаруживается статически, без необходимости выполнения анализируемой программы.

В статье отражены основные результаты исследования в области построения частичных


```
long recv ( int s ,void* buf ,unsigned len ,int flags )
{
    if ( buf == 0 )
    {
        defect ( INI-03, «Второй аргумент функции - NULL»);
    }
    if ( flags < 0 )
    {
        defect ( INI-01, «Некорректное значение аргумента функции»);
    }
    if ( state(s) != Established )
    {
        defect ( RES-06, «Получение данных из несоединенного сокета»);
    }
    set ( buf, [-inf:+inf], len );
    return [0:len];
}
```

Листинг 7: Пример аннотации функции *recv*

спецификаций программных компонентов. Проведенный анализ существующих подходов к аннотированию поведения компонентов выявил их основные недостатки. Как результат, была обоснована необходимость разработки специализированного языка, позволяющего описывать поведение компонентов. На основе требований спроектирован язык частичных спецификаций, базирующийся на синтаксисе языка С.

С помощью разработанного языка создан набор аннотирующих описаний для стандартной библиотеки С (всего поддерживается более 500 функций). Разработанные описания успешно используются

в системе обнаружения дефектов в исходном коде на языке С [10] и позволяют анализировать сложные многокомпонентные проекты, использующие функции стандартной библиотеки.

Направления дальнейших исследований связаны с расширением мощности языковых описаний, разработкой частичных спецификаций других библиотек, а также в использовании аппарата частичных спецификаций для автоматизированной трансформации программных систем.

Работа выполнена в рамках госконтракта № П2226 ФЦП «Научные и научно-педагогические кадры инновационной России на 2009-2013 гг.».

СПИСОК ЛИТЕРАТУРЫ

1. Ицкисон, В.М. Исследование средств построения моделей исходного кода программ на языках С и С++ [Текст]/В.М. Ицкисон, М.И. Глухих, А.В. Зозуля [и др.]/Научно-технические ведомости СПбГПУ. Сер. Информатика. Телекоммуникации. Управление. –2009. –№ 1 (72). –С. 122–130.
2. Baudin, P. ACSL: ANSI/ISO C Specification Language. [Текст]/ P. Baudin, P. Cuoq [et al.]. –CEA LIST and INRIA, 2009.
3. [Электронный ресурс] <http://www.splint.org/manual/>
4. [Электронный ресурс] <http://clang-analyzer.lvm.org/annotations.html>
5. [Электронный ресурс] <http://msdn.microsoft.com/enus/library/ms235402.aspx>
6. [Электронный ресурс] <http://www.bezem.de/pdf/htwpl.pdf>
7. Hallem, S. A System and Language for Building System-specific, Static Analyses [Текст]/S. Hallem, B. Chelf [et al.]. –Stanford University, 2002.
8. Bidoit, M. CASL User Manual. Introduction to Using the Common Algebraic Specification Language [Текст]/M. Bidoit, P.D. Mosses.–University of Aarhus, BRICS and Department of Computer Science, 2004.
9. Fortify Audit Workbench User Guide. Version 5.2 [Текст]. –Aug. 2008.
10. [Электронный ресурс] <http://s2a.ftk.spbstu.ru>

КЛАССИФИКАЦИОННАЯ СХЕМА ГРУППОВЫХ ПОДПИСЕЙ ДЛЯ ПОСТРОЕНИЯ РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЙ

Групповые подписи – относительно новая концепция, представленная впервые в работе [1], – могут быть использованы для скрытия организационных структур некоторой организации, т. к. обеспечивают анонимность подписывающих. Любой член группы может подписать сообщение, но подпись не позволяет узнать личность подписавшего. Обычно в схемах групповой подписи также присутствует третья сторона, которая может узнать личность подписавшего сообщение, воспользовавшись специальной лазейкой.

Благодаря своим свойствам схемы групповой подписи являются очень важным криптографическим средством, применимым во множестве специализированных областей и приложений. К примеру, групповые подписи могут быть использованы какой-либо организацией для удостоверения прайс-листов, пресс-релизов или при представлении тендера на рассмотрение; для проведения удаленной проверки программного обеспечения компьютера с сохранением конфиденциальности стороны [2], что может найти применение в таких сферах, как хоум-банкинг или онлайн-игры. Широко известный проект, в рамках которого используются групповые подписи, – организация безопасного движения транспортных средств, – совместный проект большинства

крупнейших производителей автомобилей и Министерства транспорта США [3].

Схема групповой подписи оперирует следующими сущностями (рис. 1): А – менеджер группы *GM*, который создает группу, добавляет в нее новых членов или лишает членства в группе, выпуская или отзывая сертификаты. Также он может раскрыть любую групповую подпись, тем самым отменив ее свойство анонимности. Иногда функции формирования группы и раскрытия подписи разделяются между двумя субъектами: менеджером группы *GM* и раскрывающим менеджером *OM*; Б – члены группы; В – внешние по отношению к группе субъекты, имеющие доступ к ее открытому ключу.

Обычно схема групповой подписи подразумевает выполнение следующих процедур:

- 1) формирование группы – генерация основных ключей: открытого ключа (ОК) группы и секретного ключа менеджера группы. Секретные ключи членов группы становятся известны им в результате выполнения менеджером действий определенного протокола с каждым из них;
- 2) подписывание сообщения одним из членов группы;
- 3) проверка подписи на основании сообщения и ОК группы;

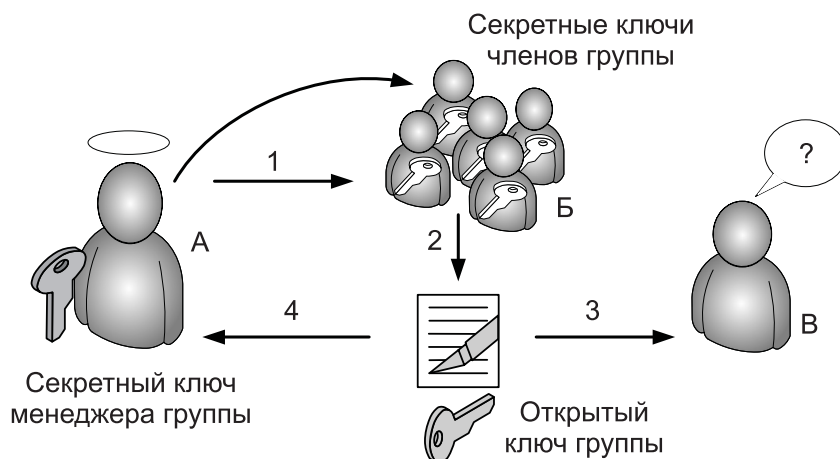


Рис. 1. Обобщенная схема групповой подписи

4) идентификация раскрывающим менеджером члена группы, подписавшего данное сообщение. Раскрытие подписи производится только в случае возникновения споров.

Групповые подписи являются обобщением механизма удостоверения личности (credential mechanism) и схем аутентификации, в которых член группы может доказать проверяющему, что он принадлежит к определенной группе, не раскрывая свой идентификатор.

Чаум и ван Хейст впервые предложили решения, связанные с групповой подписью, а именно четыре различные схемы [1]. С момента первой публикации, посвященной протоколам групповой подписи, было разработано еще несколько схем. Одни были впоследствии «взломаны», другие оказались непрактичными из-за длинных открытых ключей и/или длинных подписей, в то время как третьи предложили недоказанную безопасность [4]. Но это относится, скорее, к схемам, появившимся в течение 1990-х гг. Первой доказуемо безопасной и устойчивой к сговору является схема АСJT [5]. В 2004 г. была представлена BBS – очень эффективная схема короткой групповой подписи, использующая билинейные отображения [3].

Общая тенденция, прослеживаемая по мере создания новых схем, – сокращение вычислительной стоимости и размера итоговой подписи. В частности, BBS – одна из наиболее эффективных по этим показателям. В данной статье предложена классификационная схема групповых подписей, разработана эффективная реализация схемы BBS на основе библиотеки MIRACL, а также представлено распределенное приложение, демонстрирующее возможности этой схемы.

Классификация схем групповой подписи

Чтобы иметь возможность сравнивать схемы групповой подписи, выделим признаки, по которым их можно классифицировать:

1) криптографические предположения, лежащие в основе безопасности:

предположение о сложности разложения на множители (2-я и 3-я схемы Чаума и ван Хейста);

предположение о сложности дискретного логарифмирования (4-я схема Чаума и ван Хейста);

предположение Strong-RSA (схема АСJT);

предположение о задаче распознавания Диффи–Хеллмана – DDH (схема АСJT);

сильное предположение Диффи–Хеллмана –

Strong-DH (схема BBS);

предположение о линейной задаче распознавания Диффи–Хеллмана – DL (схема BBS);

безусловно безопасная схема;

2) роль, необходимость и возможность распределять обязанности доверенной стороны в схеме;

3) возможность динамического членства в группе;

4) тип подписи (есть ли аналог среди традиционных электронно-цифровых подписей (ЭЦП), что включается в подпись и т. п.):

классическая ЭЦП (1-я схема Чаума и ван Хейста);

неопровержимая ЭЦП – подразумевает интерактивный процесс проверки подписи и наличие протокола отказа от нее (2–4 схемы Чаума и ван Хейста);

специфическая ЭЦП – в схемах АСJT и BBS подпись включает в себя зашифрованную на открытом ключе *ОМ* группы часть сертификата пользователя, а также элементы, доказывающие знание пользователем этого сертификата, что определяет его принадлежность к группе;

5) свойства, обеспечиваемые схемой, – уровень ее безопасности:

корректность: подпись, созданная честным членом группы, принимается проверяющим, алгоритм раскрытия правильно идентифицирует подписавшего;

невозможность фальсификации: только члены группы могут подписывать сообщения от лица группы;

анонимность: идентификация члена группы, создавшего подпись, является вычислительно сложной задачей для всех, кроме *ОМ* группы;

отслеживаемость: невозможно выпустить корректную подпись, по которой честный *ОМ* не смог бы идентифицировать подписывающего;

устойчивость к сговору: даже при участии в сговоре *ГМ* и *ОМ* невозможно создать подпись, которая была бы неотслеживаемой или указывала бы на пользователя, не участвующего в сговоре;

несопоставимость: для данных двух групповых подписей невозможно определить, были ли они сформированы одним и тем же пользователем;

невозможность обвинения: ни *ГМ*, ни *ОМ* не могут создать подпись от имени других существующих членов группы;

6) эффективность: размер открытого доверенного хранилища, содержащего ОК группы,

длина результирующей подписи, количество операций, необходимых при формировании и проверке подписи;

7) механизмы, лежащие в основе схемы (например, билинейные отображения).

Такой набор признаков оказывается вполне достаточным для выбора схемы групповой подписи при решении практических задач. При этом он является общим для того чтобы производить сравнение различающихся схем.

Построить какую-либо четкую классификацию всех групповых подписей не представляется возможным из-за многогранности даже обозначенных выше простейших схем. Следует создать классификационную схему, на основании которой можно будет проводить классификацию каж-

дой конкретной групповой подписи (рис. 2).

Схема короткой групповой подписи BBS

В качестве примера использования данной классификационной схемы рассмотрим ее применение к схеме групповой подписи BBS.

Для описания схемы групповой подписи BBS будем использовать следующие обозначения: G_1 – мультипликативная циклическая группа простого порядка p ; G_2 – мультипликативная группа, порядок которой – некоторая степень числа p ; ψ – гомоморфизм из G_2 в G_1 ; $g_2 \in G_2$ – элемент порядка p ; $\langle g_1 \rangle = G_1$ – образующая группы G_1 , такая что $\psi(g_2) = g_1$.

Выбором элемента g_2 обычно ограничивают рассмотрение группы G_2 до ее циклической под-



Рис. 2. Классификационная схема групповых подписей

группы $\langle g_2 \rangle$ порядка p . При этом имеет место изоморфизм $\langle g_2 \rangle \cong G_1$.

Инициализация схемы производится менеджером группы GM . В базовом варианте он лично генерирует все используемые в схеме ключи и раздает менеджеру OM и пользователям их секретные ключи.

Алгоритм генерации ключей BBS

Вход: n – число членов группы.

Выход: gpk – открытый ключ группы, $gmsk$ – секретный ключ менеджера группы, gsk_i , где $1 \leq i \leq n$ – секретные ключи членов группы.

1. Случайным образом выбрать образующую $g_2 \in G_2$ и вычислить $g_1 \leftarrow \psi(g_2)$.

2. Выбрать $h \leftarrow G_1 \setminus \{1_{G_1}\}$, $\xi_1, \xi_2 \leftarrow Z_p^*$ и вычислить $u, v \in G_1$, такие что $u^{\xi_1} = v^{\xi_2} = h$.

3. Выбрать $\gamma \leftarrow Z_p^*$ и вычислить $w \leftarrow g_2^\gamma$.

4. Используя γ , сгенерировать для каждого пользователя – члена группы – пару (A_i, x_i) : выбрать $x_i \leftarrow Z_p^*$ и вычислить $A_i \leftarrow g_1^{1/\gamma + x_i} \in G_1$.

5. Результат: $gpk \leftarrow (g_1, g_2, h, u, v, w)$; $gmsk \leftarrow (\xi_1, \xi_2)$; $gsk_i \leftarrow (A_i, x_i)$, где $1 \leq i \leq n$.

Ни одной из сторон не разрешается знать значение γ . Оно может быть известно только центру, выпускающему секретные ключи, т. е. GM .

Идея групповой подписи BBS состоит в доказательстве знания пары (A, x) , такой что $\langle A^{x+\gamma} \rangle = G_1$.

Чтобы предоставить другому менеджеру OM возможность раскрывать подпись (идентифицировать подписавшего), доказательство только отчасти должно быть с нулевым разглашением: OM должен уметь восстанавливать значение A .

Поэтому чтобы подписать сообщение m , член группы сначала шифрует A на открытом ключе зашифрования менеджера OM – (u, v, h) , а затем предоставляет доказательство с нулевым разглашением того, что открытый текст на самом деле содержит A , для которого он знает соответствующее значение x . Для зашифрования используется алгоритм линейного шифрования, при этом пара значений (ξ_1, ξ_2) , которой обладает OM , является секретным ключом расшифрования, что дает ему возможность раскрывать групповую подпись. Полное описание схемы групповой подписи BBS приведено в работе [3].

Применим предложенную классификационную схему к схеме групповой подписи BBS.

1. Схема групповой подписи BBS основана на сильном предположении Диффи-Хеллмана

(Strong-DH) и на предположении о линейной задаче распознавания Диффи-Хеллмана (DL).

2. Схема BBS, единственная из рассматриваемых, позволяет распределить обязанности руководства группой между GM и раскрывающим менеджером OM . В обязанности GM входит создание группы, формирование всех необходимых ключей и их безопасная доставка до членов группы и OM . В обязанности OM входит раскрытие подписей в случае необходимости разрешения конфликтных ситуаций. В базовой схеме BBS предполагается, что GM самостоятельно генерирует все ключи, используемые в схеме. То есть существует сторона, которая знает секретные ключи всех пользователей и может формировать подпись от чужого имени. Однако схема BBS может быть расширена, чтобы предоставить пользователю возможность самому влиять на свой секретный ключ, делая его неизвестным для GM .

3. BBS поддерживает как динамическое вступление в группу, так и возможность эффективного отзыва членства в группе.

4. Групповая подпись BBS основана на доказательстве (с нулевым разглашением) знания решения задачи Strong-DH, т. е. знания пары (A, x) . Подпись также включает в себя зашифрованную часть сертификата пользователя на открытом ключе OM группы (безопасность линейного шифрования основана на задаче DL).

5. BBS, расширенная путем включения в нее протокола совместного формирования секретных ключей членов, удовлетворяет всем семи требованиям, предъявляемым к групповой подписи.

6. В схеме BBS длина открытого ключа не зависит от количества членов группы. Для нее не актуальна передача каких-либо значений при проверке корректности подписи. При формировании подписи выполняется восемь операций возведения в степень, при проверке подписи – шесть возведений в степень и одно вычисление билинейного отображения.

Групповая подпись BBS включает в себя три элемента группы G_1 и шесть элементов кольца Z_p . При использовании суперсингулярных эллиптических кривых в качестве p можно взять 170-битное простое число и использовать группу G_1 , в которой каждый элемент задается 171 битом. Поэтому длина всей групповой подписи будет составлять 1533 бита или 192 байта. С такими параметрами стойкость подписи будет приблизительно



но такой же, как и у стандартной подписи RSA длиной 1024 бита или 128 байтов.

Для сравнения: длина подписи ACJT, сравнимая по стойкости с 1024-битной подписью RSA, составляет 8696 бит без учета открытых ключей и параметров схемы [6].

7. Схема BBS использует билинейные отображения (спаривания).

8. Безопасность схемы BBS доказана в рамках модели со случайным оракулом. Случайный оракул (random oracle) – теоретический «черный ящик», который отвечает на каждый запрос истинно случайным ответом, выбранным из некоторого множества выходных значений, за исключением некоторых специфических запросов, на которые он отвечает одинаково каждый раз, как получает такой запрос.

Таким образом, схема групповой подписи BBS, предложенная в 2004 г., с некоторыми усовершенствованиями, позже внесенными в нее, до сих пор является одной из лучших схем групповой подписи по совокупности свойств, которыми она обладает. Поэтому в данной работе основное внимание уделяется ее рассмотрению.

С помощью приведенного выше механизма классификации можно лаконично и ёмко охарактеризовать предположительно любую схему групповой подписи, сравнить ее объективные параметры с параметрами других схем и выбрать лучшую для решения некоторой задачи. Приведенная классификационная схема не претендует на полноту – она составлена на основании шести рассмотренных типовых схем групповой подписи. Возможно, в ряде случаев ее придется некоторым образом дополнить, чтобы новая для нее схема групповой подписи адекватно в нее вписалась. Недостатком классификационной схемы является то, что она не отражает субъективную сложность понимания классифицируемой схемы подписи, что означает, что на ее основе невозможно предсказать время разработки программного продукта, реализующего заданную схему.

Реализация схемы групповой подписи BBS

На базе библиотеки NuimCrypto [7], разработанной Национальным Университетом Ирландии, а также библиотеки MIRACL [8] авторами были созданы различные реализации схемы BBS. Сравнение этих подходов между собой, а также с существующей реализацией схемы на основе библиотеки PBC – Pairing-Based Cryptography

Library [9] – показало, что реализация схемы групповой подписи BBS на основе библиотеки MIRACL является наиболее эффективной. Библиотека NuimCrypto, как и реализация схемы BBS на ее основе, – менее эффективны, но зато полностью написаны на языке Java, что обеспечивает их переносимость. Результаты тестирования временных параметров библиотек представлены в таблице.

При работе с библиотекой PBC использовались стандартные параметры классов A (нацелены на эффективность вычислений) и F (нацелены на короткую подпись) [10]. При работе с библиотеками NuimCrypto и MIRACL использовались рекомендуемые параметры: размер характеристики простого поля 512 бит, размер порядка группы точек эллиптической кривой 160 бит и 170 / 160 бит.

В таблице также приведены размеры подписи, получающиеся при использовании определенной библиотеки и размеров параметров. В ходе исследования не ставилась задачи минимизировать эти показатели.

Таким образом, переносимая реализация на Java оказалась неконкурентоспособной по сравнению с двумя другими библиотеками. Это объясняется слабой приспособленностью Java для выполнения такого рода задач, а также реализацией «в лоб» низкоуровневых алгоритмов, на базе которых уже строилась схема групповой подписи BBS.

Библиотека PBC – многоплатформенная, наиболее всеобъемлющая и проработанная из всех существующих библиотек, позволяющая реализовать криптосистему на спариваниях, имея для этого минимальные знания. Однако из-за своей достаточно узкой направленности и абстрактности интерфейса, предоставляемого разработчику, она не позволяет, не вдаваясь в детали реализации, повлиять на используемые параметры системы. По скорости PBC проигрывает библиотеке MIRACL.

Наиболее эффективная реализация – на основе библиотеки MIRACL во многом благодаря очень хорошей оптимизации низкоуровневых примитивов и ее универсальности. Поэтому разработанная библиотека групповой подписи BBS на основе MIRACL была выбрана в качестве основной. Она предоставляет высокоуровневое API для языков Си и Java, что обеспечивает удобство ее применения для создания как распределенных, так и обычных приложений, использующих возможности групповой подписи.

Сравнение реализаций схемы групповой подписи BBS

Библиотека		PBC	NuimCrypto	MIRACL
Язык		C	Java	C/C++
Платформы		Unix-like Ограниченно: Windows(x86) WinCE(ARM) Linux (ARM)	Java SE Java EE	Windows Linux x86
Время генерации параметров системы, мс	512/160	–	6093	766
	170/160	–	657	31
Время генерации ключей, мс	A / 512 / 160	453	87125	546
	F / 170 / 160	1837	13297	157
Время формирования подписи, мс	A / 512 / 160	270	8750	265
	F / 170 / 160	627	1938	78
Время проверки подписи, мс	A / 512 / 160	198	18656	250
	F / 170 / 160	477	3547	78
Время раскрытия подписи, мс	A / 512 / 160	296	20485	297
	F / 170 / 160	1571	3906	79
Размер подписи, байт	A / 512 / 160	504	509	320
	F / 170 / 160	240	254	192

Далее были рассмотрены возможные варианты архитектур распределенных приложений, построенных на базе технологии J2EE, использующих возможности разработанной библиотеки схемы групповой подписи BBS:

1. Взаимодействие по протоколу HTTP. В роли клиентов выступает Web Browser. Недостатком этого подхода является необходимость передачи по сети программного кода JavaScript и апплетов, входящих в HTML-страницу для обеспечения функциональности приложения.

2. Взаимодействие по протоколу HTTP. Клиентом является специализированное приложение Java. При сетевом взаимодействии программный код не передается. Недостатком является необходимость разработки собственного HTTP-клиента.

3. Взаимодействие по протоколу RMI. Клиентом является специализированное приложение Java. Этот подход устраняет необходимость разработки HTTP-клиента. Недостатком является обеспечение возможности удаленного доступа к EJB-контейнеру для клиентской части приложения.

4. Использование технологии Java Web Start для запуска клиентского приложения. Взаимодействие осуществляется по протоколам JNLP и RMI. В данном подходе браузер не является принципиально необходимым звеном для работы приложения. Исполняемый код за время сеанса работы с приложением загружается с сервера при необходимости и не более одного раза, при этом он защищен от постороннего вмешательства цифровой подписью.

Среди всех подходов выделим архитектуру 4, т. к. только она обеспечивает наибольшее удобство конечных пользователей таких систем, наименьшее время разработки, а также соответствует требованиям безопасности, предъявляемым к приложениям, в которых используется групповая подпись. Данную архитектуру можно применить для построения широкого класса распределенных приложений, реализующих такие функции безопасности как аутентификация пользователей, разграничение доступа к функциям приложения, обеспечение целостности передаваемых по сети данных и исполняемого кода, а также ведение журналов аудита.



В соответствии с выбранной архитектурой было разработано демонстрационное приложение, использующее ранее реализованную схему групповой подписи на основе библиотеки MIRACL. Оно моделирует отдельный компонент информационной системы, применяемой предприятиями, осуществляющими техническую поддержку выпускаемой ими продукции. Данный компонент предоставляет средство взаимодействия агентов с конечными пользователями, в котором каждое сообщение, передаваемое от агента клиенту, подписывается посредством групповой подписи. Группой в данном случае является группа агентов – сотрудников обозначенного предприятия.

Использование языка Java и технологии J2EE

позволяет несколько упростить реализацию защищенного распределенного приложения, решающего задачу, специфика которой влечет использование групповой подписи, за счет применения встроенных и проверенных механизмов обеспечения безопасности.

Предложенные в настоящей статье теоретические и практические результаты, могут эффективно применяться при разработке защищенного распределенного приложения, специфика которого предполагает использование групповой подписи как на этапе анализа требований, поиска и исследования существующих механизмов, так и на этапе реализации.

СПИСОК ЛИТЕРАТУРЫ

1. Chaum, D. Group signatures [Текст] / D. Chaum, E. van Heyst // Proc. of Eurocrypt 1991. – Springer-Verlag, Apr. 1991. – P. 257–265.
2. Организация Trusted Computing Group [Электронный ресурс] www.trustedcomputinggroup.org
3. Shacham, H. Collected Papers where Every Theorem Is Filled with Grief. Dec. 2005 [Электронный ресурс] / H. Shacham // <http://cseweb.ucsd.edu/~hovav/dist/thesis-hyperref.pdf>
4. Ateniese, G. Quasi-Efficient Revocation of Group Signatures [Электронный ресурс] / G. Ateniese, D. Song, G. Tsudik // <http://www.cs.jhu.edu/~ateniese/papers/fc02.pdf>
5. Ateniese, G. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme [Электронный ресурс] / G. Ateniese, J. Camenisch, M. Joye [et al.] // Advances in Cryptology–CRYPTO 2000. – Springer Verlag, 2000. – LNCS. – Vol. 1880. – P. 255–270.
- <http://www.zurich.ibm.com/~jca/papers/group2000.pdf>
6. Yao, D. Cascaded Authorization with Anonymous-Signer Aggregate Signatures [Электронный ресурс] / D. Yao, R. Tamassia // Proc. of the 7 Annual IEEE Systems, Man and Cybernetics Information Assurance Workshop (IAW '06). – United States Military Academy, West Point, NY. – June 2006. <http://www.cs.brown.edu/cgc/stms/papers/anonymity.pdf>
7. Библиотека NuimCrypto [Электронный ресурс] http://www.google.com.my/codesearch/p?hl=en&sa=N%20&cd=1&ct=rc#vE_ES1OPacI/software/&q=http://www.crypto.cs.nuim.ie/
8. MIRACL, Multiprecision Integer and Rational Arithmetic C/C++ Library [Электронный ресурс] <http://www.shamus.ie/>
9. The Pairing-Based Cryptography Library [Электронный ресурс] <http://crypto.stanford.edu/pbc/>

УДК 004.052.42

М.Ю. Моисеев

АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ ДЕФЕКТОВ В МНОГОПОТОЧНЫХ ПРОГРАММАХ МЕТОДАМИ СТАТИЧЕСКОГО АНАЛИЗА

Статический анализ (СА) – группа методов, использующих исходный код программы для определения требуемых свойств [1]. Одной из областей применения СА является обнаружение

программных дефектов – нефункциональных ошибок, внесенных на стадии разработки программной системы. Наиболее распространенными типами программных дефектов в последователь-

ных программах на языке С являются: разывение некорректного указателя, переполнение буфера и выход за границы объекта, использование неинициализированных переменных, утечки и повторное освобождение ресурсов. Методы СА позволяют автоматически обнаруживать дефекты всех указанных типов. Основными показателями эффективности статического анализа являются полнота и точность получаемых результатов. *Полнота* – доля обнаруженных дефектов среди всех дефектов, имеющих в программе. *Точность* – доля истинных дефектов среди всех обнаруженных дефектов.

В настоящее время существуют эффективные алгоритмы СА, позволяющие выполнять обнаружение дефектов. Однако большинство известных алгоритмов и их реализации ориентированы на анализ последовательных программ, в то время как многие из существующих и разрабатываемых программных систем используют те или иные механизмы распараллеливания. Одним из классов параллельных программ являются многопоточные программы. В них кроме дефектов последовательных программ могут присутствовать дефекты, связанные с организацией параллельного выполнения – *ошибки синхронизации*. К таким дефектам относятся: конкурентная модификация/использование разделяемых объектов из разных потоков (*Race Condition*), взаимные блокировки (*Deadlock*), некорректное использование функций синхронизации (*Blocking Call Misuse*) и др. Применение алгоритмов СА, не учитывающих специфику многопоточных программ, не позволяет обнаруживать ошибки синхронизации, а также приводит к снижению полноты и точности получаемых результатов.

В данной статье решается задача автоматического обнаружения дефектов последовательных программ в многопоточных программах на языке С на основе методов СА. Основным требованием к разрабатываемым алгоритмам является получение полных результатов. В качестве примера рассматривается организация многопоточных программ на основе потоков и объектов синхронизации определенных в стандарте POSIX (IEEE 1003.1).

Среди работ, посвященных статическому анализу параллельных (многопоточных) программ, можно выделить несколько основных направлений: используются подходы на основе извлечения частичных порядков выполнения конструи-

ций; применяется анализ наборов конструкций синхронизации на путях выполнения программы (*lockset*-анализ); используются системы типов и эффектов.

К первому направлению относится работа [2], в которой предлагается обнаруживать ошибки типа *Race Condition* на основе анализа частичных порядков – потенциальными местами ошибок являются зависимости по данным между конструкциями, для которых отсутствует строгая последовательность выполнения. В [3] учет взаимодействия потоков программы выполняется с помощью объединения состояний разделяемых объектов. Для определения актуальных значений объектов в объединяемых состояниях графа потока управления исходной программы преобразуется в факторизованный граф, в котором учитываются все возможные порядки выполнения конструкций.

К подходам на основе *lockset*-анализа относится работа [4], в которой предлагается простой алгоритм обнаружения ошибок типа *Deadlock* и *Race Condition*. Этот алгоритм использует подсчет числа выполненных операций захвата и освобождения объектов синхронизации на всех путях в графе потока управления до некоторой точки программы. В [5] решается задача определения достижимости совместного состояния параллельных потоков, для программ с рекурсией. Предлагается общий подход построения *lockset*-алгоритмов, рассматриваются алгоритмы *First Occurrence ordering*, *Label bitvector*, *Forbidden and Required sets* и *Parikh images*.

Одним из представителей подходов на основе системы типов является работа [6]. С помощью системы типов строится набор ограничений, позволяющий определить обращения к объектам многопоточной программы без использования объектов синхронизации – такие обращения считаются ошибками типа *Race Condition*.

Оригинальный подход предлагается в [7] – анализ потоков многопоточной программы заменяется на анализ путей в последовательной программе. Для этого исходная многопоточная программа преобразуется в эквивалентную последовательную программу. Эмуляция недетерминизма выполнения параллельных потоков осуществляется с помощью добавления специальных конструкций.

Существует ряд работ, в которых рассматриваются вопросы обнаружения дефектов последо-



вательных программ в многопоточных программах. Одной из таких работ является [8], в которой решается задача обнаружения ошибок разыменования нулевого указателя в многопоточных программах. Совместно используются алгоритмы анализа последовательных программ и алгоритм обнаружения влияний параллельных потоков, удаляющий решения последовательного алгоритма, которые могут быть изменены в конструкциях других потоков.

Рассмотренные работы обладают рядом недостатков: используется ограничение на число потоков, не изучаются динамически порождаемые потоки, учитываются не все взаимные влияния между алгоритмами анализа, применяются решения, приводящие к потере точности, или обладающие неприемлемо высокой вычислительной сложностью. Подход, предлагаемый в данной статье, является развитием существующих подходов и лишен указанных недостатков.

Статический анализ последовательных программ

Задачу обнаружения программных дефектов методами СА можно разделить на три стадии:

- формирование модели программы;
- извлечение необходимой информации из построенной модели;
- обнаружение дефектов на основе полученной информации.

Модель программы является упрощенным представлением исходного кода программы и содержит информацию только о тех объектах и конструкциях, которые интерпретируются алгоритмами анализа. В качестве модели программы для СА наиболее часто используются: синтаксическое дерево разбора (*Abstract Syntax Tree*), абстрактный семантический граф (*Abstract Semantic Graph*), граф потока управления (*Control Flow Graph*), модель программы со статическим однократным присваиванием (*Single Static Assignment*) [9].

На второй стадии определяются возможные *состояния программы* – значения объектов программы в заданных точках. В качестве таких точек выбираются отдельные конструкции программы или блоки конструкций, например, функции (в последнем случае точность результатов снижается). Существуют два варианта определения состояний программы, обеспечивающие получение полных результатов: раздельный анализ путей

выполнения программы и объединение состояний программы в местах слияния потоков управления (первый вариант обладает высокой вычислительной сложностью и поэтому имеет ограниченное применение).

При определении состояний программы обычно выделяют интервальный анализ, анализ указателей и ресурсный анализ. Интервальный анализ определяет значения переменных простых типов. Анализ указателей определяет объекты, на которые указывают переменные-указатели. Ресурсный анализ определяет состояния ресурсов (файлов, сокетов и т. п.). Будем называть алгоритмы интервального анализа, анализа указателей и ресурсного анализа *основными алгоритмами*.

Обнаружение дефектов выполняется для конструкций программы на основе состояния программы перед каждой конструкцией. Типы обнаруживаемых дефектов зависят от типа конструкции программы.

Основная идея предлагаемого подхода

В данной статье предлагается подход, в котором основные алгоритмы анализа дополняются *алгоритмом анализа потоков программы* – эти алгоритмы работают совместно, обмениваясь получаемыми результатами. Алгоритм анализа потоков выполняет создание потоков и анализ запускаемых в них функций, а также обеспечивает правильную последовательность анализа конструкций программы основными алгоритмами. При этом алгоритм анализа потоков использует результаты работы основных алгоритмов, в частности, для идентификации объектов синхронизации.

Модель многопоточной программы содержит *конструкции синхронизации*, с помощью которых представляются функции управления потоками и объектами синхронизации в исходной программе. Конструкции синхронизации интерпретируются алгоритмом анализа потоков программы. Этот алгоритм определяет блоки программы, которые могут выполняться параллельно, а также пары взаимодействующих конструкций синхронизации, с помощью которых обеспечиваются определенные последовательности выполнения блоков разных потоков. При обнаружении дефектов последовательных программ также выполняется обнаружение ошибок синхронизации, которые могут влиять на результаты алгоритмов анализа.

Модель многопоточной программы

При анализе многопоточной программы осуществляется привязка конструкций к потокам, в которых они выполняются. Для этого конструкции программы объединяются в *базовые блоки* – множества последовательных конструкций программы, среди которых отсутствуют конструкции синхронизации. Конструкции синхронизации, а также операторы ветвления и ϕ -функции находятся между базовыми блоками. Обозначим i -й базовый блок, выполняющийся в j -м потоке программы, – S_i^j .

Рассмотрим пару базовых блоков $S_{i_1}^{j_1}$ и $S_{i_2}^{j_2}$, которые выполняются в разных потоках программы. Блоки $S_{i_1}^{j_1}$ и $S_{i_2}^{j_2}$ являются *совместными*, если существует набор входных данных, на котором выполняются оба этих блока.

Конструкции синхронизации взаимодействуют друг с другом с помощью блокировки и освобождения объекта синхронизации – представим взаимодействие между парами конструкций синхронизации с помощью отношений синхронизации. Для конструкций синхронизации s_1 (после блока $S_{i_1}^{j_1}$) и s_2 (после блока $S_{i_2}^{j_2}$) имеет место *отношение синхронизации*, если блоки $S_{i_1}^{j_1}$, $S_{i_2}^{j_2}$ совместны и существует набор входных данных, на котором выполнение s_1 обеспечивается выполнением s_2 . Обозначим отношение синхронизации между такими конструкциями – $S_{i_2}^{j_2} \rightarrow S_{i_1}^{j_1}$.

Между блоками $S_{i_1}^{j_1}$ и $S_{i_2}^{j_2}$ существует *строгая последовательность выполнения*, если на всех наборах входных данных сначала выполняется блок $S_{i_1}^{j_1}$, затем блок $S_{i_2}^{j_2}$ или наоборот. Строгая последовательность выполнения блоков организуется с помощью взаимодействий между конструкциями синхронизации.

Базовые блоки, выполняющиеся в разных потоках программы, *могут выполняться параллельно*, если эти блоки являются совместными и отсутствует строгая последовательность их выполнения. Обозначим отношение параллельности между блоками $S_{i_1}^{j_1}$, $S_{i_2}^{j_2}$ – $S_{i_1}^{j_1} \parallel S_{i_2}^{j_2}$. Отношение параллельности – коммутативное и не является транзитивным:

$$S_{i_1}^{j_1} \parallel S_{i_2}^{j_2} \Rightarrow S_{i_2}^{j_2} \parallel S_{i_1}^{j_1}, S_{i_1}^{j_1} \parallel S_{i_2}^{j_2}, S_{i_1}^{j_1} \parallel S_{i_3}^{j_3} \not\Rightarrow S_{i_2}^{j_2} \parallel S_{i_3}^{j_3}.$$

Конструкции синхронизации выполняются над объектами синхронизации. Изменяя состояние этих объектов, можно выделить *два вида объектов синхронизации*:

объекты первой группы, для которых операции захвата и освобождения должны выполняться в одном и том же потоке;

объекты второй группы, операции над которыми могут выполняться в произвольных потоках.

Среди объектов синхронизации, определенных в стандарте POSIX, с помощью объектов первой группы представляются объекты типа `pthread_mutex_t`, `pthread_rwlock_t`, `pthread_spinlock_t`, `pthread_once_t`, с помощью объектов второй группы – объекты типа `sem_t`, `pthread_barrier_t`, с помощью объектов обеих групп – объекты типа `pthread_cond_t`. В рамках данной статьи ограничимся рассмотрением объектов типа `pthread_mutex_t` (в нерекурсивном режиме) и объектов типа `sem_t`.

Изменение состояния объекта в конструкциях синхронизации назовем *действием* над этим объектом. Для объекта синхронизации первой группы возможны два действия – блокировка (LOCK) и снятие блокировки (UNLOCK). Для объекта синхронизации второй группы действие представляется целым числом с диапазоном значений $(-\infty, +\infty)$. Отрицательные значения соответствуют операциям блокировки объекта, положительные – операциям снятия блокировки, нулевое значение – отсутствию влияния на объект синхронизации.

Состояние объекта синхронизации влияет на результат выполнения конструкций синхронизации. Объект синхронизации первой группы может иметь два состояния – заблокирован (LOCKED) и не заблокирован (UNLOCKED). Состояние объекта синхронизации второй группы – целое число с диапазоном значений $[0, +\infty)$. Значение равно нулю соответствует состоянию блокировки объекта синхронизации, положительные значения – состояниям отсутствия блокировки.

Модель многопоточной программы представляет собой модель последовательной программы, дополненную объектами и конструкциями синхронизации. В качестве модели последовательной программы используется модифицированная модель со статическим однократным присваиванием [10]. Дополнительными объектами являются объекты-потоки программы и объекты синхронизации двух видов. Конструкции синхронизации представляют типичные функции, выполняемые над этими объектами:

`create(t, pf)` – создание потока программы и запуск в нем функции, на которую указывает переменная `pf`;

`join(t)` – ожидание завершения потока программы;
`init(m, value)` – инициализация объекта синхронизации, на который указывает переменная `m`, действие данной конструкции равно `value`;
`destroy(m)` – уничтожение объекта синхронизации;
`lock(m)` – блокировка объекта синхронизации первой группы, действие данной конструкции равно `LOCK`;
`wait(m, dec)` – блокировка объекта синхронизации второй группы, действие данной конструкции равно `dec`;
`unlock(m)` – освобождение объекта синхронизации первой группы, действие данной конструкции равно `UNLOCK`;
`post(m, inc)` – освобождение объекта синхронизации второй группы, действие данной конструкции равно `inc`;

`state(m)` – определение состояния объекта синхронизации;
`eot` – последняя конструкция потока программы.

Представление основных функций над потоками и wybranнми объектами синхронизации стандарта POSIX с помощью конструкций синхронизации приведено в таблице.

Функции `pthread_mutex_trylock` и `pthread_trywait`, а также ряд других функций, определенных в POSIX выполняются атомарно. Корректность представления атомарных функций с помощью нескольких конструкций, а также способы представления остальных функций остаются за рамками данной статьи.

Алгоритм анализа потоков программы

При анализе конструкций синхронизации необходимо учитывать состояние объекта синхронизации. Расчет состояния объекта синхрони-

Исходная функция	Представление с помощью конструкций синхронизации
<code>pthread_create(t, attr, pf)</code>	<code>create(t, pf)</code>
<code>pthread_join(t, result)</code>	<code>join(t)</code>
<code>pthread_mutex_init(m, attr)</code>	<code>init(m, UNLOCK)</code>
<code>pthread_mutex_unlock(m)</code>	<code>unlock(m)</code>
<code>pthread_mutex_lock(m)</code>	<code>lock(m)</code>
<code>pthread_mutex_trylock(m)</code>	<pre> if (state(m) == LOCKED) { return false; } else { lock(m); return true; } </pre>
<code>sem_init(m, value, shared)</code>	<code>init(m, value)</code>
<code>sem_post(m)</code>	<code>post(m, 1)</code>
<code>sem_wait(m)</code>	<code>wait(m, -1)</code>
<code>sem_getvalue(m, result)</code>	<code>*result = state(m)</code>
<code>pthread_trywait(m)</code>	<pre> if (state(m) == 0) { return false; } else { wait(m, 1); return true; } </pre>

зации выполняется на основе анализа действий над этим объектом во всех потоках программы. Обозначим возможные состояния объекта m в базовом блоке $S_i^j - m(S_i^j)$. Для определения $m(S_i^j)$ строятся комбинации блоков других потоков, параллельно с которыми может выполняться блок S_i^j . Назовем все такие комбинации для блока S_i^j – *множеством допустимых комбинаций*, $\{C'\}_{S_i^j}$.

Множество допустимых комбинаций строится из блоков, которые могут быть параллельны блоку S_i^j ; в каждую комбинацию входит по одному блоку из каждого потока; все блоки комбинации должны быть попарно параллельны:

$$\frac{\forall C \in \{S_{i_1}^{j_1} : S_{i_1}^{j_1} \parallel S_i^j\} : \forall S_{i_2}^{j_2}, S_{i_3}^{j_3} \in C \ S_{i_2}^{j_2} \parallel S_{i_3}^{j_3}}{C \in \{C'\}_{S_i^j}}$$

Действия над объектами синхронизации рассматриваются отдельно для каждого потока программы. Называем *множеством действий* над объектом синхронизации в базовом блоке действия над этим объектом на всех путях от первой конструкции потока до данного блока. Обозначим множество действий над объектом m в базовом блоке $S_i^j - A(m, S_i^j)$. Правила влияния конструкций программы на множество действий выглядят следующим образом (обозначения блоков приведены на рис. 1):

$$\begin{aligned} & \frac{}{[if() S_{i_2}^{j_1} else S_{i_3}^{j_1}] : A(m, S_{i_2}^{j_1}) = A(m, S_{i_2}^{j_1}), A(m, S_{i_3}^{j_1}) = A(m, S_{i_3}^{j_1})}, \\ & \frac{}{[S_{i_3}^{j_1} = \phi(S_{i_1}^{j_1}, S_{i_2}^{j_1})] : A(m, S_{i_3}^{j_1}) = A(m, S_{i_1}^{j_1}) \cup A(m, S_{i_2}^{j_1})}, \\ & \frac{}{[create(t, pf)] : A(m, S_{i_1+1}^{j_1}) = A(m, S_{i_1}^{j_1}), A(m, S_{i_1}^{j_2}) = \{UNLOCK\} \cup \{0\}}, \\ & \frac{}{[join(t)] : A(m, S_{i_1+1}^{j_1}) = A(m, S_{i_1}^{j_1}) + A(m, S_{i_2}^{j_2})}, \end{aligned}$$

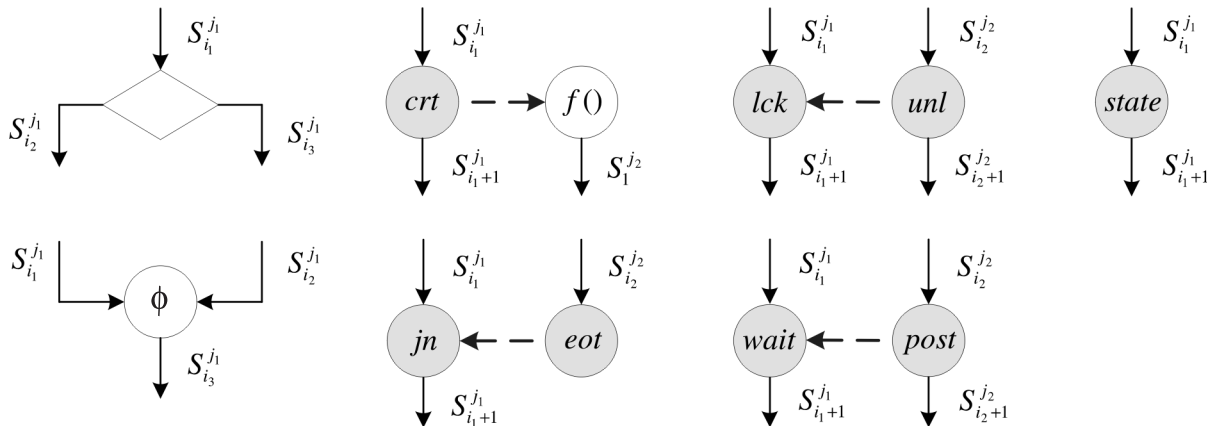


Рис. 1. Обозначения блоков программы

$$\begin{aligned} & \frac{}{[lock(m)] : A(m, S_{i_1+1}^{j_1}) = \{LOCK\}}, \\ & \frac{}{[unlock(m)] : A(m, S_{i_2+1}^{j_2}) = \{UNLOCK\}}, \\ & \frac{}{[wait(m, dec)] : A(m, S_{i_1+1}^{j_1}) = A(m, S_{i_1}^{j_1}) + dec}, \\ & \frac{}{[post(m, inc)] : A(m, S_{i_2+1}^{j_2}) = A(m, S_{i_2}^{j_2}) + inc}. \end{aligned}$$

Множество действий в первой конструкции потока равно $\{UNLOCK\}$ для объектов первой группы и $\{0\}$ для объектов второй группы. Результат операции сложения множеств действий представляет собой все возможные суммы из действий каждого множества:

$$\begin{aligned} A(m, S_{i_1}^{j_1}) &= \{a_1^1, \dots, a_{i_1}^1, \dots, a_{N_1}^1\}, \\ A(m, S_{i_2}^{j_2}) &= \{a_1^2, \dots, a_{i_2}^2, \dots, a_{N_2}^2\}, \\ A(m, S_{i_1}^{j_1}) + A(m, S_{i_2}^{j_2}) &= \{a_{i_1}^1 + a_{i_2}^2\} \quad \forall i_1 = 1 \dots N_1, i_2 = 1 \dots N_2. \end{aligned}$$

Состояния объектов синхронизации второй группы определяются с помощью следующего правила:

$$\frac{\forall C \in \{C'\}_{S_{i_1}^{j_1}}}{[state(m)] : m(S_{i_1}^{j_1}) = m(S_{i_1}^{j_1}) \cup \left[A(m, S_{i_1}^{j_1}) + \sum_{\forall S_{i'}^{j'} \in C} A(m, S_{i'}^{j'}) \right]}.$$

Начальное значение состояния $m(S_{i_1}^{j_1})$ – пустое множество.

На состояние объекта синхронизации первой группы влияют блоки других потоков, в которых этот объект может быть захвачен. Правила определения состояния объектов первой группы выглядят следующим образом:

$$\frac{\exists C \in \{C'\}_{S_i^j} : \exists S_i^j \in C \cup S_i^j, \{LOCK\} \in A(m, S_i^j)}{[state(m)]: m(S_i^j) = m(S_i^j) \cup \{LOCKED\}},$$

$$\frac{\exists C \in \{C'\}_{S_i^j} : \forall S_i^j \in C \cup S_i^j, \{UNLOCK\} \in A(m, S_i^j)}{[state(m)]: m(S_i^j) = m(S_i^j) \cup \{UNLOCKED\}}.$$

Правила построения отношений параллельности выглядят так:

$$\frac{\forall S_i^j : S_i^j \parallel S_i^j}{[if() S_i^j else S_i^j]: S_i^j \parallel S_i^j, S_i^j \parallel S_i^j},$$

$$\frac{\forall S_i^j : S_i^j \parallel S_i^j \vee S_i^j \parallel S_i^j}{[S_i^j = \Phi(S_i^j, S_i^j)]: S_i^j \parallel S_i^j},$$

$$\frac{\forall S_i^j : S_i^j \parallel S_i^j}{[create(t, pf)]: S_i^j \parallel S_i^j, S_i^j \parallel S_i^j, S_i^j \parallel S_i^j},$$

$$\frac{\forall C : C \in \{C'\}_{S_i^j}, S_i^j \in C, S_i^j \in C}{[join(t)]: S_i^j \parallel S_i^j},$$

$$\frac{\forall S_i^j \in C : C \in \{C'\}_{S_i^j}, \exists S_i^j \in C A(m, S_i^j) = \{LOCK\}}{[lock(m)]: S_i^j \parallel S_i^j},$$

$$\frac{\forall S_i^j : S_i^j \parallel S_i^j}{[unlock(m)]: S_i^j \parallel S_i^j},$$

$$\frac{\forall S_i^j \in C : C \in \{C'\}_{S_i^j}, \exists a \in \left[A(m, S_i^j) + \sum_{\forall S_i^j \in C} A(m, S_i^j) \right], a > 0}{[wait(m)]: S_i^j \parallel S_i^j},$$

$$\frac{\forall S_i^j : S_i^j \parallel S_i^j}{[post(m)]: S_i^j \parallel S_i^j}.$$

Блок после конструкции `join` параллелен блокам, входящим в комбинации для блока перед этой конструкцией, в которые также входит последний блок завершающегося потока. Блок после конструкции `lock` или `wait` параллелен всем блокам, входящим в комбинацию для блока перед этой конструкцией, при условии, что объект синхронизации в данной комбинации может быть не заблокирован.

Правила построения отношений синхронизации имеют вид:

$$\frac{}{[create(t, pf)]: S_i^j \rightarrow S_i^j},$$

$$\frac{}{[join(t)]: S_i^j \rightarrow S_i^j, [lock(m)]: S_i^j \rightarrow S_i^j},$$

$$\frac{\exists C : C \in \{C'\}_{S_i^j}, \exists a \in \left[A(m, S_i^j) + \sum_{\forall S_i^j \in C} A(m, S_i^j) \right], a = 0, S_i^j \in C}{[wait(m)]: S_i^j \rightarrow S_i^j}.$$

Отношение синхронизации для пары конструкций `wait/post` строится в случае, если в комбинации для блока перед конструкцией `wait` объект синхронизации может изменить свое состояние после выполнении конструкции `post`.

Приведенные правила алгоритма анализа потоков являются необходимыми – обеспечивают получение полных результатов.

На рис. 2 приведен пример программы, в которой используется объект синхронизации первой группы. Рассмотрим работу предложенного алгоритма для блоков потока 1, считая, что остальные

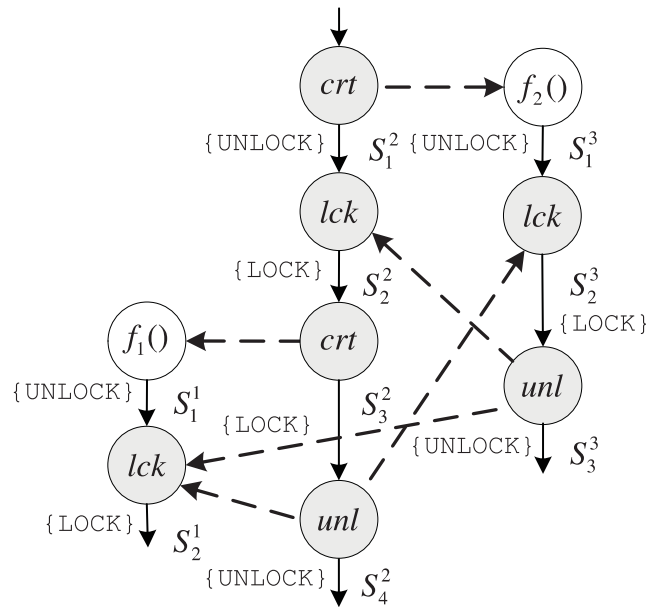


Рис. 2. Пример работы алгоритма анализа потоков программы

потоки программы уже проанализированы. Блок S_1^1 может выполняться параллельно с блоками: $S_4^2, S_4^3, S_1^3, S_2^3, S_3^3$. Между этими блоками имеют место следующие отношения параллельности: $S_3^2 \parallel S_1^3, S_3^3, S_4^2 \parallel S_1^3, S_2^3, S_3^3$. Множество допустимых комбинаций $\{C\}_{S_i^j}$ включает следующие комбинации: $\langle S_3^2, S_1^3 \rangle, \langle S_3^2, S_3^3 \rangle, \langle S_4^2, S_1^3 \rangle, \langle S_4^2, S_2^3 \rangle, \langle S_4^2, S_3^3 \rangle$. Состояние объекта синхронизации $m(S_1^1) = \{LOCKED, UNLOCKED\}$. Правилу построения отношений параллельности для конструкции `lock` удовлетворяют комбинации $\langle S_4^2, S_1^3 \rangle, \langle S_4^2, S_3^3 \rangle$, – блок S_2^1 может выполняться параллельно с S_4^2, S_1^3, S_3^3 . Так как $S_1^1 \parallel S_3^2, S_2^3$, то для конструкции `lock` строятся две дуги синхронизации: $S_3^2 \rightarrow S_1^1, S_2^3 \rightarrow S_1^1$.

Взаимодействие потоков программы

Взаимное влияние потоков в многопоточной программе заключается в изменении состояний разделяемых объектов. *Разделяемым объектом* называется объект программы, доступ к которому может осуществляться из нескольких потоков. При проведении статического анализа взаимодействие параллельных потоков учитывается путем объединения состояний разделяемых объектов. Для этого используются специальные конструкции – ψ -функции [11].

Добавление ψ -функций осуществляется для конструкций `create`, `join`, `lock` и `wait`. Каждая ψ -функция имеет один вход из текущего потока (кроме ψ -функции для конструкции `create`) и один или несколько входов из других потоков. Входы в ψ -функцию из других потоков строятся на основе дуг синхронизации – для каждой входящей дуги синхронизации добавляется один вход в ψ -функцию.

Объединение состояний в ψ -функциях обеспечивает получение полных результатов, однако точность получаемых результатов может оказаться неудовлетворительной. Это связано с тем, что при отсутствии ошибок синхронизации изменение состояния разделяемого объекта возможно только в блоках одного из параллельно выполняющихся потоков. При объединении состояний программы в ψ -функции значения разделяемого объекта либо будут одинаковы, либо актуальным является значение на одном из входов ψ -функции.

Для определения актуального значения необходимо фиксировать изменения значений разделяемых объектов в базовых блоках програм-

мы. Обозначим значение разделяемого объекта e в блоке $S_i^j - e(S_i^j)$, а изменение этого значения в данном блоке – $def(e) \in S_i^j$. Определение актуального значения для ψ -функции с двумя входами выполняется по следующим правилам (обозначения блоков приведены на рис. 3):

$$\frac{\exists S_i^{j_2} : S_i^{j_1} \parallel S_i^{j_2}, S_i^{j_2} \in Pred(S_i^{j_2}), def(e) \in S_i^{j_2}}{e(S_{i+1}^{j_1}) = e(S_{i_2}^{j_2})},$$

$$\frac{\exists S_i^{j_1} : S_i^{j_2} \parallel S_i^{j_1}, S_i^{j_1} \in Pred(S_i^{j_1}), def(e) \in S_i^{j_1}}{e(S_{i+1}^{j_1}) = e(S_{i_1}^{j_1})},$$

$$\frac{\exists S_i^j : S_i^{j_1} \parallel S_i^j, def(e) \in S_i^j}{e(S_{i+1}^{j_1}) = e(S_{i_2}^{j_2})},$$

$$\frac{\exists S_i^j : S_i^{j_2} \parallel S_i^j, def(e) \in S_i^j}{e(S_{i+1}^{j_1}) = e(S_{i_1}^{j_1})}.$$

$Pred(S_i^j)$ – множество блоков j -го потока, выполнение которых предшествует выполнению блока S_i^j ; блок S_i^j также входит в $Pred(S_i^j)$. Доказательство корректности приведенных правил остается за рамками статьи.

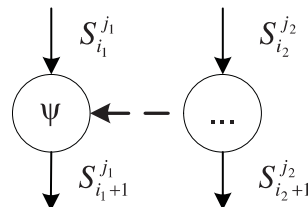


Рис. 3. Обозначения блоков для ψ -функции

Обнаружение дефектов

В данной статье решается задача обнаружения дефектов последовательных программ. Алгоритм анализа потоков рассчитан на применение для программ, в которых отсутствуют (обнаружены) ошибки типа *Blocking Call Misuse* (правила для конструкций `join` и `lock`) и *Race Condition* (правила определения актуального состояния). Поэтому, наряду с обнаружением программных дефектов последовательных программ, также необходимо обнаруживать ошибки синхронизации этих типов.

Обнаружение дефектов последовательных программ выполняется с помощью обычных правил, используемых для обнаружения таких дефектов в однопоточных программах.

К ошибкам типа *Blocking Call Misuse* относятся: отсутствие освобождения заблокированного



объекта на момент завершения потока, повторная блокировка объекта в конструкции `lock`, освобождение объекта, незахваченного в данном потоке в конструкции `unlock`. Такие ошибки обнаруживаются на основе анализа действий над объектами синхронизации с помощью следующих правил:

$$\frac{\exists m : \{LOCK\} \in A(m, S_j^i)}{[eot] : Defect},$$

$$\frac{\{LOCK\} \in A(m, S_j^i)}{[lock(m)] : Defect},$$

$$\frac{\{UNLOCK\} \in A(m, S_j^i)}{[unlock(m)] : Defect}.$$

Ошибки типа *Race Conditional* представляют собой обращения к разделяемому объекту в параллельных блоках программы, если хотя бы одно из обращений является обращением на запись. Для обнаружения таких ошибок проверяются операции над разделяемыми объектами, выполняющиеся в параллельных блоках. Операции связанные с использованием (чтением значения) разделяемого объекта e в блоке S_i^j обозначим $use(e) \in S_i^j$. Правила обнаружения ошибок типа *Race Conditional* выглядят следующим образом:

$$\frac{\exists S_{i_1}^{j_1}, S_{i_2}^{j_2} : S_{i_1}^{j_1} \parallel S_{i_2}^{j_2}, use(e) \in S_{i_1}^{j_1}, def(e) \in S_{i_2}^{j_2}}{RaceCondition},$$

$$\frac{\exists S_{i_1}^{j_1}, S_{i_2}^{j_2} : S_{i_1}^{j_1} \parallel S_{i_2}^{j_2}, def(e) \in S_{i_1}^{j_1}, def(e) \in S_{i_2}^{j_2}}{RaceCondition}.$$

В статье рассмотрены вопросы обнаружения программных дефектов в многопоточных программах на основе статического анализа. Предложен подход, расширяющий алгоритмы СА

последовательных программ на многопоточные программы. Данный подход обеспечивает получение полных результатов. В его основе алгоритм анализа потоков программы. Вычислительная сложность этого алгоритма зависит от количества конструкций синхронизации, в которых используется множество допустимых комбинаций. Сложность анализа одной такой конструкции оценивается как $O(n^k \cdot k^2 \cdot \log(k \cdot n))$, где k – число параллельно выполняющихся потоков; n – число блоков, параллельных блоку перед конструкцией синхронизации.

Сравнивая предложенный подход с известными алгоритмами анализа параллельных (многопоточных) программ, можно отметить следующие преимущества первого:

обеспечивается анализ программ с произвольным числом взаимодействующих потоков, поддерживаются динамически создаваемые потоки;

учитываются все взаимные влияния между алгоритмами анализа, что позволяет получать полные результаты;

обеспечивается совместное выполнение алгоритмов, что дает возможность получать более точные результаты по сравнению с итеративными подходами [12];

анализ состояния объектов на основе комбинаций позволяет точнее строить отношения параллельности и синхронизации по сравнению с классическими алгоритмами *lockset*-анализа;

объединение состояний разделяемых объектов выполняется с учетом актуальных значений, что гарантирует получение более точных результатов по сравнению с простым объединением состояний и обладает меньшей вычислительной сложностью по сравнению с использованием факторизованного графа потока управления.

СПИСОК ЛИТЕРАТУРЫ

1. **Nielsen, F.** Principles of Program Analysis [Текст]/F. Nielsen, H.R. Nielson, C. Hankin. –Berlin, Springer, 2005. – 452 p.
2. **Callahan, D.** Analysis of Event Synchronization in Parallel Programming Tool [Текст]/D. Callahan, K. Kennedy, J. Subhlok // Proc. of the 2 ACM SIGPLAN symp. on Principles & practice of parallel programming. – ACM NY, 1990.
3. **Stoltz, E.** Static single assignment for explicitly parallel programs: Theory and Practice [Текст]/E. Stoltz, H. Shrinivasan, J. Hook [et al.]//Proc. of the 20th ACM SIGPLAN-SIGACT symp. On Principles of programming

languages. – ACM NY, 1994.

4. **Engler, D.** Effective, Static Detection of Race Conditions and Deadlocks [Текст]/D. Engler, K. Ashcraft, X. Racer//Proc. of the 19th ACM symp. on Operating systems principles. – ACM NY, 2003.

5. **Bouajjani, A.** A Generic Approach to the Static Analysis of Concurrent Programs with Procedures [Текст]/A. Bouajjani, J. Esparza, T. Touili//Proc. of the 30th ACM SIGPLAN-SIGACT symp. on Principles of programming languages. – ACM NY, 2003.

6. **Pratikakis, P.** LOCKSMITH: Context-Sensitive Correlation Analysis for Race Detection [Текст]/P.

Pratikakis, J. Foster, M. Hicks//Proc. of the 2006 ACM SIGPLAN conf. on Programming language design and implementation. – ACM NY, 2006.

7. **Qadeer, S.** KISS: Keep It Simple and Sequential [Текст]/S. Qadeer, D.Wu//Proc. of the ACM SIGPLAN 2004 conf. on Programming language design and implementation. – ACM NY, 2004.

8. **Chugh, R.** Dataflow Analysis for Concurrent Programs using Datarace Detection [Текст]/R. Chugh, J. Young, R. Jhala [et al.]// Proc. of the 2008 ACM SIGPLAN conf. on Programming language design and implementation. – ACM NY, 2008.

9. **Ицыксон, В.М.** Исследование средств построения моделей исходного кода программ на языках С и С++ [Текст]/В.М. Ицыксон, М.И. Глухих, А.В. Зо-

зуля [и др.]//Научно-технические ведомости СПбГПУ. – 2009. – № 1.

10. **Cytron, R.** Efficiently Computing Static Single Assignment Form and the Control Dependence Graph [Текст]/R. Cytron [et al.]//ACM Transactions on Programming Languages and Systems. – ACM NY, 1991.–Vol. 13.–№ 4.

11. **Ramanujam, J.** Languages and Compilers for Parallel Computing. Analysis of Event Synchronization in Parallel Programs [Текст]/J. Ramanujam, A. Mathew. –Berlin, Springer, 1995.–P. 300–315.

12. **Моисеев, М.Ю.** Итеративный алгоритм статического анализа для обнаружения дефектов в исходном коде программ [Текст]/М.Ю. Моисеев // Информационные и управляющие системы.–2009.–№ 3.

УДК 004.021

А.А. Иванков, Д.С. Елисеев

ДИНАМИЧЕСКОЕ ОТСЛЕЖИВАНИЕ МОДУЛЕМ ИНФОРМАЦИОННО-ПОИСКОВОЙ СИСТЕМЫ ИЗМЕНЕНИЙ В СТРУКТУРЕ ИЛИ ТЕКСТЕ ИНТЕРНЕТ-РЕСУРСА

В данной статье предлагается решение задачи динамического отслеживания программными средствами информационно-поисковой системы (ИПС) изменений Интернет-ресурса на основе анализа изменений в его структуре или содержании. В качестве Интернет-ресурса (далее – документа) будем рассматривать только HTML-страницы. Опыт работы с собственной ИПС [1–3] позволяет утверждать, что подавляющее большинство веб-серверов по-прежнему предоставляют доступ к своим ресурсам в формате HTML-страниц. Мы не ставили перед собой задачу анализа документов в форматах XML, RDF и т. п. Они слишком редко встречаются в процессе обхода роботами нашей ИПС веб-серверов, изменения ресурсов которых представляют для нас интерес. Реализованные нами алгоритмы сравнительно просто расширить и на упомянутые выше форматы. Изменениями в структуре или содержании HTML-страницы будем оперировать как количественными характеристиками, задав соответствующие метрики.

Важность этой задачи вызвана тем, что БД ИПС должна, по возможности, содержать актуальную копию документа или отдельных его частей. В случае веб-сервера, который предоставляет доступ к данному документу и корректно обрабатывает запрос клиента «If-Modified-Since», такую задачу приходится решать всякий раз, когда изменяется значение атрибута «Last Modified» этого документа. Для документов, которые хранятся на веб-серверах, где администрирование организовано недостаточно профессионально, эту задачу решаем с определенной периодичностью. Когда атрибуты документа указывают, что последний претерпел какие-то изменения, естественным образом возникает еще одна задача – обнаружение, локализация этих изменений в структуре и содержании HTML-страницы. Следует подчеркнуть, что мы имеем в виду еще и алгоритмы для сбора данных, которые периодически обновляются и накапливаются на веб-серверах в течение достаточно длительного периода их существования. Примерами таких данных могут быть показатели



различных финансовых институтов (бирж, рынков), метеосводки, сообщения в блогах, а решаемыми при этом задачами – ретроспективный сбор этих данных для последующего их анализа. Еще один пример задачи, которая может быть решена с помощью наших алгоритмов, – метапоиск и исследования, требующие имитации действий пользователя, обращающегося с запросом к ИПС.

Автоматическое отслеживание изменений может быть основано на различных алгоритмах и моделях обрабатываемого документа. Хорошо известны такие алгоритмы как HtmlDiff, MMDiff, X-Diff, XYDiff. Они позволяют отслеживать изменения в частично структурированных или иерархически организованных документах и уже вошли в арсенал ряда поисковых машин. В большинстве работ для сравнения меток вершин помеченного дерева как иерархической модели структуры документа привлекается модель плоского текста. Нам представляются перспективными исследования алгоритмов сравнения иерархических моделей документов, предполагающие задание различных количественных мер семантической близости текстов в вершинах этого дерева.

Алгоритмы отслеживания семантических изменений в документах – это интенсивно развивающееся в настоящее время направление [4–7]. Мы лишь вкратце, на примере алгоритма АСОИ [6, 7], отметим присущие им достоинства и недостатки. В [6, 7] предлагается алгоритм семантического отслеживания изменений в документах, представленных в форме HTML-страницы. Автор этих публикаций, как и целого ряда подобных им, оперирует понятием семантики документа или отдельных его частей. Понятие семантики обычно интерпретируется в рамках семантической модели «сущность–отношение», ставшей уже канонической в технологиях проектирования реляционных баз данных. Такой подход, безусловно, обладает элементом новизны по сравнению с векторными моделями документов. Один из недостатков этих алгоритмов – их односторонняя ориентация только на семантический аспект. Автор АСОИ строит семантическую иерархию документа на основе иерархии некоторого подмножества HTML-тегов и тут же замечает, что «описание изменения в HTML-коде, основанное на HTML-грамматике,... не является достаточно информативным, и с его помощью тяжело отследить реальное изменение» [6]. Другими словами, для выделения семантических иерархий

в документе используется HTML-грамматика, и при этом отрицается ее, фундаментальная роль в таком алгоритме. Это противоречие вряд ли можно трактовать как диалектическое. На наш взгляд, при анализе изменений HTML-страницы HTML-грамматика может предоставить отправную точку для выявления таких изменений в структуре ресурса. Содержимое структурных элементов, которые изменились или были добавлены в новой редакции ресурса, далее может быть подвергнуто морфологическому, синтаксическому и семантическому анализу.

Постановка задачи

Динамическое отслеживание изменений в документах должно быть основано на моделях, в которых учтена как структура, так и содержание. При этом анализ структуры играет иногда не менее важную роль, чем анализ текста. Алгоритмы должны, по возможности, обеспечить автоматический режим работы. Огромные объемы информации, подлежащей мониторингу, с необходимостью предполагают, что операторы ИПС участвуют в этом процессе как можно реже. Они (операторы ИПС) должны привлекаться только в следующих исключительных случаях:

инициирование процесса мониторинга конкретного документа или их конечного подмножества; одна из главных задач на этом этапе – анализ его (их) структуры путем уточнения подмножества HTML-тегов и отношения порядка, заданных по умолчанию на элементах этого подмножества;

устранение ошибок гипертекстовой разметки в исходном документе, когда такие исправления не удастся сделать автоматически;

принятие решения о продолжении мониторинга после кардинального изменения структуры новой редакции документа, когда содержание всего документа, либо какой-то его части не удовлетворяет критерию релевантности информационной потребности.

Обсуждаемый в данной статье метод и реализованный на его основе модуль ИПС построены на основе метризации множества редакций (версий) документа как в отношении их структуры, так и в отношении содержания всего текста документа или отдельной его части. Под метризацией мы подразумеваем процедуру задания метрики на множестве элементов определенного типа.

Структура документа определяется исходя из отношения порядка, заданного нами на некотором подмножестве HTML-тегов. Это подмножество выделено нами, на основе личного опыта разработки и сопровождения веб-сайтов и веб-порталов. Стандарт языка HTML не предполагает, что его теги будут служить для определения структуры документа. Теги гипертекстовой разметки, в отличие от тегов XML, указывают только на то, как следует отображать документ, отдельные его элементы. Встречаются ресурсы, подготовленные с нарушением HTML-грамматики. В тех случаях, когда не возникает неоднозначных вариантов устранения ошибок, с ними справляется модуль ИПС. Но исправление грубых ошибок – сложная задача. Принимая во внимание ошибки верстки, которые допускают при подготовке материала к публикации в Интернете, наши допущения относительно иерархии, которую мы задаем на подмножестве HTML-тегов, могут быть оспорены, как не единственно возможные для конкретного документа. Выбранное нами подмножество HTML-тегов и отношение порядка отражают, на наш взгляд, классическую семантическую иерархию отдельных частей документа, когда заголовки раздела рассматривается в качестве семантического родителя по отношению к самому разделу. Однако такой выбор далеко не однозначен, и для каждого конкретного ресурса возможен пересмотр как элементов этого множества тегов, так и отношения порядка на них в зависимости от конкретных информационных потребностей, для удовлетворения которых иницируется мониторинг.

Это главные причины, по которым к инициации мониторинга изменений документа необходимо привлекать оператора ИПС.

Метризация на основе отношения порядка, заданного на подмножествах HTML-тегов

Для определения структуры документа нами были выделены следующие подмножества HTML-тегов (указаны вместе с отношением порядка, заданным на элементах):

$$\begin{aligned} \text{Html} &< \text{Head} < \{ \text{TITLE}, \text{META} \} < \text{Body} < \\ &< \{ \text{Hx} < \{ \text{P}, \text{PRE}, \text{DIR}, \text{MENU}, \text{DL}, \text{DT}, \text{DD}, \text{UL}, \text{OL}, \text{LI}, \text{TABLE} < \{ \text{TR} < \{ \text{TH}, \text{TD} \} \} \} \} \}, \end{aligned} \quad (1)$$

где Hx – один из элементов $\{ \text{H1}, \text{H2}, \text{H3}, \text{H4}, \text{H5}, \text{H6} \}$, и на этом подмножестве, в свою очередь, со-

гласно HTML-грамматике, задано отношение порядка $\text{H1} < \text{H2} < \text{H3} < \text{H4} < \text{H5} < \text{H6}$.

На основе этой иерархии HTML-тегов выполняется построение описания структуры документа, отчасти совпадающее с так называемым «DOM-деревом» этого документа. Здесь следует понимать, что эта иерархия используется для построения помеченного дерева, которое не изоморфно DOM-дереву лишь в силу редуцирования из гипертекстового документа тегов (вместе с их содержимым), исключенных при построении подмножеств (1). Не следует думать, что, вложенные в ячейку таблицы, $\langle \text{TD} \rangle$, списки или теги $\langle \text{P} \rangle$, $\langle \text{DIV} \rangle$ создают какие-либо трудности при построении нашего описания структуры документа. Такая ситуация разрешается модулем ИПС стандартным образом и вершина дерева, помеченная тегом $\langle \text{TD} \rangle$, будет родителем вершин с метками $\langle \text{P} \rangle$, $\langle \text{DIV} \rangle$ и т. п. В ИПС поддерживаются две формы такого описания: скобочная запись и XML.

Метрика на множестве вышеуказанных структур задается стандартным образом, как редактирующее расстояние [8].

Пусть имеется конечное (ограниченное сверху) множество сигнатур (меток) вершин Sgntr . В простейшем случае – множество HTML-тегов. Ограничение на мощность множества Sgntr объясняется вполне очевидными соображениями: обрабатываемые документы всегда конечного размера. Дополним множество Sgntr сигнатурой пустого дерева $\emptyset$, т. е. пустое подмножество, неявно принадлежащее любому множеству, в т. ч. и Sgntr , теперь указано явно.

Пусть T_1, T_2 – два помеченных дерева. При чем их представления (скобочная запись) получены с помощью одного и того же порядка перечисления вершин. Множества вершин этих деревьев, $\text{Nodes}(T_1), \text{Nodes}(T_2)$, – это подмножества Sgntr .

Введем три элементарные (атомарные) операции редактирования дерева:

(n_1, n_2) – переименование вершины;

($\emptyset, n_1$) – добавление вершины вместе с прилегающими к ней ребрами;

($n_1, \emptyset$) – удаление вершины вместе с прилегающими к ней ребрами.

Определим на декартовом произведении $\text{Nodes}(T_1) \times \text{Nodes}(T_2)$ множеств вершин деревьев T_1, T_2 функцию, которая, в конечном счете, позволит оценить стоимость преобразования дерева T_1 в дерево T_2 :

$$\text{cost}:(\text{Nodes}(T1) \times \text{Nodes}(T2)) / (\theta \times \theta) \rightarrow R. \quad (2)$$

Для этого достаточно определить стоимости элементарных преобразований, $\text{cost}(n1, n2)$, $\text{cost}(\theta, n2)$, $\text{cost}(n1, \theta)$ соответственно, а функцию cost применять рекурсивно.

Как любая другая метрика, функция cost должна удовлетворять следующим аксиомам для произвольных вершин $n1, n2, n3$:

- 1) $\text{cost}(n1, n2) \geq 0$;
- 2) $\text{cost}(n1, n2) = \text{cost}(n2, n1)$;
- 3) $\text{cost}(n1, n3) \leq \text{cost}(n1, n2) + \text{cost}(n2, n3)$.

Обозначив элементарное преобразование $op = (n1, n2)$, а последовательность элементарных преобразований дерева $T1$ в дерево $T2$ как $OP = op_1, \dots, op_i$, запишем оценку стоимости этой последовательности преобразования дерева $T1$ в дерево $T2$:

$$\text{cost}(OP) = \sum_{j=1}^i \text{cost}(op_j), \quad (4)$$

где i – число элементарных операций, преобразующих дерево $T1$ в дерево $T2$. Очевидно $i \leq |\text{Nodes}(T1)| + |\text{Nodes}(T2)|$.

Тогда метрику, или редактирующее расстояние, между двумя структурами (в нашем случае они представляют собой деревья $T1, T2$), можно задать следующим образом:

$$\delta(T_1, T_2) = \min(\text{cost}(OP) : OP \in OPs), \quad (5)$$

где OPs – множество всех последовательностей элементарных преобразований дерева $T1$ в дерево $T2$.

Нетрудно заметить, что заданная таким образом аддитивная функция тоже удовлетворяет аксиомам метрики.

При выборе алгоритма для вычисления редактирующего расстояния (5) мы рассматривали следующие возможные варианты: MMDiff, XDiff, XYDiff [8,9]. XYDiff – это приближенный алгоритм. Наши исследования и оценки, приведенные в [9], указывают на то, что время работы алгоритма XYDiff практически линейно зависит от объема документа, но всегда дает неоптимальный результат (переоценивает количество внесенных в документ изменений). Алгоритм X-Diff, разработанный для неупорядоченных деревьев, медленнее работает на сильно ветвящихся деревьях, а типичные HTML-страницы, согласно нашим собственным результатам, – это несбалансированные деревья (более чем в 50 % случаев – пере-

кос по высоте вправо, более чем в 20 % случаев – перекося по высоте влево). Кроме того, вычислительная сложность X-Diff (количество операций) выше, чем для MMDiff [9, 10].

В итоге был выбран алгоритм MMDiff, время работы которого и расходы по памяти линейно зависят от произведения мощности множеств вершин деревьев, между которыми вычисляется редактирующее расстояние.

Напомним, оценка редактирующего расстояния между структурами двух редакций одного и того же документа обеспечивает нам возможность задать критерий наличия изменений в очередной редакции документа на уровне его структуры.

Алгоритм работы модуля структурного анализа документа

1. В ходе очередной сессии робот ИПС загружает первую редакцию документа, последующие редакции которого требуется отслеживать на предмет наличия изменений. HTML-страница поступает на обработку в модуль ИПС. Последний выполняет построение так называемого «шаблона» этого документа на основе заданных по умолчанию подмножеств (1) HTML-тегов. Полученный таким образом шаблон можно рассматривать как описание по умолчанию для структуры анализируемого документа, своего рода «первое приближение».

2. Оператор ИПС редуцирует либо дополняет эту структуру:

путем уменьшения или, соответственно, расширения подмножества тегов, на основе которых она создается;

интерактивно, с помощью формы, которую ему предоставляет веб-интерфейс ИПС, указав связанные участки (подграфы, если представить структуру документа в форме графа), определив на структуре документа так называемые «семантические иерархии» (СИ). Под СИ мы будем понимать подграф в иерархической структуре документа, вершины которого содержат информацию, релевантную конкретным информационным потребностям. Другими словами, именно эта часть документа содержит данные (числовые, текст), в отношении которых далее будет осуществляться отслеживание семантических изменений. Например, анализ первой редакции некоторого документа, порожденного с помощью одного из популярных wiki-движков (www.docuwiki.org), позволяет выделить в структуре этого документа СИ, обра-

зуемую тегами <H2> (заголовок раздела) и <DIV> с атрибутом CLASS=«LEVEL2» (текст раздела). Обращаем внимание, что в одной редакции документа может иметь место несколько непересекающихся вхождений подграфа, определяемого как СИ на структуре этого документа. Таким образом определяется подмножество структурных элементов документа, изменения в содержании которых следует проверять с заданной периодичностью. СИ, заданные вышеуказанным способом, позволяют естественным образом продолжить их в структуре новой редакции документа. При условии, что структура прежней редакции документа изоморфна поддереву его новой редакции, вначале идентифицируются вхождения СИ в этом поддереве, затем ищутся вхождения СИ в оставшейся части структуры новой редакции. Для решения этой задачи в ИПС выполняется преобразование скобочной записи дерева к виду:

HTML[(TAG)*].

Грамматика:

TAG:=HTMLTAG#[{ATTRIBUTESLIST}][(TAG_CHILD)*]

ATTRIBUTESLIST:=(ATTRIBUTE_NAME=«ATTRIBUTE_VALUE»)*

TAG_CHILD:=TAG,

где метасимволы: [.] – необязательный элемент; (.* – произвольное (в т. ч. равное нулю) число вхождений элемента в скобках; # – порядковый номер HTML-тега в списке дочерних тегов (того же типа) его родителя; HTMLTAG, ATTRIBUTE_NAME, ATTRIBUTE_VALUE – термы.

Например, документ, который содержит только одну таблицу из двух строк и двух столбцов, будет (в простейшем случае) представлен так:

HTMLBODYTABLETR0TD0TD1TR1TD0TD1

Если оператор расширил множество HTML-тегов, включив в него атрибуты BORDER и ID тега TABLE:

HTMLBODYTABLE{ID=«1»;
BORDER=«2»;}TR0TD0TD1TR1TD0TD1 (6)

Указав в структуре этой же редакции документа в качестве СИ принадлежащую ему таблицу, оператор сохраняет в БД ИПС формальное определение СИ в виде строки:

TABLE{ID=«1»; BORDER=«2»; }
R0TD0TD1TR1TD0TD1 (7)

Тогда идентификация вхождений СИ в структуре документа сводится к поиску в строке вида

(6) непересекающихся подстрок вида (7).

После этого оператор может инициировать процесс мониторинга. Далее его вмешательство потребуется лишь в том случае, когда очередная редакция документа будет отличаться от шаблона.

Оперируя понятием метрики, заданной нами на структурах, вмешательство оператора будет необходимо в случаях, когда расстояние между структурой новой редакции документа и структурой, определенной шаблоном, окажется отличным от порогового (в наших исследованиях мы обычно принимали его равным нулю), и когда структура новой редакции не содержит поддерева, изоморфного структуре прежней редакции документа (шаблона).

Оперируя понятием метрики (скажем «векторная модель») на содержании элементов СИ, его вмешательство потребуется, когда расстояние до центра кластера, образованного множеством векторов, полученных по итогам лексического анализа всех предыдущих редакций документа превышает пороговую величину. Для первых k редакций в качестве центра кластера может быть указан вектор, построенный на основе подмножества лексем словаря ИПС. При построении этого вектора лексемы выбираются из словаря так, чтобы по возможности отражать терминологию релевантной предметной области.

3. Последующие загрузки новых редакций этого документа предполагают уже следующий порядок его обработки.

3.1. На вход модуля ИПС поступает новая редакция, загруженная роботом, а также идентификатор шаблона, созданный оператором на этапе инициации мониторинга этого документа. По идентификатору из БД ИПС извлекается сам шаблон (в формате, который поддерживается ИПС: скобочная запись либо XML) и подмножество HTML-тегов, на основе которых он был построен.

3.2. На основе вышеуказанного подмножества HTML-тегов модуль формирует описание структуры новой, загруженной редакции этого документа (в одном из форматов, поддерживаемых ИПС).

3.3. Вычисляется расстояние (5) между структурой, полученной на шаге 3.2, и структурой, хранящейся в шаблоне.

3.3.1. Если вычисленное расстояние (5) равно нулю, то выполняется редукция части документа, которую оператор отметил в шаблоне как нерелевантную, а содержание релевантных элементов



текста (СИ) документа обрабатывается с использованием векторной модели документа.

3.3.2. Если вычисленное редактирующее расстояние (5) отлично от нуля, то модуль пытается найти в структуре новой редакции поддерево, изоморфное структуре шаблона.

3.3.2.1. Если такое поддерево обнаружено, то аналогично выполняется его редукция для выделения СИ и обработки элементов СИ на основе векторной модели. Затем на оставшейся части структуры новой редакции модуль пытается продолжить СИ, т. е. отыскать подстроки вида (7), и, в случае успеха, сохранить содержимое таких элементов в БД ИПС с пометкой «новые данные». Затем структура новой редакции автоматически замещает прежний шаблон для этого документа в БД ИПС.

3.3.2.2. Если в структуре новой редакции документа не обнаружено поддерево, изоморфное структуре шаблона, то модуль сохраняет новую редакцию документа в БД ИПС целиком, помечая эту запись, т. е. устанавливая атрибут «Требуется вмешательство оператора».

Автоматическое отслеживание изменений на примере сайта GISMETEO

В качестве примера рассмотрим результаты, полученные по итогам мониторинга изменений на странице <http://gismeteo.ru/city/daily/4079/>. По исходному HTML-коду страницы, соответствующей этому URL, модулем ИПС был автоматически построен шаблон, который затем сократил оператор, указав, что для его построения следует использовать только подмножество HTML-тегов: HTML, BODY, DIV, TABLE, TR, TD. Скобочная запись (в виду ограниченности объема публикации мы не используем XML), определяющая структуру этого документа, имеет следующий вид:

```
(0(1361(1360(1235(1228)(1230(1229))
(1232(1231))(1234(1233)))(1248(1236)(1237)(1238)
(1239)(1240)(1241)(1242)(1243)(1244)(1245)(1246)
(1247))(1262(1249)(1250)(1251)(1252)(1253)(1254)
(1255)(1256)(1257)(1258)(1259)(1260)(1261))
(1276(1263)(1264)(1265)(1266)(1267)(1268)(1269)
(1270)(1271)(1272)(1273)(1274)(1275))(1290(1277)
(1278)(1279)(1280)(1281)(1282)(1283)(1284)(1285)
(1286)(1287)(1288)(1289))(1304(1291)(1292)(1293)
(1294)(1295)(1296)(1297)(1298)(1299)(1300)(1301)
(1302)(1303))(1318(1305)(1306)(1307)(1308)(1309)
(1310)(1311)(1312)(1313)(1314)(1315)(1316)
(1317))(1345(1320(1319))(1322(1321))(1324(1323))
```

```
(1326(1325))(1328(1327))(1330(1329))(1332(1331))
(1334(1333))(1336(1335))(1338(1337))(1340(1339))
(1342(1341))(1344(1343)))(1359(1346)(1347)(1348)
(1349)(1350)(1351)(1352)(1353)(1354)(1355)(1356)
(1357)(1358)))))
```

где числа – идентификаторы записей БД ИПС, которые хранят содержимое соответствующего тега–контейнера. Например, 1290 – это тег TR (строка таблицы), содержащая сведения о температуре, а сами прогнозируемые значения температур содержатся в тегах–контейнерах TD: 1277, 1278, 1279, 1280, 1281, 1282, 1283, 1284, 1285, 1286, 1287, 1288, 1289.

Спустя несколько месяцев владельцы сайта изменили дизайн и, тем самым, структура документа изменилась. Это изменение было выявлено модулем ИПС, который записал новую версию страницы в БД ИПС с атрибутом записи «Требуется вмешательство оператора». Повторное построение шаблона с участием оператора ИПС позволило продолжить мониторинг этого документа на основе другого шаблона. Скобочная запись, определяющая структуру новой редакции документа, теперь имеет следующий вид:

```
(0(1612(1611(1363(1362))(1378(1365(1364))
(1367(1366))(1369(1368))(1371(1370))(1373(1372))
(1375(1374))(1377(1376)))(1389(1379)(1381(1380))
(1382)(1383)(1384)(1386(1385))(1387)(1388))
(1400(1390)(1392(1391))(1393)(1394)(1395)
(1397(1396))(1398)(1399))(1411(1401)(1403(1402))
(1404)(1405)(1406)(1408(1407))(1409)(1410))
(1422(1412)(1414(1413))(1415)(1416)(1417)
(1419(1418))(1420)(1421))(1424(1423))(1426(1425))
(1441(1428(1427))(1430(1429))(1432(1431))
(1434(1433))(1436(1435))(1438(1437))(1440(1439))
(1452(1442)(1444(1443))(1445)(1446)(1447)
(1449(1448))(1450)(1451))(1463(1453)(1455(1454))
(1456)(1457)(1458)(1460(1459))(1461)(1462))
(1474(1464)(1466(1465))(1467)(1468)(1469)
(1471(1470))(1472)(1473))(1485(1475)(1477(1476))
(1478)(1479)(1480)(1482(1481))(1483)(1484))
(1487(1486))(1489(1488))(1504(1491(1490))
(1493(1492))(1495(1494))(1497(1496))(1499(1498))
(1501(1500))(1503(1502)))(1515(1505)(1507(1506))
(1508)(1509)(1510)(1512(1511))(1513)(1514))
(1526(1516)(1518(1517))(1519)(1520)(1521)
(1523(1522))(1524)(1525))(1537(1527)(1529(1528))
(1530)(1531)(1532)(1534(1533))(1535)(1536))
(1548(1538)(1540(1539))(1541)(1542)(1543)
(1545(1544))(1546)(1547))(1550(1549))(1552(1551))
(1567(1554(1553))(1556(1555))(1558(1557))
(1560(1559))(1562(1561))(1564(1563))(1566(1565))
(1578(1568)(1570(1569))(1571)(1572)(1573)
```

(1575(1574))(1576(1577))(1588(1579)(1581(1580))
(1582)(1583)(1584)(1585)(1586)(1587))(1598(1589)
(1591(1590))(1592)(1593)(1594)(1595)(1596)(1597))
(1608(1599)(1601(1600))(1602)(1603)(1604)(1605)
(1606)(1607))(1610(1609))))).

Она была построена оператором на основе того же подмножества HTML-тегов {HTML, BODY, DIV, TABLE, TR, TD}.

В нашей поисковой машине, ввиду того, что произошло кардинальное изменение структуры – одна только строка заголовка таблицы частично была вынесена за ее пределы в виде списка (теги LI), частично оформлена в виде меток строк, – потребовалось вмешательство оператора. К примеру, данные о температуре, которые в прежнем варианте структуры соответствовали тегу TR, ID 1290, и вложенных в него тегах TD (метка «Температура воздуха» в TD 1277, температуры – ID: 1278 – 1289), теперь оказались: сама метка «Температура воздуха» в TD 1369 (DIV 1368), а конкретные значения температуры в тегах TD 1383, 1394, 1405, 1416, 1446, 1457, 1468, 1479, 1509, 1520, 1531, 1542.

Невозможно использовать для динамического отслеживания изменений одни только алгоритмы на основе предварительно построенных иерархических семантик. Другими словами, автоматически построить биекцию данных на их основе не удастся, без привлечения дополнительных моделей и методов анализа содержимого тегов-контейнеров. Давайте обсудим следующую умоглядную ситуацию, которая в терминах модальной логики соответствует не «самому невероятному из возможных миров». В заголовках закладок, где сейчас явно указаны даты, владельцы сайта решили использовать существительные «сегодня», «завтра», «послезавтра». Такое изменение содержания уже не удастся динамически учесть при сохранении новых данных, если использовать лишь семантические модели типа «сущность–отношение». Потребуется привлечь референтные семантики на уровне словаря ИПС, не говоря уже о том, что интерпретация референций тоже потребует дополнительных усилий с учетом временных зон, времени суток и точки зрения владельцев сайта на эти атрибуты.

СПИСОК ЛИТЕРАТУРЫ

1. **Иванков, А.А.** Оценивание релевантности информационного ресурса при сборе информации в сети Internet с помощью информационно-поисковых систем. Краткий обзор современного состояния проблемы [Текст]/А.А. Иванков//Научные исследования и инновационная деятельность: матер. науч.-практ. конф.–СПбГПУ, 2008.–С. 202–207.
2. **Иванков, А.А.** Отслеживание семантических изменений в гипертекстовых документах. Модели, алгоритмы, опыт их применения [Текст]/А.А. Иванков //Научные исследования и инновационная деятельность: матер. науч.-практ. конф.–СПбГПУ, 2009.–С. 275–284.
3. **Иванков, А.А.** Об использовании ультраметрики, заданных на структуре Интернет-ресурса, для оценивания его релевантности [Текст]/А.А. Иванков, Д.С. Елисеев //Фундаментальные исследования и инновации в национальных исследовательских университетах: матер. XIV Всерос. конф. –СПбГПУ, 2010.–С. 88–89.
4. **Cutler, Michal** Using the Structure of HTML Documents to Improve Retrieval [Текст]/Michal Cutler, Shih Yungming, Meng Weiyi//Proc. of the USENIX symp. on Internet Technologies and Systems.–Monterey, California, Dec. 1997.
5. **Newby, G.** Information Space Based on HTML Structure [Текст]/G. Newby//NIST Special Publication 500-249: Proc. of The 9 Text REtrieval conf. (TREC 9).–Gaithersburg, Maryland, Nov. 13–16, 2000.
6. **Чеснавский, А.А.** Семантическое отслеживание изменений на веб-сайтах [Текст]/А.А. Чеснавский//Управление большими системами.–2007.–Вып. 19.–М.: ИПУ РАН.–С. 134–153.
7. **Чеснавский, А.А.** Семантическое отслеживание изменений на веб-сайтах [Текст]/А.А. Чеснавский//Информационные технологии.–2008.–№ 5.–С. 16–22.
8. **Philip, Bille** Tree edit distance, Alignment Distance & Inclusion [Текст]/ Bille Philip//Technical report TR-2003-23 in IT University Technical Report Series.–March 2003.
9. **Yuan, Wang X**-Diff: An Effective Change Detection Algorithm for XML Documents [Текст]/Yuan Wang, David J. DeWitt, Jin-yi Cai // Proc. of the 19th International conf. on Data Engineering. Bangalore, India.–March 5-8, 2003.–P. 519–530.
10. **Chawathe, Sudarshan S.** Comparing Hierarchical Data in External Memory [Текст]/Sudarshan S.Chawathe// Proc. of the 25th VLDB conf.–Edinburgh, Scotland, 1999.



УДК 004.75

Е.С. Сергеев, В.В. Гетманский, А.С. Горобцов

ПЕРЕНОС СИСТЕМЫ МНОГОТЕЛЬНОЙ ДИНАМИКИ НА ВЫЧИСЛИТЕЛЬНЫЙ КЛАСТЕР

На современном этапе развития программных систем инженерного анализа основной тенденцией является ориентация на междисциплинарные задачи моделирования, характеризующиеся одновременно наличием нескольких физических явлений и высокой степенью детализации моделей. Возрастает вычислительная сложность задач инженерного анализа и проектирования, и для их решения требуются все более значительные вычислительные ресурсы. Одно из решений этой проблемы заключается в использовании многопроцессорных систем.

Современным инструментом анализа машиностроительных конструкций являются программные системы, реализующие метод конечных элементов и метод моделирования динамики систем тел с кинематическими связями [1]. В данной статье рассмотрены возможные способы повышения эффективности анализа динамики систем тел при использовании кластерных вычислений. Программное обеспечение для проведения такого рода расчетов используется для анализа и доводки транспортных средств, расчета промышленных роботов и управляемых механизмов, получения алгоритмов управления, обеспечивающих устойчивое движение. При наличии программного средства для построения адекватной модели уменьшается время разработки и производственные риски за счет сокращения количества ошибок в проектных решениях.

При решении задач из рассматриваемой области методами математического моделирования возникают следующие проблемы:

высокая сложность и трудоемкость разработки инженерных моделей машиностроительных объектов;

ограничение на размерность расчетных схем задач многотельной динамики;

недостаточная адекватность многотельных моделей из-за их низкой детализации.

В данной статье описаны основные этапы модификации существующего программного комплекса моделирования динамики систем многих

тел ФРУНД [2] для эффективной работы на кластерных системах.

Программный комплекс ФРУНД реализует построение моделей систем тел со связями для двух общепринятых постановок: расчета динамики при наличии малых и больших движений тел. Малые движения имеют место при решении задач исследования колебаний и вибрации машин, большие движения – при моделировании динамики механизмов, управляемого движения машин. С помощью модулей системы можно получить решение уравнений математических моделей динамики конструкций машин и механизмов с учетом пространственного характера движений, упругих свойств отдельных тел, многомерности и нелинейности характеристик связей между телами, случайных и детерминированных нагрузок, динамического и кинематического способов возмущения.

Уравнение движения

Уравнение движения произвольной механической системы абсолютно твердых тел со связями имеет вид:

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{q}(\dot{\mathbf{x}}, \mathbf{x}, t), \quad (1)$$

где $\mathbf{x}$ – вектор переменных системы размерностью n ; $\mathbf{M}$ – матрица масс; $\mathbf{q}$ – вектор, содержащий внешние и неконсервативные силы при кинематических связях

$$\Phi(\dot{\mathbf{x}}, \mathbf{x}, t) = 0. \quad (2)$$

Система дифференциально-алгебраических уравнений в форме Лагранжа первого рода, используемая в рассматриваемом методе, имеет вид:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{x}} - \mathbf{D}^T \mathbf{p} = \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t); \\ \mathbf{D}\dot{\mathbf{x}} = \mathbf{h}(\dot{\mathbf{x}}, \mathbf{x}, t), \end{cases} \quad (3)$$

где $\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t)$ – вектор функция правых частей дифференциальных уравнений; $\mathbf{p}$ – вектор множителей Лагранжа; $\mathbf{D}$ – производная уравнений связи $\mathbf{D} = \Phi_{\dot{\mathbf{x}}}$; $\mathbf{h}(\dot{\mathbf{x}}, \mathbf{x}, t)$ – вектор правых частей уравнений связей

$$\mathbf{h}(\dot{\mathbf{x}}, \mathbf{x}, t) = -\dot{\Phi}_x \dot{\mathbf{x}} - \dot{\Phi}_t. \quad (4)$$

В соответствии с рассматриваемой в работе [3] расширенной постановкой Лагранжа, систему (3) можно заменить на

$$\begin{cases} \mathbf{M}\ddot{\mathbf{x}} - \mathbf{D}^T \alpha (2\mu\Omega^2 \dot{\Phi} + \Omega^2 \Phi) + \mathbf{D}^T \mathbf{p} = \mathbf{q}(\dot{\mathbf{x}}, \mathbf{x}, t), \\ \mathbf{D}\ddot{\mathbf{x}} = \mathbf{h}(\dot{\mathbf{x}}, \mathbf{x}, t). \end{cases} \quad (5)$$

где α – в общем случае диагональная матрица коэффициентов; μ – коэффициенты демпфирования; Ω – диагональная матрица собственных частот. Система (3) точно соответствует (1) т. к. $\dot{\Phi} = 0$ и $\Phi = 0$. Если добавить слагаемое вида $\mathbf{D}^T \alpha (2\mu\Omega^2 \dot{\Phi} + \Omega^2 \Phi)$ только для выбранных связей, то (5) можно привести к виду

$$\begin{cases} \mathbf{M}\ddot{\mathbf{x}} - \mathbf{D}_f^T \mathbf{p}_f + \mathbf{D}_r^T \mathbf{p}_r = \mathbf{q}(\dot{\mathbf{x}}, \mathbf{x}, t); \\ \mathbf{D}_r \ddot{\mathbf{x}} = \mathbf{h}_r(\dot{\mathbf{x}}, \mathbf{x}, t), \end{cases} \quad (6)$$

где $\Phi = (\Phi_f \ \Phi_r)^T$; $\mathbf{D}_r = \Phi_r$; $\mathbf{p}_f = \alpha(2\mu\Omega^2 \dot{\Phi}_f + \Omega^2 \Phi_f)$; Φ_f – частная производная уравнений связи по координатам податливых шарниров; Φ_r – частная производная по координатам кинематических связей; $\mathbf{p}_f$ – реакции в податливых связях; $\mathbf{p}_r$ – реакции в кинематических связях. При $\alpha \rightarrow \infty$, $(\mathbf{p}_f \ \mathbf{p}_r)^T \rightarrow \mathbf{p}$, пусть $\delta = (\mathbf{p}_f \ \mathbf{p}_r)^T$.

Решение задачи выбора $\mathbf{p}_f$ такого, которое обеспечивает минимальную величину максимальной невязки $\delta_{\max}$, приведенной в [4].

Уравнение (6) можно использовать для процедуры распараллеливания при решении задачи численного интегрирования уравнений динамики систем тел, при этом на каждом шаге интегрирования необходимо решать линейную систему алгебраических уравнений для определения множителей Лагранжа $\mathbf{p}_r$. В случае, если в системе нет жестких кинематических связей, слагаемое $\mathbf{D}_r^T \mathbf{p}_r$ в (6) пропадает и нет необходимости решения линейной задачи на каждом шаге интегрирования. В этом случае численное интегрирование системы (6) сводится к интегрированию системы обыкновенных дифференциальных уравнений (ОДУ) в форме Коши (матрицу масс $\mathbf{M}$ в большинстве случаев можно принимать диагональной). Ниже рассматриваются особенности программной реализации параллельного интегрирования системы ОДУ вида (7):

$$\mathbf{M}\ddot{\mathbf{x}} - \mathbf{D}_f^T \alpha (2\mu\Omega^2 \dot{\Phi}_f + \Omega^2 \Phi_f) = \mathbf{q}(\dot{\mathbf{x}}, \mathbf{x}, t). \quad (7)$$

В данной работе для численного интегрирования используется явный метод Рунге-Кутты 4-го

порядка. Существование условия сходимости этого метода дает преимущество над классом неявных методов интегрирования, хотя приходится использовать малый шаг интегрирования для жестких систем.

Архитектура системы

Рассмотрим более подробно систему ФРУНД (рис. 1). Как почти все комплексы моделирования систем тел со связями, ФРУНД состоит из модулей ядра и пользовательского интерфейса. Комплекс состоит из трех основных модулей: подсистемы синтеза уравнений движения, расчетного модуля, подсистемы анализа результатов. По геометрической расчетной схеме программный комплекс ФРУНД генерирует в символьном виде уравнения движения в виде программного кода на языке ФОРТРАН, который компилируется в расчетный модуль. Расчетный модуль в процессе выполнения сохраняет результаты расчета для подсистемы анализа. Результаты анализа отображаются в графическом интерфейсе.

Наличие отделенной от интерфейса подсистемы расчета позволяет дорабатывать программный комплекс за счет создания дополнительных модулей.

Для использования в комплексе ФРУНД кластерных вычислений разработаны дополнительные подсистемы:

подсистема декомпозиции модели на подмодели, которые могут рассчитываться независимо, как показано в [3];

синхронизации параллельного расчета, которая обеспечивает синхронизацию параллельных процессов на каждой итерации численного метода интегрирования уравнений движения;

объединения результатов расчета подмоделей.

Подсистема декомпозиции модели. Многотельная модель состоит из набора тел и связей между ними. Все зависимости в модели и ее структуру можно задать графом или матрицей смежности. Модели механических систем имеют преимущественно древовидную структуру графа, характеризующуюся малым числом циклов и разреженной матрицей смежности. Сортировка вершин графа позволяет получить портрет матрицы с концентрацией ненулевых элементов вблизи главной диагонали, т. е. матрица смежности приводится к ленточному виду. При декомпозиции моделей с такой структурой возникают связан-

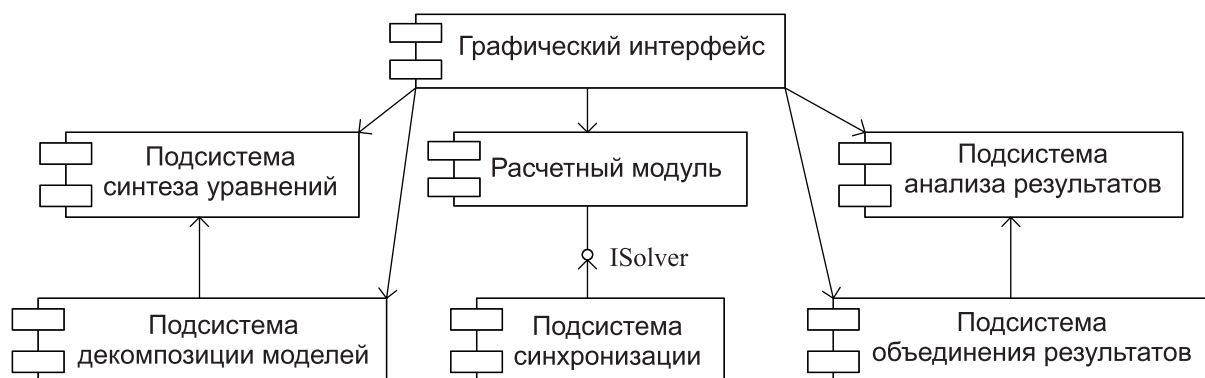


Рис. 1. Диаграмма компонентов комплекса ФРУНД с подсистемами для параллельных вычислений

ные подмодели, которые нельзя рассчитывать независимо из-за потери связей при декомпозиции. Система ОДУ (7) разделяется на несколько подсистем, которые зависят только по функциям правых частей. В результате декомпозиции формируются списки тел модели, входящие в каждую подмодель и списки граничных тел, соединенных с рассчитанными шарнирами (рис. 2). В рамках полученных подмоделей параметры граничных тел рассчитываются некорректно, требуется синхронизация параллельного расчета подмоделей. По этим спискам формируется таблица синхронизации для пересылок данных с использованием информации об адресах хранения данных подмоделей и полной модели в памяти. Это позволяет производить вычисления на каждой итерации независимо для каждой подмодели с последующей синхронизацией и обменом данными перед переходом к следующей итерации.

Подсистема синхронизации. Является одной из разработанных подсистем, необходимых для переноса системы инженерного анализа ФРУНД на кластерные вычислительные системы (см. рис. 1).

Подсистема синхронизации представляет собой MPI (Message Passing Interface) программу – множество одновременно выполняемых процессов. Каждый процесс управляет своим расчетным модулем, находящимся в динамической библиотеке, который загружается во время исполнения программы.

Для описания взаимодействия между процессами используется граф обмена, в котором каждая вершина представляет подмодель, а каждая дуга – пересылаемую информацию. Вершина графа может иметь различное количество входящих и исходящих дуг. При этом дугам графа в качестве весов назначены адреса данных в общей модели, которыми обмениваются расчетные модули.

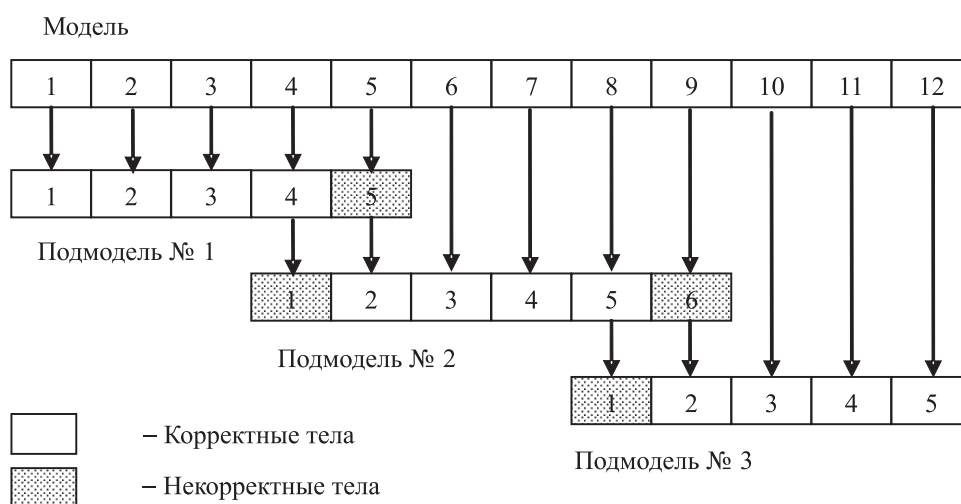


Рис. 2. Декомпозиция модели на 3 подмодели и перенумерация тел

В общем случае адреса, из которых считываются и в которые записываются данные, разные, поэтому требуется организация двух графов – для чтения и для записи. Информационный обмен между параллельными процессами осуществляется с использованием таблицы синхронизации. Подсистема основана на интерфейсе ISolver, под которым в данном случае понимается совокупность процедур, импортируемых в подсистему синхронизации из расчетного модуля. Управление процессом решения осуществляется вызовом процедур интерфейса расчетного модуля.

Для реализации подсистемы синхронизации были проанализированы два способа взаимодействия параллельных процессов.

1. С управляющим процессом: среди процессов параллельной программы выделяется процесс, который перераспределяет данные между параллельными процессами.

2. Взаимодействующие равные: процессы обмениваются данными напрямую друг с другом.

Поскольку подсистема реализована с помощью объектно-ориентированного подхода на языке C++, один и тот же набор классов может реализовывать два вышеупомянутых способа. Необходимая гибкость подсистемы достигается путем переноса методов по иерархии классов и изменением порядка вызова методов.

Для построения теоретических оценок трудоемкости коммуникационных операций используется способ вычисления временных затрат на передачу данных. Трудоемкость операции коммуникации между двумя процессорными узлами кластера оценивается в соответствии с моделью, предложенной Хокни (the Hockney model) [5]:

$$t_{\text{ид}}(m) = \alpha + m/\beta. \quad (8)$$

Время на пересылку сообщения размером m между двумя узлами определяется выражением (8), где α – латентность передачи каждого сообщения; β – пропускная способность сети.

Важно отметить, что для предварительного анализа временных затрат на выполнение коммуникационных операций точности модели Хокни может оказаться достаточно. Кроме того, данная модель имеет наиболее простой вид среди существующих моделей [6].

Рассмотрим более подробно способы реализации подсистемы синхронизации с учетом использования MPI.

Первый способ взаимодействия предполагает наличие управляющего процесса, назначение которого заключается в сборе информации со всех процессов в группе, подготовке полученных данных к пересылке и пересылке всем процессам группы предназначенных им данных. Подобное взаимодействие можно организовать с помощью коллективных операций – MPI_Gatherv и MPI_Scatterv, реализующих сбор и распределение сообщений. Операция *scatter* распределяет части данных с одного узла на все узлы. Корневой узел должен переслать все данные остальным, сетевой интерфейс корневого узла является «бутылочным горлышком». Операция *gather* собирает части данных со всех узлов на один корневой узел. Поскольку указанные операции ограничены пропускной способностью сетевого интерфейса одного узла [7], трудоемкость коммуникационных операций можно оценить так:

$$\begin{aligned} t_{\text{scatter}}(m) &= \alpha + m/\beta, \\ t_{\text{gather}}(m) &= \alpha + m/\beta. \end{aligned} \quad (9)$$

Тогда трудоемкость коммуникационных операций синхронизации процессов на одной итерации численного метода:

$$t_{\text{ScatterGather}}(m) = 2 \times [\alpha + m/\beta] + t_p, \quad (10)$$

где t_p – время обработки данных на управляющем процессе.

Второй способ взаимодействия параллельных процессов предполагает передачу данных от всех процессоров всем процессорам сети на каждой итерации численного метода. Возможный способ реализации операции множественной рассылки состоит в выполнении соответствующего набора операций одиночной рассылки. Однако такой подход не является оптимальным для многих топологий сети, поскольку часть необходимых операций одиночной рассылки потенциально может быть выполнена параллельно.

Операция множественной рассылки эквивалента выполнению всеми процессами операции *scatter* на локальных буферах. Следовательно, производительность коммуникационного взаимодействия будет ограничена производительностью сетевого интерфейса одного узла. Подобное коллективное взаимодействие можно организовать с помощью коллективных операций – MPI_Alltoall. Учитывая вышеизложенные дово-

ды, трудоемкость коммуникационных операций определяется следующим выражением:

$$t_{\text{AlltoAll}}(m) = \alpha + m/\beta. \quad (11)$$

Из анализа соотношений (10) и (11) следует, что способ взаимодействия с управляющим процессом, как минимум, в два раза проигрывает по затратам времени на коммуникационные операции, поэтому для обмена данными между процессами использован второй способ взаимодействия.

Условно подсистему синхронизации можно разбить на две логически разделенные части.

Первая часть используется для инициализации подсистемы синхронизации и отвечает за обработку данных, полученных от подсистемы декомпозиции. На основе этих данных вычисляются зависимости между подмоделями, и строятся списки объектов MPI_type для упаковки и распаковки пересылаемых данных. Они не могут использоваться ни в каких других операциях, кроме коммуникационных. Производные типы MPI следует понимать как описатели расположения в памяти элементов базовых типов, описывающих две вещи: последовательность базовых типов и последовательность смещений.

Вторая часть применяется для управления расчетным модулем и обмена данными в процессе вычислений. Она отвечает за подготовку данных расчетного модуля для их передачи с помощью функций MPI.

Подсистема объединения результатов расчета. Среда выполнения параллельной программы MPI принципиально отличается от среды выполнения обычной программы. В последовательном варианте результаты вычислений находятся в адресном пространстве одного процесса, что позволяет записывать данные на диск простым и удобным способом для дальнейшего анализа и обработки. В системах с распределенной оперативной памятью части результата вычислений распределены

между множеством процессов. Распределенность сохраняемых данных может вызвать значительное снижение производительности при записи данных в файл, т. к. данные записываются на общий сетевой диск несколькими процессами одновременно. В случае параллельного расчета производительность существенно зависит от реализации операций записи данных на диск.

Список тел в подмодели, рассчитываемой в отдельном процессе, включает внутренние тела, параметры которых рассчитываются правильно, и граничные тела, параметры которых рассчитываются неправильно. Параметры двух групп тел находятся в смежных ячейках памяти (рис. 3), что вызывает значительные трудности при обработке и сохранении данных, которые необходимо отобразить в последовательный файл.

Условное деление тел на две группы происходит путем анализа выходных данных подсистемы декомпозиции. В выходных файлах указывается соответствие между номерами тел полной модели и номерами тел подмодели, а также номера корректных тел для каждой подмодели. На основе этих данных строятся производные типы MPI.

Для реализации подсистемы объединения результата рассматривались два способа [8] записи данных в общий файл:

вложенных последовательностей (interleaved global sequentiality) на основе настраиваемого отображения файлов (MPI_File_set_view);

раздельных последовательностей (global partitioned sequentiality) на основе неблокирующего доступа к данным с точными смещениями.

Использование функции MPI_File_set_view позволяет настраивать области видимости файла отдельно для каждого процесса, участвующего в вычислениях (рис. 4, а), что дает возможность записывать параметры тел в порядке их нумерации в полной модели. Данный порядок записи используется последовательным расчетным модулем и подсистемой анализа результатов моде-

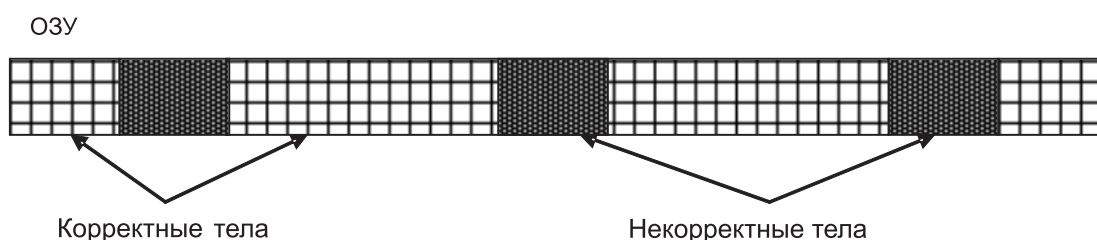


Рис. 3. Общий случай расположения параметров корректных и некорректных тел в ОЗУ процесса

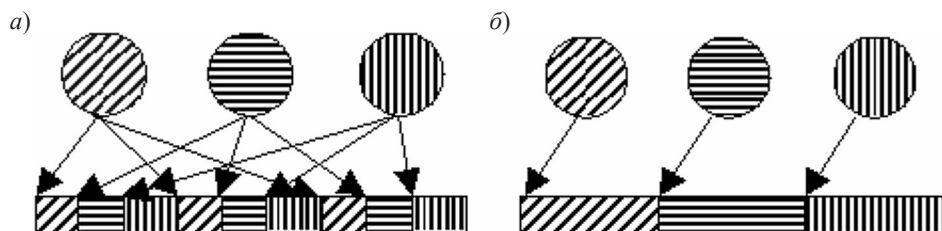


Рис. 4. Способы записи в общий файл

лирования. Отображение файла настраивается с помощью производных типов MPI. Использование MPI_File_set_view обладает существенным недостатком: значительным уменьшением производительности операций ввода–вывода [9]. Это связано с особенностями реализации записи в отображаемый файл.

После проведения серии тестов был выбран способ раздельных последовательностей (рис. 4, б) на основе неблокирующей функции MPI_iwrite_at, превосходящий по производительности способ вложенных последовательностей в 10-12 раз, что соответствует результатам, изложенным в [9]. Способ раздельных последовательностей создает файл, не совместимый с форматом файла при последовательном расчете, поэтому для анализа результатов данные предварительно конвертируются.

Результаты экспериментов

Тесты проводились на кластере из семи узлов (headnode, node01 ... node06). Каждый узел содержит два процессора Xeon Quad Core с рабочей частотой 2,5 ГГц, размер оперативной памяти 16 Гб. Узлы соединены сетью Gigabit Ethernet. Использовалась ОС Windows HPC Server 2008.

Целью экспериментов было сравнение времени моделирования одних и тех же многотельных моделей при последовательном и параллельном расчете.

В таблице представлены результаты вычислительных экспериментов моделирования многотельных систем в зависимости от количества про-

цессоров и количества тел в системе. Поскольку вычисления проводились на восьмиядерных вычислительных узлах, количество процессоров кратно восьми; если выделять процессоры неравномерно, производительность снижается. Процессоры на одном узле обмениваются сообщениями через общую память, что значительно быстрее, чем обмен через сетевой интерфейс.

Если количество тел в моделируемой системе является фиксированным, то при росте числа процессоров эффективность убывает за счет роста накладных расходов. Эффективность использования процессоров улучшается с увеличением числа тел в моделируемой системе и повышением сложности решаемой задачи. В большинстве случаев можно обеспечить определенный уровень эффективности моделирования в системе ФРУНД при увеличении детализации моделей.

В результате переноса системы ФРУНД на вычислительный кластер были получены результаты по ускорению расчета моделей систем тел, свидетельствующие о сохранении масштабируемости при увеличении размерности задачи, которая определяется количеством неизвестных в интегрируемой системе уравнений. Дальнейшее развитие проекта предполагает определение границ масштабируемости и исследование влияния характеристик размерности задачи (количества тел в модели и подмоделях, количество подмоделей, количество связей между подмоделями) на эффективность параллельного расчета. Исследованные постановки задач многотельной

Результаты вычислительных экспериментов моделирования многотельных систем

Количество тел в многотельной модели	1 процессор	16 процессоров		24 процессора		32 процессора	
	T,с	T,с	Ускорение	T,с	Ускорение	T,с	Ускорение
2000	200	42	4,76	34	5,88	30	6,67
4000	410	93	4,41	57	7,19	45	9,11
6500	800	203	3,94	99	8,08	70	11,43



динамики характеризуются слабой связанностью подмоделей, поэтому для моделирования целесобразно использовать GRID-вычисления. Эксперименты по получению результатов в GRID-среде планируются на дальнейших этапах разработки программного комплекса параллельного расчета

динамики систем тел. Предполагается разработка модуля декомпозиции сложных многотельных моделей для эффективного использования распределенных ресурсов GRID-среды. Это позволит осуществлять междисциплинарное моделирование с высокой детализацией.

СПИСОК ЛИТЕРАТУРЫ

1. **Виттенбург, И.С.** Динамика систем твердых тел [Текст]/И.С. Виттенбург.—М.: Мир.—1980.—294 с.
2. Официальный сайт пакета ФРУНД [Электронный ресурс].—2009.—Режим доступа: <http://frund.vstu.ru>.
3. **Ваю, Е.** A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems [Текст]/Е. Ваю, J. Garcia de Jalon, M.A. Serna//Computer Methods in Applied Mechanics and Engineering.—1988.—Vol. 71.—Iss. 2.—P. 183–195.
4. **Банах, Л.Я.** Условия разбиения системы дифференциально-алгебраических уравнений на слабосвязанные подсистемы [Текст]/Л.Я. Банах, А.С. Горобцов, О.К. Чесноков//Журнал вычислительной математики и математической физики.—2006.—№ 12.—С. 2223–2227.
5. **Hockney, R.** The communication challenge for MPP: Intel paragon and meiko CS-Parallel computing [Текст]/R. Hockney//Parallel Computing archive.—1994.—Vol. 20. Iss. 3.—P. 389–398.
6. **Pjesivac-Grbovic, J.** Performance analysis of MPI collective operations [Текст]/J. Pjesivac-Grbovic, T. Angskun, G. Bosilca [et al.]//Cluster Computing.—2007.—Vol. 10.—P. 127–143.
7. **Matsuda, M.** The design and implementation of MPI collective operations for clusters in long-and-fast networks [Текст]/ M. Matsuda, T. Kudoh, Y. Kodama [et al.]//Cluster Computing.—2008.—Vol. 11.—Iss. 1.—P. 45–55.
8. **Lucas, H.V.** Hint Controlled Distribution with Parallel File Systems [Текст]/H.V. Lucas, T. Ludwig//European PVM/MPI Users Group Meeting.—2005.—P. 110–118.
9. **Worringen, J.** Improving generic non-contiguous file access for MPI-IO [Текст]/J. Worringen, J.L. Traft, H. Ritzdorf//Lecture Notes in Computer Science, Recent Advances in Parallel Virtual Machine and Message Passing Interface.—2003.—P. 309–318.

УДК 004.4'23, 004.4'24

В.М. Ицыксон, Д.А. Тимофеев

ТЕХНОЛОГИЯ МОДИФИКАЦИИ ПРОГРАММНОГО КОДА, ОСНОВАННАЯ НА ПАРАМЕТРИЗУЕМЫХ ШАБЛОНАХ

Различные модификации исходного кода — одни из самых частых действий разработчиков во время всего жизненного цикла программной системы. Целями модификации могут являться внесение новой функциональности, исправление ошибок, улучшение читаемости, повышение производительности, переход на другие библиотеки и т. п. В общем случае эту задачу сложно автоматизировать, поэтому разработчики обычно решают ее вручную. Она усложняется еще и тем, что сравнительно небольшие локальные изменения могут повлечь за собой необходимость обработки большого количества зависимого программного кода.

Если модификация программного кода представляет собой изменения с сохранением поведения (рефакторинг), то задача решается относительно легко с привлечением современной среды разработки, в которой данные модификации поддерживаются и автоматизированы. Однако часто требуется вносить изменения, которые не могут быть выражены набором шаблонов рефакторинга. Кроме того, их иногда необходимо производить многократно в разных участках программы. Примером таких изменений может быть однотипная оптимизация циклов в программе.

Для автоматизации подобных операций программисту необходимо средство, позволяющее описать изначальное состояние программы до модификации и правила проведения преобразования. Существующие среды разработки таких средств не предоставляют.

Одним из способов решения указанной проблемы может быть использование существующих универсальных средств для различного рода трансформаций кода, таких как Stratego/XT [1], ASF+SDF [2], TXL [3] и т. п. Эти системы являются универсальными преобразователями структурированных текстов, управляемыми грамматикой. С их помощью можно описать довольно сложные преобразования программ, написанных на различных языках программирования.

Однако, на наш взгляд, им присущи существенные недостатки:

сложность языка трансформаций;

необходимость повторного описания грамматики целевого языка в терминах, определяемых используемым средством трансформации;

сложность интеграции с существующими средами разработки с сохранением необходимого уровня интерактивности;

необходимость описания операций преобразования программы в терминах ее модельного представления (например, в терминах абстрактного синтаксического дерева (АСД) или грамматики целевого языка);

трудоемкость и затратность внедрения в текущий процесс разработки программного обеспечения.

Предлагаемый подход. В данной статье описывается подход, упрощающий процесс описания и осуществления модификаций программного кода, ориентированный на конечного разработчика, не являющегося экспертом в области теории языков программирования. В основе этого подхода лежит

язык шаблонов, позволяющий описывать параметризуемые шаблоны кода «до» и «после» модификации. На основе данных шаблонов генерируется список операций, необходимых для модификации программного кода. Поиск участков кода, для которых необходимо произвести изменения, осуществляется с использованием шаблона «до». После завершения поиска и изменения необходимых участков, происходит запись программного кода в форматированном виде обратно в файл. На рис. 1 представлена общая схема предлагаемого подхода.

Ниже более подробно рассматриваются основные этапы осуществления модификаций кода, применяемые в этом подходе.

Язык шаблонов. Указанные выше инструменты, такие как Stratego, TXL и т. п., используют сложный язык описания трансформаций. Для того, чтобы упростить работу разработчика, необходимо создать декларативный язык описания модификаций программного кода, который является максимально близким к целевому языку программирования. В таком случае синтаксис языка трансформаций в основных чертах повторяет синтаксис языка программирования. Благодаря этому он более легок в изучении и применении.

В качестве основных требований к данному языку можно выделить следующие:

необходимость оперирования шаблонами «до» и «после» модификации;

возможность параметризации шаблонов;

наличие элементов регулярных выражений для осуществления более гибкого поиска участков кода;

иерархичность языка – не должно быть ограничений на количество вложенных конструкций.

Выбор в пользу использования шаблонов «до» и «после» обоснован тем, что в данном случае возможно осуществить автоматическое определение списка операций, необходимых для при-

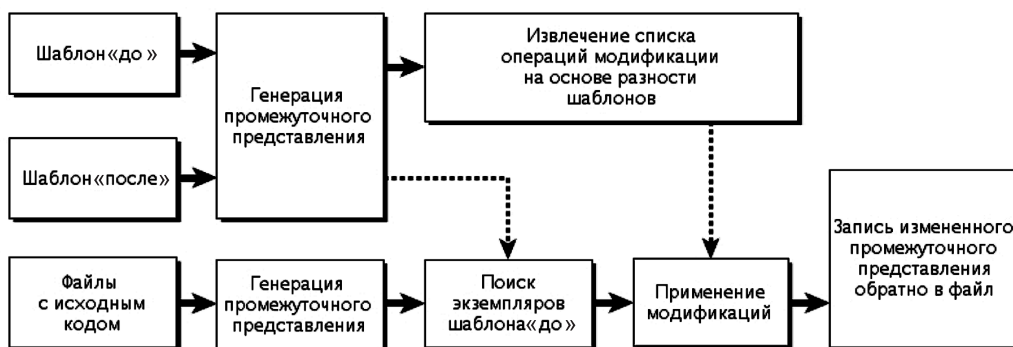


Рис. 1. Общая схема подхода



ведения шаблона «до» к виду шаблона «после». Этот список впоследствии может быть использован для преобразования участков кода, найденных на основе шаблона «до».

В статье описывается разработанный язык шаблонов для языка программирования Java 1.5. Язык Java выбран как один из самых распростра-

Шаблон «до»:

```
for( #var; #cond; #upd) {  
    #expr;  
}
```

Листинг 1: Пример параметризованных шаблонов для преобразования

Используя такие шаблоны можно за одну операцию трансформировать все операторы «for» в «while» и обратно.

Шаблон «до»:

```
for( #var; #cond; #upd) {  
    if(true) {  
        #expr;  
    }  
}
```

Листинг 2: Пример шаблонов для оптимизации

Как видно из рассмотренных примеров, разработанный язык шаблонов является достаточно прозрачным и простым в изучении.

Параметры в приведенных выше шаблонах выступают в роли элементов-«заполнителей», которые во время поиска заполняются найденными конструкциями языка программирования. Изменение местоположения параметра в шаблоне «после» позволяет указать место перемещения

```
#EXPR:1;      - количество конструкций языка = 1;  
#EXPR:1-5;    - количество конструкций языка от 1 до 5;  
#EXPR:1-;     - количество конструкций языка больше 1;
```

Листинг 3: Примеры настраиваемого параметра

Для того, чтобы сделать шаблоны модификации более универсальными, в создаваемый язык введены элементы, позволяющие находить несколько интересных конструкций в исходном

```
#method() {  
    #EXPR;  
    if(a < b) {  
        this.a = b;  
    }  
    #EXPR;  
}
```

Листинг 4: Пример шаблона для поиска методов

ненных в настоящее время языков программирования. Формальная спецификация языка шаблонов не приведена из-за ограниченного объема статьи.

В листинге 1 дан пример параметризованных шаблонов для преобразования цикла «for» в цикл «while».

Шаблон «после»:

```
#var;  
while(#cond) {  
    #expr;  
    #upd;  
}
```

В листинге 2 приведены шаблоны для оптимизации цикла «for» в том случае, если условие выражения «if» является константным.

Шаблон «после»:

```
if(true) {  
    for(#var; #cond; #upd) {  
        #expr;  
    }  
}
```

данных конструкций, а отсутствие параметра – необходимость удаления. Возможность использования элементов регулярных выражений при описании параметров позволяет увеличить гибкость поиска. В листинге 3 приведены примеры настраиваемого параметра. В данном случае ограничение на число конструкций, которыми может владеть параметр, настраивается с помощью числовых аргументов, задаваемых после его имени.

В качестве примера в листинге 4 приведен шаблон для поиска всех методов в классе, описанном на языке Java.

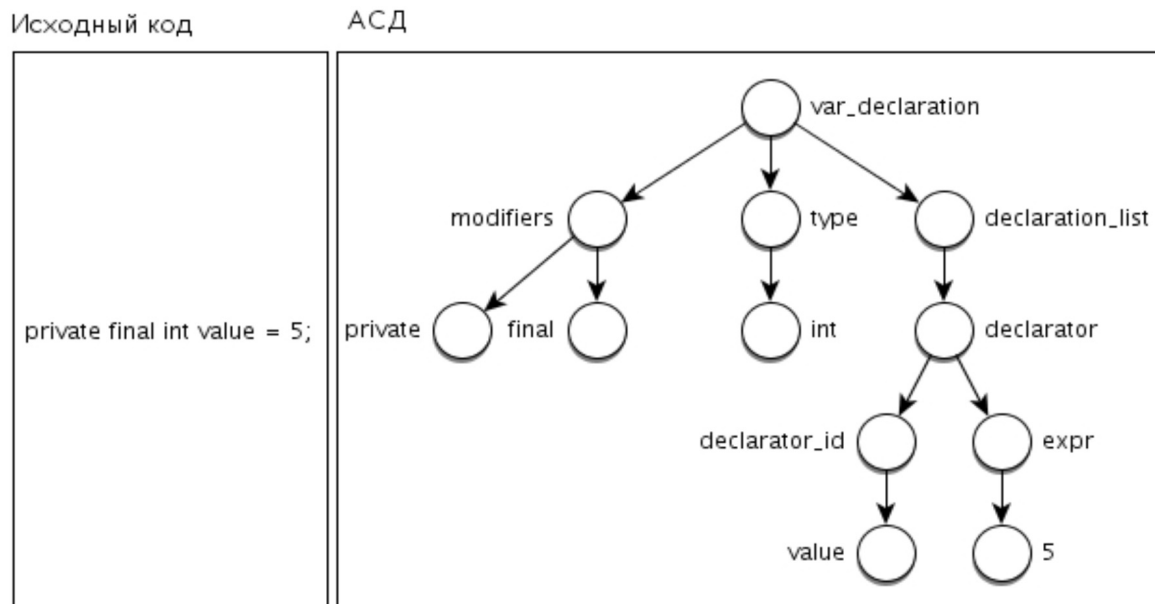


Рис. 2. АСД участка кода, отвечающего за объявление переменной

Модельное представление программы. Для осуществления операций поиска необходимых участков кода и их последующей модификации необходимо выбрать промежуточное представление программного кода, наиболее пригодное для проведения поиска и модификаций. В качестве основного представления было выбрано абстрактное синтаксическое дерево [4] (далее – АСД) в силу следующих его свойств:

расширяемость – в случае необходимости в АСД может быть включена дополнительная информация (например, различные семантические свойства, комментарии и т. п.);

эффективность – существуют эффективные алгоритмы обработки деревьев;

адаптивность – возможность настраивания уровня детализации при генерации АСД.

На рис. 2 представлен пример возможного АСД для участка кода, отвечающего за объявление переменной.

Еще одним достоинством древовидного представления является большое число свободно распространяемых парсеров, управляемых грамматикой языка, позволяющих автоматизировать процедуру построения АСД.

Модельное представление шаблонов. В качестве промежуточного представления шаблонов также используется АСД. Для его построения используется парсер, который создается на основе

расширенной грамматики целевого языка программирования. В результирующем АСД имеются узлы двух типов:

простые узлы, соответствующие конструкциям целевого языка программирования (например, «for» или «if»);

параметрические узлы, соответствующие шаблонным элементам (например, «#var» или «#expr:1»).

Так как синтаксис языка шаблонов близок к синтаксису целевого языка, при реализации можно применять единый парсер, используемый для формирования и АСД исходной программы, и АСД шаблонов.

На рис. 3 представлено АСД, построенное для шаблона, позволяющего выбрать определенную конструкцию объявления переменной независимо от инициализирующей конструкции.

Автоматическое определение операций модификации. Для того чтобы этап получения списка операций, необходимых для модификации программного кода, осуществлялся в автоматическом режиме, можно применить алгоритм определения изменений в иерархических структурах данных, описанный в [5].

Данный алгоритм работает с упорядоченными деревьями, в которых узлы поименованы (имеют идентификатор типа узла) и имеют значения. Принимая на вход дерево «до» и «после» изменений,

Исходный код шаблона

```
private final int value = #initializer;
```

АСД шаблона

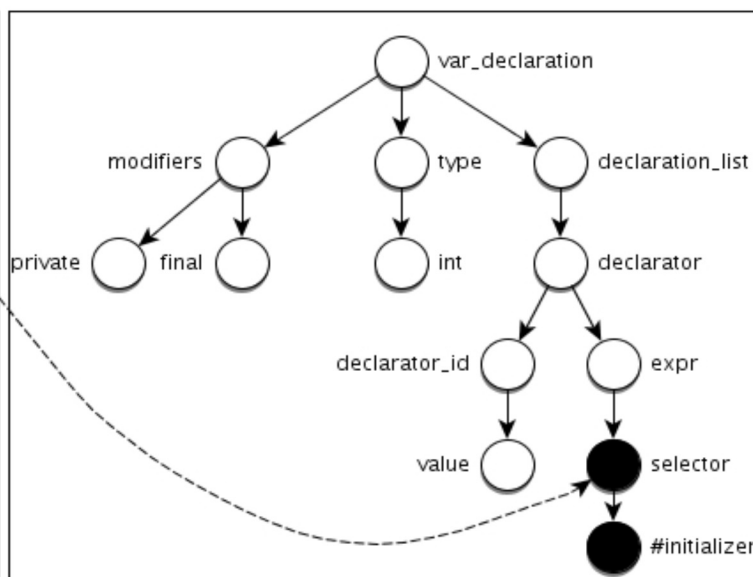


Рис. 3. АСД шаблона, позволяющего выбирать определенную конструкцию объявления переменной

алгоритм на выходе генерирует последовательность операций удаления узлов, перемещения узлов, добавления новых узлов или обновления значений узлов. Эти операции полностью покрывают совокупность изменений, который могут произойти в программном коде, представленном в виде АСД.

На рис. 4 представлены четыре типа операций, которые извлекаются с использованием алгоритма получения дифференциальной разности деревьев:

Insert. Операция вставки нового узла. Узел G

добавляется на первую позицию списка дочерних узлов узла B.

Move. Операция перемещения узла. Узел D перемещается на вторую позицию списка дочерних узлов узла B.

Update. Операция обновления значения узла. Значение «Value» узла C изменяется на значение «NewValue».

Delete. Операция удаления узла. Узел E удаляется из списка дочерних узлов узла C.

Полное описание алгоритма рассмотрено в [5].

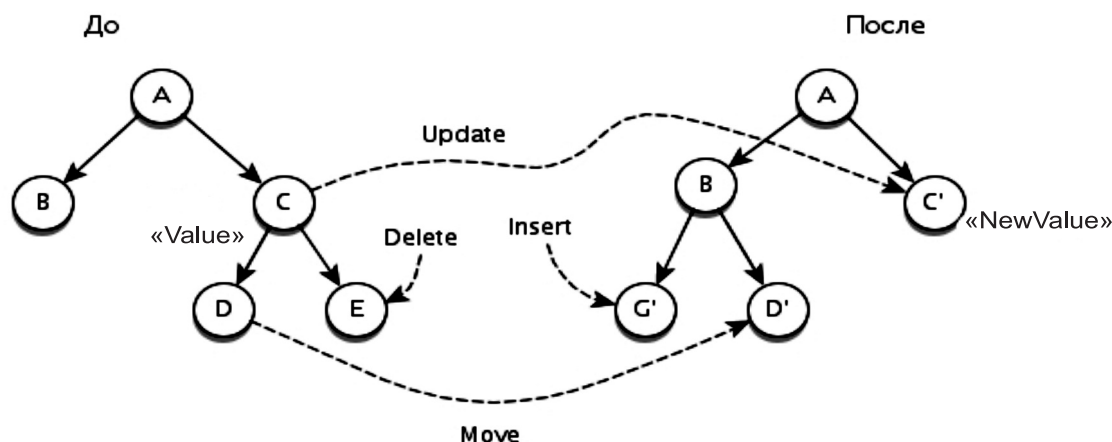


Рис. 4. Операции, извлекаемые с использованием алгоритма получения дифференциальной разности деревьев

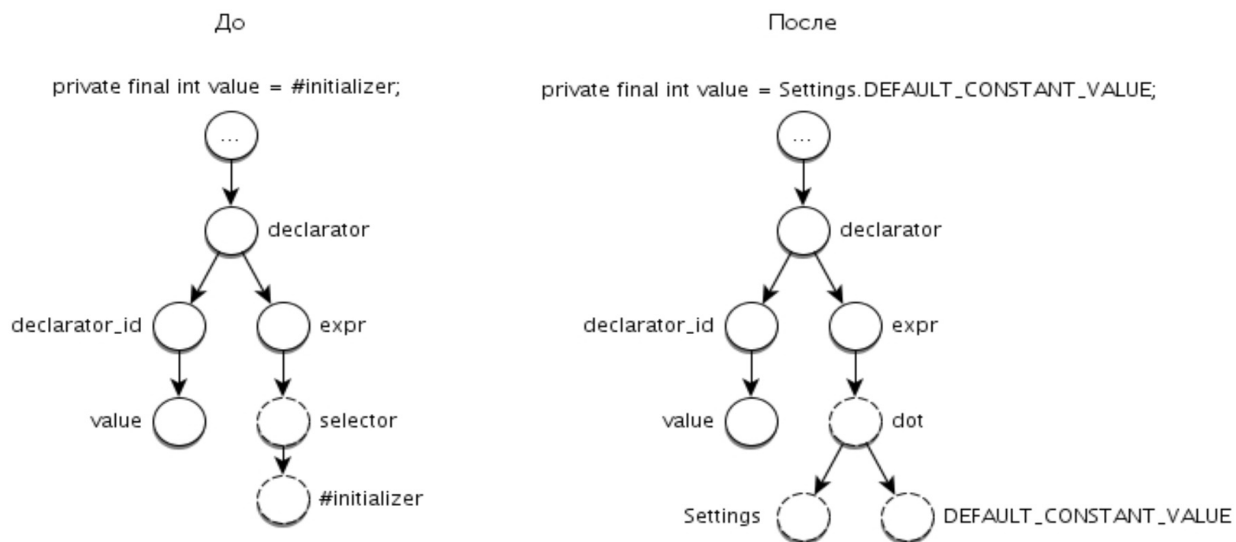


Рис. 5. АСД шаблонов «до» и «после» модификации

В качестве примера рассмотрим преобразование АСД шаблонов «до» и «после», представленных на рис. 5.

Результатом работы алгоритма построения разницы деревьев является последовательность операций модификации деревьев, представленная ниже.

1. Insert: node EXPR in node DECLARATOR in 1 position.
2. Insert: node DOT in node EXPR in 0 position.
3. Insert: node Settings in node DOT in 0 position.

4. Insert: node DEFAULT_CONSTANT_VALUE in node DOT in 1 position.

5. Delete: node EXPR.
6. Delete: node SELECTOR.
7. Delete: node #initializer.

Применение указанной последовательности операций к дереву «до» приводит к получению дерева «после». Аналогичным образом полученная последовательность может применяться и к частям дерева исходной программы, соответствующим шаблону «до».

Поиск участков кода. Поиск участков программного кода, подлежащих изменениям, осу-

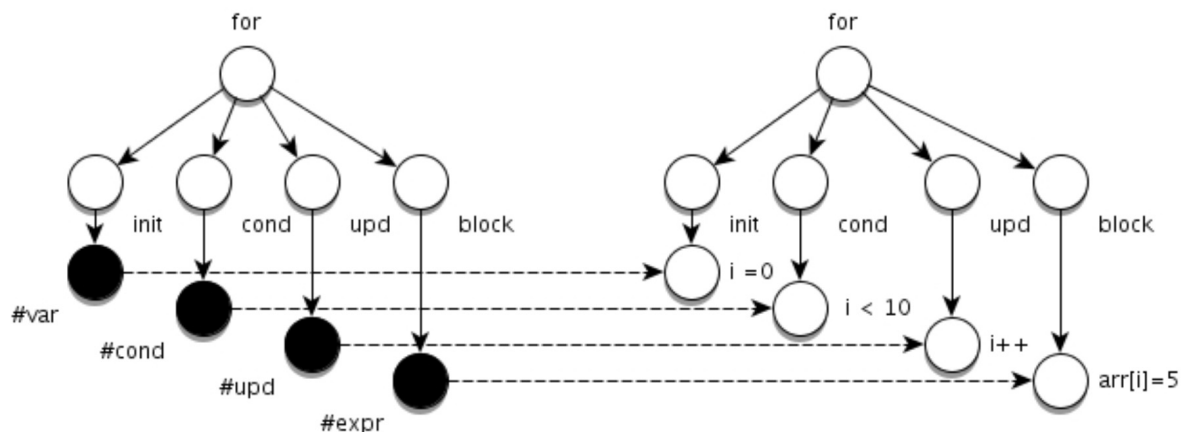


Рис. 6. Пример соответствий, построенных для параметров шаблона



ществляется в АСД исходной программы с использованием шаблона «до» модификации. Задача алгоритма поиска найти в дереве исходного кода поддеревья, соответствующие шаблону «до».

При этом используются следующие критерии соответствия.

1. Для простых узлов шаблонов:

типы и значения узлов дерева шаблона и дерева исходного кода совпадают;

выполняются критерии соответствия для всех дочерних узлов дерева шаблона и дерева исходного кода.

2. Для параметрических узлов шаблонов:

параметрический узел дерева шаблона соответствует последовательности узлов дерева исходного кода, при этом размер последовательности определяется регулярным выражением параметрического узла.

Во время поиска осуществляется построение соответствий между узлами АСД шаблона и узлами найденного поддерева АСД исходного кода. Так как параметрам шаблона может соответствовать неограниченное количество узлов, то для них строятся соответствующие списки, которые хранят найденные конструкции. На рис. 6 представлен пример построенных соответствий.

Модификация абстрактного синтаксического дерева. Все операции модификации осуществляются над АСД исходного кода путем последовательного применения операций, полученных при использовании алгоритма, описанного выше.

Операции модификации применяются во время поиска участков кода, удовлетворяющих шаблону «до», при первом же совпадении. На основе построенных соответствий между узлами исходного АСД и АСД шаблона «до» происходит проверка количества найденных узлов на соответствие количеству, указанному в свойствах параметров шаблона. В случае нахождения несоответствий поиск продолжается далее. Участки кода, которые были изменены, помечаются специальным образом для исключения их повторного нахождения. Затем поиск продолжается повторно от корня АСД. Поиск заканчивается, когда все узлы АСД проверены на соответствие шаблону «до».

Реализация. В рамках описанного подхода был разработан прототип системы модификации программного кода на основе параметризуемых шаблонов для языка Java 1.5. Для получения АСД исходного кода и шаблонов модификации используется парсер, созданный генератором парсеров ANTLR v3 [6]. Этот парсер работает на основе модифицированной грамматики языка Java 1.5. Так как модификация исходного кода осуществляется над промежуточным представлением, в качестве которого используется АСД, то при формировании текста измененного программного кода на языке Java необходимо осуществить преобразование АСД в программы на целевом языке. В разработанном прототипе используется промежуточное преобразование АСД в DOM-модель, использование которой упрощает формирование структуры и оформления (форматирования) программного кода.

Для удобства программистов, разработанный программный модуль оформлен в виде плагина для среды разработки Eclipse 3.5 [7].

Разработанная технология модификации программного кода может служить основой поддержки таких этапов жизненного цикла программного обеспечения, как разработка, тестирование и сопровождение. Использование автоматизированных преобразований существенно снижает количество ошибок, вносимых разработчиком на перечисленных этапах, и позволяет сократить время, затрачиваемое на проведение рефакторингов и оптимизаций.

Направления развития технологии связаны с расширением языка шаблонов в сторону поддержки семантических особенностей целевого языка и введения контекстно-чувствительной трансформации. Благодаря универсальному внутреннему представлению программы и шаблонов технология легко адаптируется к другим языкам программирования, таким как C/C++ и C#.

Работа выполнена в рамках госконтракта № П2226 ФЦП «Научные и научно-педагогические кадры инновационной России на 2009-2013 гг.».

СПИСОК ЛИТЕРАТУРЫ

1. Stratego/XT [Электронный ресурс]/<http://strategoxt.org/>
2. ASF+SDF [Электронный ресурс]/<http://www.meta-environment.org/>
3. The TXL Programming Language [Электронный ресурс]/<http://www.txl.ca/>
4. **Aho, A.V.** Compilers: Principles, Techniques, and Tools [Текст]/A.V. Aho, R. Sethi, J.D. Ullman.—1986.
5. **Chawathe, S.S.** Change detection in hierarchically structured information [Текст]/S.S. Chawathe, A. Rajaraman, H. Garcia-Molina [et al.]/Proc. of the 1996 ACM SIGMOD International Conf. on Management of Data.—1996.—P. 493–504.
6. ANTLR Parser Generator [Электронный ресурс]/<http://wwwantlr.org/>
7. Eclipse.org Home. The Eclipse Foundation [Электронный ресурс]/<http://www.eclipse.org/>

ВЕРИФИКАЦИЯ АЛГОРИТМА ПОДДЕРЖКИ ТРАНЗАКЦИОННОЙ ПАМЯТИ

При современном развитии многоядерных процессоров и параллельного программирования актуальны вопросы корректности программ и обнаружения в них ошибок. Разработчик не может предугадать все возможные варианты поведения параллельных взаимодействующих процессов. Поэтому любая параллельная программная система должна быть подвергнута комплексу процедур тестирования и верификации.

Это касается и одной из наиболее прогрессивных технологий, призванных упростить разработку корректных программ для многоядерных процессоров – *транзакционной памяти* [3]. Транзакционная память осуществляет управление доступом к разделяемой памяти при параллельном программировании, представляя собой аппаратно или программно реализованные ограничения доступа к памяти. Благодаря этим ограничениям, выделенные в параллельных программах участки кода могут рассматриваться как транзакции, т. е. логически неделимые программные единицы, результат параллельного выполнения которых эквивалентен результату некоторого их последовательного выполнения.

Наличие транзакционной памяти должно гарантировать полное отсутствие логических ошибок, связанных с синхронизацией процессов при выполнении параллельных программ на многоядерных процессорах. Это освобождает разработчика от необходимости использования примитивов синхронизации, неосторожное применение которых часто приводит к новым ошибкам, а также от последующих процедур тестирования и верификации готового продукта. Задача программиста сводится к тому, чтобы те участки кода, в которых происходят обращения к общей памяти, выделить как транзакции.

Реализация поддержки транзакционной памяти не тривиальна. Соответствующий алгоритм работает с параллельными процессами, поэтому в нем самом могут быть ошибки. Следовательно, такой алгоритм должен быть верифицирован. В данной статье приводятся результаты исследования «оконного» алгоритма транзакционной памяти (*window-based software transactional memory system*), *аналитическое формальное доказательство* корректности которого было предложено на X Международной конференции PaCT'09 [5].

Исследуемый алгоритм

«Оконный» алгоритм представляет собой описание трех операций: *чтения*, *записи* и *завершения*. Первые две используются для обращения транзакции к переменным, последняя фиксирует результат выполнения транзакции. Основные идеи алгоритма:

Каждая транзакция читает значения переменных (адресов) из разделяемой памяти один раз, в дальнейшем обращаясь к локальным копиям этих переменных (в операции *чтения*).

Модификация переменных всегда осуществляется локально (в операции *записи*). Реальная запись происходит, когда принято решение об успешном завершении транзакции (в операции *завершения*).

С каждой транзакцией ассоциируется *временное окно* из *временных меток*, связанных с модификацией наблюдаемых (прочитанных или записанных транзакцией) переменных. *Временные метки* – значения *логических часов*, глобальной целочисленной переменной. *Верхняя граница временного окна* для транзакции T – наименьшая величина из

значений *логических часов* перед модификацией переменных некоторыми транзакциями, отличными от T , после прочтения этих переменных транзакцией T . *Нижняя граница* – наибольшая *временная метка модификации* переменных, прочитанных T . *Временные метки модификации* записываются в ячейки памяти вместе с новыми значениями переменных (в операции *завершения*). *Временные метки модификации* представляют собой результат атомарной инструкции *Fetch&Increment*, которая увеличивает значение *логических часов* на единицу и возвращает их *старое* значение. *Верхняя* и *нижняя границы* инициализируются, соответственно, значениями $+\infty$ и 0.

Транзакция принимается или отклоняется в зависимости от состояния *временного окна*: при конечной *верхней границе* отклоняются пишущие транзакции (во время операции *завершения*), при пустом *окне* – любые (во время операции *чтения*).

Критерий согласованности параллельной программы

Описанный «оконный» алгоритм должен обеспечивать работу транзакционной памяти, при которой сохраняется согласованность параллельной программы. Для этого параллельные транзакции, выполняемые в соответствии с алгоритмом, должны удовлетворять критерию *opacity* [2], образованному двумя требованиями:

1) выполнение транзакций должно быть *сериализуемо*, т. е. результат параллельного выполнения транзакций должен быть эквивалентен результату некоторого их последовательного выполнения;

2) транзакция не должна производить никаких действий с прочитанными ею несогласованными данными.

Верификация алгоритма

Авторами [5] было представлено аналитическое формальное доказательство корректности «оконого» алгоритма, т. е. того, что транзакции, выполняемые в соответствии с ним, удовлетворяют критерию *opacity*. Однако известно, что доказательства корректности таких алгоритмов осложнены большим числом случаев, порожденных интерливингом (перекрывтием) параллельных взаимодействующих процессов, которые необходимо рассмотреть. Это может привести к ошибкам. Поэтому целью данной работы было про-

ведение автоматической верификации на основе современной технологии *model checking*, которая представляет собой исчерпывающую проверку всех возможных состояний и вычислений *модели* системы на соответствие некоторому набору формальных требований (*спецификации*) [1].

Для верификации был использован пакет комплексной верификации Spin [4]. Данный инструмент позволяет ответить на вопрос, удовлетворяет ли анализируемая система предъявляемым требованиям, а также, в случае отрицательного ответа, получить ту последовательность действий, которая иллюстрирует невыполнение требований (так называемый *контрпример*, позволяющий найти ошибку).

Основные этапы исследования:

I. Построение модели. Для алгоритма была построена *модель* на языке PROMELA [4]. Каждая транзакция в модели может находиться в одном из четырех рабочих состояний и двух заключительных (*Abort* и *Commit*) (рис. 1).

Из *состояния преобразования данных* транзакция может недетерминировано переходить в состояния *чтения*, *записи* или *завершения*, в которых выполняются алгоритмы соответствующих операций, а затем возвращаться обратно (из состояний *чтения* или *записи*) или завершать работу (в состояниях *чтения* или *завершения*), в зависимости от состояния окна.

II. Построение требований к поведению транзакций. Исходя из формальных требований критерия *opacity* [2, 5], построена спецификация этого алгоритма на языке *темпоральной логики линейного времени* (LTL) [1].

Для этого модель была дополнена генерацией *событий*, которые могут быть упорядочены во времени: BT – событие начала транзакции T ; CT , AT – принятие и отклонение соответственно; $COMPT$ – вход транзакции T в состояние преобразования данных; $rT(X)v$ – событие чтения значения v (возвращаемое значение) в переменную X из разделяемой памяти транзакцией T . Событие записи $wT(X)v$ определяется аналогично. При этом записываемое значение v – идентификатор транзакции T . Событие LT представляет собой *точку линеаризации* – момент времени, когда может быть однозначно решено, принимать транзакцию T или отклонять [5]. Событие LT может соответствовать событию CT (принятию транзакции) либо быть связанным с критическим изменением границ временных окон транзакции T , которое

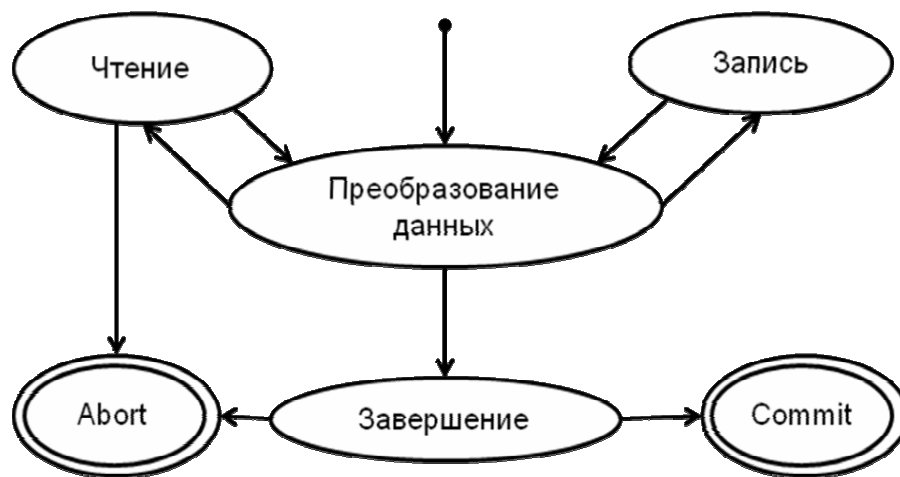


Рис. 1. Схема модели выполнения транзакции

впоследствии обязательно приведет к событию AT (отклонению транзакции). Результат сериализуемого параллельного выполнения транзакций эквивалентен результату их последовательного выполнения в порядке *точек линейаризации*.

В модели события представлены флагами, которые устанавливаются один раз для транзакции (BT , CT , AT , LT , $rT(X)v$, $wT(X)v$) или меняют свое значение в зависимости от состояния транзакции ($COMPT$).

На основании описанных событий строилось аналитическое доказательство в [5]. Эти же средства использованы в данной работе для составления формул LTL спецификации:

1) $((\Diamond CT) \rightarrow (\Diamond LT)) \wedge ((\Diamond AT) \rightarrow (\Diamond LT))$ – «принятые и отклоненные транзакции всегда имеют точку линейаризации»;

2) $(\Diamond (ET1 \wedge \neg BT2 \wedge (\Diamond BT2))) \rightarrow (\Diamond (LT1 \wedge \neg LT2 \wedge (\Diamond LT2)))$ – «если транзакция $T1$ завершилась раньше транзакции $T2$, то точка линейаризации транзакции $T1$ предшествует точке линейаризации транзакции $T2$ »;

3) $(\Diamond rT2(X)1) \rightarrow ((\Diamond ((LT1 \wedge \neg LT3 \wedge \neg LT2 \wedge (\Diamond (LT3 \wedge \neg LT2 \wedge (\Diamond LT2)))))) \wedge (\Diamond wT3(X)))$ – «если верно, что когда-нибудь транзакция $T2$ прочитает из X значение, записанное транзакцией $T1$ (транзакция $T<id>$ в построенной модели записывает в память свой идентификатор – $<id>$ (см. определение события $wT(X)v$), то неверно, что [когда-нибудь транзакция $T3$ модифицирует X и $LT1 < LT3 < LT2$]], здесь $<$ – отношение предшествования во времени на множестве событий»;

4) $(\Diamond (rT2(X)1 \wedge \neg LT2 \wedge (\Diamond LT2))) \rightarrow (\Diamond (LT1 \wedge \neg LT2 \wedge (\Diamond LT2)))$ – «если верно, что до

своей точки линейаризации транзакция $T2$ прочитает из X значение, записанное транзакцией $T1$, то $LT1 < LT2$ »;

5) $(\Diamond (LT2 \wedge \neg rT2(X)1 \wedge (\Diamond rT2(X)1))) \rightarrow \neg (\Diamond (rT2(X)1 \wedge (\Diamond COMPT2)))$ – «если транзакция $T2$ читает значение, записанное транзакцией $T1$, после своей *точки линейаризации*, то она обязательно отклоняется во время выполнения текущей операции *чтения* (не попадая в состояние *преобразования данных* из состояния *чтения*)» (см. рис. 1).

Последняя формула соответствует требованию к транзакции не производить действий с несогласованными данными (см. п. 3, требование 2), т. к. точка линейаризации $LT2$ соответствует критическому изменению границ окна (в противном случае транзакция $T2$ была бы уже принята и не могла выполнять чтение).

III. Верификация модели. Проведена верификация модели с двумя процессами, которые обращаются к двум адресам в памяти. В результате верификации обнаружена *ошибка* в алгоритме: найден пример вычисления, на котором формула (5), т. е. требование 2 критерия *opacity* (см. п. 3) не выполняется.

На рис. 2 представлен соответствующий пример некорректного выполнения транзакций, управляемых рассматриваемым алгоритмом, полученный в ходе верификации в системе Spin. Здесь R и W – события физического обращения к памяти (чтения и записи соответственно), число рядом – адрес; B , C и A – события начала, успешного завершения и отклонения транзакции; *comp* – событие входа транзакции в состояние преобразования данных.

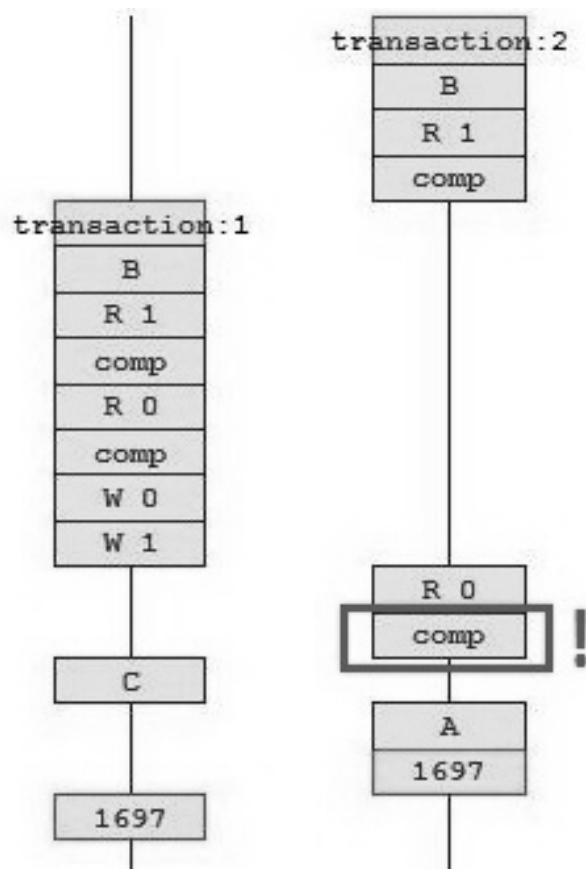


Рис. 2. Контрпример

На первый взгляд согласованность сохранена: транзакция 2, нарушающая требование сериализуемости, отклоняется (событие *A*). Однако перед этим транзакция посещает состояние преобразования данных (последнее событие *comp*), тем самым нарушая второе требование *opacity*.

Покажем на примере последствия такой ошибки. Пусть рис. 2 иллюстрирует обращения к памяти и вычисления двух транзакций: *transaction1*:: [*X* + 1; *Y* := *Y*/4; *X* := 1] и *transaction2*:: [$1/((X + 1) - Y)$], где *X* и *Y* располагаются по адресам 1 и 0 соответственно. Пусть изначально {*X* = 3, *Y* = 16}. В таблице показано соответствие событий контрпримера на рис. 2 операциям чтения из памяти, записи или последовательности вычислений рассматриваемых транзакций, а также событиям их начала и завершения.

Как видно из таблицы, перед отклонением транзакции *transaction2* будет произведена операция $1/(4 - 4)$. Таким образом, ошибка в алгоритме, пропущенная в ходе аналитического доказательства, могла привести к серьезным программным сбоям.

Интерпретация контрпримера

transaction 1		transaction 2	
		B	<i>Begin</i>
		R 1	<i>Read 3 from X</i>
		comp	3 + 1
B	<i>Begin</i>		
R 1	<i>Read 3 from X</i>		
comp	3 + 1		
R 0	<i>Read 16 from Y</i>		
comp	16/4		
W 0	<i>Write 4 to Y</i>		
W 1	<i>Write 1 to X</i>		
		R 0	<i>Read 4 from Y</i>
		comp	$1/(4 - 4)$
C	<i>Commit</i>		
		A	<i>Abort</i>



Последующий анализ алгоритма показал, что причина его некорректности – использование атомарной инструкции *Fetch&Increment(x)* внутри операции завершения, увеличивающей значение переменной *x* и возвращающей ее *старое* значение. При использовании аналогичной инструкции, возвращающей *новое* значение, некорректных поведений системы не обнаруживается.

В ходе исследования была проведена автоматическая верификация «оконного» алгоритма поддержки транзакционной памяти, описанного в [5], в результате чего в нем была найдена ошибка. Опубликованное в [5] аналитическое формальное доказательство корректности этого алгоритма также оказалось ошибочным. Анализ контрпримера, найденного системой Spin в ходе верификации, позволил выявить причину некорректности «оконного» алгоритма и исправить его.

Последующая верификация исправленного алгоритма не выявила в нем ошибок, поэтому он может быть рекомендован к реализации в много-ядерных процессорах.

Результаты проведенного исследования были признаны авторами «оконного» алгоритма в их новой статье [6], которая содержит исправленный алгоритм с предложенной модификацией.

Данная работа иллюстрирует необходимость использования процедур автоматической формальной верификации алгоритмов транзакционной памяти наряду с аналитическим доказательством их корректности. С помощью таких процедур могут быть найдены тонкие ошибки, которые проявляются редко и только в некоторых специальных случаях, поэтому они очень трудны для анализа. Но именно этим такие ошибки и опасны, т. к. могут проявиться в какой-либо критической программе и привести к сбою вычислений.

СПИСОК ЛИТЕРАТУРЫ

1. **Карпов, Ю.Г.** Model checking. Верификация параллельных и распределенных программных систем [Текст] / Карпов Ю.Г. – СПб.: БХВ, 2009.
2. **Guerraoui, R.** On the Correctness of Transactional Memory: [Текст] / R. Guerraoui, M. Kapalka // Proc. 13th ACM SIGPLAN symp. On Principles and Practice of Parallel Programming. – 2008. – P. 175–184.
3. **Herlihy, M.P.** Transactional Memory: Architectural Support for Lock-free Data Structures. [Текст] / M.P. Herlihy, J.E.B. Moss // Proc. 20th ACM Int'l symp. on Computer Architecture (ISCA'93). – 1993. – P. 289–300.
4. **Holzmann, G.** Spin Model Checker. The Primer and Reference Manual [Текст] / G. Holzmann. – Addison-Wesley, 2003. – 608 p.
5. **Imbs, D.** Software Transactional Memories: An Approach for Multicore Programming [Текст] / D. Imbs, M. Raynal. – PaCT 2009. LNCS. – 2009. – Vol. 5698. – P. 26–40.
6. **Imbs, D.** Software Transactional Memories: An Approach for Multicore Programming [Текст] / D. Imbs, M. Raynal // The journal of Supercomputing. – 09 Feb. 2010 – Springer.

УДК 025.4.03

И.С. Блеканов, Д.С. Бондаренко

ТЕМАТИЧЕСКИЙ КРАУЛИНГ НА ОСНОВЕ АЛГОРИТМА HITS

Все системы поиска информации в Веб-пространстве условно делятся на два класса по признаку наличия собственного индекса, хранящего информацию о документах, опубликованных в сети. Например, Google и Яндекс имеют собственный индекс, тогда как Quintura (<http://www.search.quintura.ru/>), Grokker (<http://www.grokker.com/>), Metacrawler (<http://www.metacrawler.com/>) и многие другие своего индек-

са не имеют и перенаправляют запросы пользователя в прочие системы (одну или несколько), но выполняют дополнительную обработку результатов и помогают пользователю формировать запрос. Для построения индекса существенной части Веб-пространства поисковая система должна иметь систему сетевых агентов (веб-краулеры), отслеживающих появление новых (и обновление старых) публикаций в Веб-пространстве и

сохраняющих индексную информацию об этих документах в индексе системы поиска. В связи с быстрым ростом Веб-пространства [5, 4] проблема построения алгоритма его обхода (как первичного, так и с целью обновления индекса) является очень сложной и актуальной.

В данной статье предлагается алгоритм обхода Веб-пространства, целью которого является посещение в первую очередь наиболее авторитетных, полезных документов. Для этого предлагается использовать модифицированный алгоритм Клеинберга HITS (в свое время был конкурентом алгоритму PageRank – базе для Google), который в относительно небольшой тематически сфокусированной части Веб-пространства может найти страницы двух типов: авторитетные и индексные. В предлагаемом алгоритме авторитетные страницы сразу же включаются в индекс, а индексные используются для расширения поиска новых авторитетных страниц. Также оценивается качество поиска данного алгоритма обхода Веб-пространства в сравнении со стандартным алгоритмом в совокупности с ранжированием его результатов.

Принципы и алгоритм работы системы

Проведенное в 1999 г. Кумаром и Бродером исследование [2] на примере 200 млн Веб-страниц выявило, что естественной моделью представления структуры Веб-пространства является ориентированный *Веб-граф* $G(V, E)$ [2], в котором вершины $v \in V$ соответствуют веб-страницам, а дуги (направленные ребра) $e(u, v)$, $u, v \in V$, $e \in E$ – соединяющим страницы гиперссылкам. Расстояние $d(u, v)$ между узлами u и v в графе без учета весов ребер равно наименьшему числу гиперссылок, по которым можно пройти из u в v . В рамках этой модели задача анализа структуры связей между страницами – это задача анализа структуры графа. Исследование Веб-графа (обход Веб-пространства) – это процесс поиска узлов и ребер графа, начиная с некоторого начального подмножества узлов (в рамках нашей терминологии узлы – это Веб-страницы, а ребра – гиперссылки, соединяющие Веб-страницы). За выполнение данной процедуры отвечает Веб-краулер (сетевой агент).

По своей сути, *Веб-краулер* – программа, написанная на языке высокого уровня, взаимодействующая с окружающей средой посредством сетевых протоколов, например, интернет-протокола

HTTP. Задачей сетевых агентов является обход Веб-графа определенным образом с целью сбора информации или понимания структуры и полезности каких-либо Веб-страниц. А также передача собранной информации для анализа другим приложениям поисковых систем.

Веб-краулеры упрощают поиск информации в Веб-пространстве и повышают его качество и эффективность. Как правило, поисковые системы используют некоторое множество сетевых агентов, которые собирают Веб-страницы, извлекая из них контент (содержание) и индексируя его [7]. Полученные сетевыми агентами результаты обхода Веб-графа обрабатываются поисковой системой на предмет выявления наиболее подходящих результатов для заданного пользователем поиска. Например, поисковая машина Google дополнительно использует алгоритм PageRank, предназначенный для ранжирования результатов поиска. В процессе ранжирования для каждой Веб-страницы вычисляется вес, страницы с наибольшими весами получают высокий рейтинг в результатах поиска.

В данной статье реализованы две модели Веб-краулеров: стандартная и сфокусированная. Стандартная модель Веб-краулера выполняет классический обход Веб-графа по всем найденным узлам, начиная с некоторого заданного множества узлов [1]. В результате своей работы он формирует коллекцию из найденных Веб-страниц. Сфокусированная модель Веб-краулера использует алгоритм Клеинберга HITS в качестве алгоритма обхода Веб-пространства. В результирующую коллекцию Веб-страниц для такой модели попадают только приоритетные по HITS.

В основе алгоритма Клеинберга HITS лежит понятие значимости страницы. Наиболее значимыми Веб-страницами в рамках заданной темы принято считать *авторитетные страницы* (*authorities*), на которые ссылаются другие страницы, относящиеся к данной теме. Это свойство позволяет выявить *индексные страницы* (*hubs*). Авторитетные страницы – это страницы, на которые ссылаются индексные страницы, а индексные страницы – это страницы, которые ссылаются на авторитетные страницы. Вместе оба типа значимых страниц образуют *отношение взаимного усиления*, т. е., на хорошую авторитетную страницу ссылаются много хороших индексных страниц, и хорошая индексная страница ссылается на хорошие авторитетные страницы. Целью алгорит-

ма HITS является поиск наиболее качественных авторитетных и индексных страниц в подмножестве S_σ Веб-графа $G(V, E)$. Качество Веб-страниц оценивается числовыми величинами.

Подмножество S_σ Веб-графа строится, например, путем отправки запроса σ поисковой машине по ключевым словам, и берутся первые n результатов. В ходе анализа подграфа S_σ каждой Веб-странице сопоставляются два веса: x – *АР-вес*, показывающий качество страницы как авторитетной, и y – *НР-вес*, показывающий качество страницы как индексной. Изначально весам присваивается максимальное значение, равное единице, и для каждой страницы p задается пара неотрицательных весов $(x^{<p>}, y^{<p>})$, где $x^{<p>}, y^{<p>}$ – веса для авторитетной и для индексной страниц, соответственно.

К весам Веб-страниц применяются две операции I и O . С помощью заданных весов $\{x^{<p>}\}$ и $\{y^{<p>}\}$ операция I вычисляет АР-вес для страницы p следующим образом [6]: $x^{<p>} \leftarrow \sum_{q:(p,q) \in E} y^{<q>}$,

а операция O вычисляет НР-вес для страницы p : $y^{<p>} \leftarrow \sum_{q:(p,q) \in E} x^{<q>}$. Операции I и O образуют

упомянутый выше процесс отношения взаимного усиления авторитетной страницы относительно индексной, и наоборот. Поэтому для нахождения стационарных значений АР и НР весов необходимо применить операции I и O в итерационном процессе. После каждой итерации полученные веса нормируются, т. е. $\sum_{p \in S_\sigma} (x^{<p>})^2 = 1$ и

$\sum_{p \in S_\sigma} (y^{<p>})^2 = 1$. В результате работы алгоритма получаем набор как авторитетных, так и индексных Веб-страниц. Далее этот набор фильтруется, и из него выбираются $c \approx 5 - 10$ Веб-страниц с наибольшими АР-весами и c страниц с наибольшими НР-весами. Так определяются качественные авторитетные и качественные индексные Веб-страницы.

В работе Клейнберга [6] обосновывается быстрая сходимость итерационного процесса операций I и O . Там указано, что для того, чтобы двадцать наивысших значений АР-весов и НР-весов становились стабильными, достаточно около двадцати итераций.

В данной статье использовалась модифицированная реализация HITS в качестве алгоритма

обхода и выбора дальнейшего пути сканирования Веб-графа сфокусированным Веб-краулером. То есть, на каждом шаге получения новых узлов Веб-краулером применялся алгоритм HITS к новому полученному множеству узлов и старому (уже обработанному) множеству. Из полученного набора Веб-страниц качественные авторитетные страницы записывались в индекс, а по качественным индексным продолжался поиск новых авторитетных. В результате обхода Веб-пространства сфокусированный краулер на основе алгоритма HITS выдаст тематическое сообщество авторитетных страниц. В работе [3] тематическим сообществом называют множество Веб-страниц, в котором каждая страница имеет больше ссылок на другие страницы сообщества, чем на страницы вне сообщества.

Эксперимент

В ходе эксперимента оценивались результаты тематического краулера на основе алгоритма HITS, полученные после обхода Веб-графа. И сравнивались с отранжированными по HITS результатами, которые получились в результате обхода того же графа стандартным Веб-краулером.

Под стандартным Веб-краулером здесь подразумевается реализация краулера с классическим сценарием поведения: нахождение большого числа страниц (в данной реализации – с помощью рекурсивного обхода всех встречающихся Веб-страниц) и затем ранжирование с помощью HITS полученных результатов в конце (в отличие от сфокусированного, где это выполняется на каждой итерации).

Эффективность работы Веб-краулеров проверялась в локальном Веб-пространстве (в рамках одного домена, в качестве которого был выбран сайт Википедия – <http://ru.wikipedia.org/>) и региональным (по русскоязычному Веб-пространству). Для эксперимента из запросов РОМИП-2007 были выбраны два запроса:

1) «Нобелевская премия». Описание: Соответствующая запросу страница должна содержать подробную информацию о Нобелевской премии. Полезной является информация об истории создания премии, создателе, научных областях, по которым присуждается премия, ее лауреатах. Упоминание о присуждении Нобелевской премии конкретному человеку не является исчерпывающим ответом на запрос.

2) «Квантовый компьютер». Описание: Релевантная страница должна содержать подробную информацию о квантовых компьютерах: историю создания, принципы работы, перспективы развития и т. п. Частично релевантной будет считаться страница с упоминанием о каком-либо достижении или событии в развитии квантовых компьютеров.

По этим запросам строится начальное множество Веб-узлов, с которого начали свою работу Веб-краулеры. Например, начальным множеством узлов для локального Веб-пространства являлся ответ Википедии на выбранный запрос. Стартовое множество для регионального Веб-пространства строилось следующим образом: посылался выбранный запрос на поисковые системы Яндекс и Google, и выбирались первые десять результатов каждой системы. В итоге получилось четыре множества Веб-страниц (по два множества на каждый выбранный запрос), с которых стартуют сфокусированный и стандартный краулеры.

Результатом работы двух Веб-краулеров на выбранных стартовых множествах является набор Веб-страниц. По этим результатам строится тестовая коллекция. Производится экспертная оценка элементов построенной коллекции, т. е. из описаний запросов, полученных из РОМИПа, выполняется бинарная классификация каждого элемента тестовой коллекции «релевантен/не релевантен» по отношению к выбранному запросу.

Для оценки эффективности разработанного тематического Веб-краулера на основе метода алгоритма HITS в данной статье использовались наиболее популярные критерии: полнота; точность (precision); 11-точечный график полноты/точности, измеренный по методике TREC (11-point matrix (TREC)) [8–10]. Если a – количество документов, найденных системой и релевантных с точки зрения экспертов, b – количество документов, найденных системой, c – количество релевантных документов, не найденных системой, но нерелевантных с точки зрения экспертов, то метрики вычисляются следующим образом:

полнота (recall) – это отношение найденных релевантных документов к общему количеству релевантных документов $r = \frac{a}{a+c}$.

точность (precision) – это отношение найденных релевантных документов к общему количеству найденных документов $p = \frac{a}{a+b}$.

В дополнение к этим метрикам приближенно оценивалась вычислительная производительность каждого Веб-краулера.

Итоги эксперимента

По запросу «Нобелевская премия» в региональном Веб-пространстве тематический краулер на основе алгоритма HITS нашел множество, состоящее из 117 авторитетных Веб-страниц. Первые 10 качественных авторитетных страниц:

<http://wikimediafoundation.org/>

<http://de.wikipedia.org/wiki/Nobelpreis>

<http://ja.wikipedia.org/wiki/%E3%83%8E%E3%83%BC%E3%83%99%E3%83%AB%E8%B3%9E>

http://fr.wikipedia.org/wiki/Prix_Nobel

http://ur.wikipedia.org/wiki/%D9%86%D9%88%D8%A8%D9%84_%D8%A7%D9%86%D8%B9%D8%A7%D9%85

http://mg.wikipedia.org/wiki/Loka_Nobel

<http://fi.wikipedia.org/wiki/Nobel-palkinto>

<http://no.wikipedia.org/wiki/Nobelprisen>

<http://nl.wikipedia.org/wiki/Nobelprijs>

<http://nds.wikipedia.org/wiki/Nobelpries>

По этому же запросу стандартный Веб-краулер в том же пространстве построил множество из 150 000 страниц. В результате ранжирования по HITS этих Веб-страниц получилось, как и в случае сфокусированного краулера, 117 страниц, первые 10 из которых:

<http://www.mediawiki.org/>

<http://wikimediafoundation.org/>

http://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D0%B1%D0%B5%D0%BB%D0%B5%D0%B2%D1%81%D0%BA%D0%B0%D1%8F_%D0%BF%D1%80%D0%B5%D0%BC%D0%B8%D1%8F

<http://de.wikipedia.org/wiki/Nobelpreis>

<http://ja.wikipedia.org/wiki/%E3%83%8E%E3%83%BC%E3%83%99%E3%83%AB%E8%B3%9E>

http://fr.wikipedia.org/wiki/Prix_Nobel

http://ur.wikipedia.org/wiki/%D9%86%D9%88%D8%A8%D9%84_%D8%A7%D9%86%D8%B9%D8%A7%D9%85

http://mg.wikipedia.org/wiki/Loka_Nobel

<http://fi.wikipedia.org/wiki/Nobel-palkinto>

<http://no.wikipedia.org/wiki/Nobelprisen>

В результате работы двух краулеров получили тестовую коллекцию из 118 Веб-страниц. На этой коллекции для тематического Веб-краулера метрика точности $p'_{11} \approx 0,93$, для отранжированных результатов стандартного Веб-краулера $p'_{12} \approx 0,93$. График зависимости полноты от точности показан на рис. 1.

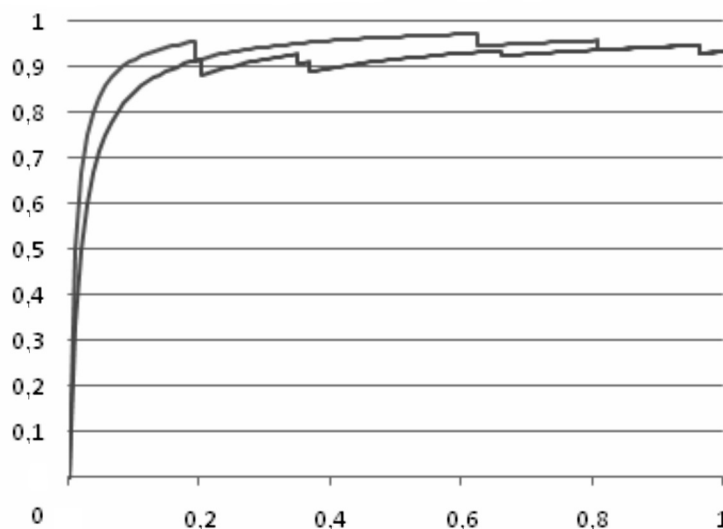


Рис. 1. 11-точечный график полноты/точности для запроса «Нобелевская премия» в региональном Веб-пространстве
(—) тематический Веб-краулер; (---) стандартный Веб-краулер

Для Википедии (Веб-пространства в рамках одного домена) по тому же запросу «Нобелевская премия» сфокусированный краулер нашел 95 Веб-страниц с метрикой точности $p'_{21} \approx 0,9$, среди которых 10 наилучших авторитетных:

<http://www.mediawiki.org/>
<http://wikimediafoundation.org/>
<http://de.wikipedia.org/wiki/Nobelpreis>
http://pl.wikipedia.org/wiki/Nagroda_Nobla
http://it.wikipedia.org/wiki/Premio_Nobel
http://fr.wikipedia.org/wiki/Prix_Nobel
http://mg.wikipedia.org/wiki/Loka_Nobel
<http://fi.wikipedia.org/wiki/Nobel-palkinto>
<http://nl.wikipedia.org/wiki/Nobelprijs>
<http://no.wikipedia.org/wiki/Nobelprisen>

Стандартный краулер после ранжирования выдал 100 Веб-страниц и точность $p'_{22} \approx 0,02$. Первые 10 авторитетных Веб-страниц с наивысшими весами:

<http://www.mediawiki.org/>
<http://wikimediafoundation.org/>
<http://km.wikipedia.org/wiki/%E1%9E%9C%E1%9E%B7%E1%9E%82%E1%9E%B8%E1%9E%97%E1%9E%B8%E1%9E%8C%E1%9E%B6>
<http://eu.wikipedia.org/wiki/Wikipedia>
<http://ur.wikipedia.org/wiki/%D9%88%DB%8C%DA%A9%DB%8C%D9%BE%DB%8C%DA%88%DB%8C%D8%A7>
<http://wa.wikipedia.org/wiki/Wikipedia>
<http://gn.wikipedia.org/wiki/Vikipet%C3%A3>
<http://nah.wikipedia.org/wiki/Huiquikipedia>

<http://ay.wikipedia.org/wiki/Wikipidiya>
<http://bar.wikipedia.org/wiki/Wikipedia>

11-точечный график полноты/точности показан на рис. 2.

По запросу «Квантовые компьютеры» в региональном Веб-пространстве тематический краулер на основе алгоритма Клейнберга получил 45 авторитетных Веб-страниц и коэффициент точности $p''_{11} \approx 0,73$. Первые 10 наилучших:

http://id.wikipedia.org/wiki/Komputer_kuantum
http://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%BD%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80
<http://www.mediawiki.org/>
<http://wikimediafoundation.org/>
http://simple.wikipedia.org/wiki/Quantum_computer
<http://de.wikipedia.org/wiki/Quantencomputer>
http://cs.wikipedia.org/wiki/Kvantov%C3%BD_po%C4%8D%C3%ADta%C4%8D
<http://ja.wikipedia.org/wiki/%E9%87%8F%E5%AD%90%E3%82%B3%E3%83%B3%E3%83%94%E3%83%A5%E3%83%BC%E3%82%BF>
<http://no.wikipedia.org/wiki/Kvantedatamaskin>
<http://hu.wikipedia.org/wiki/Kvantumsz%C3%A1lmc%C3%ADr%C3%B3g%C3%A9p>

Стандартный Веб-краулер по результатам ранжирования получил 48 Веб-страниц и коэффициент точности $p''_{12} \approx 0,71$. Первые 10 наилучших авторитетных страниц:

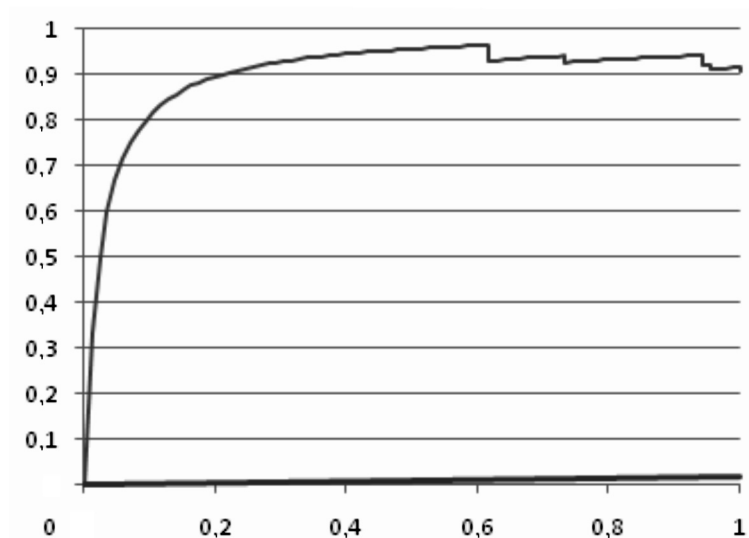


Рис. 2. 11-точечный график полноты/точности для запроса «Нобелевская премия» в локальном Веб-пространстве
(————) тематический Веб-краулер; (-----) стандартный Веб-краулер

<http://www.mediawiki.org/>
<http://wikimediafoundation.org/>
http://id.wikipedia.org/wiki/Komputer_kuantum
http://es.wikipedia.org/wiki/Computaci%C3%B3n_cu%C3%A1ntica
<http://no.wikipedia.org/wiki/Kvantedatamaskin>
http://cs.wikipedia.org/wiki/Kvantov%C3%BD_po%C4%8D%C3%ADta%C4%8D
http://ko.wikipedia.org/wiki/%EC%96%91%EC%9E%90_%EC%BB%B4%ED%93%A8%ED%84%B0

http://it.wikipedia.org/wiki/Computer_quantistico
<http://nl.wikipedia.org/wiki/Kwantumcomputer>
http://en.wikipedia.org/wiki/Quantum_computer
 Зависимость полноты от точности принимает следующие значения, показанные на рис. 3.

В локальном Веб-пространстве, на примере Википедии, по запросу «Квантовые компьютеры» тематический Веб-краулер нашел 41 Веб-страницу с коэффициентом точности $p''_{12} \approx 0,71$. Первые 10 наиболее качественных:

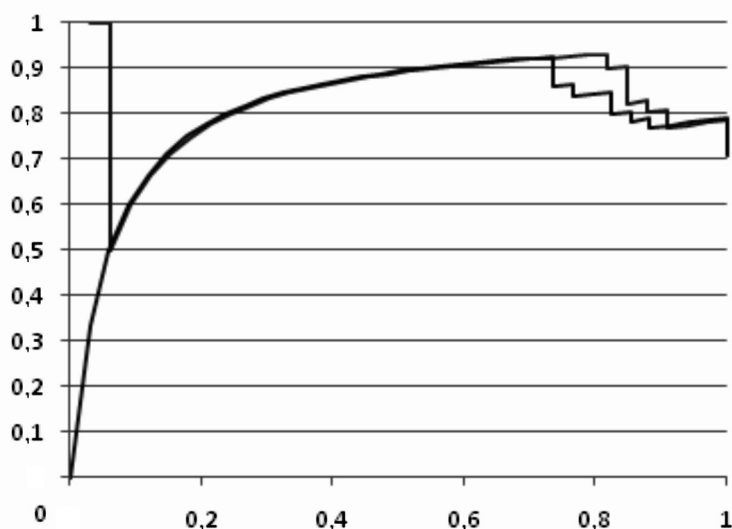


Рис. 3. 11-точечный график полноты/точности для запроса «Квантовые компьютеры» в региональном Веб-пространстве
(————) тематический Веб-краулер; (-----) стандартный Веб-краулер

http://id.wikipedia.org/wiki/Komputer_kuantum
<http://www.mediawiki.org/>
<http://wikimediafoundation.org/>
<http://tph.tuwien.ac.at/~oemer/qcl.html>
<http://www.qubit.org>
<http://www.nature.com/nature/journal/vaop/ncurrent/pdf/nature08121.pdf>
<http://www.wikimatrix.org/>
<http://opa.yale.edu/news/article.aspx?id=6764>
<http://www.newscientist.com/article/dn17736-codebreaking-quantum-algorithm-run-on-a-silicon-chip.html>
<http://jqquantum.sourceforge.net/jQuantumApplet.html>

Стандартный Веб-краулер по тому же запросу после ранжирования выдал 73 Веб-страницы с коэффициентом точности $p''_{22} \approx 0,0$. Первые 10 наилучших авторитетных страниц:

<http://sv.wikipedia.org/wiki/>
<http://pt.wikipedia.org/wiki/>
<http://it.wikipedia.org/wiki/>
<http://www.wikipedia.org>
<http://www.wikipedia.org/>
http://wikimediafoundation.org/wiki/Donate/Now/en?utm_source=donate&utm_medium=sidebar&utm_campaign=spontaneous_donation
<http://creativecommons.org/licenses/by-sa/3.0/>
http://wikimediafoundation.org/wiki/Terms_of_Use
<http://bh.wikipedia.org/wiki/%E0%A4%B5%E0%A4%BF%E0%A4%95%E0%A4%BF%E0%A4%AA%E0%A5%80%E0%A4%A1%E0%A4%BF%E0%A4%AF%E0%A4%BE>

<http://ps.wikipedia.org/wiki/%D9%88%D9%8A%DA%A9%D9%8A%D9%BE%DB%90%DA%89%D9%8A%D8%A7>

Зависимость полноты от точности принимает следующие значения, проиллюстрированные на рис. 4.

Целью данного исследования была оценка эффективности работы и качества поиска разработанного авторами тематического краулера на основе алгоритма HITS по сравнению со стандартным Веб-краулером с учетом ранжирования его результатов по HITS, а также исследование поведения обоих Веб-краулеров в локальном (на примере Википедии) и глобальном Веб-пространствах.

Тематический краулер на основе алгоритма HITS показал лучшую метрику точности ($p'_{11} \approx 0,93$, $p'_{21} \approx 0,9$, $p''_{11} \approx 0,73$, $p''_{21} \approx 0,71$) во всех проведенных опытах в сравнении со стандартным Веб-краулером ($p'_{12} \approx 0,93$, $p'_{22} \approx 0,02$, $p''_{21} \approx 0,71$, $p''_{22} \approx 0,0$). В региональном Веб-пространстве по двум запросам тематический краулер нашел более релевантное множество Веб-страниц, чем отранжированное по HITS множество, выданное стандартным Веб-краулером (рис. 1, 3). В свою очередь, стандартный Веб-краулер показал плохие результаты в Википедии (рис. 2, 4). В результирующее множество попадали только нерелевантные Веб-страницы. В локальном Веб-

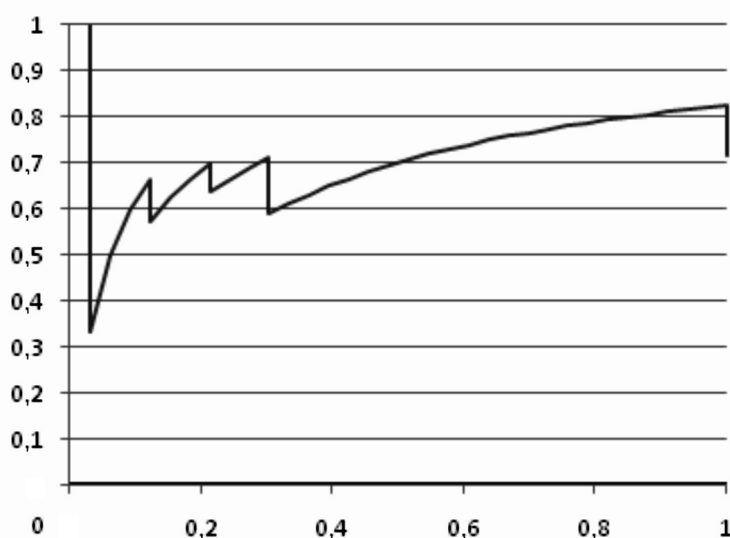


Рис. 4. 11-точечный график полноты/точности для запроса «Квантовые компьютеры» в локальном Веб-пространстве
(—) тематический Веб-краулер; (---) стандартный Веб-краулер

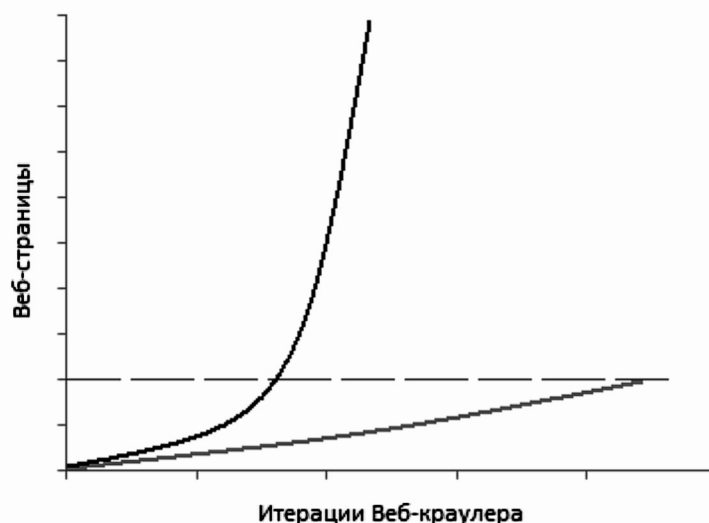


Рис. 5. Скорость роста количества Веб-страниц на каждой итерации Веб-краулера
(——) тематический Веб-краулер; (——) стандартный Веб-краулер

пространстве тематический краулер зачастую выдавал Веб-страницы, релевантные запросу. Его результирующее множество состояло из тематически близких страниц.

В ходе работы стандартного Веб-краулера наблюдается экспоненциальный рост Веб-страниц на каждой его итерации, уже на четвертой итерации насчитывалось 150–200 тысяч страниц. Обработка таких результатов, безусловно, негативно влияет на производительность. Тематический краулер на основе алгоритма HITS отличился и в этом компоненте, на каждой его итерации рост

Веб-страниц стремится к некоторой асимптоте (рис. 5). Результатом этого является увеличение производительности.

В дальнейшем планируется расширить эксперимент, сравнив тематический краулер на основе алгоритма Клейнберга, анализирующий гиперссылочную структуру Веб-пространства, и сфокусированный краулер на основе TF-IDF алгоритма [7], анализирующий содержание каждой Веб-страницы, и объединить TF-IDF и HITS алгоритмы и оценить эффективность поиска Веб-краулера, основанного на таком принципе.

СПИСОК ЛИТЕРАТУРЫ

1. **Arasu, A** Searching the web [Текст]/A. Arasu, J. Cho, H. Garcia-Molina [et al.]/ACM Transactions on Internet Technology. —Aug. 2001.—№ 1 (1).—P. 2–43.
2. **Broder, A.** Graph structure in the Web: Experiments and models [Текст]/A. Broder, R. Kumar, F. Maghoul [et al.]/In WWW9. —Amsterdam. —Elsevier Science. —May 2000. —P. 309–320.
3. **Gibson, D.** Inferring Web communities from link topology [Текст]/D. Gibson, J. Kleinberg, P. Raghavan//Proc. 9th ACM conf. on Hypertext and Hypermedia. 1998.—ACM, NY. —P. 117–133.—P. 225–234.
4. **Huberman, B.A.** Growth dynamics of the world-wide web. [Текст]/ B.A. Huberman, L.A. Adamic//Nature, 1999.—P. 131.
5. **Kahle, B.** Preserving the Internet [Текст]/B. Kahle//Scientific American, Mar. 1997.—P. 82–83.
6. **Kleinberg, J.** Authoritative sources in a hyperlinked environment [Текст]/J. Kleinberg//In Proc. of the 1998 ACM-SIAM symp. on Discrete Algorithms.—ACM Press, NY., 1998.
7. **Manning, C** An Introduction to Information Retrieval [Текст]/C. Manning, P. Raghavan, H. Schütze//Cambridge University Press, Apr. 2009.—P. 443–481.
8. Program to evaluate TREC results using SMART evaluation procedures: Documentation [Электронный ресурс]/http://www-nlpir.nist.gov/projects/trecvid/trecvid.tools/trec_eval
9. The 12 Text Retrieval conf. (TREC 2003). Appendix 1. Common Evaluation Measures [Электронный ресурс]/http://trec.nist.gov/pubs/trec12/appendices/measures.ps
10. **Zobel, J.** How reliable are the results of large-scale information retrieval experiments? [Текст]/J. Zobel//In Research and Development in Information Retrieval.—1998.—P 307–314.



ИССЛЕДОВАНИЕ РАВНОВЕСНЫХ СИТУАЦИЙ В ИГРОВОЙ ЗАДАЧЕ ДВУХ ЛИЦ С ВЕКТОРНЫМИ ВЫИГРЫШАМИ

Проблема принятия решений является одной из основных для человечества. Качество принятых решений определяет качество жизни. Повседневно сталкиваясь с необходимостью выбирать тот или иной способ действий, человек использует при этом имеющиеся в его распоряжении логические возможности, проводя различные рассуждения; применяет ассоциации; вспоминает аналогичные случаи; высказывает прогнозы, предположения, догадки; прибегает к интуиции, производит расчеты. При этом стремится к решениям, которые приводят к наилучшим результатам. Такой выбор принято называть оптимальным.

Особенно сложные управленческие проблемы возникают в сфере социально-экономических отношений. Такие явления имеют особенности, усложняющие их изучение. Обычно развитие социально-экономического явления зависит от действий нескольких лиц, каждое из которых лишь частично контролирует ситуацию. Лица, принимающие решения (ЛПР), имеют свои интересы, часто не полностью выявленные и представленные с определенной долей неопределенности. Эти интересы могут совпадать (возможно, частично), что ведет к сотрудничеству и кооперации. Но они могут и противоречить друг другу (возможно, частично), что ведет к соперничеству и конфликту. Важная задача при выработке решения – согласование интересов сторон.

В статье представлена математическая модель, учитывающая приведенные выше особенности. Это модель взаимодействия (сотрудничества и соперничества) двух лиц, каждое из которых имеет свой набор действий X и Y , соответственно. Особенностью модели является то, что результат взаимодействия для каждого лица представлен не одним показателем, а конечным набором критериев. Для простоты изложения рассматривается задача с двумя критериями у каждой стороны. Случай произвольного конечного набора критериев с соответствующими изменениями изучается по аналогичной схеме. Лица принимают решения независимо, так, чтобы получить по каждому

критерию наилучший для себя результат. Определение оптимального решения в представленной модели, изучение свойств и создание алгоритмов для вычисления составляют основное содержание статьи.

Ситуации принятия решения более чем одним лицом и конечным набором целей (по крайней мере, у одной стороны их две или более) может быть представлена многокритериальной игрой или – альтернативное название – игрой с векторными выигрышами. Игровые задачи с векторными выигрышами изучаются уже давно [1, 2]. Краткий исторический обзор и последние результаты представлены в [3]. В этой работе отмечается, что «многокритериальные игры могут быть использованы при моделировании различных явлений, когда в расчет приходится брать несколько различных критериев, например, при моделировании политики или менеджмента».

Математическая постановка задачи

Рассматривается бескоалиционная игра двух лиц с векторным двухкритериальным выигрышем у каждого из игроков:

$$\begin{aligned} \langle X, Y, F(x, y) \rangle = \\ = \langle X, Y, f^{(1)}(x, y) = (f_1^{(1)}(x, y), f_2^{(1)}(x, y)), \\ f^{(2)}(x, y) = (f_1^{(2)}(x, y), f_2^{(2)}(x, y)) \rangle. \end{aligned} \quad (1)$$

Здесь множество $X \subset R^k$ ($Y \subset R^l$) состоит из стратегий первого (второго) игрока. Набор из стратегий двух игроков $(x, y) \in X \times Y$ называется *ситуацией* и множество всех ситуаций $X \times Y$. Задана векторная функция $f^{(1)}: X \times Y \rightarrow R^2$ ($f^{(2)}: X \times Y \rightarrow R^2$), которая каждой ситуации ставит в соответствие вектор выигрышей первого (второго) игрока $f^{(1)}(x, y) = (f_1^{(1)}(x, y), f_2^{(1)}(x, y))$ ($f^{(2)}(x, y) = (f_1^{(2)}(x, y), f_2^{(2)}(x, y))$). Рассмотрим случай двухкритериального выигрыша у каждого игрока, случай большего числа критериев рассматривается аналогично.

Партия игры развивается следующим образом: каждый из двух игроков выбирает свою

стратегию, в результате чего складывается ситуация $(x, y) \in X \times Y$. После этого игроки получают свои выигрыши $f^{(1)}(x, y) = (f_1^{(1)}(x, y), f_2^{(1)}(x, y))$ и $f^{(2)}(x, y) = (f_1^{(2)}(x, y), f_2^{(2)}(x, y))$, равные значению своей векторной функции в сложившейся ситуации. Цель обоих игроков состоит в выборе такой своей стратегии, чтобы достичь возможно большего значения каждой из двух компонент своей векторной функции выигрыша, учитывая при этом выбор другого игрока.

В задаче (1) каждый игрок решает многокритериальную задачу при фиксированном выборе другого игрока. В [4, с. 58–59] отмечено, что «кандидатом» на оптимальное решение в многокритериальной задаче может являться только Парето-оптимальный исход. Однако Парето-оптимальных решений в задаче (1) может быть несколько, а в непрерывном случае даже бесконечное множество. Это связано с эффектом несравнимости исходов в векторном критериальном пространстве. Несравнимость исходов является формой неопределенности, именно, ценностной неопределенностью. Она обусловлена тем, что в условиях неполной информированности игрок стремится достичь нескольких, часто противоречивых целей. Эта проблема сохраняется и для игровой задачи с векторными выигрышами.

Достаточно общий подход к определению оценочной структуры в задачах с векторным выигрышем предлагают конусные отношения. В многокритериальных задачах такой подход представлен в [5, 6]. Будем рассматривать выпуклый, острый, многогранный (полиэдральный) пространственный конус $K \subset R^2$ [7, с. 235–255]. Конус порождает в векторном пространстве отношение порядка (векторную упорядоченность) $\geq$ по правилу:

$$f^{(1)} \geq f^{(2)} \Leftrightarrow f^{(1)} - f^{(2)} \in K. \quad (2)$$

Такой конус K называют конусом доминирования в R^2 . Часто рассматривается многогранный конус, который можно задать матрицей:

$$K = \{f \in R^2 \mid Af \geq 0_{(2)}\}. \quad (3)$$

Здесь представлена система двух однородных неравенств и $0_{(2)}$ – нулевой вектор в R^2 . Зафиксирована A – квадратная матрица второго порядка. Будем считать, что матрица A является неотрицательной, т. е. $a_{ij} \geq 0$. Кроме того, полагаем, что матрица A является невырожденной и, в специально оговорённых случаях, неразложимой [8, с. 352].

Важными примерами многогранных конусов являются

$$R_{\geq}^2 = \{x \in R^2 \mid Ex \geq 0_{(2)}\} = \{x \in R^2 \mid x_i \geq 0, i = 1, 2\} \quad (4)$$

и его внутренность

$$R_{>}^2 = \{x \in R^2 \mid Ex > 0_{(2)}\} = \{x \in R^2 \mid x_i > 0, i = 1, 2\}, \quad (5)$$

где E – единичная матрица в R^2 . Здесь и далее будем считать, что многогранный конус не содержит нулевой вектор. Использование векторной упорядоченности (2) позволяет определить в задаче (1) исходы, оптимальные по конусу K .

Определение 1. Ситуация $(x^*, y^*) \in X \times Y$ называется равновесием по конусу K в игре двух лиц с векторным выигрышем (1), если выполнены два условия:

$$\forall x \in X \quad f(x^*, y^*) - f(x, y^*) \notin K,$$

$$\forall y \in Y \quad f(x^*, y^*) - f(x^*, y) \notin K.$$

Множество таких решений будем обозначать $S_K(X, Y)$.

Определение равновесия по конусу в игровой задаче (1) является достаточно общим. Оно включает в себя, как частный случай, равновесие Нэша–Парето (Нэша–Слейтера) [9, с. 233]. Действительно, такие решения получаются в определении 1, если в качестве конуса доминирования использовать конус $R_{\geq}^2$ из (4) (конус $R_{>}^2$ из (5)). Множество равновесий Нэша–Парето (Нэша–Слейтера) будем обозначать $S_P(X, Y)$ ($S_S(X, Y)$).

Как правило, игровая задача с векторными выигрышами имеет бесконечное множество равновесий по Нэшу–Парето. Так, в [1, пример 2.3] (игра «производство – инспекция») исследуются бескоалиционные игры двух лиц с векторными выигрышами. Эта игра имеет бесконечное множество равновесных по Парето решений. Сокращение множества равновесных решений, а тем более выбор единственного наилучшего решения, является важной задачей.

Оптимальность по конусу в многокритериальной задаче

В задаче (1) каждый игрок решает многокритериальную задачу при фиксированном выборе другого игрока. Используем терминологию и обозначения из [10, 11]. Моделью является многокритериальная задача. Это система

$$\langle X, f^{(1)}(x) \rangle. \quad (6)$$

Задано множество допустимых альтернатив $x \in X \subset R^k$, среди которых ЛПР делает свой выбор. Выделен конечный набор желаемых свойств или критериев.

Обычно информацию о всех критериях объединяют в одну векторную функцию выигрыша, в рассматриваемом случае $f^{(1)} : R^k \rightarrow R^2$. Значения этой функции каждому исходу ставят в соответствие количественную оценку для двух свойств $f^{(1)}(x) = f_1^{(1)}(x), f_2^{(1)}(x)$. Не уменьшая общности, считаем, что критерии $f_1^{(1)}(x, y), f_2^{(1)}(x, y)$ являются позитивными [4, с. 55]. Тогда, на содержательном уровне, цель ЛПР состоит в выборе такого исхода, что доставляет возможно большие значения одновременно двум компонентам векторной функции выигрыша $f^{(1)}(x)$.

Определение конусного оптимума основано на векторной упорядоченности (2) и многогранный конус определен посредством матрицы (3). Именно [6, с. 170], оптимальное по конусу решение в многокритериальной задаче (6) есть стратегия $x^* \in X$ выполнения условия

$$\forall x \in X \quad f(x^*) - f(x) \notin K.$$

Свойства такого решения представлены в [6, с. 170–171]. Так, если в многокритериальной задаче (6) множество альтернатив $X \subset R^k$ компактно, векторная функция выигрыша $f^{(1)} : R^k \rightarrow R^2$ непрерывна, конус доминирования K в критериальном пространстве R^2 является выпуклым, острым, то существует альтернатива, оптимальная по конусу K . Пусть в многокритериальной задаче (6) заданы конусы доминирования K_1 и K_2 и $X_1^* \subset X, X_2^* \subset X$ множества альтернатив, оптимальных по конусу K_1, K_2 соответственно. Тогда из $K_1 \subset K_2$ следует включение $X_2^* \subset X_1^*$.

Многокритериальная задача (6) является задачей в условиях неопределенности, именно в условиях неопределенности цели. Для уточнения решения (уменьшения степени неопределенности) можно использовать дополнительную информацию. Это могут быть мнения экспертов. В качестве примера рассмотрим случай, когда два эксперта оценили важность (вес) критериев, причем, например, первый эксперт указал отношение 7:3, а второй эксперт – 1:1. Эти отношения важности критериев верны для первого и второго игроков. Такая информация позволит определить матрицу и соответствующий двухгранный конус:

$$K = \{x \in R^2 \mid Ax = \begin{pmatrix} 7 & 3 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq 0_{(2)}\}.$$

Пример. Рассмотрим первую двухкритериальную задачу, в которой множество стратегий $X = \{x \in R^2 \mid 2x_1 + 5x_2 \leq 10, x_i \geq 0, i = 1, 2\}$. Оно представлено на рис. 1 треугольником ОТР. Задан векторный двухкритериальный выигрыш $f^{(1)}(x) = f_1^{(1)}(x), f_2^{(1)}(x) = (x_1, x_2)$. В такой задаче максимальных по Слейтеру (и по Парето) решений бесконечно много и их множество $X^* = \{x^* \in R^2 \mid 2x_1^* + 5x_2^* \leq 10, x_i^* \geq 0, i = 1, 2\}$. Оно представлено отрезком ТР. В то же время оптимальным по конусу K из (7) является единственный исход $f^{(1)*}(x) = f_1^{(1)*}(x), f_2^{(1)*}(x) = (5, 0)$. По условию многокритериальной задачи оптимальной по конусу K стратегией будет такая же пара $x^* = (x_1^*, x_2^*) = (5, 0)$.

Рассмотрим вторую двухкритериальную задачу (6), в которой множество стратегий

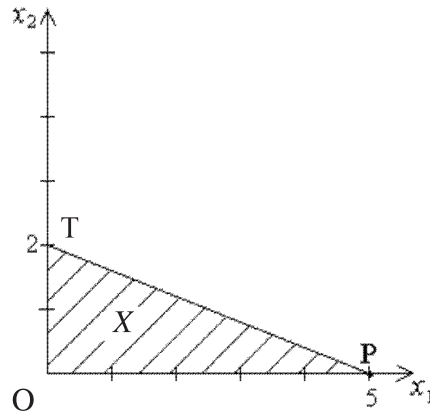


Рис. 1. Множество стратегий для первой двухкритериальной задачи

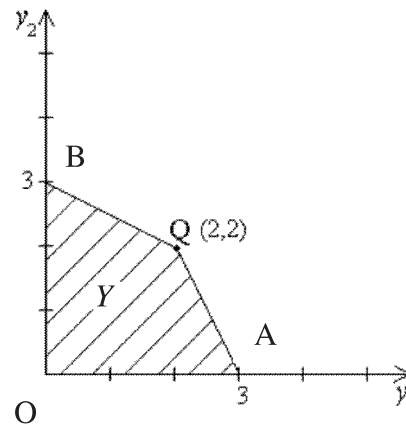


Рис. 2. Множество стратегий для второй двухкритериальной задачи

$Y = \{y \in R^2 \mid 2y_1 + y_2 \leq 6, y_1 + 2y_2 \leq 6, y_i \geq 0, i = 1, 2\}$. Оно представлено на рис. 2 четырехугольником ОАQB. Двухкритериальный векторный выигрыш задан $f^{(2)}(y) = f_1^{(2)}(y), f_2^{(2)}(y) = (y_1, y_2)$. В этой задаче максимальных по Слейтеру (и по Парето) решений бесконечное множество:

$$Y^* = Y^{1*} \cup Y^{2*} = Y^* = Y^{1*} \cup Y^{2*} = \\ = \{y \in R^2 \mid y_1 + 2y_2 = 6, 0 \leq y_1 \leq 2, i = 1, 2\} \cup \\ \{y \in R^2 \mid 2y_1 + y_2 = 6, 2 \leq y_1 \leq 3, i = 1, 2\}.$$

Множество всех таких решений представлено ломаной линией AQB на рис. 2. Оптимальными по конусу K из (7) являются стратегии, представленные отрезком АО, т. е. $Y^k = \{y \in R^2 \mid 2y_1 + y_2 = 6, 2 \leq y_1 \leq 3, i = 1, 2\}$. Таких решений бесконечно много.

Уточнение оптимального по конусу решения

Оптимальных по конусу решений в игровой задаче (1) может быть много. Тогда уточнение по конусу можно применить несколько раз, последовательно уточняя (улучшая) решение. Такой метод в многокритериальной задаче приведен в [6]. Соответствующий подход для игровой задачи можно представить в матричной форме. Рассмотрим следующую последовательность квадратных, невырожденных, неразложимых, стохастических матриц [8, с. 352; с. 381]. Отметим, что это матрицы второго порядка

$$A_1, A_2, \dots, A_i \quad i \in N. \quad (8)$$

Все элементы стохастической матрицы неотрицательны и сумма элементов каждой строки равна единице [8, с. 381]. По последовательности матриц построим новую последовательность

$$B_1 = A_1, B_2 = A_2 \cdot A_1 = A_2 B_1, B_3 = A_3 \cdot A_2 \cdot A_1 = A_3 \cdot B_2, \dots, \\ B_n = A_n \cdot A_{n-1} \cdot \dots \cdot A_1 = A_n B_{n-1}, \dots, n \in N. \quad (9)$$

Каждая матрица из последовательности (9) будет определять многогранный конус аналогично (3). Обозначим конусы этой последовательности как $K_p, i \in N$. Полученная последовательность конусов позволит построить уточнённое по конусу решение в игровой задаче (1).

Утверждение 1. Рассматривается игровая задача (1). Пусть матрицы $A_p, i \in N$ из последовательности (7) являются неотрицательными, невырожденными, неразложимыми, стохастическими. Тогда для любого натурального n :

матрица $B_n = A_n \cdot A_{n-1} \cdot \dots \cdot A_1$ из последовательности (9) является неотрицательной, невырожденной, неразложимой, стохастической;

для соответствующих конусов имеет место включение $K_n \subset K_{n+1}$;

для соответствующих множеств оптимальных по конусу решений в игровой задаче (1) имеет место включение $S_{K_n}(X, Y) \supset S_{K_{n+1}}(X, Y)$.

Каждая матрица $B_i, i \in N$ из последовательности (9) является стохастической и для нее верны условия теоремы Фробениуса [8, с. 355]. У такой матрицы максимальное собственное значение $\lambda_i = 1$. Каждому собственному значению однозначно можно выбрать левый собственный вектор:

$$\alpha^{(n)} = (\alpha_1^{(n)}, \alpha_2^{(n)}), \alpha_1^{(n)} + \alpha_2^{(n)} = 1, \alpha_1^{(n)} > 0, \alpha_2^{(n)} > 0. \quad (10)$$

Учитывая вышеизложенное, для последовательности матриц (9) верно.

Утверждение 2. Пусть матрицы $A_p, i \in N$ из последовательности (8) являются неотрицательными, невырожденными, неразложимыми, стохастическими. Тогда существует предел последовательности матриц (9), т. е.

$$\lim_{n \rightarrow \infty} B_i = \lim_{n \rightarrow \infty} A_n \cdot A_{n-1} \cdot \dots \cdot A_1 = A_0.$$

Матрица A_0 является положительной, вырожденной с рангом равным 1, обе строки матрицы равны

$$\lim_{n \rightarrow \infty} \alpha^{(n)} = \alpha^{(0)} = (\alpha_1^{(0)}, \alpha_2^{(0)}), \alpha_1^{(0)} + \alpha_2^{(0)} = 1, \alpha_1^{(0)} > 0, \alpha_2^{(0)} > 0,$$

где левый собственный вектор $\alpha^{(n)}$ из (10).

Последнее утверждение является основанием для уточнения оптимального по конусу решения игровой задачи (1).

Определение 2. Рассматривается игровая задача (1) и последовательность неотрицательных, невырожденных, неразложимых, стохастических матриц (8). Пусть набор чисел

$$\alpha^{(0)} = (\alpha_1^{(0)}, \alpha_2^{(0)}), \alpha_1^{(0)} + \alpha_2^{(0)} = 1, \alpha_1^{(0)} > 0, \alpha_2^{(0)} > 0$$

представляет строку предельной матрицы A_0 из утверждения 2. Тогда равновесное решение бескоалиционной игры двух лиц со скалярными выигрышами

$$\langle X, Y, f^{(1)}(x, y) = \alpha_1^{(0)} f_1^{(1)}(x, y) + \alpha_2^{(0)} f_2^{(1)}(x, y), \\ f^{(1)}(x, y) = \alpha_1^{(0)} f_1^{(2)}(x, y) + \alpha_2^{(0)} f_2^{(2)}(x, y) \rangle \quad (11)$$

будем называть уточнённым по последовательности матриц (8) решением игровой задачи с век-



торными выигрышами (1) и множество таких решений будем обозначать $S_O(X, Y)$.

Утверждение 3. Пусть в игровой задаче (1) множество стратегий $X \subset R^k$ и $Y \subset R^l$ компактны, векторные функции выигрыша $f^{(1)} : X \times Y \rightarrow R^2$, $f^{(2)} : X \times Y \rightarrow R^2$ непрерывны, квадратные матрицы второго порядка в последовательности (8) являются неотрицательными, невырожденными, неразложимыми, стохастическими. Тогда в задаче существует уточненное по последовательности матриц (8) решение, возможно, в смешанном расширении игры.

Существование уточненного по последовательности матриц решения в задаче (1) следует из компактности множества стратегий X и Y , существования и непрерывности скалярных функций из (11)

$$f^{(1)}(x, y) = \alpha_1^{(0)} f_1^{(1)}(x, y) + \alpha_2^{(0)} f_2^{(1)}(x, y),$$

$$f^{(2)}(x, y) = \alpha_1^{(0)} f_1^{(2)}(x, y) + \alpha_2^{(0)} f_2^{(2)}(x, y)$$

и теоремы о существовании ситуации равновесия в бескоалиционной игре двух лиц со скалярными выигрышами [12, с. 132–133].

Утверждение 4. Уточненное по последовательности матриц (8) решение в игровой задаче (1) является оптимальным (максимальным) по Парето решением, более того, оно является оптимальным (максимальным) по любому конусу, определенному матрицей из последовательности (9).

Уточнение оптимального по Парето (по конусу) решения игровой задачи (1) определяется последовательностью матриц (8). В частности такая последовательность может состоять из постоянных матриц, т. е. в (8) $A_i, i \in N$, где A – произвольная неотрицательная, невырожденная, неразложимая, стохастическая матрица. В этом случае последовательность матриц (9) составляют натуральные степени матрицы A . Суммируя результаты для данного частного случая, получаем

Утверждение 5. Пусть квадратная матрица A второго порядка является неотрицательной, невырожденной, неразложимой, стохастической. Тогда

существует предел последовательности матриц $\lim_{n \rightarrow \infty} A^n = A_0$;

предельная матрица A_0 является неотрицательной, невырожденной, неразложимой, стохастической, и все строки этой матрицы равны левому собственному вектору, относящемуся к максимальному собственному значению матрицы A , т. е.

$$\lambda = 1, \alpha^{(0)} = (\alpha_1^{(0)}, \alpha_2^{(0)}), \alpha_1^{(0)} + \alpha_2^{(0)} = 1, \alpha_1^{(0)} > 0, \alpha_2^{(0)} > 0;$$

для последовательности матриц $A^n, n = 1, 2, \dots$, соответствующая последовательность многогранных конусов $K_n, n = 1, 2, \dots$, определенная аналогично (3), удовлетворяет цепочке включений $K_1 \subset K_2 \subset K_3 \subset \dots \subset K_0$;

соответствующая последовательность оптимальных по конусу решений игровой задачи (1) удовлетворяет включениям

$$S_{K_1}(X, Y) \supset S_{K_2}(X, Y) \supset S_{K_3}(X, Y) \supset \dots$$

$$\dots \supset S_{K_n}(X, Y) \supset \dots \supset S_0(X, Y).$$

Если в определении 2 уточнение оптимального по конусу решения в игровой задаче (1) проводится по последовательности многогранных конусов, определенных степенями неотрицательной, невырожденной, неразложимой, стохастической матрицы A , то полученное решение будем называть уточненным по конусу K решением многокритериальной задачи (1).

Рассмотренные выше методы уточнения оптимального по конусу (по Парето) решения существенно зависят от дополнительной информации, связанной с критериями задачи (1). Если такая информация отсутствует, то решением в (1) может быть только Парето-оптимальный исход. Причем все такие решения равноправны, как решения задачи (1). В то же время информация о важности критериев может возникать из оценок экспертов или из отношения ЛПР к критериям. Такая информация позволяет уточнить оптимальное по Парето решение.

Оценка относительной важности критериев может быть представлена весовыми коэффициентами, т. е. набором положительных (неотрицательных) чисел, отношение между которыми представляет отношение важности соответствующих критериев для эксперта (для ЛПР). Набор положительных чисел можно нормировать так, что сумма коэффициентов равна единице, при этом отношение чисел не изменится. Такое преобразование не только приводит различные мнения экспертов к единой форме, но и позволяет представить информацию от всех экспертов в форме стохастической матрицы. Последнее допускает построение последовательности уточнений.

Если процесс выработки решения в задаче (1) развивается во времени, т. е. информация о важности критериев поступает дискретными порциями, то возникает последовательный процесс уточне-

ния решения. Если на очередном шаге новая информация о критериях отсутствует, то последняя матрица $B_k = A_k \cdot A_{k-1} \cdot \dots \cdot A_1$ уточняет оптимальное по Парето решение. Далее, в условиях отсутствия новой информации, решение уточняется последовательностью конусов, определяемых степенями последней матрицы B_k . Но такую последовательность уточнений, начиная с $k + 1$ шага, можно не выполнять, а сразу перейти к предельной матрице, как она определяется в утверждении 2. Таким образом, применив конечное число уточнений (именно, k), решение задачи (1) может быть сведено к бескоалиционной игровой задаче со скалярными выигрышами.

Модельный пример

Рассматривается игровая задача двух лиц с векторными выигрышами у каждого игрока (1). Множества стратегий первого и второго игрока заданы

$$X = \{x \in R^2 \mid 2x_1 + 5x_2 \leq 10, x_i \geq 0, i = 1, 2\},$$

$$Y = \{y \in R^2 \mid 2y_1 + y_2 \leq 6, y_1 + 2y_2 \leq 6, y_i \geq 0, i = 1, 2\}$$

и представлены на рис. 1 и рис. 2 соответственно. Стратегиями игроков в этой игре являются пары: для первого игрока $x = (x_1, x_2) \in X \subset R^2$, для второго игрока $y = (y_1, y_2) \in Y \subset R^2$. Пара стратегий $(x, y) = ((x_1, x_2), (y_1, y_2)) \in X \times Y \subset R^4$ составляет ситуацию. Векторные функции выигрыша являются двухкритериальными:

$$F(x, y) = (f_1^1(x, y) + f_1^2(x, y), f_2^1(x, y) + f_2^2(x, y)) = (x_1 + y_1, x_2 + y_2) = (x_1, x_2) + (y_1, y_2) = f^{(1)}(x) + f^{(2)}(y).$$

Цель первого (второго) игроков состоит в выборе своей стратегии, что доставляет возможно большие значения своей векторной функции выигрыша $f^{(1)}(x) = (f_1^1(x), f_2^1(x)) = (x_1, x_2)$ ($f^{(2)}(y) = (f_1^2(y), f_2^2(y)) = (y_1, y_2)$).

Задан двухгранный конус (7), который можно определить с помощью стохастической матрицы. Эту матрицу обозначим также A . Тогда конус

$$K = \{x \in R^2 \mid Ax = \begin{pmatrix} 7 & 3 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq 0_{(2)}\} = \{x \in R^2 \mid Ax = \begin{pmatrix} 0,7 & 0,3 \\ 0,5 & 0,5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq 0_{(2)}\}. \quad (12)$$

Учитывая сепарабельный характер векторных выигрышей $F(x, y)$ в игровой задаче, нахождение равновесных ситуаций в (1) сводится к решению

двух двухкритериальных задач для первого и второго игроков:

$$\langle X, f^{(1)}(x) = (f_1^1(x), f_2^1(x)) = (x_1, x_2) \rangle, \quad (13)$$

$$\langle Y, f^{(2)}(y) = (f_1^2(y), f_2^2(y)) = (y_1, y_2) \rangle. \quad (14)$$

Решение многокритериальных задач приведено в разделе «Оптимальность по конусу в многокритериальной задаче». Учитывая вышеизложенное, получаем, что множество равновесных по Нэшу–Парето ситуаций:

$$S_P(X, Y) = X^P \times Y^P = \{x^* \in R^2 \mid 2x_1^* + 5x_2^* = 10, x_i^* \geq 0, i = 1, 2\} \times \{y \in R^2 \mid y_1 + 2y_2 = 6, 0 \leq y_1 \leq 2, i = 1, 2\} \cup \{y \in R^2 \mid 2y_1 + y_2 = 6, 2 \leq y_1 \leq 3, i = 1, 2\}.$$

Среди них выделяется множество равновесных по конусу K (7) ситуаций:

$$S_K(X, Y) = X^K \times Y^K = (5, 0) \times \{y \in R^2 \mid 2y_1 + y_2 = 6, 2 \leq y_1 \leq 3\}.$$

Найдем уточненные по конусу K ситуации. Это решение будет подмножеством $S_K(X, Y)$. Конус определяется стохастической матрицей A из (12). Найдем предел последовательности матриц $\lim_{n \rightarrow \infty} A^n = A_0$. Наибольшее собственное значение матрицы A есть $\lambda = 1$. Тогда соответствующий левый собственный вектор можно выбрать $c = (\frac{5}{8}, \frac{3}{8})$. По утверждению 2 предельная матрица $A_0 = \begin{pmatrix} \frac{5}{8} & \frac{3}{8} \\ \frac{5}{8} & \frac{3}{8} \end{pmatrix}$. В соответствии с определением 2, уточненным по конусу K максимальным решением многокритериальной задачи (13) будут решения задачи математического программирования:

$$f(x) = 5x_1 + 3x_2 \rightarrow \max,$$

$$(x_1, x_2) \in X = \{x \in R^2 \mid 2x_1 + 5x_2 \leq 10, x_i \geq 0, i = 1, 2\}.$$

Аналогично для второго игрока:

$$f(y) = 5y_1 + 3y_2 \rightarrow \max,$$

$$(y_1, y_2) \in Y = \{y \in R^2 \mid 2y_1 + y_2 \leq 6, y_1 + 2y_2 \leq 6, y_i \geq 0, i = 1, 2\}.$$

В первой задаче максимум достигается в единственной точке $x^\# = (x_1^\#, x_2^\#) = (5, 0) \in X$, что доставляет векторный исход $f^{(1)}(x^\#) = (f_1^1(x^\#), f_2^1(x^\#)) = (5, 0) = f^{(1)}(X)$. Во второй задаче максимум достигается в единственной точке



$y^\# = (y_1^\#, y_2^\#) = (2, 2) \in Y$, что доставляет векторный исход $f^{(2)}(y^\#) = (f_1^{(2)}(y^\#), f_2^{(2)}(y^\#)) = (2, 2) = f^{(2)}(Y)$.

Таким образом в представленной игровой задаче двух лиц с векторными двухкритериальными выигрышами у обоих игроков имеется единственная уточненная по конусу K из (7) ситуация:

$$S_o(X, Y) = ((5, 0), (2, 2)).$$

В этой равновесной ситуации первый и второй игроки получают векторные выигрыши $f^{(1)\#}(x) = (f_1^{(1)\#}(x), f_2^{(1)\#}(x)) = (5, 0)$ и $f^{(2)\#}(y) = (f_1^{(2)\#}(y), f_2^{(2)\#}(y)) = (y_1, y_2)$ соответственно.

В данной статье векторное равновесие уточняется посредством расширения отношения предпочтения в R^i , $i \in N$. Тогда, согласно утверждению 1, множество максимальных стратегий сужается, что приводит к уточнению векторного равновесия. Это уточненное решение называется Нэш–А равновесием. Приводится пример, который показывает, что предложенная процедура позволяет выделить единственное векторное равновесное решение в задаче (1).

В статье рассмотрена игровая задача двух лиц с двухкритериальным выигрышем у каждого игрока (1). Для такой задачи имеются общепринятые подходы к определению решения. Они реализованы в концепции равновесной ситуации по Нэшу–Парето [9, с. 233].

Но, как правило, в игровой задаче с векторными выигрышами имеется бесконечное множество таких решений. Все они равноправны относительно определения равновесия по Нэшу–Парето. Возникает проблема уточнения равновесия. Решение следует уточнить так, что с новых уточненных позиций удастся выделить единственное наилучшее решение или, по крайней мере, сократить множество кандидатов на наилучший выбор.

Предложен подход к проблеме уточнения решения в игровой задаче с векторными выигрышами, основанный на расширении отношения

предпочтения. В векторном критериальном R^2 пространстве выделяется конус доминирования K и определяется векторная упорядоченность.

Конусное решение указывает приемлемый компромисс между критериями по мнению экспертов при принятии решения. Если исход оптимален по конусу и от него возможен переход к другому исходу, выигрыш по одному критерию будет большим, то найдется другой критерий и проигрыш по нему будет недопустимо большим.

Конусные решения в многокритериальной задаче имеют важное свойство: если первый конус включает второй конус как подмножество, то множество оптимальных по первому конусу решений является подмножеством решений, оптимальных по второму конусу. На этом свойстве построена процедура уточнения решения.

Конусные отношения можно задавать в форме стохастических матриц. В этом случае последовательное уточнение конусной упорядоченности соответствует умножению стохастических матриц. Предлагается следующий алгоритм: на основании мнения экспертов строится последовательность расширяющихся конусов, которой соответствует последовательность вложенных множеств конусных решений. Решение по предельному конусу называется уточненным по последовательности конусов оптимальным решением многокритериальной задачи.

На основании свойств оптимальности в многокритериальной задаче предложена концепция уточнения по конусу равновесной ситуации в игровой задаче двух лиц с векторными выигрышами у каждого игрока. Проведено обоснование алгоритма уточнения решения. Указаны свойства последовательности стохастических матриц, которые гарантируют существование предельного конуса. Рассмотрен модельный пример, в котором вычисляется единственная уточненная по конусу ситуация.

СПИСОК ЛИТЕРАТУРЫ

1. **Blackwell, O.** An analog of the Minimax Theorem for Vector Payoff [Текст]/O. Blackwell//Pacific Journal of Mathematics. –1956.–Vol. 6.–P. 1–8.
2. **Shaplay, L.S.** Equilibrium Points in Games with Vector Payoffs [Текст]/L.S. Shaplay//Naval Research Logistics Quarterly. –1959. – Vol. 6. – P. 57–61.
3. **Megen, F.** A preference concept for multicriteria game. [Текст] / F. Megen, P. Borm, S. Tijs // Mathematical

Methods of Operation Research. –1999. – Vol. 49. – № 3. P. 401–412.

4. **Розен, В.В.** Математические модели принятия решений в экономике. [Текст] / В.В. Розен – М.: Высш. шк., 2002.

5. **Yu, P.L.** Cone convexity, cone extreme points and nondominated solutions in decision problems with multiobjectives [Текст] / P.L. Yu // Journal of optimization

theory and application. –1974. Vol. 14. – № 3. – P. 319–377.

6. **Матвеев, В.А.** Исследование оптимальности по конусу в многокритериальной задаче [Текст] / В.А. Матвеев // Научно-технические ведомости СПбГПУ. – 2009. – № 4. – С.169–176.

7. **Беклемишев, Д.В.** Дополнительные главы линейной алгебры [Текст] / Д.В. Беклемишев. –М.: Наука, 1983.

8. **Гантмахер, Ф.Р.** Теория матриц [Текст] / Ф.Р. Гантмахер. –М.: Наука, 1967.

9. **Жуковский, В.И.** Линейно-квадратичные дифференциальные игры [Текст]/В.И. Жуковский, А.А. Чикрий. –Киев: Наукова думка, 1994.

10. **Подinovский, В.В.** Парето-оптимальные решения многокритериальных задач [Текст] / В.В. Подinovский, В.Д. Ногин. –М.: Наука, 1982,

11. **Ногин, В.Д.** Принятие решений в многокритериальной среде: количественный подход [Текст] / В.Д. Ногин. –М.: Физматлит, 2002.

12. **Петросян, Л.А.** Теория игр [Текст] / Л.А. Петросян, Н.А. Зенкевич, Е.А. Семина. –М.: Высш. шк. Книж. дом «Университет», 1998.

13. **Васильев, Ф.П.** Численные метода решения экстремальных задач [Текст] / Ф.П. Васильев. –М.: Наука, 1980.

УДК 658

Е.В. Бабкова, С.С. Шерстюк

МОДЕЛЬ И АЛГОРИТМ КАЛЕНДАРНОГО РАСПРЕДЕЛЕНИЯ ВЗАИМОЗАМЕНЯЕМЫХ РЕСУРСОВ С УЧЕТОМ ФРОНТА ВЫПОЛНЕНИЯ РАБОТ

Календарное планирование является конечным этапом оперативной деятельности сложной системы и поэтому в значительной мере определяет уровень использования ресурсов и время выполнения работ. Учет в одной задаче исследования этих двух элементов, а именно, затрат ресурсов и время выполнения работ, приводит к появлению понятия «фронта работ». Изначально оно связывалось только со строительной отраслью, поскольку здесь требовалось специальное оснащение и подготовка рабочего места к выполнению работ. В настоящее время во многих сферах деятельности используется понятие «фронта работ», например:

по отраслям деятельности (полевой, аграрной, машиностроения, строительства, дорожной и проч.);

в различных сферах деятельности – подразумевается объем работ, который необходимо выполнить в физическом выражении (количество единиц работы, квадратные метры покрытия и т. п.);

в организационной и управленческой деятельности (фронт управления, фронт работ министерств, фронт контроля и т. п.) [2, 4, 7–10].

Различают фронт работ, рассчитанный на этапе объемно-календарного планирования с целью формирования укрупненных планов работы объекта по календарным периодам. В данной статье будет применяться более узкое понимание «фронта работ»: текущий фронт выполнения работ – это множество работ, которые могут быть начаты в какой-то период времени с учетом работ, начатых ранее, но еще не законченных к настоящему моменту (предполагается, что работы не могут быть прерваны).

В статье рассматривается постановка задачи и модель календарного распределения взаимозаменяемых ресурсов. Кроме того, приводится эвристический алгоритм решения поставленной задачи.

Рассмотрим задачу календарного распределения взаимозаменяемых ресурсов с учетом фронта выполнения работ [1, 3, 5, 6].

Постановка задачи исследования

Имеется фронт работ, сформированный на этапе объемно-календарного планирования $\Phi_1, \Phi_2, \dots, \Phi_m, i = 1, 2, \dots, m$. Пусть при выполнении какого-то вида работ на отрезке времени $[t_1, t_2]$ используется r ресурсов. Ежедневное наличие ресур-

сов обозначим следующим образом:

$$B_1(t), B_2(t), \dots, B_r(t), \dots, B_D(t),$$

где $B_r(t)$ – наличие ресурса r -го вида в момент времени t .

Работы необходимо выполнить не позднее календарных сроков $t_1^B, t_2^B, \dots, t_n^B$. Для каждой работы из списка $R_1, R_2, \dots, R_n$ задана технологическая последовательность использования ресурсов $W_{j1}, W_{j2}, \dots, W_{jv_j}$ и известна интенсивность d_{ijv} применения v -го ресурса для j -й работы из i -го фронта.

В задаче используются следующие допущения:

длительность работ измеряется в тех же календарных единицах, что и потребность в ресурсах (дни, часы и т. п.), причем для удобства решения задачи все показатели календарного времени нормированы по одной условной шкале времени с фиксированной точкой отсчета; для представления решения задачи в календарных датах (час, день, месяц, смена) применяется специальная операция определения календарной даты;

для выполнения работ используется v видов ресурсов $S_1, S_2, \dots, S_v$;

при выполнении работ применяется оборудование, которое также является ресурсом и имеет свой v -й номер группы ресурса;

для каждой j -й работы известны коэффициенты μ_{jqqs} эквивалентной замены q -го ресурса на s -й ресурс внутри v -й группы ресурсов;

определим текущий фронт работ $\Phi(t)$ в каждый промежуток времени $[t_1, t_2]$, зная сроки выполнения фронтов и дату выполнения всего комплекса работ T^0 :

$$\Phi(t) = \Phi_H \cup \Phi_3 \cup \Phi_{II},$$

где Φ_H – множество работ, которые могут быть начаты в момент времени t ; Φ_3 – ранее начатые, но еще не законченные работы (предполагается, что работы не могут быть прерваны); Φ_{II} – работы, которые необходимо переделать.

Необходимо определить список выполнения работ на каждую текущую дату t на отрезке $[t_1, t_2]$ и календарный график выполнения работ, эффективный по заданному критерию оптимальности.

Под календарным графиком понимается список работ с указанием календарной даты начала выполнения работы (день, смена, час) и указани-

ем порядкового номера используемого ресурса (группы ресурса), объема ресурса, номера взаимозаменяемого ресурса, фронта работы.

Пусть в результате агрегирования работ всех фронтов на отрезке $[t_1, t_2]$ должны выполняться n видов работ $R_1^1, R_2^2, \dots, R_j^i, \dots, R_n^m$.

Введем следующие обозначения:

i – номер фронта работ, $i = 1, 2, \dots, m$;

j, k – номера работ, $j = 1, 2, \dots, n$; $k = 1, 2, \dots, n$;

v – код группы взаимозаменяемых ресурсов, $v = 1, 2, \dots, G_v$;

q, s – текущие номера взаимозаменяемого ресурса в v -й группе, $q = 1, 2, \dots, Q_v$; $s = 1, 2, \dots, S_v$;

t_{jv} – время выполнения j -й работы v -м ресурсом, $j = 1, 2, \dots, n$, $v = 1, 2, \dots, G_v$;

t_j^0 – нормированная дата окончания j -й работы;

x_{jv} – нормированная дата начала выполнения j -й работы с использованием ресурса из v -й группы, $j = 1, 2, \dots, n$;

d_{ijv} – интенсивность использования v -го ресурса при выполнении j -й работы из i -го фронта работ;

y_{kj} – коэффициент, учитывающий последовательность выполнения работ: j -я работа выполняется после k -й работы (или наоборот), в зависимости от значения коэффициента;

u_{jqvq} – коэффициент, учитывающий использование (или нет) q -го ресурса из v -й группы при выполнении j -й работы;

z_{qs} – коэффициент, учитывающий заменяемость q -го ресурса на s -й ресурс внутри одной группы ресурсов.

Сформулируем модель календарной задачи, используя идею, изложенную в [1, 3, 6].

Математическая модель календарной задачи

Модель. Для заданных ρ_{ijv} , μ_{jqqs} , t_{jv} и при условии целочисленности переменных y_{kj} , u_{jqvq} , z_{qs} , x_{jv} , $k = 1, 2, \dots, n$; $j = 1, 2, \dots, n$; $v = 1, 2, \dots, G_v$; $q = 1, 2, \dots, Q_v$ найти y_{kj} , u_{jqvq} , z_{qs} и x_{jv} , удовлетворяющие условиям:

$$1) y_{kj} = \begin{cases} 0, j\text{-я работа выполняется после } k\text{-й работы;} \\ 1, j\text{-я работа выполняется после } k\text{-й работы.} \end{cases}$$

$$2) u_{jqvq} = \begin{cases} 0, v\text{-й ресурс не используется в } j\text{-й работе;} \\ 1, v\text{-й ресурс используется в } j\text{-й работе.} \end{cases}$$

$$3) z_{qs} = \begin{cases} 0, q\text{-й ресурс не заменяется на } s\text{-й} \\ \text{ресурс;} \\ 1, q\text{-й ресурс заменяется на } s\text{-й ре-} \\ \text{сурс.} \end{cases}$$

4) $x_{jv} \geq 0$ – целые, y_{jk} – целые, u_{jvq} – целые; z_{qs} – целые, $j = 1, 2, \dots, n$; $k = 1, 2, \dots, n$; $v = 1, 2, \dots, G_v$; $q = 1, 2, \dots, Q_v$; $s = 1, 2, \dots, S_v$.

5) Для текущего выполнения выбирается только одна работа

$$y_{kj} + y_{jk} = 1, (\forall j, k). \quad (1)$$

6) Выполнение j -й работы v -м ресурсом должно закончиться не позднее планового периода (или смены)

$$u_{jv}(x_{jv}, t_{jv}) \leq t_j^0, (\forall j, k, v, s). \quad (2)$$

7) Ограничения по наличию ресурсов

$$\sum_{j=1}^n (u_{jvq} \cdot t_{jvq} \cdot d_{jvq}) + \sum_{s=1}^{S_v} z_{qs} u_{qs} t_{jvs} \cdot d_{jvs} \leq B_v. \quad (3)$$

8) Работа может выполняться только тогда, когда закончится выполнение начатой ранее работы

$$y_{jk} \cdot x_{jv} \geq y_{kj} \cdot x_{kv} + t_{kv}, \text{ или } y_{kj} \cdot x_{kv} \geq y_{jk} \cdot x_{jv} + t_{jv}. \quad (4)$$

9) В качестве целевой функции модели может быть выбрана минимизация времени выполнения работ по всем фронтам:

$$F = \min \left\{ \max_{vi,j} (x_{jvq} + t_{jvq}) \right\}. \quad (5)$$

Поставленная задача является целочисленной и нелинейной. Для ее решения предлагается алгоритм, основанный на вычислении индексов приоритетов для работ и ресурсов, в соответствии с которыми устанавливается очередность выполнения работ. Пересчет индексов приоритетов осуществляется после каждого такта закрепления ресурса за очередной работой.

Алгоритм решения задачи

Рассмотрим этапы алгоритма определения параметров календарной задачи распределения ресурсов по работам. Для простоты изложения опустим формулы для расчета приоритетов [1].

Этап 1. Определение фронта работ для $t = 1$ в предположении, что имеется одна начальная исходная работа и начальное время выполнения работ (использования ресурсов) $T_1 = 0$. $\Phi(t = 1) = \Phi(1) = \Phi_1$

Этап 2. Определение функций предпочтения фронтов работ нормированных циклов вы-

полнения работ; определение индексов приоритета фронтов, работ, ресурсов.

Множество фронтов, упорядоченных в соответствии с индексами приоритета, обозначим $\Phi_1^y, \Phi_2^y, \dots, \Phi_m^y$.

Упорядоченное множество работ из i -го фронта работ обозначим $R_{i1}, R_{i2}, \dots, R_{in}$.

Ресурсы, упорядоченные в соответствии с моментами освобождения от использования, обозначим $S_1, S_2, \dots, S_q$.

Этап 3. Выполнение такта закрепления очередной работы из текущего фронта за ресурсом.

При анализе очередной работы и наличия ресурса может иметь место, по крайней мере, одно из следующих соотношений:

время окончания последней работы больше планового срока выполнения работ. В этом случае перечень работ не будет выполнен – окончание алгоритма;

время освобождения ресурса больше, чем время начала выполнения работы. В этом случае работа ждет ресурс, либо подбираем ресурс из эквивалентной замены. Если имеем противоположную ситуацию (время освобождения ресурса меньше, чем время начала выполнения работы), то ресурс ожидает работу;

нет работ на выполнение – окончание алгоритма.

Этап 4. Определение потребности в ресурсах на первый период:

$$b_{1v} = \sum_{j \in \Phi(1)} p_{1jv}, \quad (6)$$

где b_{1v} – потребность в v -м ресурсе на первый период; p_{1jv} – интенсивность использования v -го ресурса для j -й работы в первый период.

Этап 5. Определение потребности в ресурсе b_t на последующие периоды $t > 1$ для выполнения ранее начатых работ из $\Phi(1)$, т. е. для каждой работы $(1, j)$, у которой $t_{1j} > 1$, изменяются параметры $b_t, t = 2, \dots, t_{1j}$:

$$b_{tv} = b_{(t-1)v} + \beta_{1jv}, \quad (7)$$

где b_{tv} – величина v -го ресурса в момент t ; $b_{(t-1)v}$ – величина v -го ресурса в предыдущий момент времени $(t - 1)$; β_{1jv} – объем использования v -го ресурса для работ $(1, j)$, $\beta_{1jv} = t_{1j} \cdot p_{1jv}$.

После пересмотра всех работ из $\Phi(1)$ переходим к общему шагу алгоритма.

Этап 6. Выполнение общего шага алгоритма. При рассмотрении очередного момента времени k определяем фронт работ $\Phi(k)$ и потребность в ресурсе S_v на период k :

$$S_v = b_v + \sum_{(j,q) \in \Phi(k)} \beta_{jq}, \quad (8)$$

где $\Phi(k)$ – фронт работ, которые могут быть начаты в момент k ; b_{kv} – потребность в v -м ресурсе в момент времени k .

Алгоритм заканчивается при просмотре конечной работы фронта работ.

Этап 7. Определение календарных дат начала и окончания выполнения очередной работы.

Этап 8. Определение составляющей критериальной функции для текущего отрезка времени $[t_1, t_2]$.

Этап 9. Оформление расписания выполнения работ с использованием v -го ресурса.

После выполнения работы R_j приоритеты работ и ресурсов изменились, их необходимо пересчитать.

Календарный график считается построенным (или прекращается его расчет), когда выполнено одно из условий:

выполнены все работы из множества $R_1, R_1, \dots, R_n$ на отрезке $[t_1, t_2]$;

время выполнения последней работы больше, чем срок окончания периода планирования;

не хватает ресурса или взаимозаменяемого ресурса для какой-нибудь работы из множества $R_1, R_1, \dots, R_n$.

В заключение отметим возможности рассмотренной задачи с позиций учета некоторых особенностей реальных календарных задач распределения ресурсов.

На практике централизованное поступление ресурса в систему осуществляется периодически (например, один раз в квартал), равными объемами. Пусть стратегия поступления ресурса на отрезке $[T^0, T]$ изменилась: происходит уменьшение или увеличение объема B_{vk} . В момент времени $t \in (T^0, T)$ возникает задача перераспределения ресурса между фронтами работ. В этом случае необходимо решать новую календарную задачу целочисленного линейного программирования по распределению ресурсов.

Рассмотренный эвристический алгоритм календарного планирования позволяет достаточно просто реализовать процедуру моделирования, если предположить, что в качестве очередного опыта моделирования будет производиться случайный розыгрыш работы на выполнение из фронта работ, а также очередной объем требуемого ресурса.

Данное исследование осуществлялось при частичной поддержке РФФИ (грант № 06-07-89228-а).

СПИСОК ЛИТЕРАТУРЫ

1. **Бабкова, Е.В.** Модель сменной загрузки раскройного оборудования [Текст]/Е.В. Бабкова//Принятие решений в условиях неопределенности: межвуз. науч. сб.–Уфа: УГАТУ, 2002.–С. 184–190.
2. **Бабкова, Е.В.** Моделирование социальных и эколого-экономических систем: учеб. пособ. [Текст]/Е.В. Бабкова, Т.О. Вишнякова, С.С. Шерстюк.–Уфа: УГАТУ, 2009.–184 с.
3. **Бабкова, Е.В.** Модель календарного распределения ресурсов [Текст]/Е.В. Бабкова, С.С. Шерстюк//Принятие решений в условиях неопределенности: межвуз. науч. сб.–Уфа: УГАТУ, 2009. –Вып. 5.–С. 249–253.
4. **Бабкова, Е.В.** Управление многоцелевым календарным распределением ресурсов с учетом фронта выполнения работ [Текст]/Е.В. Бабкова, С.С. Шерстюк //

Принятие решений в условиях неопределенности: межвуз. науч. сб. –Уфа: УГАТУ, 2009. –Вып. 6.–С. 317–326.

5. **Вагнер, Г.** Основы исследования операций. Т. 2. [Текст]/Г. Вагнер. –М.: Мир, 1973.–488 с.

6. **Babcova, E.V.** Model of changeable cutting equipment loading [Текст]/E.V. Babcova//Decision Making in the Condition of Uncertainty (Cutting-Packing Problems). –Ufa, 1997.–P. 359–366.

7. www.agro.ru/news.aspx [Электронный ресурс]

8. www.dni.ru/polit.2008.12/html [Электронный ресурс]

9. www.PromKrovlya.ru/content/view/145/47 [Электронный ресурс]

10. stroymontagh.ru/front.rabot.html [Электронный ресурс]

МОДИФИКАЦИЯ МЕТОДА ВЕТВЕЙ И ГРАНИЦ ДЛЯ ПОИСКА ОПТИМАЛЬНОГО РЕШЕНИЯ ЗАДАЧИ ОДНОМЕРНОЙ УПАКОВКИ ПРЯМОУГОЛЬНО-ОРИЕНТИРОВАННОЙ СТРУКТУРЫ

В качестве основной рассматривается следующая задача прямоугольно-ориентированной одномерной упаковки (1D Rectangular Oriented Cutting Stock Problem, 1ROCSP): в одномерные контейнеры длины W требуется упаковать предметы заданной длины w_i в количестве b_i (потребности), $i \in I := \{1, \dots, m\}$, соблюдая при этом следующие условия: в один и тот же контейнер можно упаковывать предметы только с различными номерами; если при упаковке предмета с номером $i \in I$ в текущий контейнер такие предметы ещё остались не упакованными, то предмет с этим же номером упаковывается в следующий контейнер. Требуется *минимизировать* количество упакованных контейнеров. При этом все величины – целые.

Описанная задача отвечает блочной структуре двухмерной упаковки, которая была введена Э.А. Мухачевой и описана в 1998 г. в [1]. Одновременно с блок-структурами двухмерной упаковки в [2] рассматриваются соответствующие им прямоугольно-ориентированные одномерные упаковки, уточнения свойств которых приведены в [3]. Независимо от вышеуказанных авторов, поставленная задача рассматривалась итальянским исследователем М. Monachi в формулировке задачи одномерной продолженной упаковки (1D Contiguous Bin Packing Problem, 1CBPP), которая тождественна задаче 1ROCSP. В работе [4] показано, что оптимальное решение задачи 1CBPP является нижней границей для двухмерной упаковки. А.С. Мухачевой доказано, что значение решения задачи ROCSP, отвечающей индивидуальной задаче прямоугольной упаковки, не превосходит ее решения [5]. Отсюда вытекает тот же результат, который был получен М. Monachi. На этом факте построены алгоритмы локального поиска [5, 6] для задачи двухмерной упаковки. Наряду с этим задача 1CBPP также имеет самостоятельное применение, например, в области расписания [7].

На основе результатов М. Monachi и А.С. Мухачевой автором разработан метод точного

решения задачи ROCSP для поиска улучшенной нижней границы для двухмерной упаковки. Он является модификацией метода ветвей и границ, основные процедуры которого – ветвление и подсчет нижней границы частичных упаковок. Реализация этих процедур описана в статье, также приведены результаты численного эксперимента.

Ветвления

В процессе комбинаторного поиска множество кодов задает пространство решений, и исходная задача нахождения оптимального решения сводится к поиску наилучшего элемента (кода) в этом пространстве. Необходимо, чтобы это пространство было P -допустимым.

Определение 1. Пространство представлений решений называется P -допустимым, если оно удовлетворяет следующим требованиям: 1) пространство кодов конечно; 2) каждый элемент кода соответствует допустимому решению; 3) решение задачи может быть получено по его коду за полиномиальное время; 4) наилучший элемент в пространстве кодов соответствует оптимальному решению исходной задачи.

Утверждение 1. Множество перестановок элементов списка $\langle i_1, \dots, i_m \rangle$ определяет P -допустимое пространство решений задачи 1ROCSP размерности m , которое состоит из $m!$ перестановок, каждая из которых может быть трансформирована в допустимую упаковку за время $O(m \cdot \log m)$ и по меньшей мере одна из них соответствует оптимальному решению задачи 1ROCSP.

Для восстановления упаковки по ее коду (перестановке) предлагается использовать правило «следующий подходящий» (Next Fit, NF). Рассмотрим приоритетный список $\langle i_1, \dots, i_m \rangle$, который определяет порядок последовательного размещения предметов. Пара $i_p, i_{p+1} \in I$ означает, что предмет i_{p+1} будет размещен следующим по

порядку после i_l . Если предмет i_{l+1} не уместается в текущий контейнер, то осуществляется переход к следующему контейнеру. Если при размещении предмета такой же предмет ещё не упакован, то он размещается в следующем контейнере. Обозначим $NF(\langle i_1, \dots, i_m \rangle)$ факт получения упаковки контейнера применением правила NF к приоритетному списку $\langle i_1, \dots, i_m \rangle, i_l \in I, l = \{1, \dots, m\}$.

Принцип ветвлений состоит в последовательном и направленном размещении каждого элемента множества I . Посредством этого строится дерево ветвлений $T = (V, E)$, листья которого представляют собой множество всех перестановок элементов из I . С каждым узлом $u \in V$ связана упорядоченная последовательность $u := \langle i_1, \dots, i_m \rangle$ размещенных предметов $i_l \in I, l = 1, \dots, d; d \leq m$. При этом каждому узлу $u \in V$ дерева T соответствует однозначно получаемая с помощью алгоритма $NF(u)$ частичная упаковка $\langle (\alpha^1, \chi_1), \dots, (\alpha^{\sigma(u)}, \chi_{\sigma(u)}) \rangle$, или блок-структура (по аналогии с двухмерной упаковкой [1, 2]), в которой предметы $\bar{I}(u) := I \setminus \{i_1, \dots, i_d\}$ должны быть размещены; α^j – вектор упаковки блока j , а χ_j – интенсивность (длина) блока j . Например, на рис. а имеется частичная прямоугольно-ориентированная одномерная упаковка, состоящая из четырех блоков. Ей соответствует блок-структура $SS = \langle (1, 2, 3) 2; (1, 3, 4) 1; (1, 3) 4; (3) 2 \rangle$, которая получается указанием номеров предметов, пересекающих каждый блок по направлению снизу-вверх, и интенсивности блоков. Ребро (u, v) , где $u, v \in V, u \neq v$, принадлежит множеству E , если подзадача v получается из u размещением предмета $i \in \bar{I}(u)$, т. е. $v = \langle u, i \rangle$.

Очевидно, что при таком способе кодирования порождается большое количество эквивалентных

упаковок. В процессе ветвлений все эквивалентные упаковки отсекаются из рассмотрения с помощью правил отсечения, кроме одной, которая называется доминирующей. Рассмотрим процесс исключения некоторых упаковок.

Определение 2. Пусть дана упаковка $u = \langle i_1, \dots, i_k, \dots, i_q, \dots, i_d \rangle \in V$ с ее блок-структурой $SS(u)$, где $k < q$ и $i_k, i_q \in I_j^-$, $I_j^- = \{i \in I_j : i \notin I_{j-1}\}, j \in \{1, \dots, \sigma(u)\}$. Упаковку $u' = \langle i_1, \dots, i_q, \dots, i_k, \dots, i_d \rangle \in V$ с блок-структурой $SS(u)$ назовем вертикально эквивалентной упаковке u .

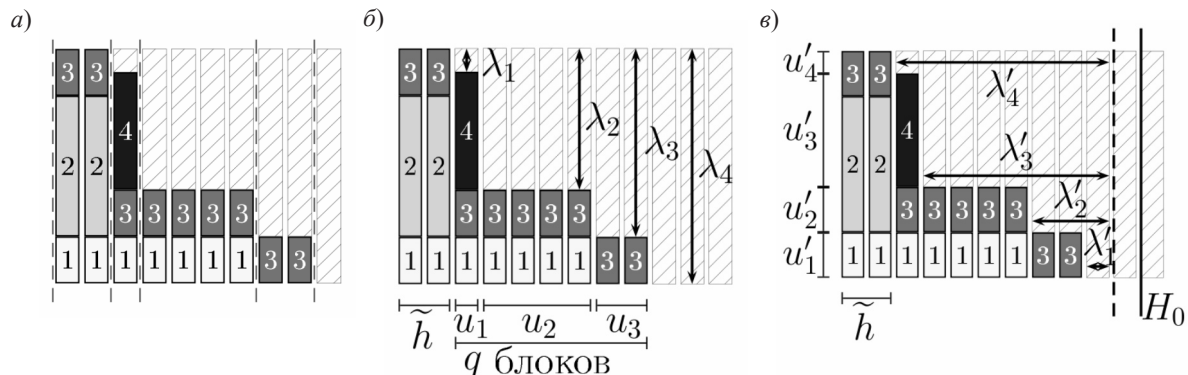
Далее вводится следующее правило отсечения. Пусть дан узел $u = \langle i_1, \dots, i_d \rangle \in V$ с блок-структурой $SS(u)$, если при размещении следующего предмета он помещается в текущий контейнер, а его номер меньше наибольшего номера предмета, который впервые начинает размещаться в этом контейнере, то узел $v = \langle i_1, \dots, i_d, i^* \rangle$ отсекается из рассмотрения. Для корректности применения введенного правила отсечения доказывается следующее утверждение.

Лемма 1. Пусть дана оптимальная упаковка $u = \langle i_1, \dots, i_d \rangle \in V$ с блок-структурой $SS(u)$. Если существует блок $S_j \in SS(u)$, в котором имеются такие элементы $k, q \in \{1, \dots, m\}$, что

$$i_k, i_q \in I_j^-, \quad k < q, \quad i_k > i_q, \quad (1)$$

то существует оптимальная упаковка $u' \in V$, для которой условия (1) не выполняются.

Таким образом, после применения правила и отсечения частичной упаковки, ветвлением которой могла быть получена оптимальная упаковка, останется хотя бы одна частичная, для которой условия леммы не выполняются. Ветвлением оставшейся после отсечения частичной упаковки будет получена оптимальная упаковка.



Частичная прямоугольно-ориентированная одномерная упаковка:
 а – блок-структура упаковки; б, в – горизонтальная и вертикальная релаксации

Аналогично описанному правилу в рассмотрение вводятся правила отсечения горизонтально эквивалентных упаковок, эквивалентных упаковок для идентичных предметов и другие правила отсечения.

Нижние границы

На рис. 6 изображена полученная с помощью алгоритма $NF(u)$ блок-структура $SS(u)$ узла $u = \langle i_1, \dots, i_d \rangle \in V$. Остатки в контейнерах имеют длины λ_k в количестве u_k .

Сформулируем для узла $u \in V$ следующую задачу. Имеются прутки материала различной длины λ_k в комплектности u_k , из которых необходимо вырезать предметы длины w_i в количестве b_i , $i \in I(u)$, при этом соблюдая два условия: каждый предмет w_i вырезается только один раз из прутка и комплектность u_k каждого типа прутка не должна быть превышена заданной. Необходимо минимизировать количество затраченных прутков типа $\lambda_{q(u)}$. Эта задача известна как проблема одномерного раскроя с различными длинами прутков (1D Cutting Stock Problem with Multiple Stock Lengths, 1CSPM) [8]. Для ее решения предложена релаксация на основе линейного программирования $L1CSPM(u)$, которая решается с использованием метода генерации столбцов. Обозначим z_{LP}^* значение ее оптимального решения. Имеет место следующее утверждение.

Лемма 2. Пусть дан узел $u = \langle i_1, \dots, i_d \rangle \in V$, $i_l \in I$, $l = 1, \dots, d$, $d \leq m$. Число

$$lb_{LP}(u) := \sum_{j=1}^{\sigma(u)} \chi_j + \left[z_{LP}^*(u) \right] \quad (2)$$

является нижней границей для значений решений, которые могут быть получены ветвлением узла u .

С другой стороны, если известна верхняя граница H_0 (например, полученная приближенно любым эвристическим методом), то можно проверить, уместятся ли оставшиеся предметы в контейнеры до $H_0 - 1$. Если нет, то данный узел может быть удален из рассмотрения. Аналогично $L1CSPM(u)$ формируется задача вертикальной релаксации $L1CSPM'(u, H_0)$, где предметы b_i в потребности λ'_k вырезаются из прутков длины в комплектности u'_k (см. рис. 6). При этом контейнеры последнего типа доступны в ограниченном количестве. Пусть $z_{LP}^*(u, H_0)$ – значение оптимального решения задачи $L1CSPM'(u, H_0)$, тогда справедливо следующее утверждение.

Лемма 3. Пусть дан узел $u = \langle i_1, \dots, i_d \rangle \in V$, $i_l \in I$, $l = 1, \dots, d$, $d \leq m$. Если

$$z_{LP}^*(u, H_0) \geq \varepsilon, \quad \varepsilon > 0, \quad (3)$$

то решение, которое будет получено ветвлением узла u , будет иметь значение не лучше, чем верхняя граница H_0 для значения целевой функции.

В процессе ветвления узла $u \in V$ создается узел v , при этом $(u, v) \in E$, только если нижняя граница (2) меньше, чем верхняя граница для дерева T , и не выполнено условие (3). Имеет место следующее утверждение.

Теорема 1. Алгоритм, основанный на методе ветвей и границ на приоритетных списках с учетом нижней границы (2) и условия (3), является точным.

Численный эксперимент

В связи с тем, что решение задачи 1ROCSP является нижней границей для 2SPP, численный эксперимент проводился на задачах 2SPP библиотеки задач Bortfeld [9]. Результаты сравнивались с известными нижними границами. Программное обеспечение было реализовано на C++ и скомпилировано gcc 4.1.2. Вычислительные эксперименты проводились на AMD Opteron 250 (2,4 GHz).

Библиотека состоит из 500 примеров, которые были случайно сгенерированы и поделены на десять классов. Задачи из первых шести классов предоставлены Berkey и Wang. Остальные четыре класса предоставили Martello и Vigo. Ширина полосы для каждого класса $W = 100$. Число предметов в каждом классе 20, 40, 60, 80 и 100. Для каждого числа предметов сгенерировано по десять задач.

Для каждого примера Bortfeld предложил нижнюю границу, среднее значение которой для каждого класса приведено в табл. 1 в столбце Bortfeld(B). Для сравнения приводится значение оптимального решения непрерывной релаксации задачи одномерного раскроя прутков (столбец LP), округленное сверху. В столбце 1ROCSP(C) приведено среднее значение решений 1ROCSP, из которых в шестом столбце указано число задач, решенных оптимально. Для остальных задач, которые не были решены оптимально, приведено минимальное значение нижней границы среди всех необработанных узлов. На каждый пример налагалось ограничение времени счета в 3 мин.

Случаи, когда одна из границ превышает другие, приведены в табл. 2, из которой видно, что в 178 и 184 случаях нижние границы LP (столбец

Таблица 1

Среднее значение нижних границ для задач Berkey & Wang и Martello & Vigo

Класс	N	Bortfeld(B)	LP	1ROCSP(C)	Кол-во опт.
cl01	50	187,18	187,68	187,74	33
cl02	50	60,52	60,52	60,52	33
cl03	50	504,12	507,62	507,84	12
cl04	50	193,50	193,50	193,50	0
cl05	50	1613,16	1630,66	1631,72	26
cl06	50	506,40	506,40	506,40	0
cl07	50	1577,82	1588,76	1591,24	46
cl08	50	1397,92	1399,58	1399,84	0
cl09	50	3343,10	3344,56	3346,16	50
cl10	50	909,16	917,58	917,70	6
	500	1029,29	1033,69	1034,27	206

Таблица 2

Сравнение абсолютных значений нижних границ
Bortfeld(B), LP и 1ROCPS(C)

Класс	B>LP	B>C	LP>B	LP>C	C>B	C>LP
cl01	1	0	15	0	16	3
cl02	0	0	0	0	0	0
cl03	0	0	33	0	36	6
cl04	0	0	0	0	0	0
cl05	1	0	44	0	45	10
cl06	0	0	0	0	0	0
cl07	9	0	27	0	27	16
cl08	0	0	15	0	16	6
cl09	3	0	5	0	5	4
cl10	0	0	39	0	39	1
	14	0	178	0	184	46

LP>B) и точное решение задачи 1ROCSP (столбец C>B) соответственно больше, чем нижняя граница приведенная Bortfeld. В 46 случаях 1ROCSP оказалась лучше, чем нижняя граница LP (столбец C>LP). Из столбцов B>C и LP>C видно, что нижняя граница, основанная на точном решении 1ROCSP, на данных задачах строго лучше, чем две другие, известные из литературы, нижние границы.

Результаты численного эксперимента:

На классах cl04, cl06, cl08 не удалось решить оптимально ни одного примера. Эти классы можно обозначить как наиболее сложные для разработанного алгоритма. И, напротив, практически все задачи из классов cl07, cl09 были решены оптимально. В целом чуть меньше половины задач было решено оптимально, что говорит об удовлетворительной производительности алгоритма.

Нижняя граница на основе решения задачи 1ROCSP всегда не меньше границ Bortfeld, LP на примерах библиотеки задач Bortfeld.

В данной статье предложен точный алгоритм на основе метода ветвей и границ с использованием линейного программирования. Основываясь на том факте, что значение решения прямоугольно-ориентированной одномерной задачи является нижней границей для задачи двухмерной упаковки, исследование эффективности проводилось на задачах двухмерной упаковки. Несмотря на большую размерность задач (20–100), 42 % из них были решены оптимально. Для сравнения качества полученных решений согласно сравнения нижних границ для двухмерной упаковки использовались известные границы. Лучшие результаты показал предложенный метод.

СПИСОК ЛИТЕРАТУРЫ

1. **Мухачева, Э.А.** Модели и методы расчета раскроя-упаковки геометрических объектов [Текст] / Э.А. Мухачева, М.А. Верхотуров, В.В. Мартынов. – Уфа, 1998.
2. **Мухачева, Э.А.** Генетический алгоритм блочной структуры в задачах двумерной упаковки [Текст] / Э.А. Мухачева, А.С. Мухачева, А.В. Чиглинцев. – М.: Информационные технологии. – 1999. – № 11. – С.13–18.
3. **Мухачева, Э.А.** Задача прямоугольной упаковки: методы локального поиска оптимума на базе блочных структур [Текст] / Э.А. Мухачева, А.С. Мухачева. – М.: Автоматика и телемеханика. – 2004. – № 2. – С. 10–15.
4. **Monachi, M.** Algorithms for packing and scheduling problems. PhD Thesis [Текст] / M. Monachi. – University of Bologna, 2001.
5. **Житников, В.П.** Задача прямоугольной упаковки в полубесконечную полосу: поиск решения в окрестности локальной нижней границы [Текст] / В.П. Житников, А.С. Филиппова. – М.: Информационные технологии. – 2007. – № 5. – С. 55–62.
6. **Картак, В.М.** Локальный поиск ортогональных упаковок с использованием нижних границ [Текст] / В.М. Картак, М.А. Месягутов, Э.А. Мухачева [и др.]. – М.: Автоматика и телемеханика. – 2009. – № 6. – С. 167–180.
7. **Hartmann, S.** Project Scheduling under limited resources. Models, methods and applications [Текст] / S. Hartmann. – Springer, Berlin, 1999.
8. **Belov, G.** A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths [Текст] / G. Belov, G. Scheithauer // European Journal of Operational Research. – 2002. – № 141 (2). – P. 274–294.
9. **Bortfeld, A.** A genetic algorithm for the two-dimensional strip packing problem with rectangular prices [Текст] / A. Bortfeld // European Journal of Operational Research. – 2006. – № 172 (3). – P.814–837.

УДК 519.6; 519.17

А.А. Кочкаров, Л.Х. Хапаева

АЛГОРИТМ РАСПОЗНАВАНИЯ ПРЕДФРАКТАЛЬНОГО ГРАФА С ЧЕРЕДОВАНИЕМ ЗАТРАВОВ

Теория графов – незаменимый инструмент в проектировании сложных систем и синтезе структур. Применение методов и подходов теории графов показали свою результативность в различных областях от медицины и биологии до экономики и менеджмента [1].

Особое внимание стоит обратить на использование методов и подходов теории графов и дискретной математики в моделировании сложных многоэлементных систем. Задачи, которые возникли при исследовании таких систем, как электроэнергетические, социальные и информационные сети, дали существенный толчок для нового этапа развития и применения идей теории графов [2–3].

Интересные и оригинальные результаты были получены при моделировании сложных иерархических систем самоподобными или фрактальными графами [4]. Своим рождением фрактальные (предфрактальные) графы обязаны синтезу идей

синергетики [5] и нелинейной динамики [6], фракталов [10] и теории графов [1].

Термин «сеть» широко распространен в современной научной и бизнес-литературе. На слуху такие выражения, как «розничная или торговая сеть (сеть магазинов)», «сетевой маркетинг», «филиальная сеть», «сеть трубопроводов», «железнодорожная сеть», «социальная сеть», «компьютерная сеть», «информационная сеть», «телефонная сеть» и т. д. Часто этот термин используется для обозначения совершенно различных понятий. В настоящем диссертационном исследовании термин «сеть» понимается, во-первых, как совокупность путей доставки товаров или услуг до конечного получателя, а, во-вторых, как совокупность связей между элементами многоэлементной системы. Системы, в основе функционирования которых лежит сеть, принято называть *сетевыми системами* [8].



На протяжении довольно длительного времени техническая и экономическая науки считали аксиомой стационарность структуры всякой сетевой системы. Под структурой системы понимали совокупность исключительно устойчивых связей между элементами системы. На этом понимании выросли научные школы в области теории графов, дискретной математики, комбинаторной оптимизации и теории систем.

Развивающаяся экономика и глобализационные процессы вынуждают сетевые системы развиваться, адаптировать, оптимизировать свою сетевую структуру под сильно изменчивую конкурентную среду и под новую геополитическую конъюнктуру. В такой ситуации в регулярных изменениях сетевых структур прослеживаются закономерности. Сетевые структуры не только теряют свою стационарность (фиксированность), но и приобретают признаки динамических систем, свойства иерархических и масштабно-инвариантных структур. Процессы изменения, развития, поведения сетевых структур можно объединить общим понятием *структурная динамика*.

Исследования в области структурной динамики ведутся в научных школах профессора В.В. Кульбы [9], члена-корреспондента РАН Д.А. Новикова [10], профессора А.М. Кочкарова [4] в таких научно-исследовательских институтах и ведущих вузах России, как Институт проблем управления имени В.А. Трапезникова РАН, Институт прикладной математики имени М.В. Келдыша РАН, Вычислительный центр имени А.А. Дородницына РАН, Карачаево-Черкесская государственная технологическая академия. Работы школ профессора В.В. Кульбы и члена-корреспондента Д.А. Новикова посвящены, в основном, вопросам взаимодействия между элементами сложных иерархических систем. В работах школы профессора А.М. Кочкарова первичное внимание уделено именно вопросам поведения–развития сетевых структур. В качестве моделей структурной динамики сетевых систем предлагаются различные классы масштабно-инвариантных графов, называемых *предфрактальными*.

Очевидно, что при исследовании сетевых систем необходимо решать не только задачу распознавания структуры уже существующей сетевой системы, но и задачу распознавания самого процесса развития–изменения структуры сетевой системы. Задачу, объединяющую две указанные, назовем задачей *структурного распознавания*.

Алгоритмы распознавания, строящие решение задачи структурного распознавания, во-первых, устанавливают, что процесс развития сетевых структур соответствует тем или иным правилам порождения предфрактальных графов, а, во-вторых, определяют какие типы затравок при порождении были использованы. В данной статье описан и обоснован алгоритм предфрактального графа, порожденного парой регулярных затравок с чередованием.

Фрактальные и предфрактальные графы: основные понятия. В настоящей работе определен особый класс предфрактальных графов: предфрактальный граф, порождаемый множеством затравок с чередованием. Именно он может наиболее адекватно описывать структуры сложных многоэлементных сетевых систем при построении моделей. Поэтому решается задача распознавания предфрактальных графов, порожденных множеством затравок с чередованием. Предложен и обоснован алгоритм распознавания предфрактального графа, порожденного двумя полными затравками с чередованием.

Предфрактальный граф будем обозначать через $G_L = (V_L, E_L)$, где V_L – множество вершин графа; E_L – множество его ребер. Определим его рекуррентно, поэтапно, заменяя каждый раз в построенном на предыдущем этапе $l = \{1, 2, \dots, L-1\}$ графе G_l каждую его вершину связным графом $H = (W, Q)$, который будем называть *затравка*. На первом этапе предфрактальному графу соответствует затравка. При этом об описанном процессе говорят, что предфрактальный граф $G_L = (V_L, E_L)$ порожден затравкой $H = (W, Q)$. Ребра, появившиеся на этапе $l, l = \overline{1, L}$, порождения предфрактального графа, будем называть *ребрами ранга l* . Новыми ребрами предфрактального графа G_L назовем ребра ранга L , а все остальные ребра назовем *старыми*. Процесс построения предфрактального графа, по существу, является процессом построения последовательности предфрактальных графов, которую назовем *траекторией*. *Фрактальный граф* определяется бесконечной траекторией. Обобщением описанного процесса порождения предфрактального графа G_L является такой случай, когда вместо единственной затравки H используется множество затравок $\mathbf{H} = \{H_i\} = \{H_1, H_2, \dots, H_p, \dots, H_T\}, T \geq 2$. Суть этого обобщения состоит в том, что при переходе от графа G_{l-1} к графу G_l каждая вершина замещается некоторой затравкой $H_i \in \mathbf{H}$, которая выбирается

случайно или согласно определенному правилу, отражающему специфику моделируемого процесса или структуры. Если при переходе от графа G_{l-1} к графу G_l каждая вершина графа G_{l-1} замещается одной конкретной случайно выбранной затравкой $H_t \in \mathbf{H}$, то будем говорить, что *предфрактальный граф G_L порожден множеством затравок $\mathbf{H} = \{H_t\}, t = 1, 2, \dots, T, T \geq 2$, с чередованием*. Если же при порождении предфрактального графа G_L множеством затравок $\mathbf{H} = \{H_t\}, T \geq 2$, с чередованием задано некоторое правило выбора затравок из $\mathbf{H}$, например, неубывание числа вершин или ребер выбираемых затравок, то будем говорить, что *предфрактальный граф G_L порожден множеством затравок $\mathbf{H} = \{H_t\}, t = 1, 2, \dots, T, T \geq 2$, с упорядоченным чередованием*.

Очевидно, что порождение фрактального графа $G = (V, E)$ (т. е. когда траектория является бесконечным множеством $G_1, G_2, \dots, G_p, \dots, G_L, G_{L+1}, \dots$) с чередованием затравок, возможно только при бесконечном числе замещений затравок $\mathbf{H} = \{H_t\}, t = 1, 2, \dots, T, T \geq 2$.

Если при порождении предфрактального графа с чередованием для замещения вершин на последующих шагах порождения выбираются затравки с возрастанием числа вершин, то такой предфрактальный граф будем называть *порожденным с упорядоченным возрастанием затравок*. Если для замещения вершин на последующих шагах порождения выбираются затравки с убыванием числа вершин, то такой предфрактальный граф будем называть *порожденным с упорядоченным убыванием затравок*. Использование одной и той же затравки для порождения предфрактального графа чередованием затравок, на различных этапах порождения исключается.

В случае порождения предфрактального графа G_L с упорядоченным возрастанием (с упорядоченным убыванием) затравок $\mathbf{H} = \{H_t\}, t = 1, 2, \dots, T, T \geq 2$, если $L > T$, то переход на $l = T + 1$ шаге порождения от предфрактального графа G_T к G_{T+1} осуществляется заменой всех вершин графа G_T затравкой с наименьшим (наибольшим) числом вершин из $\mathbf{H} = \{H_t\}, t = 1, 2, \dots, T, T \geq 2$. На последующих шагах $l = T + 2, T + 3, \dots, L$ порождения предфрактального графа G_L затравки из $\mathbf{H} = \{H_t\}, t = 1, 2, \dots, T, T \geq 2$, используются последовательно по очередности возрастания (убывания) числа вершин. В случае порождения предфрактального графа G_L число этапов порождения больше числа затравок $L > T$, целесообразно говорить о периоде

замещения вершин затравками. Период замещения вершин затравками в процессе порождения предфрактального графа G_L с возрастанием или убыванием вершин затравок $\mathbf{H} = \{H_t\}, t = 1, 2, \dots, T, T \geq 2$ обозначим через P .

Алгоритм распознавания. Рассмотрим задачу, когда требуется распознать предфрактальный граф, порожденный парой полных затравок с возрастанием, и при сохранении смежности старых ребер.

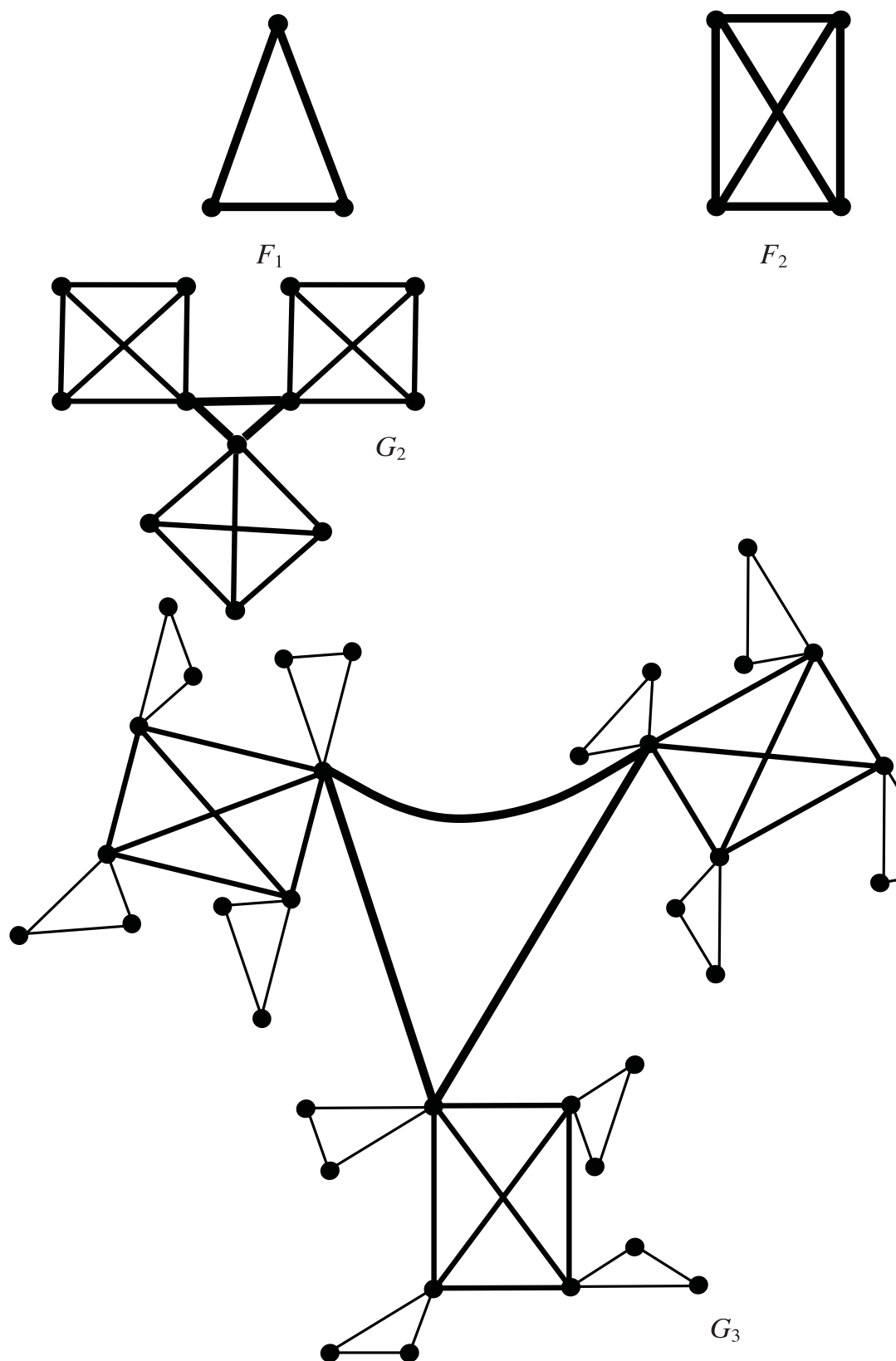
Переход в траектории $G_1, G_2, \dots, G_p, \dots, G_L$ графа с чередованием затравок от текущего графа $G_r = (V_r, E_r)$ к следующему графу G_{r+1} всякий раз подчиняется основным правилам порождения предфрактального графа с чередованием затравок при сохранении смежности старых ребер.

Распознавание предфрактального графа $G_L = (V_L, E_L)$, порожденного парой полных затравок $F_1 = (V_1, E_1)$ и $F_2 = (V_2, E_2)$ с числом вершин m_1 и $m_2 (m_1 < m_2)$ соответственно, с возрастанием и при сохранении смежности старых ребер можно осуществить алгоритмом β . Суть алгоритма β заключается в идентификации графов $F_1 = (V_1, E_1)$, $F_2 = (V_2, E_2)$ как затравок предфрактального графа $G_L = (V_L, E_L)$, а также идентификации траектории самого предфрактального графа.

На рисунке изображена траектория предфрактального графа $G_3 = (V_3, E_3)$, порожденного парой полных затравок – трехвершинным графом $F_1 = (V_1, E_1)$ и четырехвершинным $F_2 = (V_2, E_2)$, с возрастанием и при сохранении смежности старых ребер.

Алгоритм β состоит из $k = 1, 2, \dots, l, \dots, L$ этапов. На каждом из этапов алгоритма β выполняются три ключевые операции: окрашивание (выделение) вершины, и – ребра, стягивание ребра, стягивание цепи. На вход первого $k = 1$ этапа алгоритма предъявляется предназначенный для распознавания граф $G = (V, E)$. На вход каждого из этапов $k = 1, 2, \dots, l, \dots, L$ алгоритма β предъявляется граф $G^k = (V^k, E^k)$, как результат работы предыдущего этапа.

Определим число вершин графа $G = (V, E)$ и обозначим их через N . Разделим число вершин графа $G = (V, E)$ на произведение $m_1 m_2$ последовательно необходимое количество раз, чтобы получить представление $N = (m_1 m_2)^{kr}$. Возможны два случая: $r = 1, r = m_1$. Рассмотрим каждый из них подробно. Если остаток r не удовлетворяет ни одному из указанных вариантов, то алгоритм β_3 завершает свою работу с отрицательным результатом.



Траектория предфрактального графа, порожденного парой полных затравок

Пусть $r = 1$. Тогда $G = G^L$, а число этапов алгоритма β равно $L = 2K$. Затем у графа G^L проводим поиск всех вершин степени $m_2 - 1$ и объединяем их в множество $V_{m_2}^L$. Затем полученное множество $V_{m_2}^L$ разбивается на подмножества с учетом взаимной смежности составляющих его вершин следующим образом. Выделяем вершину $v \in V_{m_2}^L$ и смежные ей $(m_2 - 2)$ вершины из множества $V_{m_2}^L$. Если вершина $v \in V_{m_2}^L$ не имеет в множестве $V_{m_2}^L$ смежных с ней $(m_2 - 2)$ вершин, то алгоритм β заканчивает свою работу с отрицательным результатом. В процессе разбиения множества $V_{m_2}^L$ на подмножества, вершины не выделяются более одного раза. После разбиения множества $V_{m_2}^L$ на подмножества, в каждом подмножестве проводится проверка на взаимную смежность всех ее $(m_2 - 2)$ вершин. Далее выделяются ребра, обеспечивающие смежность вершин в каждом подмножестве. Их число в каждом из подмножеств равно $\frac{(m_2 - 1)(m_2 - 2)}{2}$, в противном случае алгоритм заканчивает свою работу с отрицательным результатом.

Далее проверяем смежность всех $(m_2 - 1)$ вершин каждого выделенного подмножества с одной вершиной, степень которой больше, чем $m_2 - 1$. Если таковая имеется, выделяя ее и все ребра, соединяющие эту вершину с уже выделенными, получим выделенный полный подграф с m_2 вершинами. Такие полные подграфы должны быть выделены на графе G^L в соответствии с числом подмножеств множества $V_{m_2}^L$, в противном случае алгоритм заканчивает свою работу с отрицательным результатом.

Далее стягиваем все выделенные ребра. И на вход следующего этапа алгоритма β передается граф G^{L-1} . На графе G^{L-1} выделяем все вершины степени $m_1 - 1$ и объединяем в множество $V_{m_1}^{L-1}$. С множеством $V_{m_1}^{L-1}$ проделываем те же операции относительно вершин со степенью $m_1 - 1$, что и с вершинами множества. Далее стягиваем все выделенные ребра. На вход следующего этапа алгоритма β передается граф G^{L-2} . На всех последующих этапах проводим операции по стягиванию выделяемых полных подграфов, чередуя число их вершин m_1 и m_2 в продолжении уже начатой последовательности на первых двух шагах. На выходе последнего шага получим полный граф G^1 с числом вершин m_1 , в противном случае алгоритм β завершает свою работу с отрицательным результатом.

Пусть $r = m_1$. Тогда $G = G^L$, а число этапов алгоритма β равно $L = 2K + 1$. Затем на графе G^L выделяем все вершины степени $m_1 - 1$. Далее процесс распознавания графа $G = G^L$ при $r = m_1$ идентичен предыдущему случаю, когда $r = 1$.

Результатом работы алгоритма β_3 в случаях, когда $r = 1, r = m_1$ является траектория $\bar{G}^k = (V^k, E^k) = G_k = (V_k, E_k)$, $k = 1, 2, \dots, l, \dots, L$, предфрактального графа $G_L = (V_L, E_L)$, порожденного парой полных затравок $F_1 = (V_1, E_1)$, $F_2 = (V_2, E_2)$ с числом вершин m_1 и m_2 ($m_1 < m_2$) соответственно, с возрастанием и при сохранении смежности старых ребер.

Не налагая дополнительных ограничений на входные данные с алгоритм β_3 , кроме изменения условия $m_1 < m_2$ на $m_1 > m_2$, можно переориентировать алгоритм β_3 на распознавание предфрактального графа $G_L = (V_L, E_L)$, порожденного парой полных затравок $F_1 = (V_1, E_1)$, $F_2 = (V_2, E_2)$ с числом вершин m_1 и m_2 ($m_1 > m_2$) соответственно, с убыванием и при сохранении смежности старых ребер.

Теорема 1. Всякий предфрактальный граф $G_L = (V_L, E_L)$, порожденный парой полных затравок $F_1 = (V_1, E_1)$ и $F_2 = (V_2, E_2)$, $|V_1| = m_1$ и $|V_2| = m_2$ ($m_1 > m_2$), с упорядоченным возрастанием и сохранением смежности старых ребер распознается алгоритмом β с полиномиальной трудоемкостью $O(|E_L| + L|V_L|)$.

Доказательство разделим на две части. В первой будет доказано соответствие выполняемой алгоритмом β работы заявленным целям, т. е. распознаванию предфрактального графа $G_L = (V_L, E_L)$, порожденного парой полных затравок $F_1 = (V_1, E_1)$, $F_2 = (V_2, E_2)$ с числом вершин m_1 и m_2 ($m_1 < m_2$) соответственно, с возрастанием и при сохранении смежности старых ребер. Во второй будет подсчитана трудоемкость самого алгоритма β .

1. При порождении предфрактального графа $G_L = (V_L, E_L)$ парой полных затравок $F_1 = (V_1, E_1)$ и $F_2 = (V_2, E_2)$ с числом вершин m_1 и m_2 ($m_1 < m_2$) соответственно, число вершин самого графа будет равно произведению с чередующимися множителями $N = m_1 m_2 m_1 m_2 \dots \cdot r$, где каждый множитель соответствует этапу порождения предфрактального графа. Формула $N = m_1 m_2 m_1 m_2 \dots \cdot r$ позволяет вычислять число вершин предфрактального графа, порожденного парой полных затравок с упорядоченным возрастанием. Множитель r , если число этапов порождения нечетно, равен m_1 . Если число этапов порождения четно, то множитель r



равен 1. Рассмотрим оба случая в отдельности.

Пусть число этапов порождения предфрактального графа $G_L = (V_L, E_L)$ нечетно, т. е. $r = m_1$. Это значит, что при порождении предфрактального графа $G_L = (V_L, E_L)$ число его вершин будет равно $N(G_L) = (m_1 m_2)^M m_1$, а $L = 2M + 1$. Кроме того, это значит, что последней затравкой, использованной для замещения вершин предфрактального графа $G_{L-1} = (V_{L-1}, E_{L-1})$, является полный m_1 -вершинный граф $F_1 = (V_1, E_1)$.

Всякий предфрактальный граф, порождаемый полными затравками с сохранением смежности старых ребер, можно получить склеиванием новых затравок с вершинами предфрактального графа предыдущего этапа. Это следует из общего описания процесса порождения предфрактального графа с сохранением смежности старых ребер. Поэтому только лишь одна вершина каждой новой подграф-затравки предфрактального графа $G_L = (V_L, E_L)$ будет инцидентна старым ребрам, значит, только степень этой вершины подграф-затравки будет отличаться от $m_1 - 1$.

Выделив на предфрактальном графе $G_L = (V_L, E_L)$ все вершины степени $m_1 - 1$ и объединив их в множество $V_{m_1}^L$, можно сгруппировать их в подмножества, исходя из взаимной смежности выделенных вершин. Эти подмножества представляют собой $(m_1 - 1)$ смежные вершины каждой новой подграф-затравки предфрактального графа $G_L = (V_L, E_L)$. В каждом из сгруппированных подмножеств множества $V_{m_1}^L$ не достает по одной вершине, чтобы каждое из подмножеств соответствовало множеству вершин отдельно взятой подграф-затравки. Для добавления этих подмножеств последними вершинами достаточно выделить все инцидентные вершинам подмножеств ребра, часть из которых $(m_1 - 1)$ ребро) окажется инцидентными недостающей вершине.

После выделения все новые ребра стягиваются. На вход следующему этапу порождения подается предфрактальный граф $G_{L-1} = (V_{L-1}, E_{L-1})$.

Далее на каждом этапе распознавания проводится, чередуя, выделение и стягивание новых подграф-затравок, соответствующих затравкам $F_2 = (V_2, E_2)$ и $F_1 = (V_1, E_1)$, по аналогии с описанным и обоснованным первым этапом распознавания алгоритма β .

Пусть число этапов порождения предфрактального графа $G_L = (V_L, E_L)$ четно, т. е. $r = 1$. Это значит, что число замещений, учитывая чередующийся характер процесса порождения, вер-

шин затравкой $F_1 = (V_1, E_1)$ и затравкой $F_2 = (V_2, E_2)$ будет равным. Поэтому число вершин предфрактального графа $G_L = (V_L, E_L)$ вычисляется в соответствии с произведением $N(G_L) = (m_1 m_2)^K$, а $L = 2K$. Кроме того, это значит, что последней затравкой, использованной для замещения вершин предфрактального графа $G_{L-1} = (V_{L-1}, E_{L-1})$, является полный m_2 -вершинный граф $F_2 = (V_2, E_2)$. Тогда с выделения подграф-затравок, соответствующих затравке $F_2 = (V_2, E_2)$, и целесообразно начинать распознавание предфрактального графа $G_L = (V_L, E_L)$. Последующие шаги в алгоритме β при $r = 1$ аналогичны шагам при $r = m_1$. Поэтому и обоснование этой части алгоритма аналогично предъявленному выше.

2. На каждом из $k = 1, 2, \dots, l, \dots, L$ этапов алгоритм β просматривает все вершины графа $G = (V, E)$ и выделяет соответствующие требованиям текущего этапа. То есть на каждом этапе алгоритма β вершины степени $m_2 - 1$ и $m_1 - 1$ соответственно, поочередно объединяются в множества $V_{m_2}^l$ и $V_{m_1}^l$. На каждом шаге алгоритма эти действия соответствуют $|V_L|$ операциям. На протяжении всего алгоритма эти действия соответствуют $L|V_L|$ операциям. Кроме того, на протяжении всего алгоритма β каждое ребро графа $G = (V, E)$ выделяется лишь единожды; общая вычислительная сложность алгоритма β распознавания предфрактального графа $G_L = (V_L, E_L)$, порожденного парой полных затравок $F_1 = (V_1, E_1)$, $F_2 = (V_2, E_2)$ с числом вершин m_1 и m_2 ($m_1 < m_2$) соответственно, с возрастанием и при сохранении смежности старых ребер равна $O(|E_L| + L|V_L|)$.

Работа [4] целиком посвящена распознаванию фрактальных (предфрактальных) графов, порожденных одной затравкой. Вопрос о распознавании фрактальных (предфрактальных) графов, порожденных множеством затравок, оставался открытым до недавнего времени.

В данной статье приведен пример адаптации (изменения правила) процесса порождения предфрактальных графов для моделирования развивающихся сетевых структур. Результатом адаптации стал новый класс предфрактальных графов, порождаемый множеством затравок с чередованием. Как следствие этого, рассмотрена частная задача распознавания предфрактального графа, порожденного парой полных затравок чередованием. Предложенный алгоритм распознавания решает эту задачу за полиномиальное время.

СПИСОК ЛИТЕРАТУРЫ

1. **Емеличев, В.А.** Лекции по теории графов [Текст]/В.А. Емеличев, О.И. Мельников, В.И. Сарванов [и др.]. – М.: Наука, 1990.
2. **Strogatz, S.** Exploring complex networks [Текст]/S. Strogatz//Nature. –2001. –№ 410. –Р. 268–276.
3. **Krön, B.** Growth of self-similar graphs [Текст]/B. Krön//J. Graph Theory. –2004. –№ 45 (3). –Р. 224–239.
4. **Кочкаров, А.М.** Распознавание фрактальных графов. Алгоритмический подход [Текст]/А.М. Кочкаров. –Нижний Архыз: РАН САО, 1998.
5. **Малинецкий, Г.Г.** Математические основы синергетики. Хаос, структуры, вычислительный эксперимент [Текст]/Г.Г. Малинецкий. –М.: КомКнига, 2005.
6. **Малинецкий, Г.Г.** Нелинейная динамика и проблема прогноза [Текст]/Г.Г. Малинецкий, С.П. Курдюмов//Вестник РАН. –2001. –Т. 71. –№3. –С. 210–224.
7. **Фракталы в физике** [Текст]/Под ред. Л. Пьетро-нро, Э. Тозатти. –М.: Мир, 1988.
8. **Малашенко, Ю.Е.** Модели неопределенности в многопользовательских сетях [Текст]/Ю.Е. Малашенко, Н.М. Новикова. –М.: Эдиториал УРСС, 1999.
9. **Кульба, В.В.** Управление и контроль реализации социально-экономических целевых программ [Текст]/В.В. Кульба, С.С. Ковалевский, В.А. Уткин [и др.]. –М.: Книж. дом «Либриком», 2009.
10. **Новиков, Д.А.** Сетевые структуры и организационные системы [Текст]/Д.А. Новиков. –М.: ИПУ РАН, 2003. –102 с.

УДК 519.632.4

А.В. Пашковский

ПОВЫШЕНИЕ ТОЧНОСТИ РАСЧЕТОВ МАГНИТНОГО ПОЛЯ И СИЛОВЫХ ХАРАКТЕРИСТИК ЭЛЕКТРОМЕХАНИЧЕСКИХ УСТРОЙСТВ КОМБИНИРОВАННЫМ МЕТОДОМ СТАНДАРТНЫХ И КОНЕЧНЫХ ЭЛЕМЕНТОВ

Развитие практически любой отрасли современного производства в значительной мере зависит от степени автоматизации инженерных разработок, гибкости технологий производства и присутствия в них обратной связи по контролю качества продукции. Именно поэтому в одном ряду по важности создания стоят системы диагностики, контроля качества, автоматизированные системы управления технологическими процессами и другие подобные системы. Значительная роль, определяющая качество функционирования систем, принадлежит математическим методам, применяющимся в мониторинге, диагностике и управлении технологическими процессами.

Современные численные методы, такие, как метод конечных элементов (МКЭ), комбинированный метод граничных и конечных элементов (КМГиКЭ) и т. д. позволяют осуществлять разнообразные типы прикладных расчетов. Тем не менее, использование новых технологий и научных разработок в производстве электромехани-

ческих, электромагнитных и тепловых устройств определило целый ряд задач, в которых существующие методы не обеспечивают достаточной точности. В частности, это относится к расчету полей, особенно трехмерных, в устройствах, которые рассматриваются как неоднородные или кусочно-однородные среды (КОС) с нелинейными включениями, узкими порами, тонкими пленками или мелкозернистой структурой. Ситуация усугубится при использовании в проектировании и производстве устройств микро-, микро- и нанотехнологий. Следует отметить, что даже в инженерных и научных задачах средней сложности при наличии нелинейности характеристик материалов, особенностей решения в окрестностях угловых точек сред, тонких включений, осцилляции и неудовлетворительных свойств решений в расчетных средах у вышеперечисленных методов возникают значительные потери точности.

Рассмотрим проблему такого рода на примере расчета магнитного поля и силовых характе-

ристик асинхронного тягово-подъемного модуля (рис. 1) комбинированным методом граничных и конечных элементов. Для последующей оценки точности расчетных значений КМГиКЭ воспользуемся экспериментальными данными и результатами расчета на основе комбинированного метода стандартных и конечных элементов (КМСФиКЭ), разрабатываемого автором.

Исследуемый асинхронный тягово-подъемный модуль совмещает в себе функции электромагнитного подвеса и одностороннего линейного асинхронного двигателя. Особенностью его конструкции является наличие пазов в полюсах П-образного магнитопровода индуктора и электромагнитного путевого элемента. Обмотка намагничивания в данной системе располагается на ярме индуктора. Длина электромагнитной системы вдоль полюсов равна 940 мм. Задача расчета магнитного поля сведена к плоскопараллельной. Механический зазор принят $z = 0,015$ м. Намагничивающая сила iW варьируется в пределах [6440; 12880] А. Экспериментальные значения нормального усилия определены для одного из шести асинхронных тягово-подъемных модулей тележки массой 6 т. Подъемная сила измерена с помощью четырех упругих элементов с тензорезисторами, установленных на шпильках крепления путевого элемента к продольной балке станда. Механический зазор фиксировался диэлектрическими прокладками, расположенными между исследуемым индуктором и путевыми элементами.

Результаты экспериментальных исследований подъемной силы F_y как функции намагничивающей силы обмотки намагничивания iW при раз-

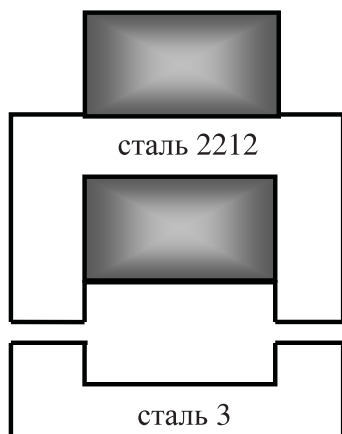


Рис. 1. Сечение асинхронного тягово-подъемного модуля

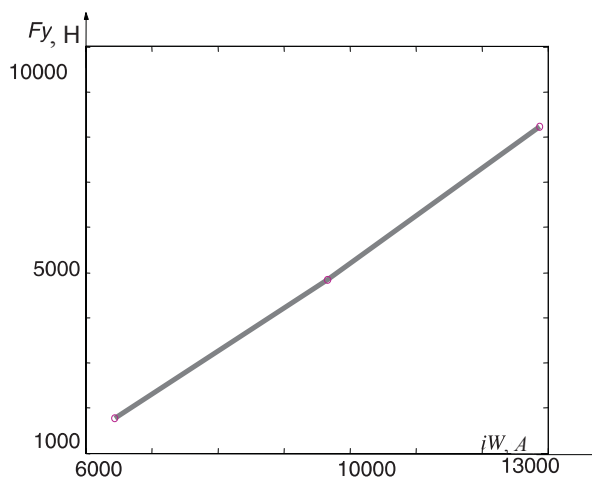


Рис. 2. Экспериментальные значения подъемной силы F_y при механическом зазоре $z = 0,015$ м

ных величинах механического зазора подробно рассмотрены в [1].

На рис. 2 приведены результаты, полученные при величине механического зазора $z = 0,015$ м.

КМГиКЭ в расчете силовых характеристик модуля

Результаты расчета значений подъемной силы F_y , полученные КМГиКЭ при варьировании механического зазора подробно рассмотрены в [2]. Зависимости $B(H)$ материалов магнитопровода нелинейные. В расчетах использованы кривые намагничивания сталей № 3 и № 2212 (рис. 3). Остановимся отдельно на методике расчета магнитного поля и силовых взаимодействий в рассматриваемой электромагнитной системе при комбинировании метода граничных элементов и метода конечных элементов. Так как векторный потенциал $\vec{A} = A(x, y)\vec{e}_z$ предполагается в каждом конечном элементе линейно зависящим от координат x и y , а магнитная индукция $\vec{B} = \text{rot } \vec{A}$ является результатом дифференциальной операции от потенциала, то индукцию считают постоянным вектором на каждом конечном элементе:

$$\vec{B}^{(r)}(x, y) = B_x^{(r)}\vec{e}_x + B_y^{(r)}\vec{e}_y,$$

где r – номер элемента.

Составляющие индукции в r -м конечном элементе выражаются через векторный потенциал следующим образом:

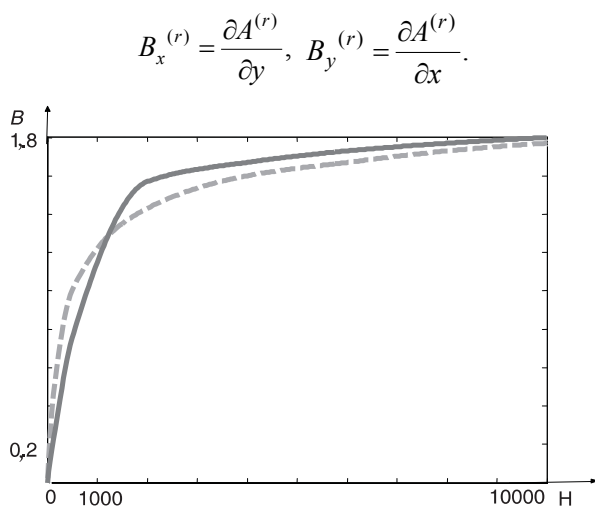


Рис. 3. Кривые намагничивания стали в ферромагнетиках
(—) сталь 3; (---) сталь 2212

Учитывая финитное представление потенциала в каждом конечном элементе, составляющие индукции можно переписать в виде:

$$B_x^{(r)} = \frac{1}{2\Delta S^{(r)}} [A_i(x_k - x_j) + A_j(x_i - x_k) + A_k(x_j - x_i)];$$

$$B_y^{(r)} = \frac{1}{2\Delta S^{(r)}} [A_i(y_k - y_j) + A_j(y_i - y_k) + A_k(y_j - y_i)],$$

где $\Delta S^{(r)}$ – площадь r -го треугольника; i, j, k – узлы r -го элемента; A_i, A_j, A_k – значения потенциала в соответствующих узлах элемента.

По такому алгоритму после нахождения распределения потенциала A вычисляются составляющие индукции $\vec{B}$ в каждом конечном элементе. По известным значениям магнитной индукции $\vec{B}$ можно в дальнейшем найти электромагнитную силу (в плоском случае), действующую на ферромагнитное тело:

$$\vec{F} = \frac{1}{\mu_0} \oint_L [\vec{B}(\vec{B} \cdot \vec{n}) - \frac{1}{2} B^2 \vec{n}] dl,$$

где L – контур, охватывающий ферромагнетик и не содержащий внутри себя других намагничивающихся тел; $\vec{n}$ – единичный вектор нормали, направленный из ферромагнетика.

Силу ищем в виде $\vec{F} = F_x \vec{e}_x + F_y \vec{e}_y$, где составляющие определяются по формулам:

$$F_x = \frac{1}{\mu_0} \oint_L \left\{ \frac{1}{2} [(B_x^+)^2 - (B_y^+)^2] n_x + B_x^+ B_y^+ n_y \right\} dl,$$

$$F_y = \frac{1}{\mu_0} \oint_L \left\{ \frac{1}{2} [(B_y^+)^2 - (B_x^+)^2] n_y + B_x^+ B_y^+ n_x \right\} dl.$$

В качестве контура L взята граница ферромагнетика $\Gamma_i (i = 1, n)$.

В рассматриваемой задаче расчета магнитного поля и силовых характеристик модуля заранее известно, что границы ферромагнетиков состоят из прямых, образующих между собой прямые углы. Поэтому составляющие силы можно определить отдельно для каждого участка контура, представляющего прямую, а затем просуммировать по принципу суперпозиции. Экспериментальные и расчетные значения КМГиКЭ подъемной силы F_y для механического зазора $z = 0,015$ м приведены на рис. 4 и в табл. 1.

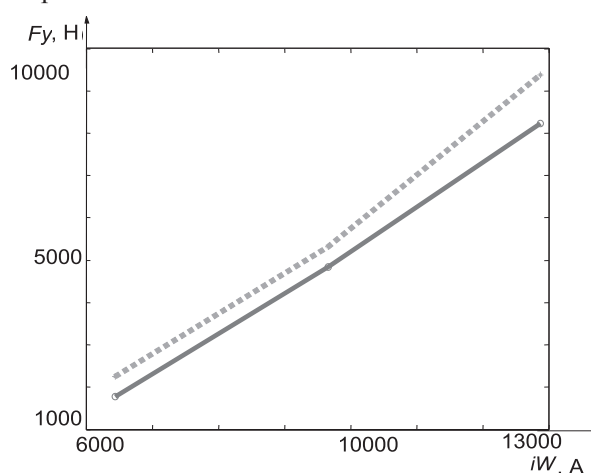


Рис. 4. Экспериментальные и расчетные значения подъемной силы F_y по КМГиКЭ
(—) КМГиКЭ; (---) эксперимент

Таблица 1

Экспериментальные и расчетные значения F_y

iW, A	F_y, H		$\delta_{\text{КМГиКЭ}}, \%$
	Эксперимент	КМГиКЭ	
6440	1774	2258	27,28
9660	4838	5323	10,03
12880	8226	9387	14,11

КМСФиКЭ в расчете силовых характеристик модуля

Следует отметить, что ранее выдвинутое предположение о линейности векторного потенциала $\vec{A} = A(x, y) \vec{e}_z$ в каждом конечном элементе от координат x и y является одним из недостатков методики МКЭ. Оно приводит к высоким погрешностям в вычислении производных потенциала, что в дальнейшем определяет и повышенную погрешность расчетов силовых взаимодействий в ЭМС.

Разрабатываемый КМСФиКЭ позволяет использовать коэффициенты разложения потенциала и его нормальной производной в ряд Фурье, вычисленные на границах стандартных элементов, ничего не предполагая о характере поведения потенциала. Можно заключить, что в такой возможности просматривается еще одно преимущество КМСФиКЭ, полезное при вычислении силовых взаимодействий в ЭМС.

Остановимся кратко на методике КМСФиКЭ по расчету силовых взаимодействий в ЭМС. Очевидно, что в расчетной области $\vec{B} = B_x \vec{e}_x + B_y \vec{e}_y$, $\vec{B} = \text{rot } \vec{A} = \frac{\partial A}{\partial y} \vec{e}_x + \frac{\partial A}{\partial x} \vec{e}_y$. То есть $B_x = \frac{\partial A}{\partial y}$, $B_y = \frac{\partial A}{\partial x}$.

Очевидно, $\frac{\partial A}{\partial y}$, $\frac{\partial A}{\partial x}$ – производные векторного потенциала вдоль и по нормали к вертикальному участку границы и, соответственно, наоборот, для горизонтального.

Таким образом, решая полевую задачу КМСФиКЭ и имея как результат – представление векторного потенциала на сторонах СЭ в виде разложения в ряд Фурье, можем определить B_x и B_y . Следовательно, знаем все значения, входящие в формулы проекций силы.

Задача расчета магнитного поля подъемно-тягового модуля при использовании КМГиКЭ в [2] была сведена к плоскопараллельной. Решим КМСФиКЭ эквивалентную ей краевую задачу Дирихле для уравнения Пуассона, в постановке:

$$\text{div}\left(\frac{1}{\mu} \text{grad} A\right) = f, \quad A|_s = 0. \quad (1)$$

Внешняя граница расчетной области установлена на удалении 0,06 м от границ рассчитываемой системы. Обратим внимание, что один из возможных вариантов заполнения сечения асинхронного тягово-подъемного модуля стандартными элементами при линейных характеристиках материалов и использовании разработанного ранее МСЭФ, приведен на рис. 5.

Дискретизация характерна полным отсутствием конечных элементов, а для заполнения сечения использовано лишь 54 прямоугольных СЭ типа SERECD1234. Проведенная дискретизация подобна введенной ранее для П-образного электромагнита [3] и линейного двигателя [4]. С целью оценки точности КМСФиКЭ в задаче (1) прямоугольные СЭ, введенные в статорах с нелинейными магнитными свойствами, заменены конечно-элементной сеткой. При этом узлы, расположенные на их границе, использованы для склейки конечно-элементной сетки с оставшимися СЭ.

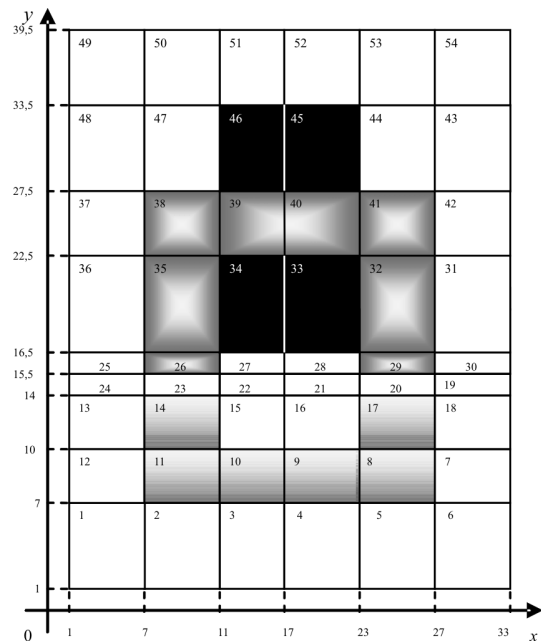


Рис. 5. Заполнение сечения асинхронного тягово-подъемного модуля СЭ

Один из возможных вариантов заполнения сечения асинхронного тягово-подъемного модуля стандартными и конечными элементами, при нелинейных характеристиках материалов, приведен на рис. 6. Разбиение содержит 25 СЭ и 864 конечных элемента. Для расчета силовых взаимодействий применен оригинальный программный пакет SEMF-CF.

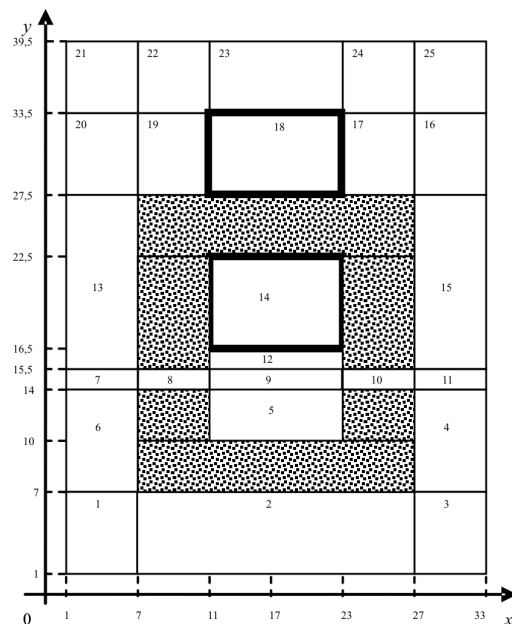


Рис. 6. Комбинированная дискретизация сечения ЭМС по методике КМСФиКЭ

Таблица 2

Расчетные и экспериментальные значения F_y

iW, A	F_y, H			$\delta, \%$	
	Эксперимент	КМГиКЭ	КМСФиКЭ	КМГиКЭ	КМСФиКЭ
6440	1774	2258	1856	27,28	4,62
9660	4838	5323	5032	10,03	4,01
12880	8226	9387	8620	14,11	4,79

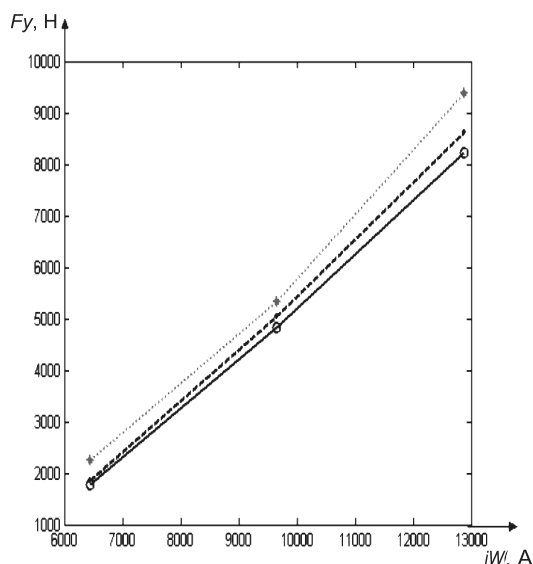


Рис. 7. Сопоставление расчетных значений подъемной силы F_y с экспериментальными (—○—) эксперимент; (---◇---) КМГиКЭ; (- - -◇ - -) КМСФиКЭ

После нахождения распределения потенциала A , если итерационный процесс не закончен, пакетом вычисляются соответствующие составляющие и модуль индукции в каждом конечном элементе. По кривой намагничивания, аппроксимируемой кубическими сплайн-функциями, находится значение магнитной напряженности $H^{(r)}$, также предполагаемой постоянной на каждом конечном элементе, и магнитной проницаемости

$\mu^{(r)} = B^{(r)} / H^{(r)}$. Осуществляется пересчет магнитной проницаемости стальных элементов магнитопровода. Если итерационный процесс закончен, то составляющие индукции вычисляются на сторонах СЭ и используются для дальнейшего анализа и вычисления силы. Сопоставление результатов расчета значений подъемной силы F_y на основе КМСФиКЭ с результатами по КМГиКЭ и экспериментальными значениями, проведено в табл. 2 и на рис. 7.

Таким образом, КМСФиКЭ позволяет:

1. Значительно сократить степень дискретизации расчетной области.
2. В качестве неизвестных системы линейных алгебраических уравнений рассмотреть совместно коэффициенты Фурье решения на сторонах СЭ и значения в узлах конечно-элементной сетки.
3. Для повышения точности расчетов использовать коэффициенты разложения потенциала и его нормальной производной в ряд Фурье, вычисленные на границе СЭ, ничего не предполагая о поведении потенциала в СЭ.
4. Сократить размерность расширенной матрицы решаемой системы уравнений, не прибегая к специальным алгоритмам ее формирования (ленточной матрице).
5. Значительно повысить точность расчетов по сравнению с используемыми численными методами и обеспечить приемлемое совпадение результатов расчета с экспериментальными данными.

СПИСОК ЛИТЕРАТУРЫ

1. Исследовательские испытания характеристик магнитного поля, силовых взаимодействий и тепловых процессов макета электромагнитного подвеса опытного вагона транспортной линии Ереван-Абавян [Текст]/Протокол № ВС-ТМ2-49-87/ВЭлНИИ.—Новочеркасск, 1987.—100 с.
2. Косиченко, М.Ю. Численное моделирование электромагнитных полей и силовых характеристик в электромагнитных и магнитоэлектрических системах комбинированным методом конечных и граничных

- элементов дис. ... канд. техн. наук [Текст]/М.Ю. Косиченко.—Новочеркасск. 2003.—181 с.
3. Пашковский, А.В. МСЭФ в моделировании стационарного поля в области с П-образным сердечником [Текст]/А.В. Пашковский, И.В. Пашковская//Изв. вузов. Сер. Электромеханика.—2009.—№ 2.—С. 10–12.
4. Пашковский, А.В. Метод стандартных элементов в расчете магнитного поля линейного двигателя с постоянными магнитами [Текст]/А.В. Пашковский//Научно-технические ведомости СПбГПУ.—2010.—№ 2.—С. 130–136.



АЛГОРИТМ ПОИСКА ВНЕШНЕ-ВНУТРЕННЕГО ЦЕНТРА ПРЕДФРАКТАЛЬНОГО ГРАФА С СОХРАНЕНИЕМ СМЕЖНОСТИ СТАРЫХ РЕБЕР

В практической деятельности постоянно возникают задачи «наилучшего» размещения оборудования (или средств обслуживания) в сетях или графах. В частности, если граф представляет сеть дорог и вершины соответствуют отдельным районам, то можно поставить задачу оптимального размещения больниц, полицейских участков, пожарных частей и многих других крайне необходимых предприятий и служб. В таких случаях критерий оптимальности может состоять в минимизации расстояния (или времени проезда) от пункта обслуживания до самой отдаленной вершины графа, т. е. в оптимизации «наихудшего варианта». В более общей задаче требуется разместить несколько таких пунктов обслуживания. При этом самая отдаленная вершина графа должна находиться на минимально возможном расстоянии, по крайней мере, от одного пункта обслуживания. К таким задачам относятся задачи размещения аварийных служб. Поэтому объективным требованием здесь является минимизация наибольшего расстояния от произвольной вершины графа до ближайшего к ней пункта обслуживания. Полученные при решении этих задач места размещения пунктов обслуживания называются *центрами* графа.

Термином *затравка* условимся называть какой-либо связный граф $H = (W, Q)$. Для определения *фрактального (предфрактального) графа* нам потребуется операция *замены вершины затравкой (ЗВЗ)*. Суть операции ЗВЗ заключается в следующем. В данном графе $G = (V, E)$ у намеченной для замещения вершины $\tilde{v} \in V$ выделяется множество $\tilde{V} = \{\tilde{v}_j\} \subseteq V, j = 1, 2, \dots, |\tilde{V}|$, смежных ей вершин. Далее, из графа G удаляется вершина $\tilde{v}$ и все инцидентные ей ребра. Затем каждая вершина $\tilde{v}_j \in \tilde{V}, j = 1, 2, \dots, |\tilde{V}|$, произвольно (случайным образом) или по определенному правилу (при необходимости), соединяется ребром с одной из вершин затравки $H = (W, Q)$.

Предфрактальный граф будем обозначать через $G_L = (V_L, E_L)$, где V_L – множество вершин

графа; E_L – множество его ребер. Определим его рекуррентно, поэтапно, заменяя каждый раз в построенном на предыдущем этапе $l = 1, 2, \dots, L - 1$ графе $G_l = (V_l, E_l)$ каждую его вершину затравкой $H = (W, Q)$. На этапе $l = 1$ предфрактальному графу соответствует затравка $G_1 = H$. Об описанном процессе говорят, что *предфрактальный граф* $G_L = (V_L, E_L)$ порожден затравкой $H = (W, Q)$. Процесс порождения предфрактального графа G_L , по существу, есть процесс построения последовательности предфрактальных графов $G_1, G_2, \dots, G_p, \dots, G_L$, называемой *траекторией*. Фрактальный граф $G = (V, E)$, порожденный затравкой $H = (W, Q)$, определяется бесконечной траекторией.

Предфрактальный граф $G_L = (V_L, E_L)$ условимся называть (n, q, L) -графом, если он порожден n -вершинной q -реберной связной затравкой $H = (W, Q)$, и (n, L) -графом, если затравка $H = (W, Q)$ – регулярный граф.

Использование операции ЗВЗ в процессе порождения предфрактального графа G_L для элементов $G_l = (V_l, E_l), l \in \{1, 2, \dots, L - 1\}$ и его траектории позволяет ввести отображение $\varphi: V_l \rightarrow V_{l+1}$ или $\varphi(V_l) = V_{l+1}$, а в общем виде $\varphi'(V_l) = V_{l+p}, t = 1, 2, \dots, L - l$. В последнем выражении множество V_{l+t} – образ множества V_l , а множество V_l – прообраз множества V_{l+t} .

Для предфрактального графа G_L ребра, появившиеся на l -м ($l \in \{1, 2, \dots, L - 1\}$) этапе порождения, будем называть *ребрами ранга l* . Новыми ребрами предфрактального графа G_L назовем ребра ранга L , а все остальные ребра назовем *старыми*.

Для любой вершины v_i графа $G_l = (V_l, E_l)$ пусть $R_\lambda^\alpha(v_i)$ есть множество тех вершин v_j графа G_p , которые достижимы из вершины v_i с помощью путей со взвешенными длинами $d(v_i, v_j)$, не превосходящими величины λ . Через $R'_\lambda(v_i)$ будет обозначаться множество тех вершин v_j графа G , из которых вершина v_i может быть достигнута с использованием путей, имеющих взвешенные длины $d(v_j, v_i) \leq \lambda$. Таким образом,

$$R_{\lambda}^o(v_i) = \{v_j \mid d(v_i, v_j) \leq \lambda, v_j \in V_l\},$$

$$R_{\lambda}'(v_i) = \{v_j \mid d(v_j, v_i) \leq \lambda, v_j \in V_l\}.$$

Для каждой вершины v_i графа $G_l = (V_l, E_l)$ определим следующие числа:

$$s_o(v_i) = \max_{v_j \in V'} [d(v_i, v_j)],$$

$$s_i(v_i) = \max_{v_j \in V'} [d(v_j, v_i)],$$

$$s_{ot}(v_i) = \max_{v_j \in V'} [d(v_i, v_j) + d(v_j, v_i)].$$

Числа $s_o(v_i)$, $s_i(v_i)$ и $s_{ot}(v_i)$ называются, соответственно, *числом внешнего разделения*, *числом внутреннего разделения вершины* и *числом внешне-внутреннего разделения вершины* v_i .

Вершина v_o^* , для которой $s_o(v_o^*) = \min_{v_i \in V'} [s_o(v_i)]$, называется *внешним центром* графа G_l ; вершина v_i^* , для которой $s_i(v_i^*) = \min_{v_i \in V'} [s_i(v_i)]$, называется *внутренним центром* графа G_l , а вершина v_{ot}^* , для которой $s_{ot}(v_{ot}^*) = \min_{v_i \in V'} [s_{ot}(v_i)]$, называется *внешне-внутренним центром* графа G_l .

У графа может быть несколько внешних, внутренних и внешне-внутренних центров, которые образуют множества внешних, внутренних и внешне-внутренних центров, соответственно.

Рассмотрим взвешенный предфрактальный граф $G_L = (V_L, E_L)$, порожденный ориентированной затравкой $H = (W, Q)$, смежность старых ребер которого сохраняется [1].

Предположим, что затравка $H = (W, Q)$ является сильно связным графом, т. е. каждая вершина достижима из всякой другой вершины [2].

Алгоритм α_3 поиска внешне-внутреннего центра предфрактального графа представляет собой синтез двух алгоритмов: α_1 (поиска внешнего центра) и α_2 (поиска внутреннего центра). Опишем далее принцип работы алгоритма α_3 .

Алгоритм α_3 поиска внешне-внутреннего центра

Алгоритм начинает свою работу с подграф-затравок L -го ранга $z_{S_L}^{(L)}$, $s_L = 1, n^{L-1}$. Рассмотрим подграф-затравку $z_1^{(L)}$, найдем кратчайшие пути от «общей» вершины $x_1^{(L)}$ до оставшихся $(n-1)$ вершин $v_{j_1}^{(L)}$ подграф-затравки $z_1^{(L)}$, и далее – кратчайшие пути от $(n-1)$ вершин $v_{j_1}^{(L)}$ до вершины $x_1^{(L)}$. Или по-другому найдем пару кратчайших путей: от вершины $x_1^{(L)}$ до вершины $v_{j_1}^{(L)}$ и обратно до $x_1^{(L)}$. Среди сумм кратчайших путей найдем максимальную и определим число $s_o(x_1^{(L)}) : s_{ot}(x_1^{(L)}) =$

$= \max_{j_1=1, n-1} [d(x_1^{(L)}, v_{j_1}^{(L)}) + d(v_{j_1}^{(L)}, x_1^{(L)})]$, которое назовем *числом внешне-внутреннего разделения* вершины $x_1^{(L)}$ подграф-затравки $z_1^{(L)}$. Далее, поочередно найдем числа внешне-внутреннего разделения $s_{ot}(x_{S_L}^{(L)}) = \max_{j_L=1, n-1} [d(x_{S_L}^{(L)}, v_{j_L}^{(L)}) + d(v_{j_L}^{(L)}, x_{S_L}^{(L)})]$ «общих» вершин всех подграф-затравок L -го ранга $z_{S_L}^{(L)}$, $s_L = 1, n^{L-1}$, где $d(x_{S_L}^{(L)}, x_{S_L}^{(L)}) = 0$. Поиск кратчайших путей осуществляется с помощью известного алгоритма Флойда [3].

Рассмотрим далее подграф-затравки $(L-1)$ -го ранга $z_{S_{L-1}}^{(L-1)}$, $s_{L-1} = 1, n^{L-2}$. Каждая подграф-затравка $(L-1)$ -го ранга $z_{S_{L-1}}^{(L-1)}$ также имеет одну общую вершину с соответствующими затравками $(L-2)$ -го ранга $z_{S_{L-2}}^{(L-2)}$, $s_{L-2} = 1, n^{L-3}$. Найдем числа внешне-внутреннего разделения для каждой общей вершины $x_{S_{L-1}}^{(L-1)}$ подграф-затравок $(L-1)$ -го ранга $z_{S_{L-1}}^{(L-1)}$, $s_{L-1} = 1, n^{L-2}$ следующим образом: $s_{ot}(x_{S_{L-1}}^{(L-1)}) = \max_{j_{L-1}} [d(x_{S_{L-1}}^{(L-1)}, v_{j_{L-1}}^{(L-1)}) + s_{ot}(x_{S_L}^{(L)}) + d(v_{j_{L-1}}^{(L-1)}, x_{S_{L-1}}^{(L-1)})]$, где $d(x_{S_{L-1}}^{(L-1)}, x_{S_{L-1}}^{(L-1)}) = 0$. То есть находим кратчайшие пути от общей вершины $x_{S_{L-1}}^{(L-1)}$ до оставшихся $(n-1)$ вершин подграф-затравки $z_{S_{L-1}}^{(L-1)}$. К длине кратчайшего пути $d(x_{S_{L-1}}^{(L-1)}, v_{j_{L-1}}^{(L-1)})$ добавляем соответствующее число разделения $s_{ot}(x_{S_L}^{(L)})$ и длину кратчайшего пути $d(v_{j_{L-1}}^{(L-1)}, x_{S_{L-1}}^{(L-1)})$. Среди получившихся сумм выбираем максимальную. Сумма $d(x_{S_{L-1}}^{(L-1)}, x_{S_{L-1}}^{(L-1)}) + s_{ot}(x_{S_L}^{(L)}) + d(x_{S_{L-1}}^{(L-1)}, x_{S_{L-1}}^{(L-1)}) = s_{ot}(x_{S_L}^{(L)})$, так как $d(x_{S_{L-1}}^{(L-1)}, x_{S_{L-1}}^{(L-1)}) = 0$.

Указанным способом находим числа внешне-внутреннего разделения $s_{ot}(x_{S_l}^{(l)})$ для общих вершин $x_{S_l}^{(l)}$ подграф-затравок $z_{S_l}^{(l)}$, $s_l = 1, n^{l-1}$ до 2-го ранга включительно, т. е. для всех $l = L, L-1, \dots, 2$.

На последнем шаге $l = 2$ найдены числа разделения $s_{ot}(x_{S_2}^{(2)})$, $s_2 = 1, n$ для общих вершин $x_{S_2}^{(2)}$ подграф-затравок 2-го ранга $z_{S_2}^{(2)}$ и одной подграф-затравки первого ранга $z_{S_1}^{(1)} = z_1^{(1)}$. Подграф-затравка $z_1^{(1)}$, по сути, соответствует графу G_1 из траектории $G_1, G_2, \dots, G_L$. Тогда для каждой вершины подграф-затравки $z_1^{(1)}$ найдено число $s_{ot}(x_{S_2}^{(2)})$, $s_2 = 1, n$.

Далее рассмотрим подграф-затравку $z_1^{(1)}$ как отдельный граф и найдем для каждой ее вершины $z_{S_1}^{(1)}$ число внешне-внутреннего разделения следующим образом: $s_{ot}(x_{S_1}^{(1)}) = \max_{j_1=1, n-1} [d(x_{S_1}^{(1)}, v_{j_1}^{(1)}) + s_{ot}(x_{S_2}^{(2)}) + d(v_{j_1}^{(1)}, x_{S_1}^{(1)})]$. Вершина x_{ot}^* , для кото-

рой число внешне-внутреннего разделения $s_{ot}(x_{ot}^*) = \min_{s_1=1, 2, \dots, n} [s_{ot}(x_{s_1}^{(1)})]$ минимальное, является внешне-внутренним центром предфрактального графа G_L .

Представим далее алгоритм α_3 , где для поиска пар кратчайших путей между любыми вершинами графа используется процедура Флойда.

Алгоритм α_3

Вход: взвешенный предфрактальный граф $G_L = (V_L, E_L)$.

Выход: x_{ot}^* – внешне-внутренний центр предфрактального графа G_L .

Шаг 1. Для каждой вершины $x_{s_L}^{(L)}$ подграф-затравки L -го ранга $z_{s_L}^{(L)}$, $s_L = \overline{1, n^{L-1}}$ найти число внешне-внутреннего разделения $s_{ot}(x_{s_L}^{(L)}) = \max_{j_L=1, n-1} [d(x_{s_L}^{(L)}, v_{j_L}^{(L)}) + d(v_{j_L}^{(L)}, x_{s_L}^{(L)})]$, где $d(x_{s_L}^{(L)}, x_{s_L}^{(L)}) = 0$. Поиск кратчайших путей между вершинами осуществляется с помощью процедуры Флойда.

Для всех $l = L - 1, L - 2, \dots, 2$ выполнить:

Шаг $L - l + 1$. Для каждой вершины $x_{s_l}^{(l)}$ подграф-затравки l -го ранга $z_{s_l}^{(l)}$ найти число внешне-внутреннего разделения $s_{ot}(x_{s_l}^{(l)}) = \max_{j_l=1, n-1} [d(x_{s_l}^{(l)}, v_{j_l}^{(l)}) + s_{ot}(x_{j_l}^{(l+1)}) + d(v_{j_l}^{(l)}, x_{s_l}^{(l)})]$, $s_l = \overline{1, n^{l-1}}$, где $d(x_{s_l}^{(l)}, x_{s_l}^{(l)}) = 0$. Поиск кратчайших путей между вершинами осуществляется с помощью процедуры Флойда.

Шаг L . Для каждой вершины $x_{s_1}^{(1)}$ подграф-затравки первого ранга $z_{s_1}^{(1)}$ найти число внешне-внутреннего разделения $s_{ot}(x_{s_1}^{(1)}) = \max_{j_1=1, n-1} [d(x_{s_1}^{(1)}, v_{j_1}^{(1)}) + s_{ot}(x_{j_1}^{(2)}) + d(v_{j_1}^{(1)}, x_{s_1}^{(1)})]$, $s_1 = \overline{1, n^{l-1}}$, где $d(x_{s_1}^{(1)}, x_{s_1}^{(1)}) = 0$. Поиск кратчайших путей между вершинами осуществляется с помощью процедуры Флойда.

Шаг $L + 1$. Из всех вершин $x_{s_1}^{(1)}$, $s_1 = 1, 2, \dots, n$ в качестве внешне-внутреннего центра предфрактального графа G_L выбрать вершину x_{ot}^* с наименьшим числом разделения: $s_{ot}(x_{ot}^*) = \min_{s_1=1, 2, \dots, n} [s_{ot}(x_{s_1}^{(1)})]$.

Процедура Флойда.

Вход: взвешенный граф $G = (V, E)$.

Выход: кратчайшие пути $d(v_i, v_j)$, $i, j = 1, 2, \dots, n$.

Теорема 1. Алгоритм α_3 выделяет внешне-внутренний центр предфрактального графа G_L , смежность старых ребер которого сохраняется.

Доказательство. Доказательство теоремы строится конструктивно. На каждом шаге алгоритма применяется процедура Флойда, не требующая доказательства.

На рисунке представлен пример поиска внешне-внутреннего центра предфрактального графа G_4 , смежность старых ребер которого сохраняется. Порождающая затравка H – ориентированный граф.

Предфрактальный граф G_4 взвешен в соответствии с правилом взвешивания ребер, где начальный отрезок $[a; b] = [27; 54]$, а весовой коэффициент $\theta = 1/3$. Веса ребер указаны рядом с ребрами обычным шрифтом. Алгоритм поиска внешне-внутреннего центра начинается свою работу с подграф-затравок 4-го ранга. Внутри подграф-затравок 4-го ранга мелким шрифтом указаны числа внешне-внутреннего разделения для вершин $x_{s_4}^{(4)}$, $s_4 = 1, 2, \dots, 27$. Число внешне-внутреннего разделения для вершины $x_1^{(4)}$ найдено следующим образом:

$$\begin{aligned} s_{ot}(x_1^{(4)}) &= \max(d(v_1^{(4)}, x_1^{(4)}) + d(v_1^{(4)}, x_1^{(4)}); \\ d(x_1^{(4)}, v_2^{(4)}) + d(v_2^{(4)}, x_1^{(4)})) &= \max(2 + 2, 6; 3, 4 + 1, 2) = \\ &= \max(4, 6; 4, 6) = 4, 6. \end{aligned}$$

Далее переходим к подграф-затравкам 3-го ранга, внутри каждой указаны числа внешне-внутреннего разделения для вершин $x_{s_3}^{(3)}$, $s_3 = 1, 2, \dots, 9$. Число внешне-внутреннего разделения вершины $s_{ot}(x_1^{(3)})$ вычисляется соответственно правилу:

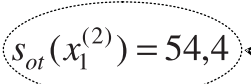
$$\begin{aligned} s_{ot}(x_1^{(3)}) &= \max(d(x_1^{(3)}, v_1^{(3)}) + s_{ot}(x_4^{(4)}) + d(v_1^{(3)}, x_1^{(3)}); \\ d(x_1^{(3)}, v_2^{(3)}) + s_{ot}(x_5^{(4)}) + d(v_2^{(3)}, x_1^{(3)}); \\ d(x_1^{(3)}, v_3^{(3)}) + s_{ot}(x_6^{(4)}) + d(v_3^{(3)}, x_1^{(3)})) &= \\ &= \max(0 + 3, 6 + 0; 4, 5 + 4, 3 + 9; 9, 5 + 4, 7 + 4) = \\ &= \max(3, 6; 18, 8; 18, 2) = 18, 8. \end{aligned}$$

Заметим, что длины кратчайших расстояний $d(x_1^{(3)}, v_1^{(3)})$ и $d(v_1^{(3)}, x_1^{(3)})$ равны нулю, т. к. $d(x_1^{(3)}, v_1^{(3)}) = d(v_1^{(3)}, x_1^{(3)}) = d(x_1^{(3)}, x_1^{(3)}) = 0$.

Поиск чисел внешне-внутреннего разделения вершин $x_{s_2}^{(2)}$, $s_2 = 1, 2, 3$, подграф-затравок второго ранга осуществляется аналогично, а кратчайшие расстояния $d(x_1^{(2)}, v_1^{(2)}) = d(v_1^{(2)}, x_1^{(2)}) = d(x_1^{(2)}, x_1^{(2)}) = 0$:

$$\begin{aligned} s_{ot}(x_1^{(2)}) &= \max(d(x_1^{(2)}, v_1^{(2)}) + s_{ot}(x_7^{(3)}) + d(v_1^{(2)}, x_1^{(2)}); \\ d(x_1^{(2)}, v_2^{(2)}) + s_{ot}(x_8^{(3)}) + d(v_2^{(2)}, x_1^{(2)}); \\ d(x_1^{(2)}, v_3^{(2)}) + s_{ot}(x_9^{(3)}) + d(v_3^{(2)}, x_1^{(2)})) &= \\ &= \max(0 + 16, 4 + 0; 10 + 16 + 28; 23 + 16, 4 + 15) = \\ &= \max(16, 4; 54; 54, 4) = 54, 4. \end{aligned}$$

На последнем шаге алгоритма находим числа внешне-внутреннего разделения для вершин $x_{s_1}^{(1)}$, $s_1 = 1, 2, 3$ и выбираем среди них минимальное:



Поиск внешне-внутреннего центра предфрактального графа G_4



$$s_{ot}(x_{ot}^*) = \min_{s_1=1, 2, 3} [s_{ot}(x_{s_1}^{(1)})] = \min[s_{ot}(x_1^{(1)}); s_{ot}(x_2^{(1)}); s_{ot}(x_3^{(1)})] = \\ = \min[174,4; 177,6; 179,8] = 174,4.$$

Таким образом, внешне-внутренним центром является вершина $x_1^{(1)}$, для которой число внешне-внутреннего разделения минимальное: $x_{ot}^* = x_1^{(1)}$. Внешне-внутренний радиус предфрактального графа G_4 равен числу внешне-внутреннего разделения вершины x_{ot}^* : $\rho_{ot} = s_{ot}(x_{ot}^*) = 174,4$.

Заметим, что внешний центр $x_o^* = x_1^{(1)}$ совпадает с внешне-внутренним центром $x_{ot}^* = x_1^{(1)}$, а внутренним центром является другая вершина $x_i^* = x_3^{(1)}$. Внешне-внутренний радиус $\rho_{ot} = 174,4$.

Теорема 2. Вычислительная сложность алгоритма α_3 равна $O(2n^3 \cdot N)$.

Доказательство. На первом шаге алгоритм работает на подграф-затравках L -го ранга $z_{s_L}^{(L)}$, где находит числа внешне-внутреннего разделения вершин $x_{s_L}^{(L)}$, $s_L = 1, 2, \dots, n^{L-1}$. Поиск числа внешне-внутреннего разделения на отдельно взятой подграф-затравке состоит из: 1) поиска пар кратчайших путей $d(x_{s_L}^{(L)}, v_{j_L}^{(L)})$ и $d(v_{j_L}^{(L)}, x_{s_L}^{(L)})$ и 2) выбора максимального элемента из сумм кратчайших путей $d(x_{s_L}^{(L)}, v_{j_L}^{(L)}) + d(v_{j_L}^{(L)}, x_{s_L}^{(L)})$. Поиск кратчайших путей осуществляется с помощью процедуры Флойда, вычислительная сложность которого равна n^3 , а выбор максимального элемента или,

в худшем случае, сортировка элементов по возрастанию, требует выполнения также n^2 операций [3]. Тогда поиск числа внешне-внутреннего разделения на одной подграф-затравке требует в сумме $n^3 + n^2 \leq 2n^3$ операций. Число всех подграф-затравок L -го ранга равно n^{L-1} , для выполнения шага 1 или поиска всех $s_{ot}(x_{s_L}^{(L)})$, $s_L = 1, 2, \dots, n^{L-1}$ требуется $2n^3 \cdot n^{L-1} = 2n^{L+2}$ операций.

На следующем шаге осуществляется поиск чисел внешне-внутреннего разделения для вершин $x_{s_{L-1}}^{(L-1)}$, $s_L = 1, 2, \dots, n^{L-2}$ что требует $2n^3 \cdot n^{L-2} = 2n^{L+1}$ операций. Продолжая поиск чисел внешне-внутреннего разделения вершин до второго ранга включительно, получаем: $2n^{L+2} + 2n^{L+1} + 2n^L + \dots + 2n^4$. На последнем шаге осуществляется поиск чисел внешне-внутреннего разделения для вершин $x_{s_1}^{(1)}$, $s_1 = 1, 2, \dots, n$, что требует $n^3 + n^3 = 2n^3$ операций плюс поиск максимального элемента n^2 операций.

В сумме получаем:

$$2n^{L+2} + 2n^{L+1} + 2n^L + \dots + 2n^4 + 2n^3 + n^2 \leq \\ \leq 2n^{L+2} + 2n^{L+1} + 2n^L + \dots + 2n^4 + 2n^3 + 2n^2 = \\ = 2 \cdot \frac{n^{L+2} \cdot n - n^2}{n-1} \leq 2n^{L+3} - n^2 \leq 2n^{L+3}. n^L = 2n^3 \cdot N.$$

Таким образом, вычислительная сложность алгоритма α_3 равна $O(2n^3 \cdot N)$.

СПИСОК ЛИТЕРАТУРЫ

1. Кочкаров, А.М. Распознавание фрактальных графов. Алгоритмический подход [Текст]/А.М. Кочкаров. – Нижний Архыз: РАН CAO, 1998.
2. Кристофидес, Н. Теория графов. Алгоритмиче-

ский подход [Текст]/Н. Кристофидес. – М.: Мир, 1978.

3. Асанов, М.О. Дискретная математика: графы, матроиды, алгоритмы [Текст]/М.О. Асанов, В.А. Баранский, В.В. Расин. – Ижевск: НИЦ «РХД», 2001.

УДК 621.391

С.С. Владимиров

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ МАЖОРИТАРНОГО ДЕКОДИРОВАНИЯ КОМБИНАЦИИ ЭКВИДИСТАНТНОГО КОДА ПО К ЛИНЕЙНО-НЕЗАВИСИМЫМ ЭЛЕМЕНТАМ

В системах передачи информации широко используется помехоустойчивое кодирование, повышающее вероятность правильного приема информации в реальном канале связи. Все помехоустойчивые коды обладают избыточностью, что

уменьшает скорость передачи информации в канале. Вместе с тем, алгоритмы их декодирования имеют ограниченную способность исправления ошибок. В связи с этим возникает необходимость разработки и исследования новых алгоритмов

декодирования, которые позволят увеличить вероятность правильного декодирования принятой комбинации при сохранении избыточности кода или уменьшить избыточность при той же помехоустойчивости.

В статье рассмотрен алгоритм декодирования комбинаций эквидистантного двоичного циклического кода с длиной $n = 2^k - 1$ по k линейно-независимым элементам последовательности, а также приведены результаты, полученные при моделировании данного алгоритма. Выполнено сравнение этого алгоритма с другими методами декодирования указанных эквидистантных кодов. В качестве критериев сравнения выбраны вероятности правильного и ложного декодирования.

Комбинации эквидистантного циклического кода (n, k) представляют собой М-последовательности или коды максимальной длины, порождаемые некоторым примитивным многочленом степени k . В общем случае комбинация такого кода для простого поля $GF(p)$ имеет длину $n = p^k - 1$, первые k элементов которой являются информационными. Все ненулевые комбинации эквидистантного циклического кода над простым полем $GF(p)$ имеют одинаковый вес, равный $\omega = p^{k-1}(p-1)$. Следовательно, минимальное кодовое расстояние равно $d_{\min} = p^{k-1}(p-1)$, а для двоичных кодов $d_{\min} = 2^{k-1}$. В дальнейшем, в статье вместо термина «комбинация эквидистантного циклического кода» будет использоваться термин «М-последовательность» или термин «последовательность максимальной длины» (ПМД).

Математический аппарат исследуемого алгоритма подробно описан в [1]. Поэтому рассмотрим этот вопрос очень кратко.

Зададим некоторую рекуррентную последовательность $\{s\}$ максимальной длины в общем виде через функцию-след:

$$\{s\} = [s_0 s_1 s_2 s_3 \dots s_{p^k-2}] = [T(\epsilon^m)T(\epsilon^{m+1}) \dots T(\epsilon^{m+p^k-2})] = [T(c)T(c\epsilon) \dots T(c\epsilon^{p^k-2})]. \quad (1)$$

Последовательность $\{s\}$ полностью определяется видом характеристического многочлена и начальной фазой $c = \epsilon^m$. Соответственно, если характеристический многочлен задан заранее, то задача декодирования М-последовательности сводится к определению начальной фазы $c = \epsilon^m$.

Для решения поставленной задачи выберем k линейно-независимых элементов последовательности $\{s\}$, а именно $s_{i_1}, s_{i_2}, s_{i_3}, \dots, s_{i_k}$, из которых

составим систему уравнений:

$$\begin{aligned} s_{i_1} &= T(\epsilon^{m+i_1}) = T(c\epsilon^{i_1}), \\ s_{i_2} &= T(\epsilon^{m+i_2}) = T(c\epsilon^{i_2}), \\ &\vdots \\ s_{i_k} &= T(\epsilon^{m+i_k}) = T(c\epsilon^{i_k}). \end{aligned} \quad (2)$$

Данную систему уравнений путем некоторых преобразований можно привести к виду:

$$S = \Theta \cdot A, \quad (3)$$

где

$$S = \begin{bmatrix} s_{i_1} \\ s_{i_2} \\ \vdots \\ s_{i_k} \end{bmatrix}; \quad \Theta = F^j \theta_0 = \begin{bmatrix} (F^{i_1} \theta_0)^T \\ (F^{i_2} \theta_0)^T \\ \vdots \\ (F^{i_k} \theta_0)^T \end{bmatrix} = \begin{bmatrix} f_{0,0} & f_{0,1} & \dots & f_{0,k-1} \\ f_{1,0} & f_{1,1} & \dots & f_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{k-1,0} & f_{k-1,1} & \dots & f_{k-1,k-1} \end{bmatrix}; \quad A = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{bmatrix}.$$

Если эта система уравнений линейно-независима, то ее решение относительно элементов $a_0, a_1, \dots, a_{k-1}$ дает векторное представление элемента ϵ^m , определяющего начальную фазу последовательности $\{s\}$.

Для решения системы уравнений (3) используем следующую формулу:

$$A = \Theta^{-1} \cdot S, \quad (4)$$

где Θ^{-1} — обратная матрица для матрицы Θ .

Перебирая таким образом все k -элементные линейно-независимые комбинации, можно организовать мажоритарную обработку М-последовательности.

При этом возникает задача определения количества возможных линейно-независимых k -элементных комбинаций и самих этих комбинаций.

Существует два способа решения данной задачи. Это аналитический метод, который позволяет определить лишь количество линейно-независимых комбинаций, и метод программного моделирования, который определяет также и сами комбинации.

Аналитический метод подробно описан в [1], поэтому рассмотрим основные моменты метода.

Произвольные k -элементы М-последовательности $\{s\}$ представим в виде многочлена:

$$U(x) = s_{j_1} x^{j_1} + s_{j_2} x^{j_2} + \dots + s_{j_k} x^{j_k}. \quad (5)$$

Выборка элементов $s_{j_1}, s_{j_2}, \dots, s_{j_k}$ будет линейно зависимой в том случае, если среди ненулевых элементов многочлена $U(x)$ найдутся такие, что их сумма будет представлять собой ненулевой многочлен $\varphi_1(x)$ сравнимый с нулем по модулю характеристического многочлена $P(x)$ степени k .

Так как многочлен $\varphi_1(x)$ делится без остатка на многочлен $P(x)$, то он будет соответствовать определенной разрешенной кодовой комбинации эквидистантного циклического кода. Поэтому для определения количества k -элементных сочетаний M -последовательности, среди которых будут линейно зависимые, достаточно найти весовой спектр такого циклического кода.

Для двоичных циклических кодов весовой спектр находится по формуле:

$$f(z) = \frac{1}{n+1} \left\{ (1+z)^n + n(1+z)^{\frac{n-1}{2}} (1-z)^{\frac{n+1}{2}} \right\}. \quad (6)$$

Общее число линейно-независимых k -элементных сочетаний элементов принимаемой M -последовательности равно:

$$B = C_N^k - A = C_N^k - \sum_{\omega=\omega_{\min}}^k C_{N-\omega}^{k-\omega} A(\omega). \quad (7)$$

Для примера определим число линейно-независимых сочетаний для двоичной M -последовательности с характеристическим многочленом $P(x) = 1 + x + x^4$ и периодом $N = 2^4 - 1 = 15$ ($k = 4$).

В соответствии с формулой (6), распределение весов $A(\omega)$ кодовых слов будет следующим:

$$\begin{aligned} f(z) &= \frac{1}{16} [(1+z)^{15} + 15(1+z)^7 (1-z)^8] = \\ &= 1 + 35z^3 + 105z^4 + 168z^5 + 280z^6 + 435z^7 + 435z^8 + \\ &\quad + 280z^9 + 168z^{10} + 105z^{11} + 35z^{12} + z^{15}. \end{aligned}$$

Таким образом, согласно формуле (7) число линейно-независимых сочетаний в рассматриваемой M -последовательности будет равно:

$$\begin{aligned} B &= C_{15}^4 - \sum_{\omega=3}^4 C_{15-\omega}^{4-\omega} A(\omega) = 1365 - (12A(3) + A(4)) = \\ &= 1365 - (12 \cdot 35 + 105) = 840. \end{aligned}$$

Теперь рассмотрим решение задачи определения количества возможных линейно-независимых k -элементных комбинаций путем моделирования. В данном случае моделирование представляет собой перебор всех возможных k -элементных комбинаций и проверку их на линейную независимость.

В случае, если комбинация линейно-независима, она записывается в память для последующего использования в процессе декодирования.

На рис. 1 показана блок-схема алгоритма определения возможных линейно-независимых k -элементных комбинаций и их количества.

Вычисления, проведенные по данному алгоритму, показали такое же количество k -элементных линейно-независимых комбинаций, как и вычисления, проведенные по аналитическому методу.

Учитывая все вышесказанное, составим алгоритм мажоритарного декодирования M -последовательности с характеристическим многочленом $P(x)$ степени k и периодом $N = 2^k - 1$. Фактически этот алгоритм представляет собой полный перебор всех возможных линейно-независимых k -элементных комбинаций и выполнение для каждой из них вышеприведенного алгоритма. После перебора всех комбинаций правильным считается ответ, полученный большее число раз (имеющий наибольший вес).

Каждой k -элементной линейно-независимой комбинации соответствует своя система уравнений (3) и, соответственно, своя обратная матрица для данной системы. Поэтому для отдельно взятого характеристического многочлена можно один раз вычислить список k -элементных линейно-независимой комбинаций и соответствующих им обратных матриц, записать их в память и при декодировании просто считывать их из памяти.

Линейно-независимые k -элементные комбинации удобно записывать в память в виде так называемых «масок» длиной $N = 2^k - 1$, в которых линейно-независимым элементам $s_{i_1}, s_{i_2}, \dots, s_{i_k}$ соответствуют единицы на позициях $i_1, i_2, \dots, i_k$, а на остальных позициях стоят нули. В этом случае матрица S в системе (3) вычисляется путем перемножения «маски» и принятой M -последовательности.

Чтобы начать декодирование принимаемой M -последовательности, не дожидаясь приема всей последовательности, маски при записи следует сортировать по позиции i_k .

Таким образом, декодирование может быть начато сразу же после получения первых k элементов M -последовательности.

Еще одной особенностью данного алгоритма является то, что при количестве «масок» S для определения правильного ответа достаточно получить $C/2 + 1$ одинаковых ответов. То есть декодирование может быть завершено еще до того,

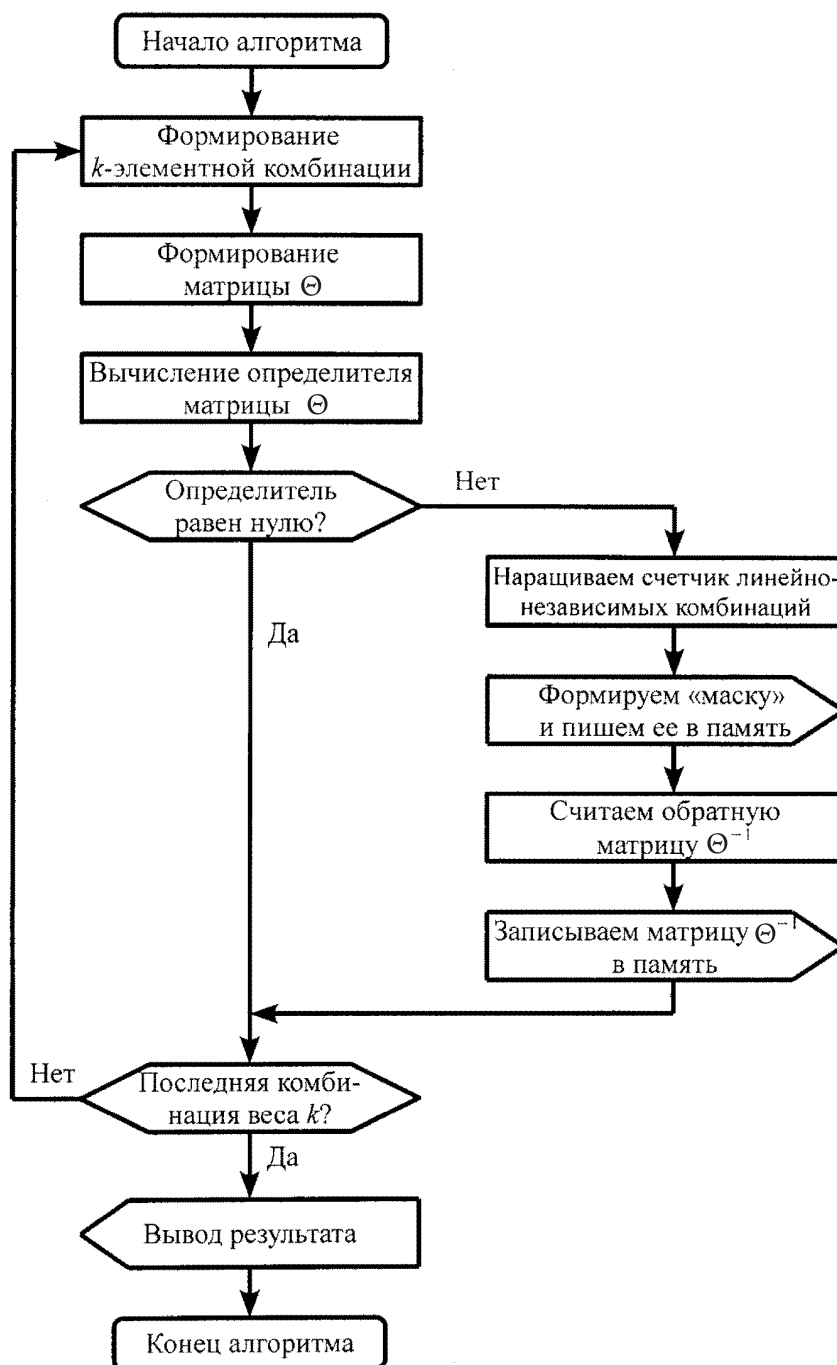


Рис. 1. Алгоритм определения линейно-независимых k -элементных комбинаций и их количества

как будет получена вся M -последовательность и перебраны все «маски».

Если число одинаковых ответов меньше $C/2 + 1$, то может возникнуть ситуация, когда есть несколько ответов с одинаковым весом, при этом решение об ответе принимать нельзя, т. е. будет отказ от декодирования.

Учитывая все вышесказанное, состав-

вим алгоритм мажоритарного декодирования M -последовательности.

Необходимо отметить, что в алгоритме, показанном на рис. 2, подразумевается, что «маски» и соответствующие им обратные матрицы определены заранее по алгоритму, показанному на рис. 1, и хранятся в памяти декодера.

Проведем проверку корректирующих

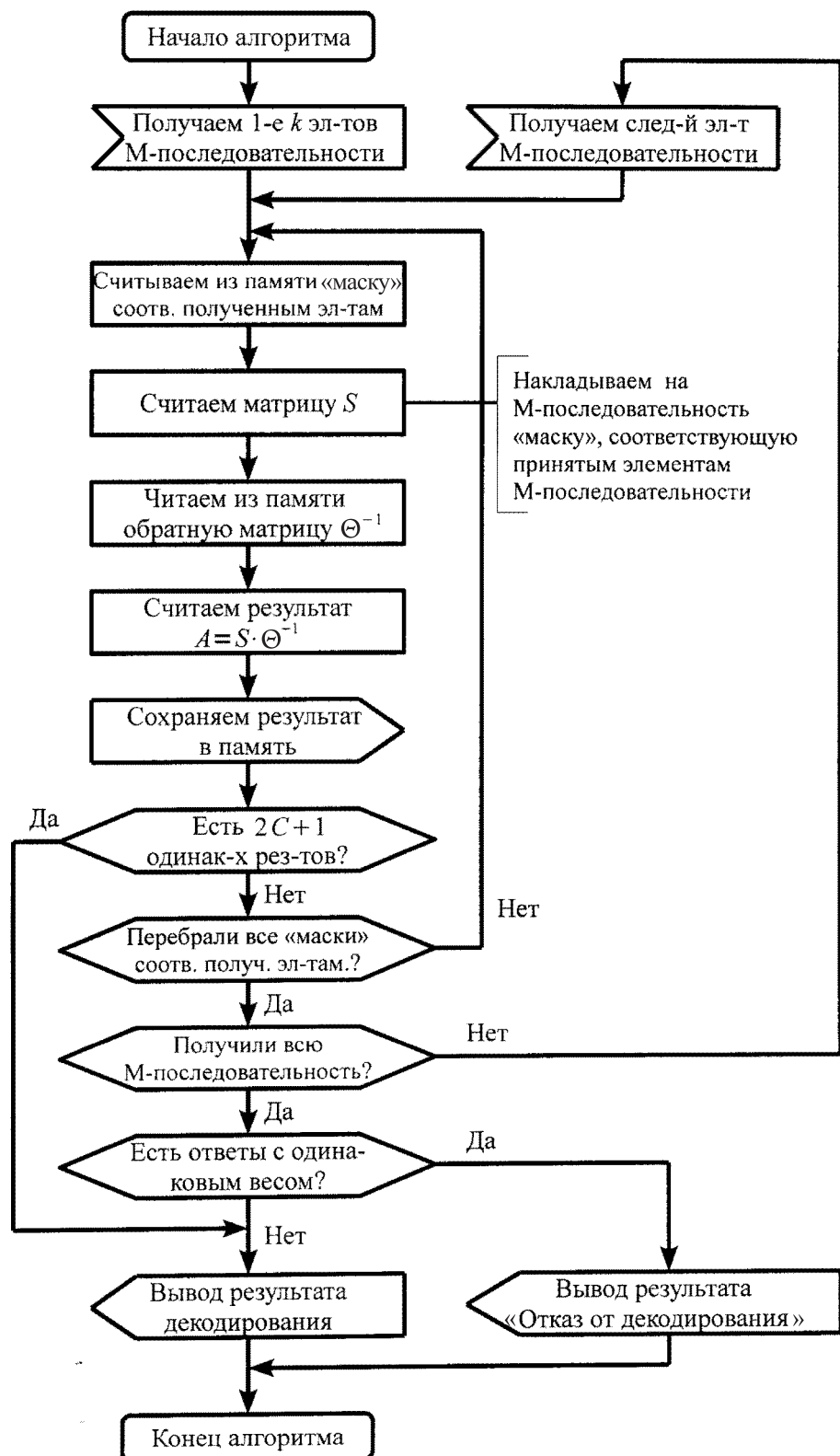


Рис. 2. Алгоритм мажоритарного декодирования M -последовательности по k -элементным линейно-независимым комбинациям

свойств алгоритма мажоритарного декодирования последовательности максимальной длины для М-последовательности с характеристическим многочленом $P(x) = 1 + x + x^4$ и периодом $N = 2^4 - 1 = 15$.

В качестве критерия выберем долю исправляемых ошибок различной кратности.

Для проверки была написана программа, моделирующая алгоритм, показанный на рис. 2. В данной модели на заданную М-последовательность накладываются по модулю 2 ошибки, начиная с ошибок кратности 1 и заканчивая ошибками той кратности, на которой ошибки перестают исправляться. Результатом выполнения модели является таблица соответствия количества верно исправленных ошибок их кратности.

В табл. 1 приведены результаты моделирования в сравнении с такими алгоритмами декодирования, как корреляционный (согласованные фильтры), табличный (на основании таблицы остатков от деления на образующий многочлен) и метод двойственного базиса (МДБ) с децимациями.

Как видно из таблицы, алгоритм мажоритарного декодирования по k произвольным линейно-независимым элементам по своим корректирующим свойствам совпадает с такими методами декодирования как корреляционный и табличный, а метод двойственного базиса с децимациями превосходит все эти методы по количеству исправляемых ошибок. При этом реализация алгоритма МДБ существенно проще из-за того, что в нем используется лишь 15 комбинаций-«масок» против 840 «масок» в исследуемом алгоритме. Результаты, полученные для исследуемого алгоритма, объясняются тем, что при числе ошибок большем трех возникает большое количество отказов в декодировании, вызванных появлением

максимумов одинакового веса и появлением не обнаруживаемых ошибок.

Теперь проведем проверку корректирующих свойств исследуемого алгоритма при исправлении ошибок и стираний. Стирание представляет собой символ, о котором невозможно сказать на приемной стороне, равен он единице или нулю. Например [2], если при передаче двоичных символов по каналу с АБГШ с двоичной фазовой манипуляцией, т. е. с отображением $0 \rightarrow +1$ и $1 \rightarrow -1$, значение принятого сигнала слишком близко к нулю. Соответственно, стертую позицию невозможно использовать при «жестком» декодировании.

Для исправления ошибок и стираний необходимо изменить исследуемый алгоритм. При декодировании стираний, те «маски», которые попадают хотя бы на одну стертую позицию, отбрасываются. Поэтому при исправлении ошибок и стираний использование алгоритма МДБ с децимациями невозможно, поскольку каждое стирание исключает из процесса декодирования k «масок», что при наличии дополнительных ошибок не позволит получить верный результат. Использование табличного и корреляционного методов при стираниях также затруднено. Исследуемый же алгоритм, благодаря использованию большого числа «масок», позволяет получить выборку значений, по которым и выбирается результат. Блок-схема измененного алгоритма представлена на рис. 3.

Для проверки исправляющей способности исследуемого алгоритма при исправлении ошибок и стираний была написана программа, моделирующая алгоритм, показанный на рисунке.

В табл. 2 приведены результаты моделирования исследуемого алгоритма при исправлении ошибок и стираний.

Таблица 1

Сравнение исправляющей способности исследуемого алгоритма с существующими алгоритмами декодирования

Кратность ошибок	Общее количество ошибок	Исследуемый метод	Корреляционный метод	Табличный метод	МДБ с децимациями
1	15	15 (100 %)	15 (100 %)	15 (100 %)	15 (100 %)
2	105	105 (100 %)	105 (100 %)	105 (100 %)	105 (100 %)
3	455	455 (100 %)	455 (100 %)	455 (100 %)	455 (100 %)
4	1365	476 (34,9 %)	476 (34,9 %)	419 (30,7 %)	990 (72,5 %)
5	3003	0	0	0	825 (27,5 %)
6	5005	0	0	0	240 (4,8 %)
7	6435	0	0	0	75 (1,1 %)
8	6435	0	0	0	15 (<1 %)

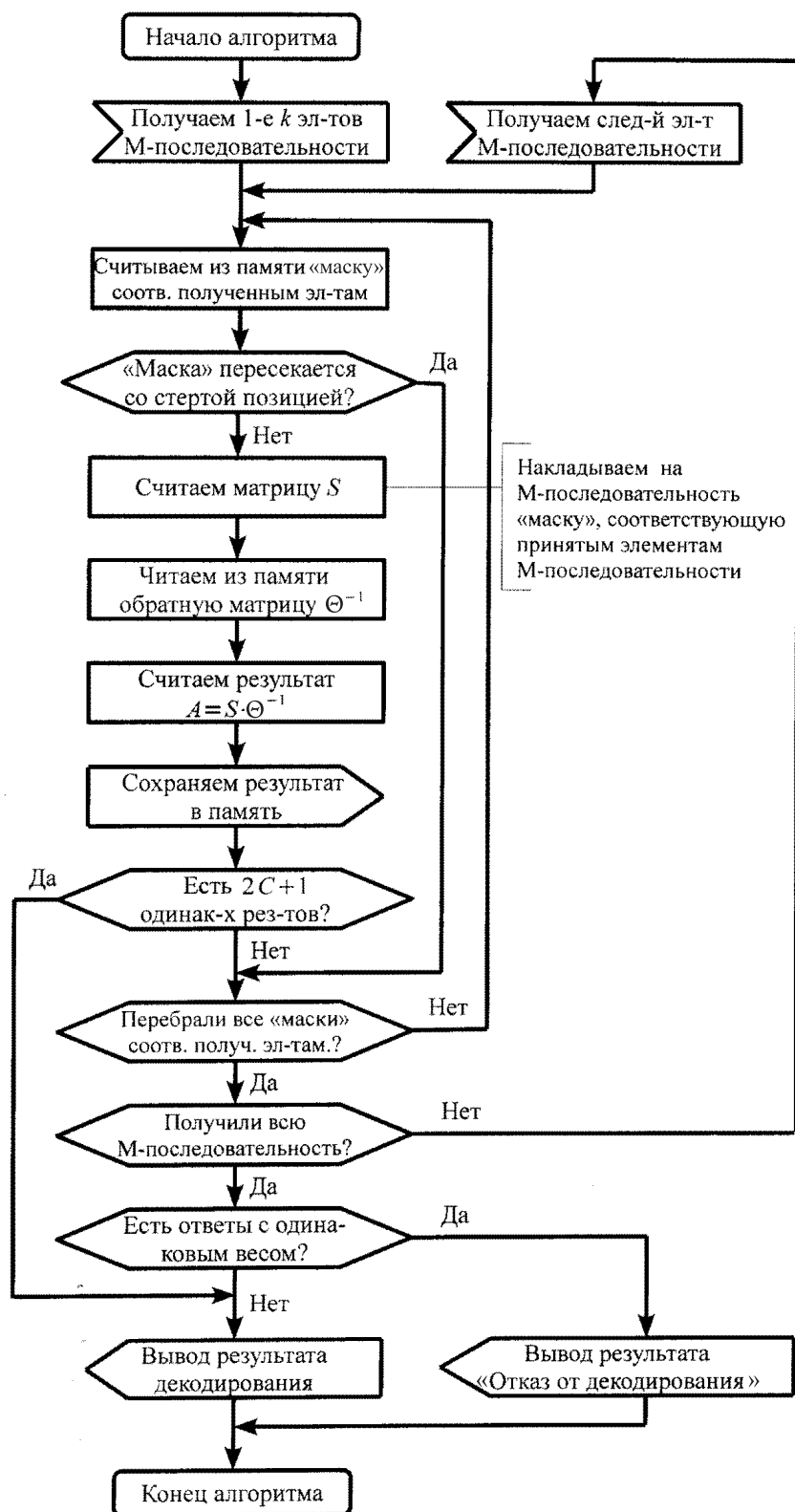


Рис. 3. Алгоритм мажоритарного декодирования М-последовательности по k -элементным линейно-независимым комбинациям для исправления ошибок и стираний

Таблица 2

Исправляющая способность исследуемого алгоритма
при исправлении ошибок и стираний для (n, k) кода (15, 4)

Кратность стираний	Кратность ошибок									
	0		1		2		3		4	
	Всего	Исправл.	Всего	Исправл.	Всего	Исправл.	Всего	Исправл.	Всего	Исправл.
0	1	1 (100 %)	15	15 (100 %)	105	105 (100 %)	455	455 (100 %)	1365	476 (35 %)
1	15	15 (100 %)	14	14 (100 %)	91	91 (100 %)	364	364 (100 %)	1001	336-364 (33-36 %)
2	105	105 (100 %)	13	13 (100 %)	78	78 (100 %)	286	206-226 (72-79 %)	715	40-76 (5-10 %)
3	455	455 (100 %)	12	12 (100 %)	66	66 (100 %)	220	40-128 (18-58 %)	495	0-19 (0-3 %)
4	1365	1365 (100 %)	11	11 (100 %)	55	49-55 (89-100 %)	165	25-53 (15-32 %)	330	0
5	3003	3003 (100 %)	10	10 (100 %)	45	15-36 (33-80 %)	120	0-17 (0-14 %)	210	0
6	5005	5005 (100 %)	9	7-9 (77-100 %)	36	2-23 (5-63 %)	84	0-2 (0-2 %)	126	0
7	6435	6435 (100 %)	8	6-8 (75-100 %)	28	0-9 (0-32 %)	56	0	70	0
8	6435	6420 (99.8 %)	7	1-7 (14-100 %)	21	0-2 (0-9 %)	35	0	35	0
9	5005	4900 (98 %)	6	0-3 (0-50 %)	15	0	20	0	15	0
10	3003	2688 (90 %)	5	0	10	0	10	0	5	0
11	1365	840 (61 %)	4	0	6	0	4	0	1	0
12	455	0	3	0	3	0	1	0	0	0

Подводя итог, можно сделать ряд выводов о применимости алгоритма мажоритарного декодирования по k произвольным линейно-независимым элементам.

Результаты моделирования процессов мажоритарного декодирования комбинации эквидистантного кода по k линейно-независимым элементам, приведенные в табл. 1, показывают, что при использовании исследованного алгоритма только для исправления ошибок, его корректирующие свойства оказываются аналогичными корректи-

рующим свойствам корреляционного алгоритма декодирования. Моделирование работы алгоритма мажоритарного декодирования по k линейно-независимым элементам при исправлении ошибок и стираний показывает, что данный алгоритм позволяет восстанавливать достоверную кодовую комбинацию при потере до $(n-1)/2$ элементов, что подтверждается данными моделирования, представленными в табл. 2. Таким образом, исследованный алгоритм можно рекомендовать к применению для исправления ошибок и стираний.

СПИСОК ЛИТЕРАТУРЫ

1. Когновицкий, О.С. Двойственный базис и его применение в телекоммуникациях [Текст]/О.С. Когновицкий.—СПб: Линк, 2009. — 424 с.

2. Морелос-Сарагоса, Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение [Текст]/Р. Морелос-Сарагоса; пер. с англ. В.Б. Афанасьева.—М.: Техносфера, 2005.—320 с.

ПРИМЕНЕНИЕ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ В СИСТЕМЕ ОХРАНЫ НА ОСНОВЕ МЕЖМОДОВОЙ ИНТЕРФЕРЕНЦИИ В ВОЛОКОННОМ СВЕТОВОДЕ

Волоконные интерферометрические датчики применяются для измерения или регистрации изменения различных физических величин, например, температуры, давления. Одним из них является межмодовый волоконный интерферометр (МВИ) – датчик, использующий интерференцию мод на выходе многомодового световода. Он состоит из лазера, многомодового световода и фотоприемника, принимающего часть выходного излучения световода (рис. 1). Внешние возмущения световода, такие как изменение температуры, давление, растяжение, изгибы вызывают изменения разности фазовых задержек мод, вследствие чего изменяется интерференционная картина на выходе световода, представляющая собой набор характерных ярких и темных пятен. Чтобы регистрировать эти изменения, на фотоприемник подается часть интерференционной картины, поскольку общая интенсивность излучения остается неизменной.

Данный датчик может применяться для регистрации внешних воздействий (возмущений), когда не требуется точного измерения величины

воздействия, а необходимо только его обнаружение. Подобные задачи возникают при разработке систем охраны периметра объектов, технического мониторинга протяженных конструкций и т. п.

Основной проблемой при регистрации воздействий на световод с использованием сигнала межмодовой интерференции является фединг амплитуды сигнала. Причем, учитывая особенности поведения сигналов вследствие связи мод, это явление фактически невозможно контролировать или компенсировать за счет применения вспомогательных модулирующих сигналов и подобных методов обработки сигнала МВИ.

Одним из эффективных методов борьбы с федингом является использование многоканального приема сигнала. Ранее применялось аналоговое устройство обработки сигналов десяти каналов. В статье рассматриваются вопросы, связанные с разработкой аналогичного устройства с цифровой обработкой сигналов (ЦОС) для охранной сигнализации периметров объектов. ЦОС позволяет существенно расширить возможности устройства по сравнению с ранее использованной

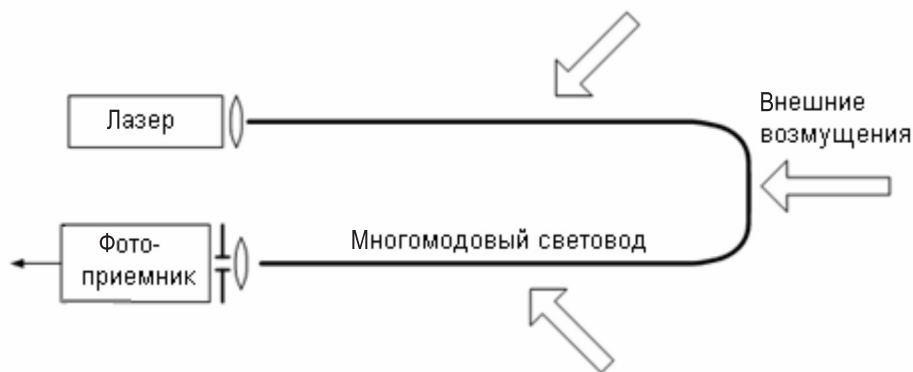


Рис. 1. Схема межмодового волоконного интерферометра

аналоговой обработкой, сделать устройство более универсальным и упростить принципиальную схему.

Теоретические основы

Для обоснования свойств сигналов МВИ необходимо опираться на подробную математическую модель, которая была разработана в лаборатории волоконной оптики кафедры радиофизики. Приведем основные математические выражения, необходимые для описания сигналов МВИ.

Рассмотрим интерференцию мод на выходном торце многомодового световода длиной L , возбуждаемого когерентным источником с частотой ω . Запишем известное выражение для распределения интенсивности на выходе многомодового световода [1, 2]:

$$I(\vec{r}) = \frac{1}{2} \sum_{i=1}^N A_i^2 E_i^2 + \sum_{i=1}^N \sum_{\substack{k=1 \\ i > k}}^N (A_i A_k^*) (E_i E_k^*) \cos \varphi_{ik},$$

где N – число распространяющихся мод; $E_i = E_i(r, \varphi)$, $E_k = E_k(r, \varphi)$ – нормированные модовые функции i -й (k -й) моды (распределение поля моды в поперечном сечении световода); r – радиус-вектор точки поперечного сечения световода; A_i (A_k) – амплитуды полей i -й (k -й) моды, определяемые мощностью источника и условиями возбуждения мод; φ_{ik} – разность фаз i -й и k -й мод

$$\begin{aligned} \varphi_{ik} &= (\beta_i - \beta_k)L + (\Phi_i - \Phi_k) = \\ &= \Delta\beta_{ik}L + (\Phi_i - \Phi_k) = \Delta\beta_{ik}L + (\Phi_{ik}), \end{aligned}$$

где β_i – постоянная распространения i -й моды, Φ_i – учитывают дополнительные сдвиги фазы, вызванные изгибами и другими неоднородностями реального световода, что обычно интерпретируется в рамках «взаимодействия мод» [3, 4], или как следствие анизотропных свойств сердцевин световода.

При регистрации света на выходе волокна, фотоприемное устройство принимает излучение с некоторой площадки s . При этом интенсивность светового потока, регистрируемого фотоприемником, можно записать в виде

$$I(\vec{r}) = \frac{1}{2} \sum_{i=1}^N A_i^2 \int_s E_i^2 ds + \sum_{i=1}^N \sum_{\substack{k=1 \\ i > k}}^N (A_i A_k^*) \left(\int_s E_i E_k^* ds \right) \cos \varphi_{ik}.$$

Введем обозначения:

$$\begin{aligned} \int_s E_i^2 ds &= C_i^2, \quad \int_s E_i E_k^* ds = C_{ik}, \\ \frac{1}{2} \sum_{i=1}^N A_i^2 \int_s E_i^2 ds &= \frac{1}{2} \sum_{i=1}^N A_i^2 C_i^2 = I_0. \end{aligned}$$

Тогда интенсивность, регистрируемую приемником, можно записать как

$$\begin{aligned} I(s) &= \frac{1}{2} \sum_{i=1}^N A_i^2 C_i^2 + \sum_{i=1}^N \sum_{\substack{k=1 \\ i > k}}^N A_{ik} C_{ik} \cos \varphi_{ik} = \\ &= I_0 + \sum_{i=1}^N \sum_{\substack{k=1 \\ i > k}}^N A_{ik} C_{ik} \cos \varphi_{ik}, \end{aligned}$$

где $A_{ik} = A_i A_k$.

Первое слагаемое представляет собой сумму интенсивностей мод (по площадке s). Предполагая постоянство условий возбуждения световода и отсутствие воздействий, заметно изменяющих потери мощности в волокне. Это слагаемое будем считать константой. Второе слагаемое – интерференционная составляющая – результат интерференции полей мод. Оно изменяется в результате воздействий на световод, меняющих разность фаз мод φ_{ik} , и описывает собственно сигнал МВИ.

В реальных условиях МВИ подвергается многочисленным воздействиям $V(t)$, которые изменяют разность фаз мод φ_{ik} и приводят к изменению регистрируемой фотоприемником выходной интенсивности МВИ $I(t)$. Прежде всего, это механические (вибрационные) воздействия на волокно. Они разделяются на «полезные воздействия» δV , которые несут в себе информацию, и паразитные воздействия, «помехи», которые необходимо отфильтровать. Отклик МВИ на «полезные воздействия» обозначен δI (рис. 2).

Реальные полезные механические воздействия, как правило, находятся в частотном диапазоне от единиц до нескольких сотен Гц. В большинстве случаев нет информации о форме сигнала полезного воздействия или времени, когда оно будет происходить, поэтому составляющая сигнала МВИ, вызванная помехами, попадающими в «полезный» частотный диапазон, фактически неотличима от составляющей, связанной с информационным воздействием, и мешает его регистрации.

Другим классом воздействий ΔV , вызывающих значительные сдвиги в разности фаз мод φ_{ik} , являются изменения внешних условий, в которых находится световод, например, температуры, давления. Они, как и механические помехи, вызывают паразитные, неинформативные изменения выходной интенсивности. Однако существенное отличие данного типа воздействий состоит в том, что изменения внешних условий являются квазистатическими (рис. 2). В реальных системах

частотный диапазон изменений, определяемых внешними условиями, обычно находится в области ниже сотых долей Гц, что и определяет их влияние на работу МВИ. Отклик МВИ на квазистатические изменения внешних условий обозначен ΔI .

Обозначим квазистатические, медленные изменения разности фаз мод – $\Delta\varphi_{ik}$, а колебания φ_{ik} в частотном диапазоне полезных воздействий («быстрые» изменения) – $\delta\varphi_{ik}$. Обозначим также начальные значения разности фаз мод φ_{ik}^0 , после чего можем записать полные разности фаз в виде

$$\varphi_{ik}(t) = \varphi_{ik}^0 + \Delta\varphi_{ik}(t) + \delta\varphi_{ik}(t),$$

где первые два слагаемых образуют квазистатическую часть разностей фаз мод, а последнее слагаемое описывает переменную составляющую.

Для образования сигнала межмодовой интерференции имеет значение не изменение самих фазовых задержек мод, а изменения их разности.

Такой процесс в литературе называют дифференциальной фазовой модуляцией (ДФМ) [1, 5, 6]. Выделяются два механизма формирования ДФМ. Первый из них предполагает изменение фазового набега моды, как результат изменения оптической длины участка, подвергнутого воздействию; второй является следствием взаимодействия мод [1, 4, 6], которое вызывает изменение постоянных распространения.

Для ДФМ первого рода получено выражение для реакции МВИ на слабое переменное воздействие [7, 8]:

$$\begin{aligned} \delta I(t) &= - \sum_{i=1}^N \sum_{k=1}^N A_{ik} C_{ik} \sin(\varphi_{ik}^0 + \Delta\varphi_{ik}(t)) \cdot \delta\varphi_{ik}(t) = \\ &= - \left(\sum_{i=1}^N \sum_{k=1}^N A_{ik} C_{ik} \sin(\varphi_{ik}^0 + \Delta\varphi_{ik}(t)) \cdot \Delta\beta_{ik} \right) \cdot l \cdot K \cdot \delta V(t), \end{aligned}$$

где $\Delta\beta_{ik}$ – изменение постоянной распространения; l – длина участка, подвергаемого воздействию;

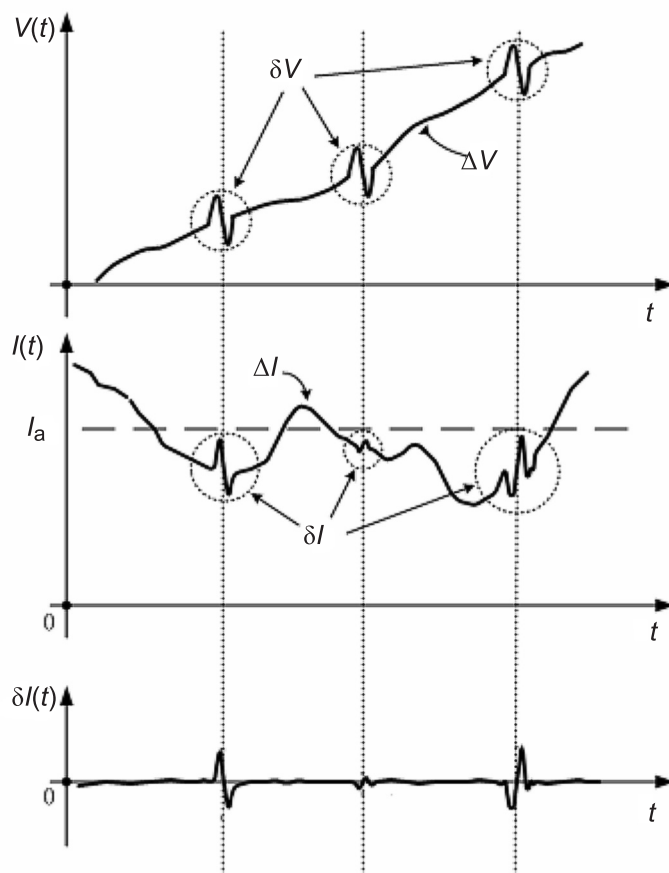


Рис. 2. Примерный вид сигнала МВИ в результате внешних воздействий на световод: верхний график – внешнее воздействие; средний график – сигнал МВИ; нижний график – отклик МВИ на переменное воздействие (после его выделения ФВЧ)

K – коэффициент относительного приращения фазового набег для данного воздействия.

Сигнал МВИ, возникающий за счет механизма связи мод, выглядит следующим образом:

$$\delta I(t) = - \sum_{i=1}^N \sum_{k=1}^N A_{ik} C_{ik} \sin(\varphi_{ik}^0 + \Delta\varphi_{ik}(t)) \cdot \delta\varphi_{ik}(t) =$$

$$= - \left(\sum_{i=1}^N \sum_{k=1}^N A_{ik} C_{ik} (\beta_i K_i - \beta_k K_k) \sin(\varphi_{ik}^0 + \Delta\varphi_{ik}(t)) \right) \cdot l \cdot \delta V(t).$$

Здесь невозможно вынести из-под суммы коэффициенты K_i , что приводит к существенным отличиям в свойствах фединга сигналов и в работе устройств на основе МВИ.

Наличие этого математического аппарата позволило моделировать свойства реальных сигналов МВИ. Было обосновано утверждение о невозможности применения вспомогательных модулирующих сигналов и т. п. методов для борьбы с федингом. Все это привело к решению использовать многоканальный прием сигнала МВИ [9, 10]. Он заключается в том, что несколько фотоприемников регистрируют разные части интерференционной картины, и по их сигналам формируется общий выходной сигнал. При этом маловероятно одновременное появление неудачного наложения спекл-картины и диафрагмы на всех участках регистрации.

Устройство обработки сигнала МВИ

Как уже отмечалось, ранее применялись многоканальные аналоговые устройства обработки сигнала МВИ. Но они имели ряд ограничений. Во-первых, было трудно реализовать сложные алгоритмы обработки, во-вторых, для каждого алгоритма необходимо спроектировать свою принципиальную схему.

Наш новый подход подразумевает создание многоканального устройства с ЦОС. Цифровая обработка позволяет реализовывать различные алгоритмы принятия решений в одной схеме, а также существенно усложнять их. Следовательно, такая система становится универсальной для различных объектов охраны. Все это позволяет выйти на серийное производство схем обработки, что существенно снижает их стоимость.

Устройство состоит из предварительной аналоговой части и непосредственно цифровой части. Аналоговая часть (рис. 3) предназначена для приема, фильтрации и предварительного усиления сигнала МВИ. Фильтр верхних частот предназначен для удаления квазистатической составляющей сигнала.

Одним из важных вопросов является выбор числа каналов и, соответственно, фотоприемников. С учетом предварительного анализа и проведенных экспериментов [9, 10] было принято решение использовать восемь каналов. Это близко к оптимальному числу, с точки зрения свойств спекл-картины. С другой стороны, при этом задействованы все восемь входов АЦП микроконтроллера.

Фототранзисторы расположены в форме круга, чтобы максимально эффективно использовать площадь спекл-картины.

Аналоговый сигнал фотодетектора преобразуется в цифровой с помощью АЦП, встроенного в микроконтроллер (рис. 4). При этом реализованы несколько вариантов обработки сигналов: обработка внутри микроконтроллера, передача сигнала на персональный компьютер (ПК) и дальнейшая обработка в ПК, разделение функций между микроконтроллером и ПК. Выбор варианта обработки зависит от сложности алгоритмов и

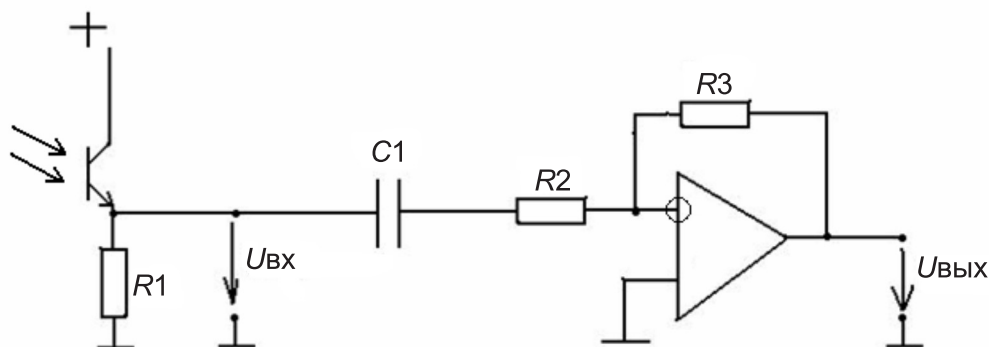


Рис. 3. Схема одного канала приема и усиления сигнала МВИ

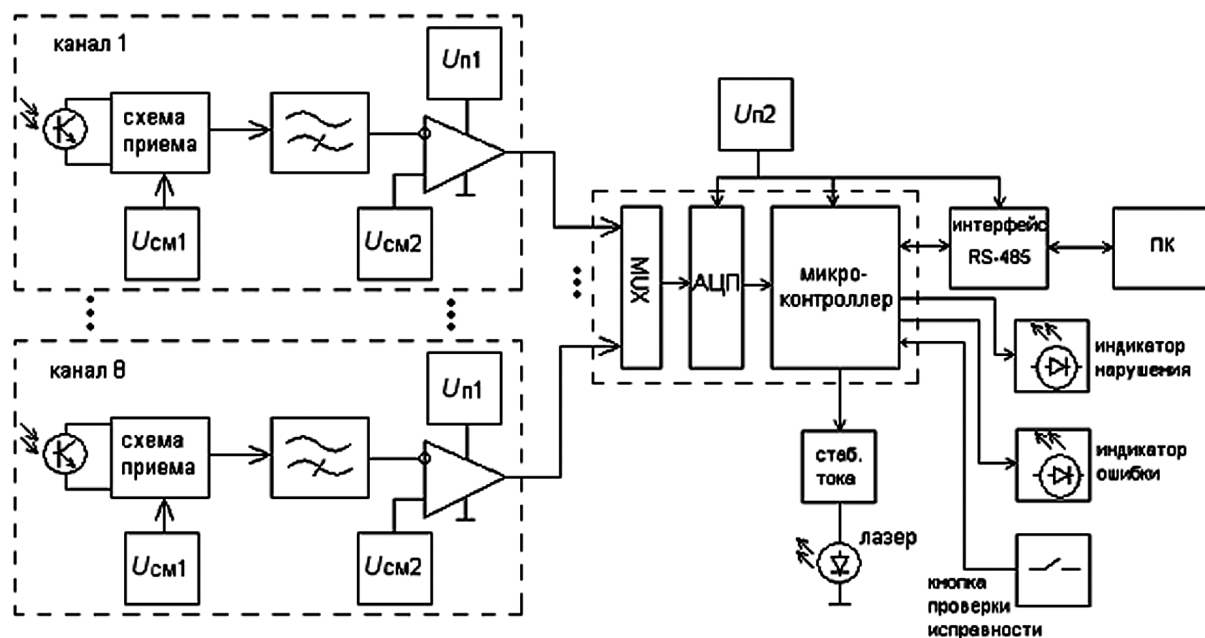


Рис. 4. Структурная схема устройства

от требований заказчика. Связь устройства с ПК реализуется с помощью интерфейса RS-485, позволяющего нескольким приемникам работать с одним компьютером.

На основе описанной математической модели были развиты и опробованы несколько алгоритмов принятия решения о тревоге. Для примера, можно привести самые простые варианты, когда осуществляется суммирование модулей сигналов по всем каналам приема или выбор максимального по модулю сигнала:

$$U(t) = \sum_{i=1}^N |U_i(t)| \text{ или } U(t) = \max |U_i(t)|.$$

Результирующий сигнал сравнивается с пороговым уровнем.

Необходимо учитывать, что на чувствительный элемент воздействуют не только полезные сигналы (нарушитель), но и паразитные помехи, например, проезжающие автомобили. В качестве простейшего метода борьбы с ложными тревогами используется переменный порог срабатывания, отслеживающий общий уровень помех.

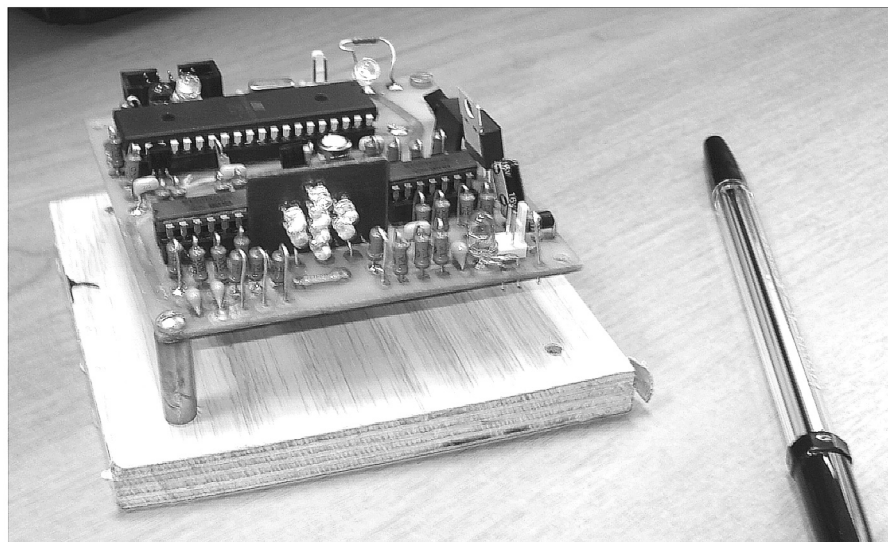


Рис. 5. Фотография экспериментальной платы приемника МВИ

При разработке устройства была предусмотрена функция периодической автоматической проверки работоспособности датчика.

Нами был изготовлен блок регистрации и обработки сигнала МВИ. Его фотография приведена на рис. 5.

В настоящее время ведется тестирование блока с регистрацией сигналов в реальных условиях.

По итогам тестирования предполагается сделать вывод об эффективности используемых алгоритмов в различных условиях и, соответственно, об области их применения. Также рассматривается вопрос объединения данной системы с другими системами охраны, такими, как, например, видеонаблюдение.

СПИСОК ЛИТЕРАТУРЫ

1. **Бутусов, М.М.** Волоконная оптика и приборостроение [Текст]/М.М. Бутусов, С.Л. Галкин, С.П. Оробинский [и др.]. –Л.: Машиностроение, 1987. –328 с.
2. **Кульчин, Ю.Н.** Распределенные волоконно-оптические измерительные системы [Текст]/Ю.Н. Кульчин. –М.: Физматлит, 2001.–272 с.
3. **Снайдер, А.** Теория оптических волноводов [Текст]/А. Снайдер, Дж. Лав. –М.: Радио и связь, 1987. –656 с.
4. **Унгер, Х.Г.** Планарные и волоконные оптические волноводы [Текст]/Х.Г. Унгер. –М.: Мир, 1980. – 656 с.
5. **Гуляев, Ю.В.** Модуляционные эффекты в волоконных световодах [Текст]/Ю.В. Гуляев, М.Я. Меш, В.В. Проклов. –М: Радио и связь, 1991. –152 с.
6. **Котов, О.И.** Фазовая модуляция и связь мод в двухмодовых волоконных световодах [Текст]/О.И. Котов, О.Л. Марусов, В.М. Николаев//Письма в ЖТФ. –1990. –Т. 16. –№ 7. –С. 48–52.
7. **Котов, О.И.** Регистрация воздействий на световод с использованием межмодовой интерференции [Текст]/О.И. Котов, Л.Б. Лиокумович, С.И. Марков//Тезисы Межд. конф. «Лазеры. Измерения. Информация». СПб, 25-26 июня 2003. –С. 25–26.
8. **Косарева, Л.И.** Два механизма фазовой модуляции в многомодовых волоконных световодах [Текст]/Л.И. Косарева, О.И. Котов, Л.Б. Лиокумович [и др.]//Письма в ЖТФ. –2000. –Т. 26. –№ 2. –С. 52–63.
9. **Косарева, Л.И.** Многоканальный прием сигналов межмодового интерферометра в условиях помех [Текст]/Л.И. Косарева, О.И. Котов, Л.Б. Лиокумович [и др.]//Письма в ЖТФ. –2000. –Т. 26. № 22. –С.26–35.
10. **Косарева, Л.И.** Особенности многоканального приема сигналов межмодового волоконно-оптического интерферометра: снижение амплитудного фединга [Текст]/Л.И. Косарева, О.И. Котов, Л.Б. Лиокумович [и др.]//Письма в ЖТФ. –2000. –Т. 26. –№ 18. –С.85–94.

Международный конкурс-семинар «Таймырские чтения»



Уважаемые читатели!

В марте 2010 года в Норильском индустриальном институте состоялась IX Международная научно-практическая конференция «Таймырские чтения 2010». Современные тенденции развития экономики, направленные на глобализацию производства, повышение качества выпускаемой продукции, максимальное удовлетворение требований заказчика напрямую связаны с всесторонней интеллектуализацией предприятий. В новой, информационно-ориентированной экономике эффективная деятельность приводит к необходимости синтеза инновационных организационных структур и внедрения новых методов управления, нацеленных на наиболее полное использование возможностей современных информационных технологий для создания конкурентного преимущества на рынке. Информационные технологии позволяют существенно повысить устойчивость развития промышленного региона в криолитозоне.

Основная цель проведения конференции – актуализация научных исследований в области социально-экономических, экологических и производственных проблем, возникающих при функционировании организаций и предприятий на Крайнем Севере.

В этом году особое внимание было уделено разработке и применению современных технологий управления в экстремальных условиях жизнедеятельности, экологическим аспектам природопользования, а также вопросам генезиса коренных народов Таймыра, Сибири и всего мира. В работе форума приняли участие представители 29 организаций, университетов и предприятий из Финляндии, Венгрии, Германии и России. Постоянными и активными участниками конференции являются руководители и специалисты предприятий группы компаний «Норильский никель».

На страницах журнала «Научно-технические ведомости СПбГПУ» мы представляем вашему вниманию доклады, активно обсуждавшиеся в рамках данного форума и получившие наибольшее количество положительных откликов.

Проректор по научной работе НИИ

Михаил Антонович Маркеев

ИСПОЛЬЗОВАНИЕ МЯГКИХ ВЫЧИСЛЕНИЙ ДЛЯ ОПТИМИЗАЦИИ ПАРАМЕТРОВ ДЕВИАЦИИ ПРИЗНАКОВ ПРИ РАСПОЗНАВАНИИ ОБРАЗОВ

В системах распознавания образов основными задачами являются извлечение идентификационных признаков исходного образа и проведение этапа классификации и кластеризации над идентификационными признаками. На практике на этапе классификации для получения более качественного результата распознавания необходимо нелинейно варьировать параметр отклонения идентификационных признаков. В данной статье предлагается метод оптимизации параметра девиации идентификационных признаков при распознавании образов на основе нечетких нейронных сетей (ННС).

Быстрые алгоритмы обучения и интерпретируемость накопленных знаний – эти факторы сделали сегодня ННС одним из самых перспективных и эффективных инструментов мягких вычислений, в т. ч. при решении задач классификации. ННС осуществляют выводы на основе аппарата нечеткой логики, однако параметры функций принадлежности настраиваются с использованием алгоритмов обучения нейронных сетей. Поэтому для подбора параметров таких сетей применим метод обратного распространения ошибки, изначально предложенный для обучения персептрона. Для этого модуль нечеткого управления представляется в форме многослойной нейронной сети (МНС) [1].

Постановка задачи. Обзор подходов к прогнозированию девиации признаков

Пусть имеется вектор признаков $X = \{x_1^o, x_1^o, \dots, x_n^o\}$ исследуемого объекта O . Имеется вектор идентификационных признаков каждого из известных классов $P = \{p_1(x_1^1, x_2^1, \dots, x_n^1), \dots, p_m(x_1^m, x_2^m, \dots, x_n^m)\}$. Необходимо определить параметр q , который позволит максимизировать отклонение x_k^o от x_l^d для O , при этом отклонение x_k^o для классов отличных от d будет минимальным. Другими словами, $Deviation(X, p_d) \rightarrow \max$, при $Deviation(X, p_e) \rightarrow \min$, где d – наиболее близкий класс для O .

На настоящий момент существует несколько методов оптимизации данного параметра:

1. Вручную, людьми, которые являются экспертами в определенной области. Основной недостаток данного метода состоит в том, что при различных входных данных параметр девиации для фиксированного класса образов будет иметь отличную от предыдущей величину, поэтому необходимо постоянное присутствие эксперта. Второй недостаток – человеческий фактор, который влечет к накоплению ошибок классификации.

2. С использованием нейронных сетей различных архитектур. При этом следует отметить два существенных недостатка: сложность подбора архитектуры сети для конкретной задачи и сложность в понимании процесса при выборе того или иного решения, поскольку его принятие хранится в виде массивов синаптических весов.

3. Методы на базе систем нечеткого вывода. Основной их недостаток заключается в необходимости вручную, на основе экспертных оценок, сформировать большое количество правил и параметров сети для каждой задачи.

Перечисленные методы в большинстве случаев справляются со своей задачей, но из-за имеющихся недостатков предполагают трудности при разработке унифицированных систем распознавания образов.

В данной статье предложена система нейро-нечеткого вывода типа Сугено, реализующая прогнозирование параметра q , избавленная от недостатков, характерных для вышеуказанных методов.

Архитектура нейро-нечеткой сети

Предлагаемая структура системы нейро-нечеткого вывода имеет тип Сугено и состоит из четырех слоев. Каждый слой ННС имеет строгое предназначение: первый слой – фаззификация входных переменных; второй – агрегирование значений активации условия; третий – агрегиро-



вание нечетких правил; четвертый слой – дефаззификация выхода системы. Сформированная база правил представлена совокупностью следующих правил для каждой функции принадлежности:

Если $(in_1 \text{ is } mf_{1j})$ И $(in_2 \text{ is } mf_{2j})$ И $(in_3 \text{ is } mf_{3j})$...
... И $(in_k \text{ is } mf_{kj})$ ТО $(out \text{ is } omf_j)$, (1)

где in_i – i -я входная переменная; mf_{ij} – j -я функция принадлежности i -й входной переменной; out – выходная переменная; omf_j – j -я функция принадлежности выходной переменной; $j = 1, k$. Максимальное количество k функций принадлежности определяется с помощью алгоритма субстративной кластеризации. Данный алгоритм используется в качестве экспертной оценки термов лингвистических переменных для формирования количества функций принадлежности.

На первом шаге субстративной кластеризации определяют точки, которые могут быть потенциальными центрами кластеров. На втором шаге для каждой такой точки рассчитывается значение потенциала, показывающего возможность формирования кластера в ее окрестности. Чем плотнее расположены объекты в окрестности потенциального центра кластера, тем выше значение его потенциала. После этого итерационно выбираются центры кластеров среди точек с максимальными потенциалами. Для алгоритма субстративной кластеризации число потенциальных центров кластеров должно быть конечным. В качестве начальных центров кластеров могут быть выбраны все объекты кластеризации.

Потенциалы центров кластеров рассчитываются следующим образом:

$$P(C_i) = \sum_{j=1}^N \exp(-\alpha \cdot D(C_i, X_j)), \quad i = \overline{1, M}, \quad (2)$$

где M – количество начальных центров; N – количество объектов кластеризации; $C_i = (c_{1,i}, c_{2,i}, \dots, c_{n,i})$ – потенциальный центр i -го кластера; α – положительная константа; $D(C_i, X_k)$ – расстояние между потенциальным центром кластера C_i и объектом кластеризации X_k .

Центром первого кластера выбирается точка с наибольшим потенциалом. Прежде чем выбрать следующий центр кластера необходимо исключить влияние только что найденного кластера. Для этого значения потенциала для оставшихся возможных центров кластеров пересчитываются с учетом вклада центра только что найденного кластера V_1 :

$$P(C_i) = P(C_i) - P(V_1) \cdot \exp(-\beta \cdot D(C_i, V_1)), \quad (3)$$

где $V_1 = \arg \max_{C_i, i=\overline{1, M}} (P(C_i))$, β – положительная константа.

Центр второго кластера определяется по максимальному значению обновленных потенциалов оставшихся точек. Итерационная процедура пересчета потенциалов и выделения центров кластеров продолжается до тех пор, пока максимальное значение потенциала превышает изначально заданный порог [3].

Сформированная таким образом база правил ННС далее используется для оптимизации идентификационных признаков при помощи функций принадлежности, реализующих *первый слой* ННС; определяется степень принадлежности для каждой входной переменной. Функции принадлежности имеют вид Гаусса:

$$f_i(x) = \exp\left(-\frac{1}{2} \left(\frac{x - \beta_i}{\alpha_i}\right)^2\right), \quad (4)$$

где параметры α_i и β_i для каждой функции принадлежности $f_i(x)$ после обучения. Для авторской системы предметно-независимого распознавания графических образов, описанной в [2], количество функций принадлежности $k = 6$, а значения параметров α_i и β_i приведены в табл. 1. Диапазоны значений входных переменных для данной системы представлены в табл. 2.

На *втором слое* сети определяются уровни отсечения для левой части каждого правила, согласно операции минимизации:

$$a_i = \min_j \{mf_{ij}(in_j)\}, \quad (5)$$

где $i = \overline{1, 5}$, $j = \overline{1, 6}$.

На *третьем слое* вычисляются индивидуальные выходы продукционных правил (1):

$$z_i = \sum_{j=1}^m v_{ij} \cdot in_j, \quad (6)$$

где $i = \overline{1, 5}$, $j = \overline{1, 6}$; v_{ij} – некоторый параметр.

На *четвертом слое* выполняется дефаззификация выходной переменной для системы Сугено традиционно методом центра тяжести для одноточечных множеств (*Center of Gravity for Singletons*), по формуле:

$$y = \frac{\sum_{i=1}^n a_i \cdot z_i}{\sum_{i=1}^n a_i}, \quad (7)$$

Таблица 1

Значения параметров функций принадлежности

Номер входной переменной	Номер функции принадлежности											
	1		2		3		4		5		6	
	α	β	α	β	α	β	α	β	α	β	α	β
1	0,132	0,032	0,269	0,18	0,504	0,137	0,242	0,163	0,016	0,153	0,181	0,181
2	0,548	0,266	0,282	0,214	0,931	0,225	0,763	0,247	0,058	0,263	0,346	0,226
3	0,511	0,247	0,378	0,209	0,998	0,213	0,961	0,159	0,795	0,221	0,346	0,226
4	0,388	0,252	0,757	0,194	0,907	0,25	0,619	0,261	0,064	0,257	0,225	0,29
5	0,447	0,191	0,212	0,153	0,47	0,163	0,221	0,184	0,017	0,187	0,054	0,113

Таблица 2

Диапазоны значений входных переменных

Номер входной переменной	Диапазон значений	
	Нижний	Верхний
1	0,02	0,5267
2	0,06	0,8967
3	0,2833	0,9833
4	0,06667	0,89
5	0,02	0,6233

где n – число одноточечных нечетких множеств, каждое из которых характеризует единственное значение рассматриваемой выходной переменной.

Выходная переменная ННС определена на интервале $[0; 1]$. Для алгоритма, описанного в [2], реализующего предметно-независимое распознавание графических образов, выходные параметры обучающей выборки для прогнозирования параметра девиации находились на промежутке $[0,01; 0,2]$. Для графических образов (см. рис.) с максимальным потенциальным варьированием признаков оптимизирование значения q близким к 0,2, а для образов, которые мало подвержены модификации в пространстве – близким к 0,01.

В качестве сравнения отметим, что при необходимости прогнозирования девиации признаков с применением МНС на диапазоне 0,01 ... 0,2 с шагом 0,01 используется 20 различных состояний, что можно закодировать 5 нейронами с бинарными выходами выходного слоя в случае многослойного перцептрона. Входной слой можно смоделировать 15 нейронами (по правилам построения МНС, последующий слой должен содержать около 30 % от количества нейронов предыдущего слоя). Функция активации имеет сигмоидальный тип:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (8)$$

Ошибка классификации E при этом может быть определена, как

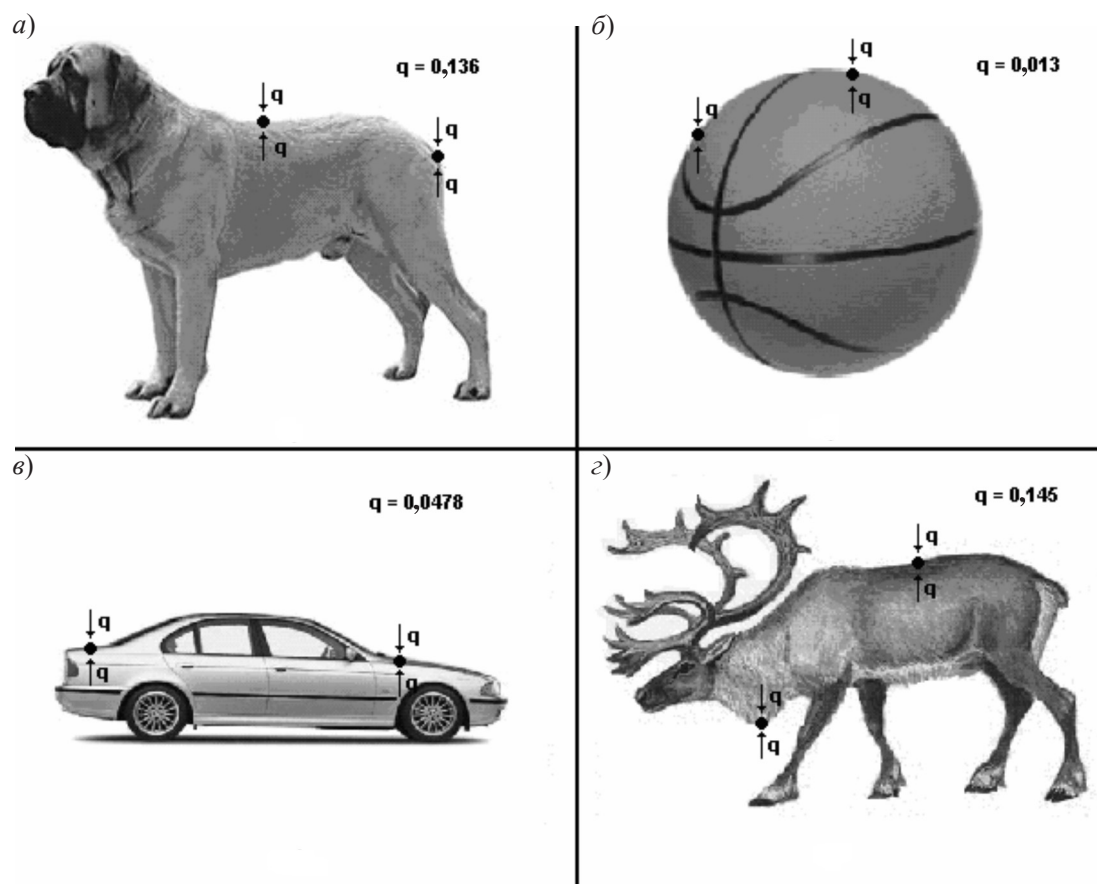
$$E(\{w_{i,j}\}) = \sqrt{\frac{1}{n} \cdot \sum_{k \in \text{Outputs}} (t_k - o_k)^2}, \quad (9)$$

где $w_{i,j}$ – вес, стоящий на ребре, соединяющем i -й и j -й узлы; o_i – выход i -го узла. Если нам известен обучающий пример, то t_k – правильные ответы сети; $k \in \text{Outputs}$.

Анализ полученных результатов

Система нейро-нечеткого вывода обучена и реализована в алгоритмах экспертной системы «*Ident Smart Studio*». Экспертная система спрогнозировала параметр девиации q для образов, изображенных на рисунке а, б, в и г, соответственно равным 0,136, 0,013, 0,0478, 0,145. Данный параметр позволяет признакам отклоняться на определенную величину при минимальной ошибке классификации.

В табл. 3 приведены основные характеристики методов прогнозирования девиации признаков на примере задачи идентификации, описанной в [2] для предметно-независимого распознавания



Исследуемые образы

Таблица 3

Сравнительная характеристика методов

Подход	Критерий				
	Скорость обучения	Ошибка обучения	Точность прогнозирования q на тестовых данных, %	Ошибка классификации на тестовых данных	Уровень автоматизации
Ручной способ	Использует опыт эксперта (от 1 дня до нескольких месяцев)	$\sim 0,05$	75	0,1	Отсутствует
Нейронная сеть	~ 8 с	$5 \cdot 10^{-5}$	87	0,087	Высокий
Нечеткий вывод	Использует опыт эксперта (от 1 дня до нескольких месяцев)	$\sim 0,01$	83	0,09	Низкий
Нейро-нечеткая сеть	~ 10 с	$3 \cdot 10^{-7}$	91	0,073	Высокий

графических образов. Тестовые данные содержали 1700 векторов идентификационных признаков. Значение выходной переменной ННС является вещественным числом, поэтому для оценки точности прогнозирования допускалось варьирование в пределах до 0,01 в большую и меньшую стороны. Ошибки обучения и классификации рассчитывались по формуле (9).

Как показывают экспериментальные исследования, подход с использованием ННС является

наиболее оправданным, поскольку при расширении диапазона девиации архитектура сети не изменится, дополнительный объем вычислений будет незначительным, по сравнению с МНС. Предлагаемый метод не требует ручной настройки параметров и конфигурации, как это требует нечеткий вывод, а также не требует постоянного присутствия эксперта в работе. Данный подход можно использовать для распознавания различных образов: визуальных и звуковых сигналов; финансовых и социальных ситуаций.

СПИСОК ЛИТЕРАТУРЫ

1. BaseGroups Labs. Технологии анализа данных [Электронный ресурс] <http://www.basegroup.ru>
2. **Фомичева, С.Г.** Нейро-нечеткий анализ графических образов экосистемы [Текст] / С.Г. Фомичева, С.А. Мальчевский // Научно-технические ведомости СПбГПУ. – 2009. – № 2. – С. 115–122.
3. **Демидова, Л.А.** Алгоритмы и системы нечеткого вывода в задачах диагностики городских инженер-

ных телекоммуникаций [Текст] / Л.А. Демидова, В.В. Кираковский, А.Н. Пылькин. – М.: Радио и связь – Горячая линия – Телеком. – 2005. – 592 с.

4. **Леоненков, А.В.** Нечеткое моделирование в среде Matlab и FuzzyTECH [Текст] / А.В. Леоненков. – СПб.: БХВ-Петербург, 2005. – 736 с.

УДК 681.5.017

А.И. Морозов, А.И. Писарев

МЕТОД СОЗДАНИЯ ЛИНГВИСТИЧЕСКИХ ПЕРЕМЕННЫХ НЕЧЕТКОЙ МОДЕЛИ

Алгоритм построения «нечеткой» модели Сугено благодаря виду подзаклучений [1, 2] своих правил позволяет получить адекватные результаты при наличии необходимых статистических данных. Существующие методы настройки базы правил Сугено основаны на компенсировании «мгновенной ошибки» (разницы между рассчитанным и фактическим значениями) [2]. Главными недостатками этих методов являются:

количество термов, выбор вида функций принадлежности и структура правил устанавливается «экспертом» и не изменяется при настройке «нечеткой» модели;

статистика используется только для расчета настроечных параметров функций принадлежности.

В данной статье предлагается метод построения «нечетких» моделей, лишенный этих недостатков.

Формализация задачи

Введем понятие *строки статистических данных* – совокупность значений входных и выходных параметров в фиксированный момент времени t_v , обозначим $Z_v = \{x_i(t_v), y_j(t_v)\}$, где $x_i(t_v)$ – значение i -го входного параметра в момент времени t_v , и $i = 1 \div N$, $v = 1 \div K$; $y_j(t_v)$ – значение j -го выходного параметра в момент времени t_v , где $j = 1 \div L$; $M = \{Z_v\}$ – множество строк статистических данных.

Статистические данные должны удовлетворять следующим требованиям:

количество строк статистических данных (K) и количество входных параметров (N) должны удовлетворять условию $K \gg N$;

строки статистических данных не должны повторяться;

статистические данные должны содержать информацию только об установившихся режимах работы объекта моделирования.

Практическое использование теории нечеткого вывода предполагает наличие функций принадлежности, которыми описываются входные параметры.

В литературе [1–3] предлагается несколько видов функций принадлежности и методов их построения. Наиболее часто используется функция Гаусса. Ее достоинство в том, что она задается всего двумя параметрами. В то же время функция никогда не принимает нулевое значение, что является существенным недостатком, т. к. на этапе фаззификации становятся активными все правила. Этот недостаток приводит к множеству дополнительных расчетов, а, следовательно, и к повышению трудоемкости.

Введем следующие обозначения:

U_i – количество термов i -го входного параметра; $\bar{x}_{i\gamma}$ – координата максимума γ -го терма i -го входного параметра, где $\gamma = 1 \div U_i$; $\Delta_{i\gamma}$ – носитель γ -го терма i -го входного параметра; $X_i = \{x_i(t_v)\}$ – множество значений i -го входного параметра; $\mu_{i\gamma}(x_i)$ – функция принадлежности γ -го терма i -го входного параметра.

Решением задачи будет считаться определение конечного числа i термов U_i лингвистической переменной с конкретными значениями $\bar{x}_{i\gamma}$ и $\Delta_{i\gamma}$, а также коэффициентов функций подзаклучений, максимально правдоподобно отображающих исходные статистические данные.

Для решения данной задачи предлагается применять следующую функцию принадлежности (пример функций принадлежности для трех термов T_{i1} , T_{i2} , T_{i3} показан на рис. 1).

Такая функция задается формулой:

$$\mu_{i\gamma}(x_i) = \begin{cases} \frac{\sin\left(\frac{2\pi}{\Delta_{i\gamma}}x_i + \frac{\pi}{2} - \frac{2\pi\bar{x}_{i\gamma}}{\Delta_{i\gamma}}\right) + 1}{2}, \\ \text{при } x_i \in \left(\bar{x}_{i\gamma} - \frac{\Delta_{i\gamma}}{2}; \bar{x}_{i\gamma} + \frac{\Delta_{i\gamma}}{2}\right); \\ 0, \text{ при } x_i \in \left(-\infty; \bar{x}_{i\gamma} - \frac{\Delta_{i\gamma}}{2}\right] \cup \left[\bar{x}_{i\gamma} + \frac{\Delta_{i\gamma}}{2}; +\infty\right). \end{cases} \quad (1)$$

Примем следующие обозначения:

$x_i^{\min}$, $x_i^{\max}$ – нижнее и верхнее значения i -го входного параметра; $x_{i\gamma}^{\min}$, $x_{i\gamma}^{\max}$ – координаты нижнего и верхнего граничных значений γ -го терма i -го входного параметра.

Алгоритм формирования термов лингвистических переменных

Для формирования термов входных параметров необходимо определить значения $\bar{x}_{i\gamma}$ и $\Delta_{i\gamma}$ для каждого из них. Процедура вычисления параметров следующая.

1. Введем две промежуточные переменные α и λ , начальное значение которых единица.
2. Произведем сортировку строк статистических данных Z_v во множестве M по возрастанию входного параметра X_i .
3. Определим количество термов для каждого входного параметра по формуле:

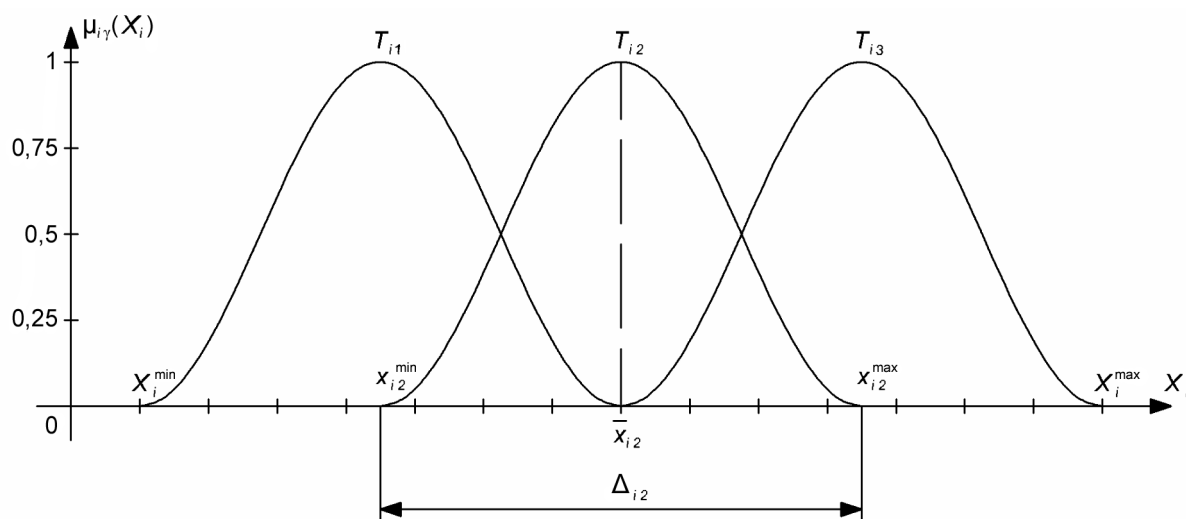


Рис. 1. Пример функций принадлежности

$$U_i = K - (\alpha(N + 1) - 1). \quad (2)$$

4. Выполним выборку V_γ^1 строк статистических данных Z_v по формуле:

$$V_\gamma^1 = \{M \mid x_\lambda(t_\gamma) \leq x_\lambda(t_v) \leq x_\lambda(t_{\gamma+\alpha(N+1)-1})\}. \quad (3)$$

5. Выполним выборку $V_{i\gamma}^2$ значений i -х входных параметров по формуле:

$$V_{i\gamma}^2 = \{X_i \mid x_i(t_v) \in V_\gamma^1\}. \quad (4)$$

6. Определим множества $V_{i\gamma}^{\min}$ и $V_{i\gamma}^{\max}$ значений i -х входных параметров, удовлетворяющих условиям:

$$\begin{aligned} V_{i\gamma}^{\min} &= \{X_i \mid x_i(t_v) < \min_v V_{i\gamma}^2\}, \\ V_{i\gamma}^{\max} &= \{X_i \mid x_i(t_v) > \max_v V_{i\gamma}^2\}. \end{aligned} \quad (5)$$

7. Определим границы термов $x_{i\gamma}^{\min}$ и $x_{i\gamma}^{\max}$ по формулам:

$$x_{i\gamma}^{\min} = \begin{cases} \max_v V_{i\gamma}^{\min}, & \text{при } \min_v V_{i\gamma}^2 > \min_v X_i; \\ X_i^{\min}, & \text{при } \min_v V_{i\gamma}^2 = \min_v X_i. \end{cases} \quad (6)$$

$$x_{i\gamma}^{\max} = \begin{cases} \min_v V_{i\gamma}^{\max}, & \text{при } \max_v V_{i\gamma}^2 < \max_v X_i; \\ X_i^{\max}, & \text{при } \max_v V_{i\gamma}^2 = \max_v X_i. \end{cases} \quad (7)$$

8. Объединим термы, у которых носители совпадают, либо один носитель находится внутри другого.

9. Определим для каждого терма значения координаты центра термов $\bar{x}_{i\gamma}$ и носителя $\Delta_{i\gamma}$ по формулам:

$$\bar{x}_{i\gamma} = x_{i\gamma}^{\min} + \frac{x_{i\gamma}^{\max} - x_{i\gamma}^{\min}}{2}, \quad \Delta_{i\gamma} = x_{i\gamma}^{\max} - x_{i\gamma}^{\min}. \quad (8)$$

10. Сформируем базу правил путем перебора всех возможных комбинаций термов входных параметров:

№ 1: Если $x_1(t) \in T_{11}$ и $x_2(t) \in T_{21}$ и ...
и $x_N(t) \in T_{N1}$, То $S_{11}(t)$, $S_{21}(t)$, ..., $S_{L1}(t)$;

№ 2: Если $x_1(t) \in T_{11}$ и $x_2(t) \in T_{21}$ и ...
и $x_N(t) \in T_{N2}$, То $S_{12}(t)$, $S_{22}(t)$, ..., $S_{L2}(t)$;
...

№ R: Если $x_1(t) \in T_{1U_1}$ и $x_2(t) \in T_{2U_2}$ и ...
и $x_N(t) \in T_{NU_N}$, То $S_{1R}(t)$, $S_{2R}(t)$, ..., $S_{LR}(t)$.

Здесь: $T_{i\gamma}$ – γ терм i -го входного параметра; S_{jp} – j -е подзаклучение p -го правила, где $p = 1 \div R$.

11. Определим количество правил R по формуле:

$$R = \prod_{i=1}^N U_i. \quad (9)$$

12. Проверим следующие условия (для того, чтобы в каждое из R правил попадало по $(N + 1)$ строк статистических данных):

1) если хотя бы одному из правил соответствует менее $(N + 1)$ строк, тогда переменная α увеличивается на единицу и повторяются все действия с п. 3;

2) если ни одному правилу не соответствует число строк менее $(N + 1)$, но хотя бы одному правилу соответствует число строк более $(N + 1)$, то такие правила необходимо делить следующим образом:

делению подвержены только термы λ -го входного параметра;

при делении термов структура правила (термы остальных входных параметров и подзаклучения) не изменяется;

количество новых правил определяется как разность между количеством строк статистических данных, соответствующих правилу, и числом входных параметров;

терм $T_{\lambda\gamma}$, соответствующий правилу с избыточным количеством строк, разделяется на термы ($T'_{\lambda\gamma}$, $T''_{\lambda\gamma}$ и т. д.) так, чтобы каждому новому правилу соответствовало $(N + 1)$ строк статистических данных;

после деления всех правил объединяются только одинаковые термы λ -го входного параметра; повторяющиеся правила сокращаются.

При определении количества термов и их параметров $\bar{x}_{i\gamma}$ и $\Delta_{i\gamma}$ одновременно формируется база правил.

Далее необходимо определить значения коэффициентов в подзаклучениях. Формула для расчета подзаклучений Сугено имеет вид:

$$S_{jp}(t) = B_{pj0}^x + \sum_{i=1}^N x_i(t) \cdot B_{pji}^x, \quad (10)$$

где $B_{pj\zeta}^x$ – коэффициенты j -го подзаклучения p -го правила, где $\zeta = 0 \div N$.

Преобразовав формулу (10), получим уравнение $(N + 1)$ -мерной плоскости в N координатах $x_i(t)$ и $S_{pj}(t)$:

$$\sum_{i=1}^N x_i(t) \cdot A1_{pji} + S_{pj}(t) \cdot A2_{pj} + A3_{pj} = 0, \quad (11)$$

где $A1_{pji}$, $A2_{pj}$, $A3_{pj}$ – множества параметров $(N + 1)$ -мерной плоскости.



При этом значения коэффициентов уравнения (10) будут определяться следующим образом:

$$B_{\rho ji}^x = \frac{A1_{\rho ji}}{-A2_{\rho j}}, \quad B_{\rho j0}^x = \frac{A3_{\rho j}}{-A2_{\rho j}}. \quad (12)$$

Для однозначного определения параметров $(N+1)$ -мерной плоскости достаточно знать координаты $(N+1)$ точек этой плоскости. Действия (п. 12) выполнялись для того, чтобы каждому правилу соответствовало $(N+1)$ строк статистических данных. Полученные в п.12 статистические данные используются для расчета коэффициентов уравнения (11).

Расчет коэффициентов $B_{\rho \zeta}^x$ уравнения (10) выполняется в следующей последовательности.

1. Определение функции принадлежности термов, соответствующих ρ -му правилу (множество B_ρ).

2. Выборка V_ρ^Z строк статистических данных по формуле:

$$V_\rho^Z = \{M \mid (0 < \mu_{1\gamma}(x_1) \in B_\rho) \text{ и } (0 < \mu_{2\gamma}(x_2) \in B_\rho) \text{ и } \dots (0 < \mu_{N\gamma}(x_N) \in B_\rho)\}, \quad (13)$$

где $\gamma = 1 \div U_\rho$, $v = 1 \div K$.

3. Составление трех квадратных матриц $V_{\rho ji}^{A1}$, $V_{\rho j}^{A2}$ и $V_{\rho j}^{A3}$ размером $(N+1) \times (N+1)$ из множества V_ρ^Z :

$V_{\rho ji}^{A1}$ включает N первых столбцов множества V_ρ^Z (столбец $(N+1)$ делается единичным), послед-

ний столбец с номером соответствует $(N+j)$ -му столбцу множества V_ρ^Z ,

$V_{\rho j}^{A2}$ включает N первых столбцов множества V_ρ^Z , последний столбец с номером $(N+1)$ единичный;

$V_{\rho j}^{A3}$ включает N первых столбцов множества V_ρ^Z , последний столбец с номером $(N+1)$ соответствует $(N+j)$ -му столбцу множества V_ρ^Z .

4. Вычисление значения коэффициентов уравнения (11) по формулам

$$A1_{\rho ji} = |V_{\rho ji}^{A1}|, \quad A2_{\rho j} = |V_{\rho j}^{A2}|, \quad A3_{\rho j} = -|V_{\rho j}^{A3}|. \quad (14)$$

Подставив выражения (14) в (12), получим конечный вид формул для расчета коэффициентов уравнения (10):

$$B_{\rho ji}^x = \frac{|V_{\rho ji}^{A1}|}{-|V_{\rho j}^{A2}|}, \quad B_{\rho j0}^x = \frac{|V_{\rho j}^{A3}|}{|V_{\rho j}^{A2}|}. \quad (15)$$

Следует отметить, что если точность полученной модели не удовлетворяет требованиям, то необходимо увеличить параметр λ на единицу и повторить алгоритм формирования термов, начиная с п. 2. Параметр λ можно увеличивать до значения N включительно.

Если ни с одним значением λ точность модели не удовлетворяет требованиям, то, возможно: между рассматриваемыми входными и выходными параметрами нет взаимосвязи; не выполняются указанные выше требования к статистическим данным; статистических данных недостаточно.

Таблица 1

Строки статистических данных

ν	$x_1(t_\nu)$	$x_2(t_\nu)$	$y_1(t_\nu)$
1	15,712	135,383	62171,5
2	15,772	143,583	62185,7
3	15,835	142,467	63714,7
4	15,842	128,867	60803,17
5	15,857	130,517	59265,7
6	15,867	138,85	63784,7
7	15,890	153,733	71616,7
8	15,906	132,25	73654,7
9	15,914	138,717	65466,7
10	16,047	135,917	63392,17
11	16,149	133,867	62385,17
12	16,172	141,317	70042,7
13	16,310	136,3	65640,5
14	16,465	145,333	65323,7
15	16,501	150,883	72295,7

Таблица 2

Аналитическая таблица при $\lambda = 1$

ν	$x_1(t_\nu)$	$x_2(t_\nu)$	$y_1(t_\nu)$	$y_1^{\text{расч}}(t_\nu)$	$\Delta y_1(t_\nu)$	$\varepsilon_1(t_\nu)$
1	15,712	135,383	62171,5	62171,5	0	0
2	15,772	143,583	62185,7	62185,7	0	0
3	15,835	142,467	63714,7	63714,7	0	0
4	15,842	128,867	60803,17	60803,17	0	0
5	15,857	130,517	59265,7	59265,7	0	0
6	15,867	138,85	63784,7	63784,7	0	0
7	15,890	153,733	71616,7	71616,7	0	0
8	15,906	132,25	73654,7	73654,7	0	0
9	15,914	138,717	65466,7	65466,7	0	0
10	16,047	135,917	63392,17	63392,17	0	0
11	16,149	133,867	62385,17	62385,17	0	0
12	16,172	141,317	70042,7	70042,7	0	0
13	16,310	136,3	65640,5	65640,5	0	0
14	16,465	145,333	65323,7	65323,7	0	0
15	16,501	150,883	72295,7	72295,7	0	0

Применение алгоритма формирования

Рассмотрим пример построения нечеткой модели Сугено для случая двух входных параметров x_1 , x_2 и одного выходного y_1 , т. е. $N = 2$ и $L = 1$. Данный пример лишь демонстрирует порядок действий, в статье не производится обработка статистических данных в соответствии с поставленными к ним требованиями. Предположим, что

между входными и выходными параметрами есть зависимость, задаваемая строками статистических данных (табл. 1).

Тогда применение предлагаемого алгоритма приведет к результатам, представленным в табл. 2.

В предлагаемом алгоритме формирование термов и базы правил происходит неразрывно друг от друга, поэтому ситуации (одновременно

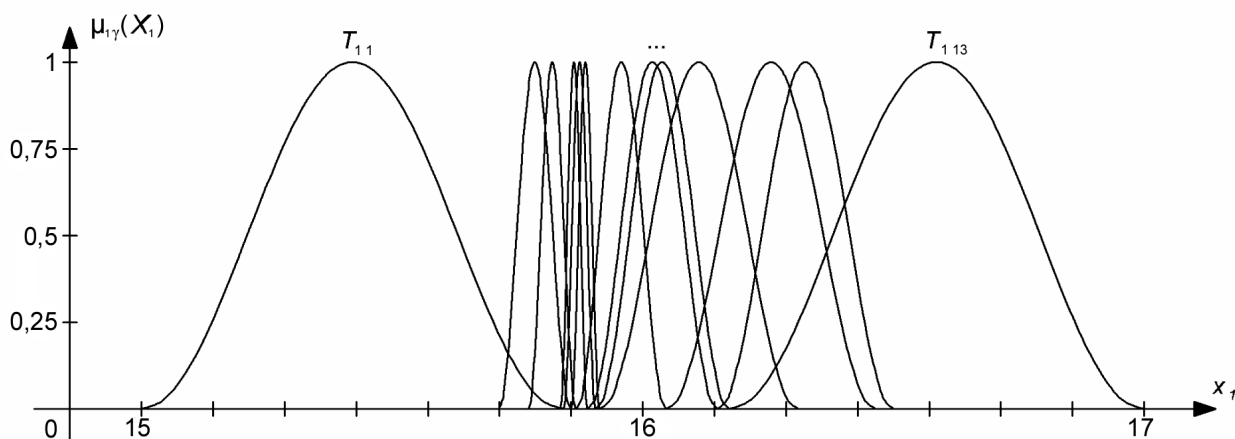


Рис. 2. Термы первого входного параметра

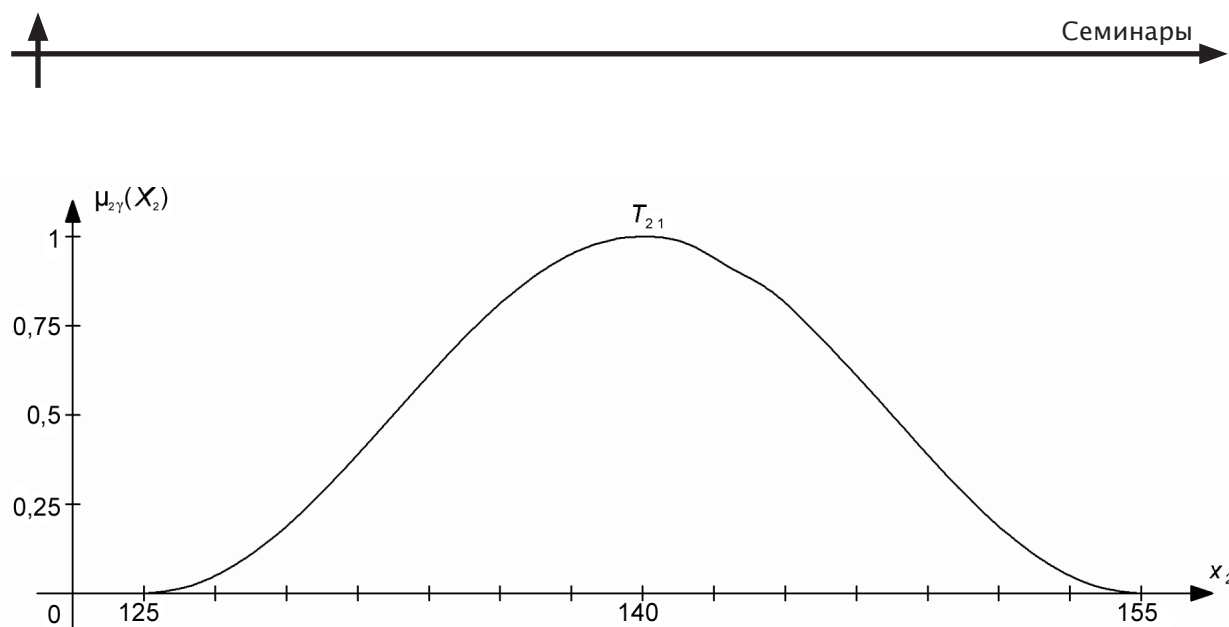


Рис. 3. Терм второго входного параметра

у двух и более термов значения функций принадлежности превышают 0,5) противоречивости не вносят, а лишь говорят об избыточности статистических данных на конкретных участках. Количество статистических данных (15 строк) недостаточно, чтобы предусмотреть все режимы любого объекта моделирования, результат – участок, где ни у одного терма значение функции принадлежности не превышает 0,5.

В табл. 1 значения абсолютной и относительной ошибки расчета указаны нулевые, т. к. реальные их значения имеют порядок 10^{-10} . Этими значениями ошибок можно пренебречь, поскольку ни один прибор контроля технологических параметров не обладает такой точностью измерения.

На рис. 2 и 3 показаны термы первого и второго входных параметров.

Рассмотренный в данной статье алгоритм автоматического формирования лингвистических переменных может использоваться при априорном формировании вектора управляющих переменных для сложных технологических процессов. Например, при управлении процессом электролиза меди. Данный алгоритм применим при достаточном количестве статистических данных, описывающих процесс, и не требует присутствия эксперта

Приводимые алгоритмы не предназначены для ручного расчета (к вопросу о трудоемкости). Предлагаемый метод создания нечетких моделей реализован в виде программы. Производительность расчетов зависит от технических характеристик ЭВМ. В частном случае, получение результатов построения модели в примере произошло мгновенно.

СПИСОК ЛИТЕРАТУРЫ

1. Леоненков, А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH [Текст]/А.В. Леоненков. –СПб.: БХВ-Петербург, 2005.–736 с.
2. Штовба, С.Д. Проектирование нечетких систем средствами MATLAB [Текст]/С.Д. Штовба. –М.: Горячая линия–Телеком, 2007.–288 с.
3. Саати, Т.Л. Взаимодействие в иерархических системах [Текст]/Т.Л. Саати//Техническая кибернетика.–1979.–№ 1.–С. 68–84.
4. Воеводин, В.В. Матрицы и вычисления [Текст]/В.В. Воеводин, Ю.А. Кузнецов.–М.:Наука,1984.–320 с.

НЕЙРО-НЕЧЕТКОЕ МОДЕЛИРОВАНИЕ СТРАТЕГИЧЕСКОГО ВЗАИМОДЕЙСТВИЯ УЧАСТНИКОВ РЫНКА ТРУДА МОНОИНДУСТРИАЛЬНОГО РЕГИОНА

В настоящее время многие регионы России сталкиваются с проблемами формирования и адекватного управления рынком труда по причине исторически сложившихся структурных диспропорций. В частности, моноиндустриальные регионы и города из-за их специфики имеют сильно ограниченные адаптационные возможности рынка труда, что требует создания механизмов его мягкого внешнего регулирования. Например, Норильский промышленный регион (НПР), где замедление темпов социально-экономического роста привело к тотальному дефициту высококвалифицированных кадров. На данный момент наблюдается значительный недостаток кадров в одних отраслях экономики региона и существенный переизбыток в других, что усугубляется невозможностью привлечения кадров из других регионов. Принципиальными отличиями Норильска в сфере кадровой политики на региональном уровне являются:

вхождение Норильска в сферу краевого подчинения и одновременно – существенная удаленность от краевых структур, территориальная замкнутость;

зависимость кадровой политики в регионе от основного работодателя – Заполярного филиала ОАО «ГМК «Норильский никель» (ЗФ) и его корпоративной политики в отношении кадровых ресурсов;

высокая инерционность и слабая эластичность рынка труда, обусловленные территориальной обособленностью региона;

слабая легитимная связь между общим средним и высшим профессиональным образованием;

особенность экстремальных климатических условий региона.

Формирование и управление региональным рынком труда в закрытом монорегионе требует создания информационно-управляющей системы, которая позволит для решения таких задач объединить усилия не только регионального центра

занятости, но и основного работодателя, местных органов власти и руководства учебных заведений региона. В данной статье рассматривается математическая модель оценки состояния регионального рынка труда на базе иерархических нейро-нечетких систем.

Формализация входных и управляемых параметров нейро-нечеткой модели

В сложившихся условиях приоритетной задачей становится подготовка в регионе квалифицированных кадров своими силами, с использованием всех ресурсов образовательных учреждений, с учетом предпочтений и запросов основного работодателя.

Реализация поставленной задачи предполагает проведение анализа демографической ситуации в регионе, системы управления кадрами градообразующего предприятия и структуры специальностей образовательных учреждений. Данные исследования были проведены в рамках НИОКР [1], где помимо всего прочего отмечено, что ни минимум уровня безработицы, ни максимум уровня занятости населения не могут рассматриваться в качестве критериев оптимальности функционирования рынка труда. Однако слишком высокий уровень безработицы и, наоборот, низкий уровень занятости, чем бы это не было вызвано, однозначно являются негативными показателями состояния рынка труда как социальной системы. Основным критерием оптимальности ($f(x_i)$) для рынка труда предлагается принять минимум совокупного отклонения между спросом предприятия и предложением квалифицированных кадров от учебных заведений по всем направлениям подготовки. Так как численность населения НПР имеет стойкую тенденцию к снижению, а критерий эффективности рынка труда следует определять только в динамике, расчет целевой функции в абсолютных величинах не покажет адекватных значений. Поэтому предлагается данный показатель относить к общей численности

населения региона для получения относительных величин:

$$f(x_i) = \min_i \frac{\sum_{i=1}^N (ЧЗН_i^i - x_i^i)}{ЧН_i}, \quad (1)$$

где $i = \overline{1, N}$ – номер профессиональной группы в общей структуре специальностей и профессий, требуемых на рынке труда региона; $ЧЗН_i^i$ – численность работников по группе i в расчетном году t , чел.; $ЧН_i$ – численность населения региона в расчетном году t , чел.; x_i^i – предложение квалифицированных кадров от учебных заведений по группе подготовки i (количество выпускников) в расчетном году t , чел.

Ограничение на параметры целевой функции (1), отражающее демографические показатели населения, выражается в том, что количество выпускников учебных заведений региона не может быть больше количества абитуриентов региона в году их поступления в вуз (ссуз) (ЧА):

$$\sum_{i=1}^N x_i^i \leq ЧА_{t-5}.$$

Кроме того необходимо учитывать ограничения по прогнозным значениям структуры персонала предприятия ($ЧЗН_{пр t}^i$):

$$x_{t-k}^i \geq ЧЗН_{пр t}^i, \quad k = 1, \dots, 5, \quad i = 1, \dots, N,$$

т. е. численность учащихся в определенном году по каждой специальности не должна быть меньше, чем прогнозная численность персонала предприятия в расчетном году.

Для эффективного функционирования рынка труда необходимо таким образом управлять показателем x^i , чтобы значение $f(x_i)$ было минимальным. Главным при этом является фактор времени t , выражающийся в том, что процесс обучения студентов занимает несколько лет, поэтому при наборе на первый курс необходимо прогнозировать будущую структуру спроса на кадры по основным направлениям подготовки, а за два года до окончания обучения – по специализациям подготовки.

Процесс определения оптимальной структуры набора в учебные заведения региона укрупненно представлен в виде IDEF0 модели на рис. 1. Для реализации поставленной цели необходимо решить три основные задачи: спрогнозировать численность и состав населения региона, численность и структуру кадров на предприятии, рассчитать оптимальные показатели набора в учебные заведения и разработать систему гибкого регулирования параметров обучения.

В настоящее время в демографических процессах территории необходимо отметить следующие тенденции: высокий миграционный отток населения, рост рождаемости на фоне снижения смертности, постоянное увеличение доли населения пенсионного возраста.

Высокая миграционная активность населения указывает на снижение привлекательности территории как места проживания. Огромное влияние

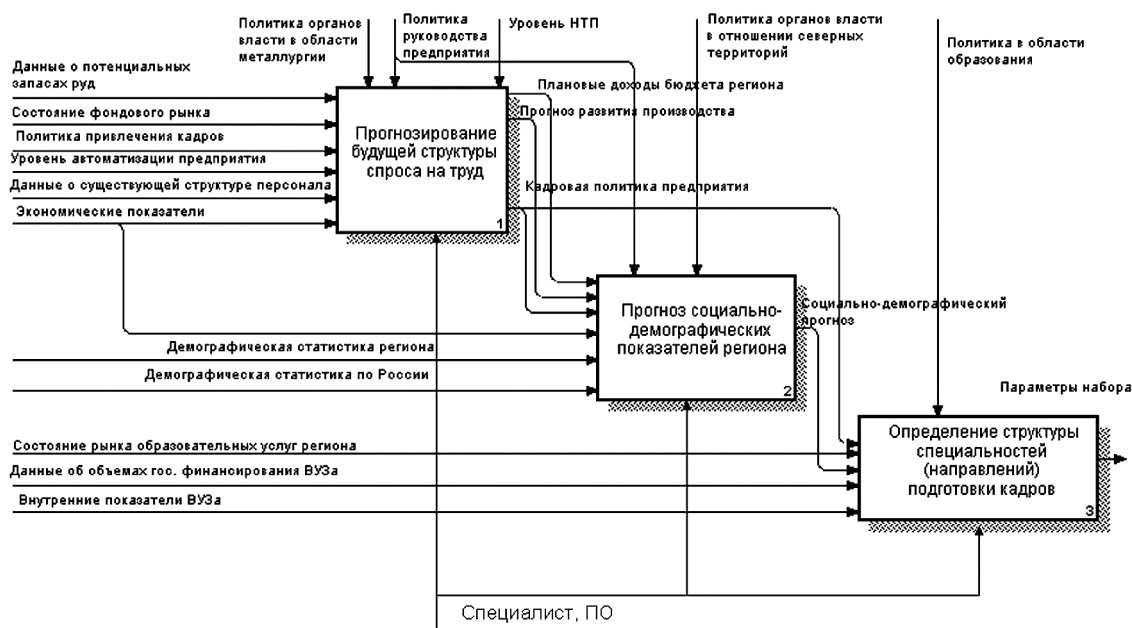


Рис. 1. Определение оптимальной структуры подготовки кадров

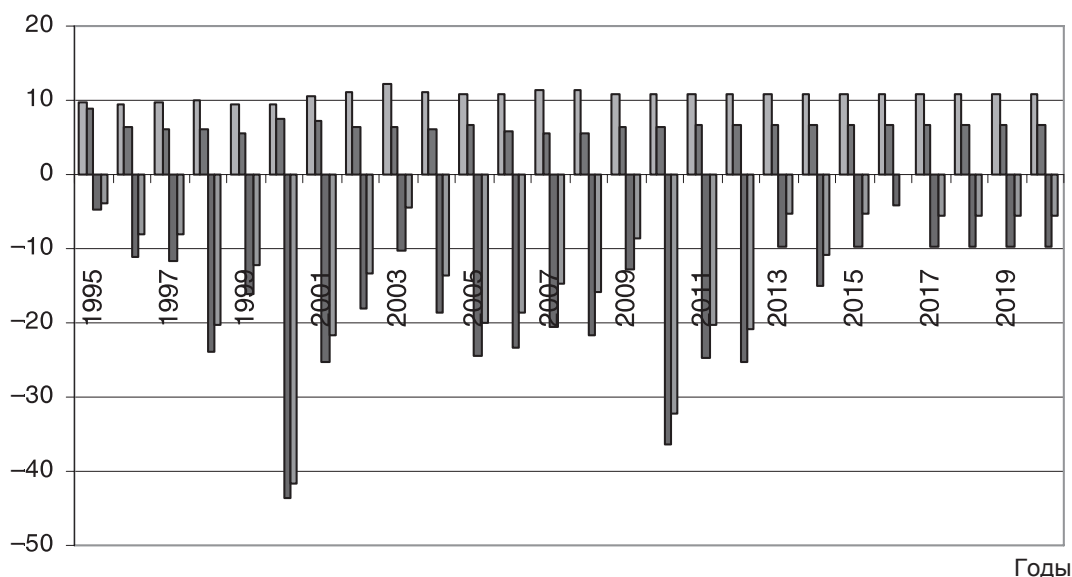


Рис. 2. Прогноз значений демографических показателей Норильска
() коэффициент рождаемости; () коэффициент смертности;
() коэффициент миграции; () коэффициент прироста

на демографический процесс оказывают суровые климатические условия, неблагоприятная экологическая обстановка и рост уровня жизни населения в других регионах России. В настоящее время значительный отток населения из НПР пока еще не оказывает существенного влияния на экономику и рынок труда, но уже в ближайшем будущем территория может испытать катастрофический дефицит трудовых ресурсов.

Для оценки показателей естественного прироста населения использовался метод «передвижки возрастов» [2]. В результате определен прогноз (рис. 2) естественного движения населения Норильска без учета миграционного оттока. Естественный прирост населения положительный, что подтверждается соотношением рождаемости и смертности в Норильске – одном из немногих регионов России, где рождаемость значительно превышает смертность. Тем не менее, численность населения города несомненно будет снижаться. Основной причиной этого является отрицательный миграционный поток, который так же необходимо спрогнозировать.

Изучая уровень миграции, в первую очередь, необходимо учитывать специфику региона. Можно выделить следующие факторы:

- динамика развития производства на территории;
- динамика численности персонала градообразующего предприятия;

уровень жизни в регионе в сравнении с другими регионами, более комфортными для проживания.

Для оценки показателей миграции проведен статистический анализ взаимосвязей демографических показателей НПР с показателями ЗФ на основе перекрестных данных [1]. При этом аппаратом исследования является корреляционно-регрессионный анализ. Для построения модели в каждом году рассматривались значения четырех абсолютных показателей: численность населения региона (ЧН); численность населения, занятого в ЗФ (ЧЗН); среднедушевой доход в месяц (СДД); объем производства руды, тыс. т (ПР). Построенная матрица парных коэффициентов корреляции выявила сильную связь между двумя парами показателей:

- ЧН и ЧЗН – коэффициент парной корреляции 0,932.
- СДД и ПР – коэффициент парной корреляции 0,952.

Полученные значения подтверждают гипотезу о том, что численность населения Норильска очень зависит от кадрового состава градообразующего предприятия. Учитывая стратегию развития производства ЗФ, до 2020 г. кардинальных изменений не произойдет, и миграционный поток останется отрицательным (рис. 2).



Из проведенного анализа деятельности ЗФ в области обеспечения кадрами можно сделать вывод, что его руководство уделяет очень большое внимание процессу формирования кадрового потенциала. Это объясняется высокими требованиями к квалификации работников. Кроме того, данная политика вызвана растущим дефицитом квалифицированных кадров в регионе расположения основных предприятий ЗФ.

Формирование кадрового резерва предполагает составление планов по набору кадров, опирающихся на прогнозирование кадрового состава предприятия в средне- и долгосрочной перспективе. Использование таких планов позволит учебным заведениям корректировать структуру подготовки по различным специальностям и направлениям и формировать такую структуру выпускников, которые будут востребованы на рынке труда региона с минимальным периодом адаптации.

Прогнозирование будущей структуры спроса ЗФ на трудовые ресурсы состоит из двух направлений:

1. С помощью описанного ранее статистического метода «Передвижки возрастов» определяется движение работников с учетом их половозрастной структуры. Для реализации этой задачи описывается текущая структура работников в разрезе их возраста, профессии и пола. Затем, учитывая возраст выхода на пенсию, определяется численность вакантных мест по каждой профессии в каждый год прогноза. Пополнение кадрового состава предприятия осуществляется из резерва и учитывается в прогнозе, как в демографической модели учитывалась рождаемость.

2. На структуру кадров предприятия оказывает большое влияние развитие производства, внешнеэкономические показатели и другие факторы, учесть которые с помощью статистических

Факторы нейро-нечеткой модели

Группа	Лингвистические переменные
Состояние внешней среды ($L_{внеш}^{вх}$)	Состояние фондового рынка ($L_1^{вх}$)
	Экономические показатели ($L_2^{вх}$)
	Потенциальные запасы руд ($L_3^{вх}$)
	Политика органов власти в области металлургии ($L_4^{вх}$)
	Политика руководства предприятия ($L_5^{вх}$)
	Естественные демографические показатели региона ($L_9^{вх}$)
	Политика в области образования ($L_{10}^{вх}$)
Внутреннее состояние вуза ($L_{внутр}^{вх}$)	Состояние учебной деятельности ($L_6^{вх}$)
	Состояние научной работы ($L_7^{вх}$)
	Техническое и программное оснащения учебного процесса ($L_8^{вх}$)
Показатели I уровня (L^I)	Развитие производства (L_1^I)
	Статус вуза (L_2^I)
Показатели II уровня (L^{II})	Кадровая политика предприятия (L_1^{II})
	Социальная политика предприятия (L_2^{II})
Показатели III уровня (L^{III})	Численность работников предприятия (L_1^{III})
	Структура кадров высшей квалификации предприятия (L_2^{III})
	Механическая миграция населения региона (L_3^{III})
	Внебюджетное финансирование вуза (L_4^{III})
	Бюджетное финансирование вуза (L_5^{III})
Показатели IV уровня (L^{IV})	Численность населения региона (L_1^{IV})
	Половозрастная структура населения региона (L_2^{IV})
	Возможности вуза по приему студентов (L_3^{IV})
Показатели V уровня (L^V)	Численность абитуриентов вуза (L_1^V)

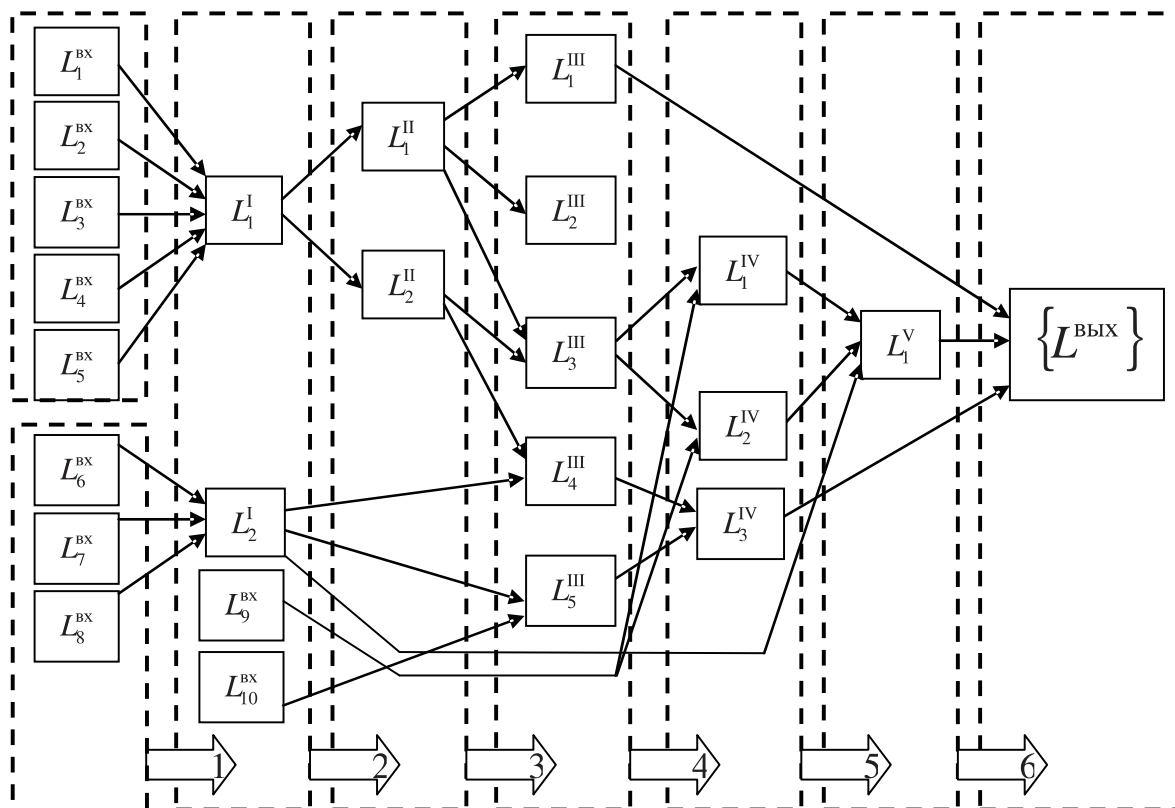


Рис. 3. Схема связей нейро-нечеткой модели

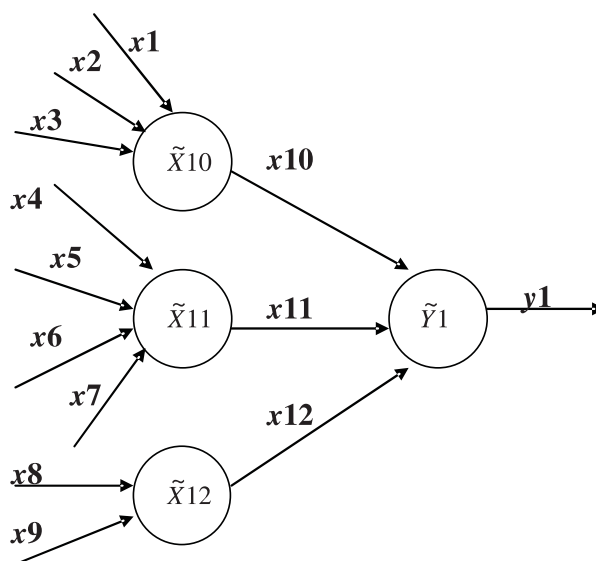


Рис 4. Структура иерархической системы нечеткого вывода

x_1 – индекс РТС; x_2 – индекс ММВБ; x_3 – цены на нефть (Brent); x_4 – объем ВВП; x_5 – уровень доходов населения;
 x_6 – инвестиции в основной капитал; x_7 – уровень инфляции; x_8 – уровень безработицы;
 x_9 – коэффициент естественного прироста

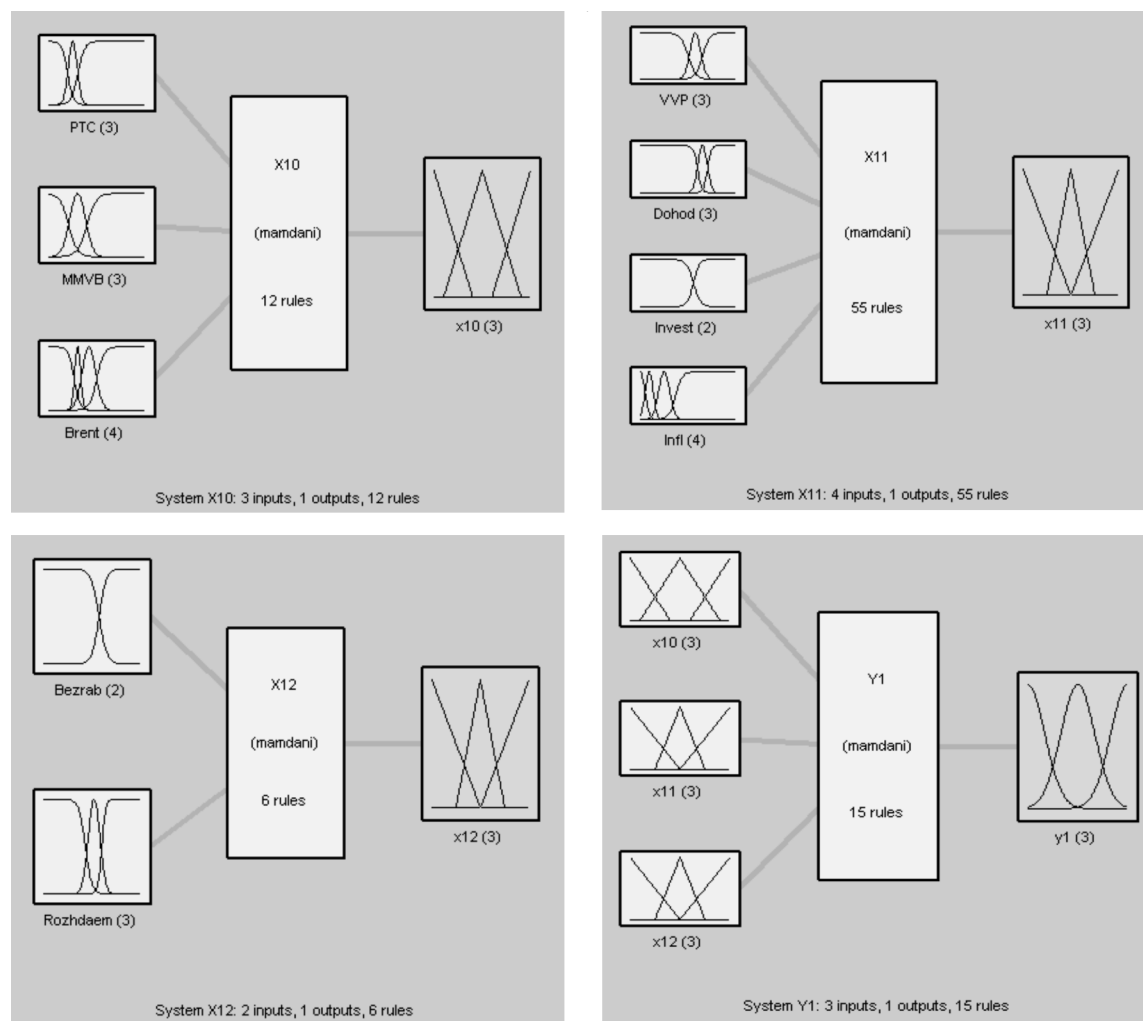


Рис. 5. Структура систем нечеткого вывода

методов невозможно из-за дефицита информации и неточности входных данных, а также из-за качественного характера некоторых параметров. Задача прогнозирования может быть выполнена средствами нейро-нечеткого моделирования.

Архитектура нейро-нечеткой сети

Описываемая нейро-нечеткая система, состав параметров которой (см. табл.) был подвергнут экспертной оценке и ранжирован, представлена в виде каскада нечетких систем (рис. 3).

Рассмотрим создание модели на базе иерархической нейро-нечеткой системы (рис. 4) оценки состояния внешней среды ($\tilde{Y}1$): состояние фондового рынка ($\tilde{X}10$), основные внешнеэкономические показатели ($\tilde{X}11$), состояние рынка труда региона ($\tilde{X}12$).

На основе экспертных оценок построены системы нечеткого вывода типа Мамдани (рис. 5). Функции принадлежности термов входных переменных $x1$ – $x9$ сигмоидного и гауссовского типов, промежуточных переменных $x10$ – $x12$ треугольного типа, выходной переменной $y1$ гауссовского типа.

Параметры термов выходной переменной $y1$ получены по результатам нечеткой кластеризации алгоритмом с-средних. В результате определены три возможных состояния внешней среды: «благоприятное», «среднее», «неблагоприятное» [3].

В структуре нейро-нечеткой сети выделены пять слоев: термы входных переменных, antecedentes нечетких правил, нормализация степеней выполнения правил, заключения правил, агрегирование результата.

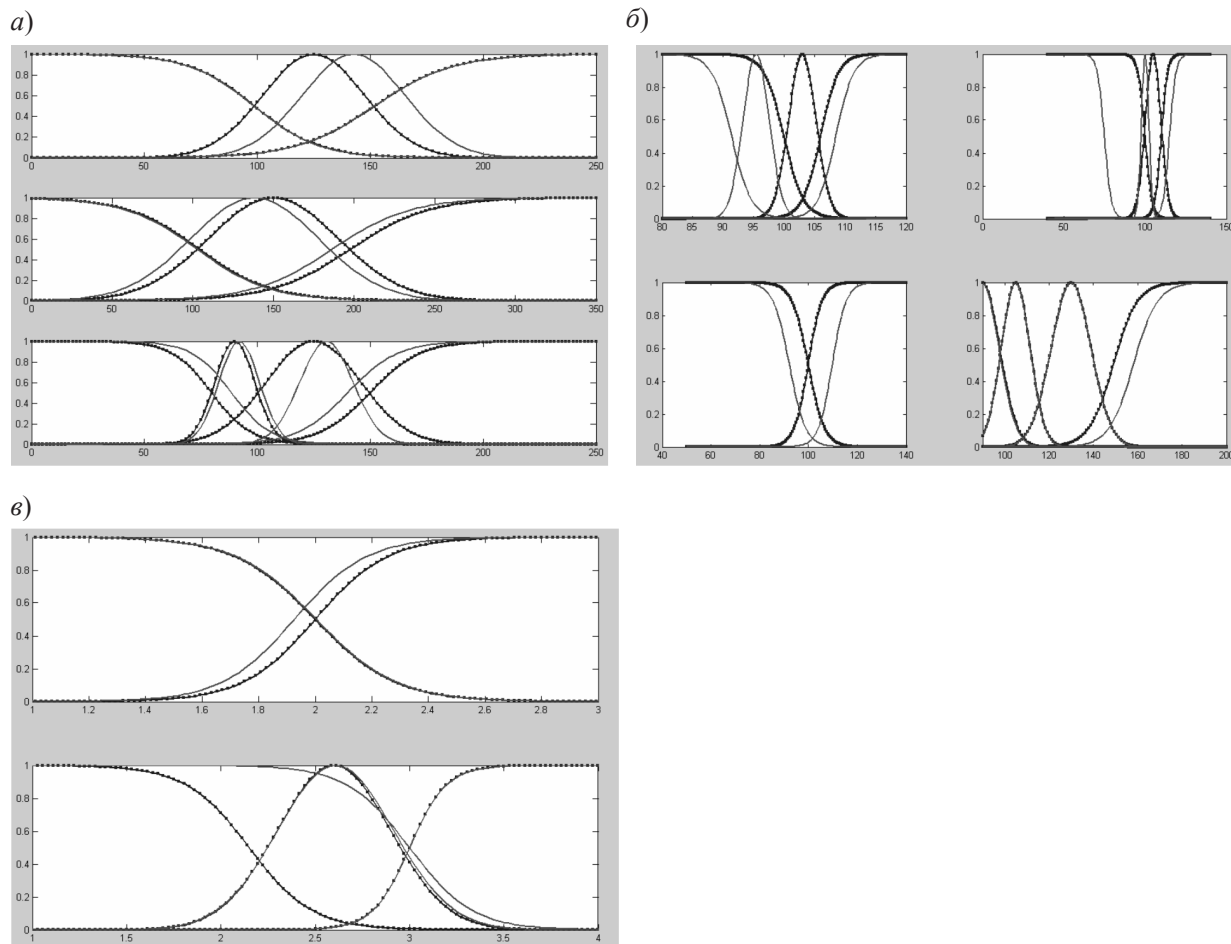


Рис. 6. Функции принадлежности сети до и после обучения:
а – система $\tilde{X}10$; б – система $\tilde{X}11$; в – система $\tilde{X}12$

Для обучения иерархической сети использовался модифицированный алгоритм [4, 5], адаптированный для систем нечеткого вывода типа Мамдами. Параметры сети, отобранные для настройки в процессе обучения, следующие: коэффициенты концентраций функций принадлежности входных и выходной переменных (P_k); координаты максимума функций принадлежности входных и выходной переменных (P_m); веса правил всех нечетких систем (W).

Обучение сети сводится к решению следующей задачи: найти такой вектор (P_k, P_m, W), чтобы

$$RMSE = \sqrt{\frac{1}{M} \cdot \sum_{r=1, M} (y_r - F(P_k, P_m, W, X_r))^2} \rightarrow \min,$$

где X_r – входной вектор в r -й паре обучающей выборки; y_r – соответствующий выход; $F(P_k, P_m, W, X_r)$ – результат вывода по иерархической системе.

На рис. 6 представлены функции принадлежности систем нечеткого вывода до обучения (жирная линия) и после обучения (светлая линия).

Результат обучения сети представлен на рис. 7. После обучения сети значение ошибки на тестовых данных, объем выборки которых – 70 значений, сократилось с 42 до 12 %.

Построение иерархической нейро-нечеткой сети всей системы (рис. 3) позволит оценить ее состояние и сделать качественный прогноз параметров рынка труда региона.

Реализация информационно-управляющей системы на основе описанной модели позволит:

- предприятию снижать затраты на формирование резервов и привлечение персонала; уменьшать кадровые риски; получать молодые квалифицированные кадры, подготовка которых проведена с учетом дальнейшей работы на пред-

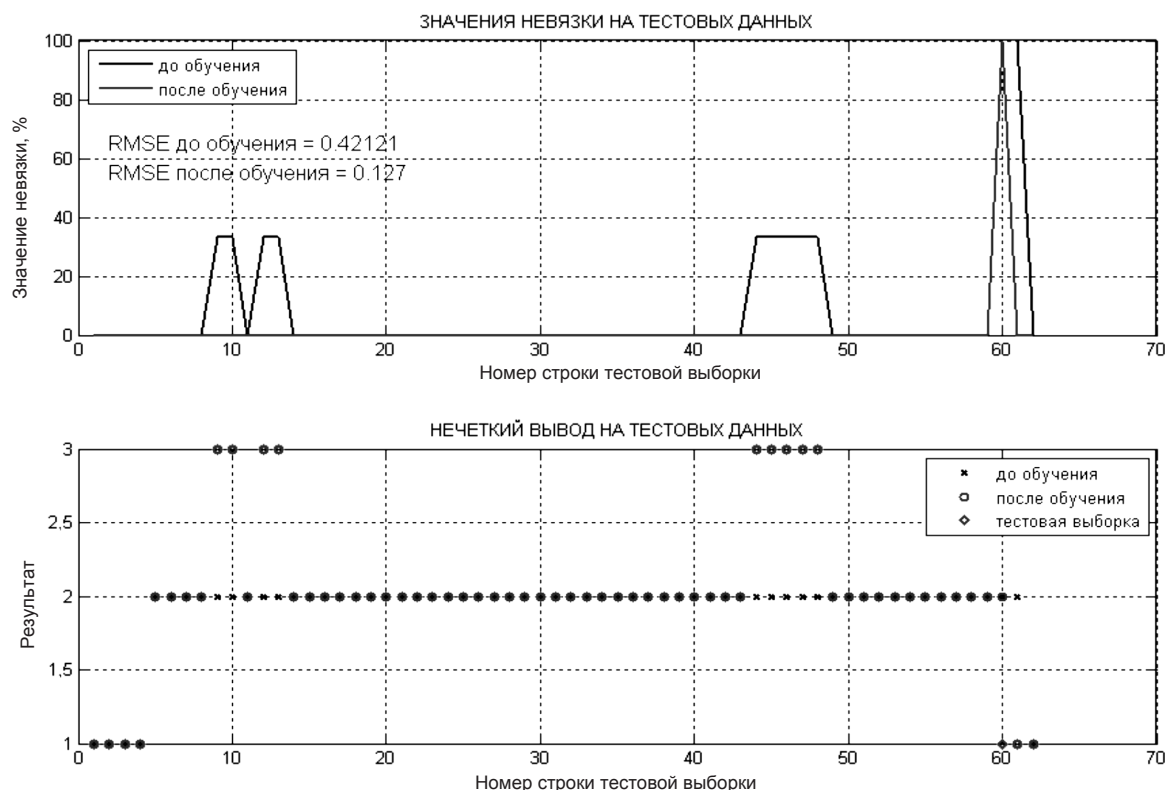


Рис. 7. Результаты обучения нейро-нечеткой сети

приятии; влиять на учебные планы, корректируя направления и специфику подготовки студентов.

- учебным заведениям увеличивать показатели трудоустройства выпускников, тем самым повышая свой статус; получать во временное или постоянное пользование современные программные продукты, используемые на предприятии; улучшать свою материальную базу; проводить консультации работников ЗФ; организовывать производственную практику на предприятиях.
- студентам и выпускникам заключать до-

говоры по оплате обучения ЗФ; выбирать свою будущую специальность в соответствии с актуальными требованиями экономики региона; знакомиться с новейшими производственными технологиями, оборудованием и информационными технологиями во время учебы; трудоустроиться.

- администрации региона уменьшить отток молодых перспективных кадров; снизить уровень безработицы; установить более тесные связи с ЗФ; снизить объемы финансовой поддержки учебных заведений из муниципального бюджета.

СПИСОК ЛИТЕРАТУРЫ

1. Разработка основных принципов стратегического управления образовательным пространством НПП и сопряженных с ними инженерных и экономических проблем развития территориального производственного комплекса: отчет о НИР [Текст] / Норильский индустриальный институт; рук. Фомичева С.Г.; исп. Мыльников В.А., Елина Т.Н. и др. – 220 с., № 020302329 03 03, № рег. 01200900213, инв. 02201000236.
2. Басалаева, Н.А. Моделирование демографических процессов и трудовых ресурсов [Текст] / Н.А. Басалаева. – М.: Наука, 1978. – 230 с.
3. Елина, Т.Н. Оценка состояния внешней среды

- металлургического предприятия методом нечеткой кластеризации [Текст] / Т.Н. Елина. – Межд. конф. «Информационная среда вуза»: сб. науч. тр. – Иваново, ИГАСУ, 2009. – с. 23–28.
4. Штовба, С.Д. Проектирование нечетких систем средствами MATLAB [Текст] / С.Д. Штовба. – М.: Горячая линия – Телеком, 2007. – 288 с.
5. Фомичева, С.Г. Система управления временем актуальности защищаемой информации [Текст] / С.Г. Фомичева // Научно-технические ведомости СПбГТУ. Сер. Информатика, телекоммуникации, управление. – 2009. – № 1 (72), – С. 77–87.

УДК 330.43

А.А. Попкова, С.Г. Фомичева

МЕТОДЫ АПРИОРНОЙ ОЦЕНКИ СТЕПЕНИ АГРЕГАЦИИ И ПРОИЗВОДИТЕЛЬНОСТИ ЗАПРОСОВ В ХРАНИЛИЩАХ ДАННЫХ

Многие предприятия и организации при управлении своими бизнес-процессами интенсивно используют системы поддержки принятия решений (СППР) на базе OLAP-технологий. Для СППР требуются хронологические данные: факты за определенные интервалы времени. При этом OLAP-системы (*On-Line Analytical Processing*) построены на двух базовых принципах: данные, необходимые для принятия решений, предварительно агрегированы на всех соответствующих уровнях и организованы так, чтобы обеспечить максимально быстрый доступ к ним; язык манипулирования данными основан на использовании бизнес-понятий.

Фундаментом любой OLAP-системы является единое корпоративное хранилище данных (ХД), разработка которого является долгим и дорогостоящим процессом. При проектировании структуры хранилища или витрин данных часто возникает желание использовать как можно больше агрегатов и за счет этого повысить производительность системы. Другая крайность состоит в использовании слишком малого числа агрегатов, а это может привести к необходимости выполнять агрегирование динамически, что заметно снижает эффективность запросов.

В данной статье предлагается подход к априорной оценке степени агрегации данных на уровне гиперкуба и методика увеличения скорости выполнения агрегатных запросов с низкой и средней избирательностью за счет уменьшения вычислительных затрат, связанных с агрегированием обобщенных данных.

Обоснование необходимости априорной оценки степени агрегации данных

Корпоративное хранилище данных, как правило, формируется постепенно, выстраиваясь в структуру гиперкуба как совокупность взаимосвязанных витрин данных. Например, на рис. 1 показано, что каждый процесс, связанный с деятельностью системы качества высшего образовательного учреждения, оперирует несколькими витринами данных. Под Витриной Данных (ВД) здесь понимается специализированное хранилище, обслуживающее одно из направлений деятельности вуза.

Часть данных одной витрины может включать агрегированные значения других витрин, затрагивая результаты аналитической деятельности других подразделений вуза. Несмотря на то, что практически невозможно априорно предсказать,

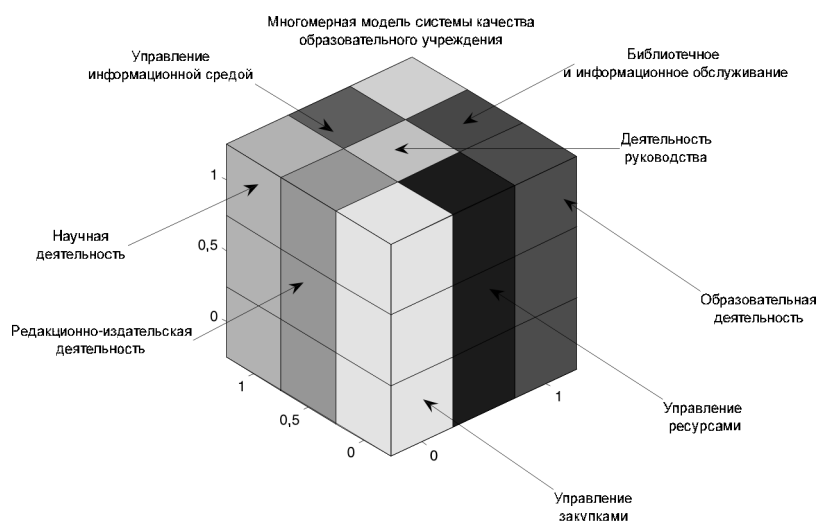


Рис. 1. Единое хранилище (Data Warehouse) системы качества вуза



какую именно информацию и в каком виде захочет получить пользователь, работая с СППР, измерения, по которым проводится анализ, достаточно стабильны, что позволяет на этапе проектирования ХД выделить плановые аналитические запросы. Помимо этого, существуют и незапланированные формы отчетов на основе запросов с низкой избирательностью, на формирование которых уходит намного больше времени. Отметим, что оптимизация соответствующих группирующих запросов OLTP-систем к существенному улучшению производительности не приводит. На рис. 2 приведена сравнительная характеристика времени выполнения плановых запросов для формирования сводных отчетов с использованием оптимизированных OLTP-решений и OLAP-технологий. По времени выполнения запросов с различной избирательностью, наблюдается значительная экономия машинного времени при использовании специализированных аналитических инструментов на основе OLAP-технологий.

Одним из основных свойств систем оперативного анализа данных является способность формирования агрегатных значений показателей с целью минимизации времени отклика при построении динамических отчетов. При этом необходимо разрешить дилемму между производительностью и объемом, занимаемым агрегатными значениями. Большое количество измерений и уровней агрегации внутри них приводят к эффекту, характеризующемуся чудовищным ростом объема дискового пространства, необходимого для хранения агрегатных значений показателей. Заблаговременное формирование и сохранение агрегатов обеспечивает уменьшение времени отклика на пользовательский запрос и является основным свойством систем поддержки оперативного анализа. Данные вопросы уже затрагивались в работах таких авторов, как Е.В. Хрусталева [1], М.С. Заботнев [2], но не приводилась оценка степени агрегации данных на уровне гиперкуба, рассматриваемая в данной статье.

Оценка степени агрегации данных в гиперкубе

Введем ряд обозначений и решим задачу априорной оценки степени агрегации данных гиперкуба, первоначально определяемой следующим выражением:

$$\alpha = \frac{a}{a^*}, \quad (1)$$

где $a = \sum_{i=1}^{C_n^t} \left(\prod_{i=1}^{n-i} k_i \right)$ – реальное количество агрегированных значений показателей (мер); a^* – максимально возможное количество агрегатных значений исходных данных гиперкуба.

Пусть $D = \{d_1, d_2, \dots, d_n\}$ – множество измерений гиперкуба, где n – количество упорядоченных измерений гиперкуба, включая иерархические измерения. Зададим множество меток i -го измерения – $M_{d_i} = \{m_1, m_2, \dots, m_{k_i}\}$ и $i = 1, n$, где k_i – количество различных кортежей (меток) i -го измерения.

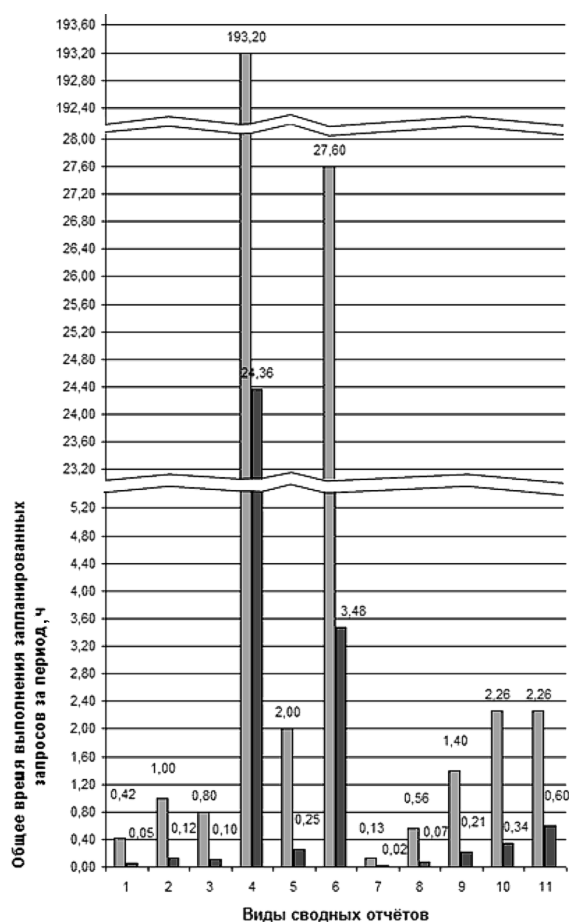


Рис. 2. Сравнительные характеристики времени выполнения плановых запросов для формирования сводных отчетов за период, ч (■) OLTP-запросы; (■) OLAP-запросы

Выражение для расчета полного числа агрегатов может быть представлено как сумма всевозможных неповторяющихся произведений количества меток по всем измерениям гиперкуба с последующим исключением исходных данных:

$$a^* = \sum_{i=1}^n \left(\sum_{j=1}^{C_n^i} \prod_{i=1}^{n-i} k_i \right) = \prod_{i=1}^n (k_i + 1) - \prod_{i=1}^n k_i. \quad (2)$$

Для случая, когда для i -го измерения существует l_i^* уровней иерархии, введем k_{ij} – количество меток в j -м уровне i -го измерения. Количество агрегатов для иерархического измерения определяется как сумма меток по всем уровням иерархии, т. е.: $k_i = \sum_{j=1}^{l_i^*} k_{ij}$. Следовательно, выражение (2) в случае определения общего количества агрегатов гиперкуба с иерархическими измерениями, исключая множество исходных данных, относящихся к первому уровню иерархии, примет вид:

$$a^* = \prod_{i=1}^n \left(\sum_{j=1}^{l_i^*} k_{ij} + 1 \right) - \prod_{i=1}^n k_{i1}. \quad (3)$$

Отметим, что для поддержки механизма адресации к элементам многомерного массива типа MOLAP, в нем должны храниться все ячейки куба, в т. ч. и пустые. Это приводит к тому, что целиком заполненные данными и сколь угодно разреженные кубы занимают одинаковое дисковое пространство. ROLAP-хранилища не содержат «пустых» значений, т. к. записи в таблице фактов существуют только для заполненных ячеек куба. Но следует учитывать, что на каждое значение меры приходится несколько значений координат измерений, образующих внешний ключ в таблице фактов. Иначе говоря, каждая ячейка куба при использовании ROLAP-хранилища требует больше пространства памяти по сравнению с MOLAP [5].

Исходя из сказанного, можно предположить, что при высоких значениях степени агрегации данных куба многомерные хранилища требуют меньше внешней памяти, чем реляционные, а при низких – наоборот. Следовательно, существует некоторое пороговое значение степени агрегации α_{kr} , при превышении которого объем данных, занимаемый гиперкубом в случае его MOLAP-реализации, становится меньше по сравнению с ROLAP-реализацией. Выражение для расчета α_{kr} приведено в [5].

Найдем зависимость объема данных хранилища от плотности его заполнения. Для этого введем в рассмотрение следующие обозначения. Пусть s – размер одной ячейки куба в байтах; s_k – размер ключевого атрибута в таблице фактов в байтах.

Тогда объем данных S_m для MOLAP-хранилища может быть вычислен по формуле:

$$S_m = V \cdot s. \quad (4)$$

Аналогичная формула для объема ROLAP-хранилища S_r с учетом хранения ключевых атрибутов в таблице факта имеет вид:

$$S_r = v(s + s_k \cdot n) = \alpha \cdot V(s + s_k \cdot n). \quad (5)$$

Для нахождения пороговой степени агрегации приравняем правые части выражений (4) и (5). В результате получим:

$$\alpha_{kr} = \frac{s}{s + s_k \cdot n}. \quad (6)$$

Экспериментальным путем показано [6], что при $\alpha > \alpha_{kr}$ MOLAP-реализация данного куба будет более эффективной с точки зрения занимаемого пространства внешней памяти, чем ROLAP.

Для рассматриваемого примера, в AS-кубе информационно-библиотечного фонда (ИБФ) вуза (рис. 3) присутствуют следующие измерения: заглавие книги (d_1), дата издания (d_2), тип литературы (d_3), издательство (d_4), город издания (d_5), автор документа (d_6), документ (d_7), форма обучения (d_{10}), курс обучения (d_{11}), семестр обучения (d_{12}) [3]. Указанным измерениям для частного случая Норильского индустриального института соответствуют следующие размерности множеств меток измерений: $|M_{d_1}| = k_1 = 18656$, $|M_{d_2}| = k_2 = 92$, $|M_{d_3}| = k_3 = 21$, $|M_{d_4}| = k_4 = 1675$, $|M_{d_5}| = k_5 = 202$, $|M_{d_6}| = k_6 = 11752$, $|M_{d_7}| = k_7 = 26000$, $|M_{d_{10}}| = k_{10} = 5$, $|M_{d_{11}}| = k_{11} = 6$, $|M_{d_{12}}| = k_{12} = 2$.

В рассматриваемом кубе присутствуют измерения со вложенными иерархиями, к ним относятся: *учебный план* (d_8): цикл дисциплин (d_{82}) – дисциплина (d_{81}); *структурная единица вуза* (d_9): факультет (d_{94}) – кафедра (d_{93}) – специальность (d_{92}) – группа (d_{91}). При этом, количество уровней иерархии измерения *учебный план* (d_8) равно двум, т. е. $l_8^* = 2$, а количество уровней иерархии измерения *структурная единица вуза* (d_9) равно четырем, т. е. $l_9^* = 4$. Общее число меток измерения *учебный план* (d_8) и *структурная единица вуза* (d_9) равно, соответственно, $|M_{d_7}| = k_{81} + k_{82} = 678 + 6 = k_8 = 684$ и $|M_{d_9}| = k_{91} + k_{92} = k_{93} = k_{94} = k_9 = 410$.

Количество факультетов, кафедр, специальностей и групп в вузе равно, соответственно, $k_{94} = 4$, $k_{93} = 17$, $k_{92} = 21$ и $k_{91} = 368$. Отметим, что в си-

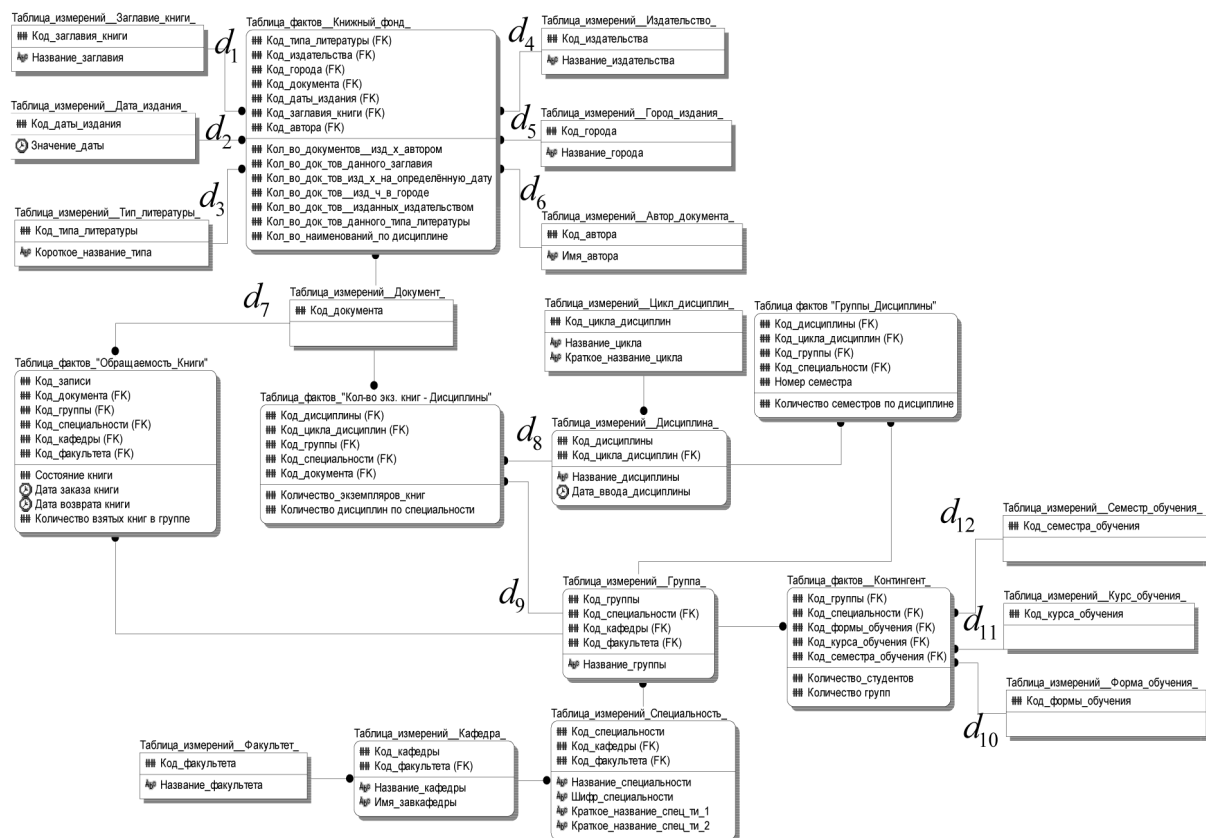


Рис. 3. Логическая структура витрины данных «ИБФ вуза»

стеме оперативного управления ИБФ вуза выполняется многомерный анализ не только по общей структуре гиперкуба Φ , но и по отдельным AS-кубам: «Книжный фонд» Φ_1 , «Количество экземпляров книг – Дисциплины» Φ_2 , «Группы – Дисциплины» Φ_3 , «Контингент» Φ_4 и «Обращаемость книги» Φ_5 . Поэтому после вычисления степеней

агрегации $\alpha_1 \div \alpha_5$ по отдельным AS-кубам $\Phi_1 - \Phi_5$ определяется степень агрегации α по совокупному по гиперкубу Φ :

$$\alpha_1 = \frac{8,4 \cdot 10^8}{2,4 \cdot 10^{20}} = 3,5 \cdot 10^{-12}; \quad \alpha_2 = \frac{13 \cdot 10^6}{8 \cdot 10^8} = 2 \cdot 10^{-2};$$

$$\alpha_3 = \frac{684}{32031} = 2 \cdot 10^{-2}; \quad \alpha_4 = \frac{5740}{29706} = 0,18 \approx 0,2;$$

Значения степени агрегации данных гиперкуба

Название куба	Количество измерений куба l	Размер одной ячейки куба s , байт	Размер ключевого атрибута в таблице факта Sk , байт	Степень агрегации куба α	Общее количество ячеек куба V	Объем данных для режима MOLAP Sm	Объем данных для режима ROLAP Sr	Пороговая степень агрегации куба α_{kr}	Рекомендуемый тип OLAP
Куб (Ф1)	7	4	4	3,50E-12	3,73E+21	1,49E+22	4,17345E+11	0,13	HOLAP
Куб (Ф2)	3	4	4	2,00E-02	7291440000	2,92E+10	2333260800	0,25	HOLAP
Куб (Ф3)	2	4	4	2,00E-02	280440	1,12E+06	67305,6	0,33	HOLAP
Куб (Ф4)	4	4	4	1,80E-01	24600	9,84E+04	88560	0,20	HOLAP
Куб (Ф5)	2	4	4	9,90E-01	10660000	4,26E+07	126640800	0,33	MOLAP
Гиперкуб (Ф)	12	4	4	5,10E-16	6,27001E+28	2,51E+29	1,66281E+15	0,25	HOLAP

$$\alpha_5 = \frac{10660000}{10686411} = 0,99 \approx 1,$$

$$\alpha = \frac{\sum_{i=1}^n \alpha_i}{n} = \frac{8,2 \cdot 10^{12}}{16 \cdot 10^{27}} \approx 5,1 \cdot 10^{-16}.$$

В таблице показаны рассчитанные реальные значения степени агрегации и пороговой степени агрегации данных в кубах ($\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5$) и в гиперкубе (Φ) рассматриваемого примера, что позволяет обосновать выбор типа OLAP-режима с учетом того, что детальные данные хранятся в OLTP-базах данных.

Методика оптимизации процедур формирования агрегатов

Методика оптимизации процедур формирования агрегатов основана на представлении гиперкуба в виде ориентированного графа $G(V, E(V))$, где $V = \bigcup_{i=1}^m G_i$ – множество узлов, каждый из которых G_i в свою очередь также является многомерным кубом; $E(V)$ – соответствующее упорядоченное множество направленных дуг $E(V) = \bigcup_{i=1}^m E(G_i)$. Для оптимизации запросов следует последовательно решить задачи локальной оптимизации в виде определения минимальных критических путей в графах G_i , а затем найти «глобальные» минимальные критические пути в графе $G(V, E(V))$.

Последовательность действий для оптимизации запросов может быть описана следующим алгоритмом.

1. Для i -го куба ($i = \overline{1, m}$) по каждому аналитическому измерению j ($j = \overline{1, n_i}$) следует построить ориентированный граф $G_i^{(j)}$, в котором множество агрегатов ассоциируется с вершинами ориентированного графа определенного уровня агрегации. Начальной вершиной такого графа является узел множества исходных данных, а конечной – узел полного агрегата (All). Ребра графа представляют собой аналог процедуры агрегации элементов определенного уровня и имеют соответствующий вес (ρ_{ij}).

2. Сформировать граф $G_i = \triangleright \triangleleft_{j=1}^{n_i} G_i^{(j)}$, где $\triangleright \triangleleft$ – означает операцию соединения узлов подграфов $G_i^{(j)}$, которые представляют один и тот же факт и соответствуют элементам одного и того же уровня агрегации каждого из измерений j ($j = \overline{1, n_i}$), использующихся для анализа факта.

3. Обозначим Π_i – упорядоченное множество всех маршрутов p_i для всех узлов графа G_i ; ρ_{il} – вес дуги $l \in E(G_i)$, определяемый как количество элементарных операций для выполнения процедур агрегации между связанными узлами графа G_i . Поскольку каждый граф G_i ациклический и является связанным, то для него существует хотя бы один ненулевой критический путь (маршрут) с минимальным суммарным весом ребер (дуг) $p_i^{(\min)}$:

$$p_i^{\min} = \min_{p_i \in \Pi_i} p_i = \min_{\substack{l \in E(G_i) \\ l \in p}} \sum \rho_{il} = \min_i (C(i, A_{l_1 \dots l_n})). \quad (7)$$

В (7) $C(i, A_{l_1 \dots l_n})$ определяется следующим образом:

$$C(i, A_{l_1 \dots l_n}) = (k_{ij-1} - k_{ij}) \prod_{t=1, t \neq i}^n k_{ij}, \quad (8)$$

где $A_{l_1 \dots l_n}$ – множество агрегатов, полученных агрегированием по всем элементам измерений (упорядоченным по номеру); n – количество измерений и $l_i \in \{0, 1\}$; $A_{1 \dots 1}$ – где индексы во всех позициях равны единицы, является множеством исходных данных; $A_{0 \dots 0}$ – где индексы равны нулю, содержит единственное значение полного агрегата. Упорядоченную последовательность $l_1, \dots, l_p, \dots, l_n$, соответствующую множеству агрегатов $A_{l_1 \dots l_n}$, будем называть состоянием агрегации.

Очевидно, что разные состояния агрегации, а, следовательно, и множество агрегатов могут соответствовать одному и тому же уровню детализации, т. е. $l = l_1 + \dots + l_i + \dots + l_n$. Обозначим количество агрегатов множества $A_{l_1 \dots l_n}$, как $a_{l_1 \dots l_n} = |A_{l_1 \dots l_n}|$. Количество агрегатов, полученных агрегированием по некоторым измерениям, равно произведению числа элементов всех остальных измерений, то есть

$$a_{l_1 \dots l_n} = \prod_{i=1}^n k_i. \quad (9)$$

В общем случае обозначим $C(i, A_{l_1 \dots l_n})$ затраты на выполнение агрегирования множества $A_{l_1 \dots l_n}$ по i -му измерению.

В случае определения количества агрегатов по i -му измерению с l_i^* уровнями иерархии, необходимо использовать следующее утверждение. Количество элементарных операций суммирования значений показателя, соответствующих l_i -му уровню, с целью получения агрегированных значений показателя нижнего l_{i-1} -го уровня, равно $k_{ij-1} - k_{ij}$ для $\forall j = \overline{1, l_i^*}$.

Таким образом, при суммировании множеств агрегатов по i -му измерению с l_{i-1} -го уровня на l_i -й уровень необходимо для каждой комбинации из элементов текущих уровней всех остальных $n-1$ измерений провести $k_{ij-1} - k_{ij}$ элементарных операций суммирования, т. е. получаем $C(i, A_{l_1 \dots l_{i-1} l_{i+1} \dots l_n}) = (k_{ij-1} - k_{ij}) \prod_{t=1, t \neq i}^n k_{ij}$.

4. Выполнить слияние G_i и сформировать результирующий гиперкуб в виде графа $G(V, E(V)) = \bigcup_{i=1}^m G_i$, где $i = \overline{1, m}$ – количество объединяемых кубов. В силу ацикличности G_p , результирующий граф $G(V, E(V))$ также будет ацикличен. Ребрами между кубами будут выступать факты формирования соответствующих ациклических запросов на уровне гиперкуба.

5. Для $G(V, E(V))$ необходимо найти вес ребер в соответствии с выражением (10). Учитывая, что одни факты могут быть вычислены только на основании других фактов, для получения оптимизированных запросов вычисления фактов, введем выражение для расчета вычислительных затрат на агрегирование данных в гиперкубе. При вычислении факта гиперкуба в разрезе некоторого измерения многомерного куба, для каждой результирующей строки t -й таблицы фактов потре-

буется вычислять агрегатные значения по обозначенному i -му измерению, т. е.

$$C'' = C(t, A_{l_1 \dots l_{i-1} l_{i+1} \dots l_n}) \cdot \prod_{t=1}^m \left(\prod_{i=1}^n C(i, A'_{l_1 \dots l_{i-1} l_{i+1} \dots l_n}) \right), \quad (10)$$

где t – порядковый номер таблицы фактов Φ_t гиперкуба «ИБФ вуза»; $t = \overline{1, m}$ и $m = 5$. Выражение $C(i, A_{l_1 \dots l_{i-1} l_{i+1} \dots l_n})$ вычисляется по формуле (8). Минимальный критический путь в $G(V, E(V))$ при этом будет соответствовать наименее затратному агрегированному запросу для формирования полного агрегата гиперкуба.

Применим указанную методику для ориентированного графа, описывающего куб «Книжный фонд» (Φ_1), изображенный на рис. 4. Он включает подграф аналитического измерения «Дата издания» (d_2) с иерархиями «Месяц издания» (M_2), «Год издания» (Y_2) и «Весь период» (A_2). Причем, $|M_2| = k_{21} = 12$, $|Y_2| = k_{22} = 80$, $|A_2| = |M_{d_2}| = k_2 = 92$. Аналитическое измерение «Учебный план» (d_8) с иерархиями «Дисциплина» (d_{81}), «Цикл дисциплин» (d_{82}) и «Различные направления квалификационной деятельности вуза» (A_8) представлено на рис. 3. В том числе, $|M_{A_8}| = |M_{d_{81}}| = k_8 = 684$.

Граф, описывающий весь гиперкуб «ИБФ вуза» Φ в обобщенном виде приведен на рис. 5, где в качестве измерений выступают уже группы мер, построенные по вышеобозначенным фактам

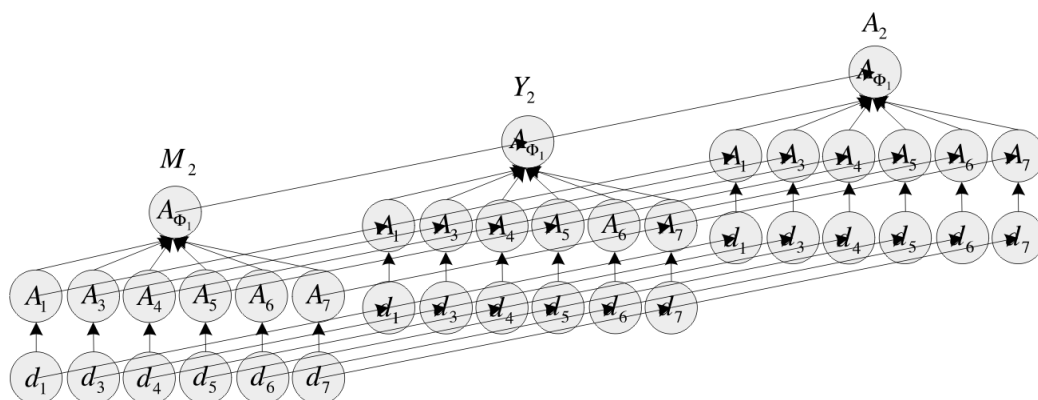


Рис. 4. Ориентированный граф куба «Книжный фонд»

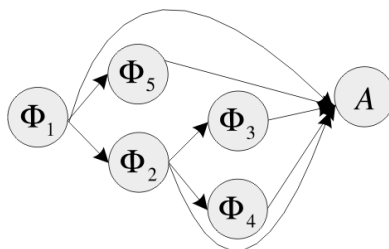


Рис. 5. Ориентированный граф гиперкуба «ИБФ вуза»

$(\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5)$, где $|\Phi_1| = k_{\Phi_1} = 26015$, $|\Phi_2| = k_{\Phi_2} = 31629$, $|\Phi_3| = k_{\Phi_3} = 2293$, $|\Phi_4| = k_{\Phi_4} = 363$ и $|\Phi_5| = k_{\Phi_5} = 1563$.

Для рассматриваемого примера существуют следующие множества агрегатов, соответствующие таблицам фактов: $A^1_{xxxxxx} = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$, $A^2_{xxx} = \{d_7, d_8, d_9, P\}$, $A^3_{xxx} = \{d_8, d_9, P\}$, $A^4_{xxx} = \{d_9, d_{10}, E\}$ и $A^5_{xxx} = \{d_7, d_9, T\}$. Воспользуемся введенными обозначениями для расчета вычислительных затрат на агрегирование данных (9) по факту «Книжный фонд». Определим вычислительные затраты, связанные с вычислением факта *объем фонда по вузу* (A_{Φ_1}) в разрезе «Месяц издания» (M_2) временного измерения «Дата издания» (d_2). Получение множества $A^1_{0100000} = \{d_i | i = 1, n\}$, которому соответствуют агрегированные данные по измерениям d_i куба «Книжный фонд», где $n = 7$, возможно несколькими способами:

$A^1_{0100000} = A^1(1, A^1_{1100000})$ агрегирование множества $A^1_{1100000}$ по измерению «Наименование книги» (d_1), тогда затраты на данную операцию равны $C(1, A^1_{1100000}) = (k_1 - 1)k_{21} = (186556 - 1)12 = 18655 \cdot 12 = 223860$;

$A^1_{0100000} = A^1(3, A^1_{0110000})$ агрегирование множества $A^1_{0110000}$ по измерению «Тип литературы» (d_3), тогда затраты на данную операцию равны

$C(3, A^1_{0110000}) = k_{21}(k_3 - 1) = 12(21 - 1) = 12 \cdot 20 = 240$;

$A^1_{0100000} = A^1(4, A^1_{0101000})$ агрегирование множества $A^1_{0101000}$ по измерению «Издательство» (d_4), тогда затраты на данную операцию равны $C(4, A^1_{0101000}) = k_{21}(k_4 - 1) = 12(1675 - 1) = 12 \cdot 1674 = 20088$;

$A^1_{0100000} = A^1(5, A^1_{0100100})$ агрегирование множества $A^1_{0100100}$ по измерению «Город издания» (d_5), тогда затраты на данную операцию равны $C(5, A^1_{0100100}) = k_{21}(k_5 - 1) = 12(202 - 1) = 12 \cdot 201 = 2412$;

$A^1_{0100000} = A^1(6, A^1_{0100010})$ агрегирование множества $A^1_{0100010}$ по измерению «Автор документа» (d_6), тогда затраты на данную операцию равны $C(6, A^1_{0100010}) = k_{21}(k_6 - 1) = 12(11752 - 1) = 12 \cdot 11751 = 141012$;

$A^1_{0100000} = A^1(7, A^1_{0100001})$ агрегирование множества $A^1_{0100001}$ по измерению «Документ» (d_7), тогда затраты на данную операцию равны $C(7, A^1_{0100001}) = k_{21}(k_7 - 1) = 12(26000 - 1) = 12 \cdot 25999 = 311988$.

Для вычисления факта в разрезе «Месяц издания» требование минимальных затрат на агрегирование данных выполняется в случае формирования множества агрегатов $A^1_{0100000}$ по измерению «Тип литературы». Очевидно, что данный вывод

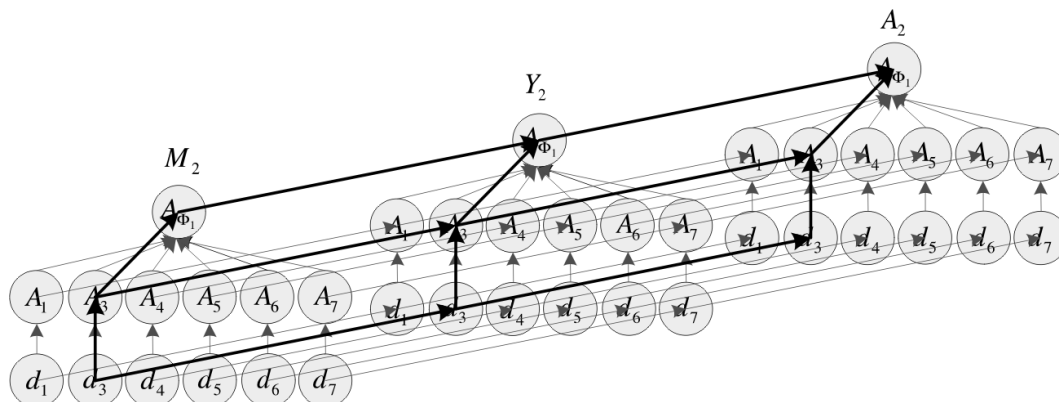


Рис. 6. Наименьший критический путь в ориентированном графе куба Φ_1

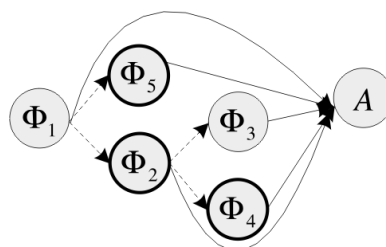


Рис. 7. Ориентированный граф факта «ИБФ вуза» с обозначенными фактами хранения в режиме MOLAP

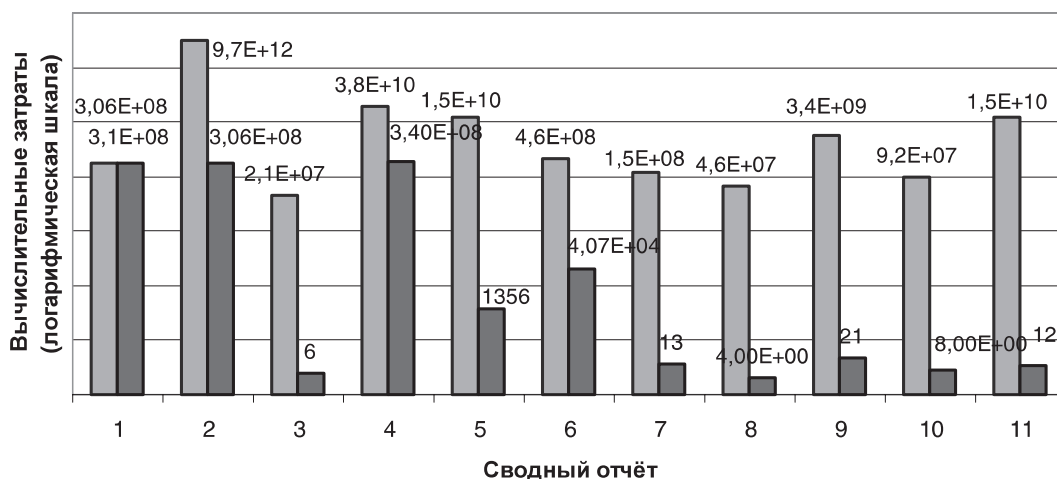


Рис. 8. Вычислительные затраты, связанные с формированием множества агрегатов в режимах хранения данных HOLA* и MOLAP (■) HOLA*; (■) MOLAP

подтвердится и при вычислении факта в разрезах «Год издания» (Y_2) и «Весь период» (A_2). Соответственно, наименьшие критические пути в ориентированном графе для выполнения запроса по факту «Книжный фонд» (Φ_1) будут иметь вид, представленный на рис. 6.

Для формирования наиболее часто используемых и востребованных сводных отчетов для рассматриваемого примера в основном проводится анализ данных по таким фактам, как Φ_2 и Φ_4 . Обратим внимание, что по сравнению с другими данными, таблица фактов Φ_5 характеризуется лавинным нарастанием данных, т. к. накапливает информацию по учету изданий, выдаваемых (возвращаемых) ежедневно.

Поэтому в данном случае принимаем решение: факты Φ_2 , Φ_4 и Φ_5 хранить в режиме MOLAP (рис. 7), тогда затраты на агрегирование данных в оперативном режиме будут минимальными, а выигрыш в вычислительном эффекте составит несколько порядков (рис. 8).

При этом объем дискового пространства, требуемый для хранения агрегированных MOLAP

значений составляет 0,127 Гб, (0,008 % от объема детализированных данных).

Предварительно сформированные по описанной выше методике агрегированные значения показателей, во-первых, минимизируют вычислительные затраты и время на оперативное формирование агрегатов и, во-вторых, обеспечивают максимальную скорость выполнения запросов при фиксированном объеме дискового пространства, занимаемого агрегатами, и при ограничении времени выполнения сводных отчетов. При этом параллельно необходимо отслеживать увеличение степени агрегации данных многомерного куба (например, с помощью предлагаемых в статье решений), анализирующего совокупные данные системы качества вуза. Если она будет превышать α_{kr} , тогда необходимо решать задачу выделения дополнительного дискового пространства и переопределения различных режимов хранения данных в секциях хранилища данных, в зависимости от частоты их использования.

СПИСОК ЛИТЕРАТУРЫ

1. Хрусталеv, Е.В. Агрегация данных в OLAP-кубах [Электронный ресурс]/Е.В. Хрусталеv//Алеф Консалтинг & Софт, 2003.—http://www.iteam.ru/publications/it/section_92/article_1759.pdf
2. Заботнев, М.С. Многомерная модель представления данных по образовательной статистике [Текст]/М.С. Заботнев//Телемеханика—2003: Тр. X Всерос. НМК.—СПб, 2003.—С. 245–246.
3. Попкова, А.А. Коэффициент книгообеспеченности как критерий многомерного анализа обеспеченности учебного процесса [Текст]/А.А. Попкова, С.Г. Фомичева//Научно-технические ведомости СПбГПУ. Сер. Информатика. Телекоммуникации. Управление.—2009.—№ 6.—С. 206–220.
4. Висков, А.В. Средства описания многомерных моделей данных [Текст]/А.В. Висков, М.Б. Фомин//

Вестник РУДН. Сер. Прикладная и компьютерная математика. 2003.–Т.2. –№ 1.–С. 128–139.

5. **Григорьев, Ю.А.** Сравнение технологий хранения данных в OLAP-системах [Электронный ресурс]/ Ю.А Григорьев//Организация баз данных.–2001. –№ 1.

www.amursu.ru/attachments/ics/N01_01.pdf

6. **Zhao, Y.H.** On the performance of an Array-Based ADT for OLAP Workloads [Текст]/Y.H. Zhao, K. Tufte, J.F. Naughton//Technical Report CS-TR-96-13-13, University of Wisconsin-Madison. CS Department, May 1996.

VII Международный научно-практический семинар «Проблемы современной аналоговой микросхемотехники»

Семинар «Проблемы современной аналоговой микросхемотехники» является традиционным и проводится с 2001 года на базе Южно-Российского государственного университета экономики и сервиса с участием ведущих российских и зарубежных ученых и специалистов. В октябре 2009 года в городе Шахты состоялся VII Международный научно-практический семинар.

Общая стратегия проектирования современных изделий микроэлектронной техники направлена на создание под определенную технологию базового комплекта сложных функциональных блоков, решающих не только определенные самостоятельные задачи, но и являющихся фундаментом для построения широкого класса БИС, включая и системы на кристалле (IP-блоки). Именно такой подход существенно сокращает сроки разработки конечной продукции, расширяет номенклатуру изделий массового применения и снижает общие издержки отрасли. Как следствие, оказывается экономически целесообразным проектирование и изготовление изделий микроэлектронной техники под конкретную задачу. Настоящая особенность текущего развития микроэлектроники продлевает «моральный срок» традиционных технологий в секторе крупносерийных изделий. Однако их «живучесть» во многом определяется практическим применением схемотехнических новаций, которые в конечном итоге «маскируют» их несовершенство. Можно с высокой достоверностью утверждать, что указанная тенденция развития аналоговой микросхемотехники сохранится и в отдаленной перспективе.

Доклады участников семинара подводят итоги исследованиям, направленным на развитие принципа собственной и взаимной компенсации влияния технологических несовершенств производства полупроводниковых компонент на основные параметры базовых электронных узлов, составляющих основу любой ИС и БИС. Результаты моделирования предлагаемых авторами принципиальных схем, в частности, показывают, что такие схемотехнические новации позволяют на базе отечественной технологии микронного уровня обеспечить производство вполне конкурентоспособных ИС и БИС.

Исследования, касающиеся решения как теоретических, так и практических вопросов синтеза высокочастотных схем на базе SiGe технологии (в настоящее время ее приобретают ведущие отечественные предприятия отрасли), способствуют созданию нового поколения принципиальных схем и повышению эффективности работы многих предприятий радиоэлектронного профиля.

Д.т.н., профессор кафедры
систем автоматического управления
Южного федерального университета



Сергей Георгиевич Крутчинский

АНАЛОГО-ЦИФРОВЫЕ ПРЕОБРАЗОВАТЕЛИ ДЛЯ СМЕШАННЫХ СИСТЕМ НА КРИСТАЛЛЕ

Для решения задачи адаптивного управления в рамках принципа самоорганизующихся оптимальных регуляторов с экстраполяцией в [1] предложена архитектура смешанных «систем на кристалле» (СнК) с многоканальным параллельным вводом непосредственно в область оперативной памяти собственно процессора цифровых данных, а в [2] – структура аналого-цифровых интерфейсов импульсно-потенциального типа, обеспечивающих указанный выше цикл преобразования с оценкой производной измеряемой величины.

В импульсно-потенциальных АЦ-преобразователях [2] физическая величина посредством коэффициента преобразования чувствительного элемента и опорного напряжения преобразуется в переменное напряжение и длительность импульсов, которые непосредственно счетчиком преобразуются в двоичный код. В этом случае длительность преобразования удваивается по сравнению с обычным АЦП последовательного приближения. Однако максимальная частота тактовых импульсов традиционных преобразователей определяется не быстроедействием цифровой схемы пересчета, а длительностью переходных процессов в ЦАП и длительностью фронта аналогового ком-

паратора. Именно поэтому максимальная частота тактовых импульсов для существующих технологий (0,18–0,25 мкм) в импульсно-потенциальных АЦП в 7–10 раз больше по сравнению с АЦП поразрядного уравнивания.

Можно также утверждать, что при реализации смешанных СнК в виде системы на подложке (SoP) или системы в корпусе (SiP), указанная тенденция будет не только сохраняться, но и позволит создать многоканальные СнК без перевода центрального процессорного элемента в «sleep»-режим. Однако совокупность предложенного не позволяет исключить прецизионный ограничитель спектра, необходимый для обеспечения предельной точности АЦ-преобразования. В статье предлагается развитие подхода, направленного на повышение точности соответствующих АЦ-интерфейсов как сложно-функциональных блоков (СФ блоков) указанных СнК.

Существенное повышение точности вычисления при одновременном упрощении аналоговых портов ввода и преобразования аналоговых величин возможно при пересмотре принципа АЦ-преобразования, когда используется жесткая корреляция опорных источников (баз сравнения). На рис. 1 приведена архитектура контроллера

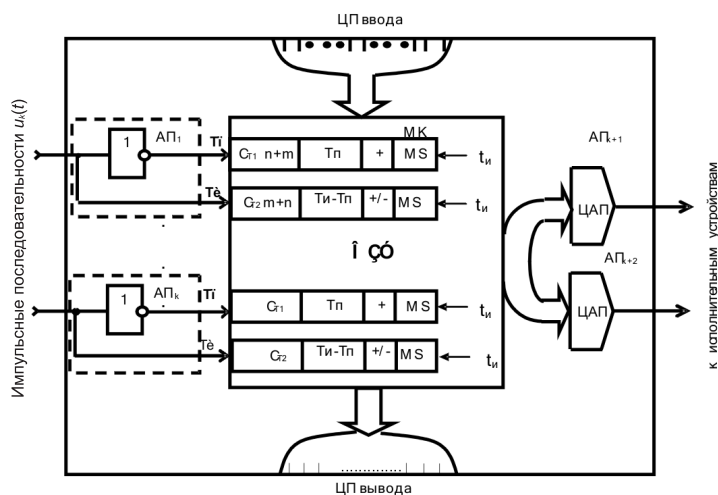


Рис. 1. Организация прямого доступа к памяти результатов
АЦ-преобразования

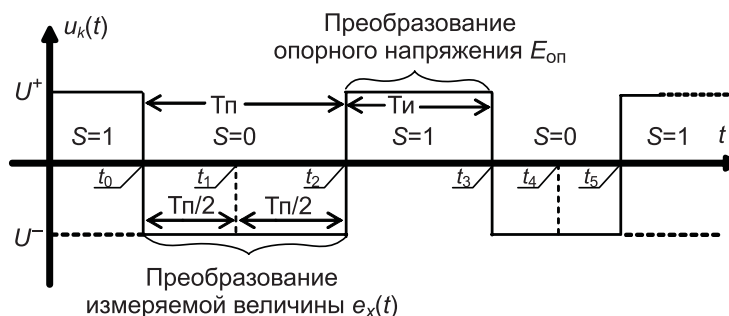


Рис. 2. Базовая временная диаграмма работы АЦП

смешанной СнК, ориентированная на обработку сигналов импульсных датчиков [2].

В этом случае преобразование длительности паузы (T_n) и длительности импульса (T_i) в цифровой код осуществляется в счетчиках СТ1 и СТ2 разрядности $n+m$ с методической погрешностью, определяемой периодом тактовых импульсов ($2ti$) кварцевого генератора. При этом из структуры порта ввода исключаются собственно АЦП и ограничитель спектра входного аналогового сигнала. Потенциально высокая точность преобразования достигается низкой погрешностью T_n и T_i , вызванной действием параметров аналоговых компонентов схемы импульсного преобразователя. Важным свойством такой структуры является возможность прямого доступа к памяти контроллера при любом числе входных каналов, а также их параллельная работа вместе с МК.

Повышение точности АЦ-преобразования в анализируемой структуре (рис. 1) достигается интегрированием измеряемого напряжения $e_x(t)$ на интервале формирования отрицательного импульса компаратора напряжения (рис. 2) [3].

Анализ схемы (рис. 3) при $e_x(t) = e_{x0} + V \cdot t$ приводит к следующему результату:

$$e_x(t) = e_x(t_1) = e_x(t_0 + T_n/2); \quad (1)$$

$$T_n = \tau \cdot k \cdot \frac{U^- + U^+}{e_x(t) + E_{on}}; T_u = \tau \cdot k \cdot \frac{U^- + U^+}{E_{on}}; \quad (2)$$

$$T_n = T_u \frac{E_{on}}{e_x(t) + E_{on}}, \quad e_x(t) = \frac{T_u - T_n}{T_n} E_{on}; \quad (3)$$

$$\dot{e}_x(t_c) = (e_x(t_4) - e_x(t_1)) / (t_4 - t_1), \quad (4)$$

где $t_c = t_1 + (t_4 - t_1)/2$.

Таким образом, жесткая корреляция длительности импульсов на каждом периоде ($T = T_n + T_u = t_3 - t_0$) позволяет практически исключить влияние пассивных элементов на точность АЦ-преобразования. Для уменьшения параметрической чувствительности

$$S_{T_u}^{e_x} = -S_{T_n}^{e_x} = T_n / (T_u - T_n) \quad (5)$$

в структуре ОЗУ микроконтроллера (МК) счетчик СТ₂ необходимо реализовать в двухканальном варианте, причем один из каналов должен работать в реверсивном режиме. Перевод счетчика из ре-

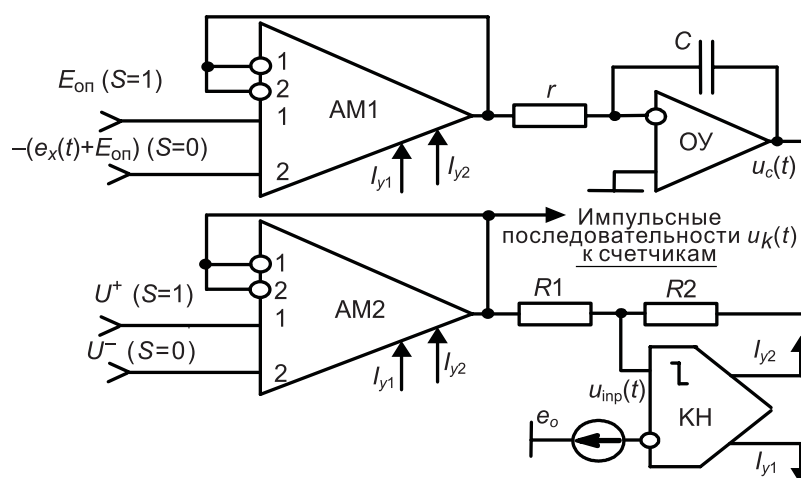


Рис. 3. Структурная схема АЦП импульсно-потенциального типа

жима суммирования в режим вычитания осуществляется отрицательным фронтом $u_k(t)$. Тогда в первом канале счетчика фиксируется длительность T_n , а во втором – $T_n - T_n$. В этом случае чувствительности измерения не превышает единицы, и вычисление $e_x(t)$ не сопровождается увеличением погрешности, которая сохраняет свое методическое значение ($2t_n$).

Можно показать, что основными факторами, влияющими на точность АЦ-преобразования, являются: статический коэффициент усиления (μ) и частота единичного усиления (f_1) ОУ, входящего в состав интегратора (рис. 3), напряжение гистерезиса аналогового компаратора (ΔU_k) и ЭДС смещения ($E_{см}$) аналогового тракта. Что касается влияния параметров мультиплексора, то рассматриваемая структура предполагает использование специально спроектированных мультидифференциальных операционных усилителей (МОУ) [4], которые обладают потенциально высокими показателями качества (высокий (135дБ) коэффициент ослабления синфазного сигнала; высокий коэффициент усиления дифференциального сигнала, обеспечивающий глубокую обратную отрицательную связь; низкое значение дрейфа нуля; широкий диапазон рабочих частот как для синфазного, так и для дифференциального сигнала) и низкой величиной $E_{см}$.

Потенциально высокие метрологические качества импульсно-потенциальных АЦП требуют специального подхода к схемотехническому проектированию его базовых узлов. В первую очередь взаимосвязь интервалов (2) невозможна без высококачественных проходных ключей (рис. 3).

Для современного этапа развития технологии полупроводниковых компонентов наиболее эффективное решение проблемы осуществляется на базе МОУ [4], когда дифференциальные входы 1 и 2 (см. рис. 3) используются автономно в режиме интегрирования $E_{оп}$, ($S = 1$) или $-(e_x(t) + E_{оп})$, ($S = 0$).

Результаты моделирования 16-разрядного АЦП на системном уровне, с учетом влияния изложенных выше факторов, представлены на рис. 4.

Системный анализ АЦП показал, что доминирующими факторами, определяющими точность преобразования, является ЭДС смещения аналоговых узлов (аналоговый мультиплексор (АМ) и операционных усилитель (ОУ)), а также статический коэффициент усиления ОУ интегратора. Это и конкретизирует задачи схемотехнического проектирования указанных узлов. Следует отметить, что низкое влияние частоты единичного усиления ОУ интегратора (рис. 4) позволяет в рамках традиционной схемотехники создать энергоэкономичный ОУ с высоким (более 100 дБ) статическим коэффициентом усиления. Кроме этого допустимое напряжение гистерезиса аналогового компаратора (КН) в 100 мкВ позволяет также при его схемотехнической разработке существенно уменьшить потребляемую мощность.

На основе приведенных результатов возможно оценить точность АЦ-преобразования с учетом влияния указанных параметров активных элементов. Применение 16-разрядного АЦП для преобразования сигналов стандартных чувствительных элементов в диапазоне рабочих частот до 1 кГц и длительностью тактовых импульсов

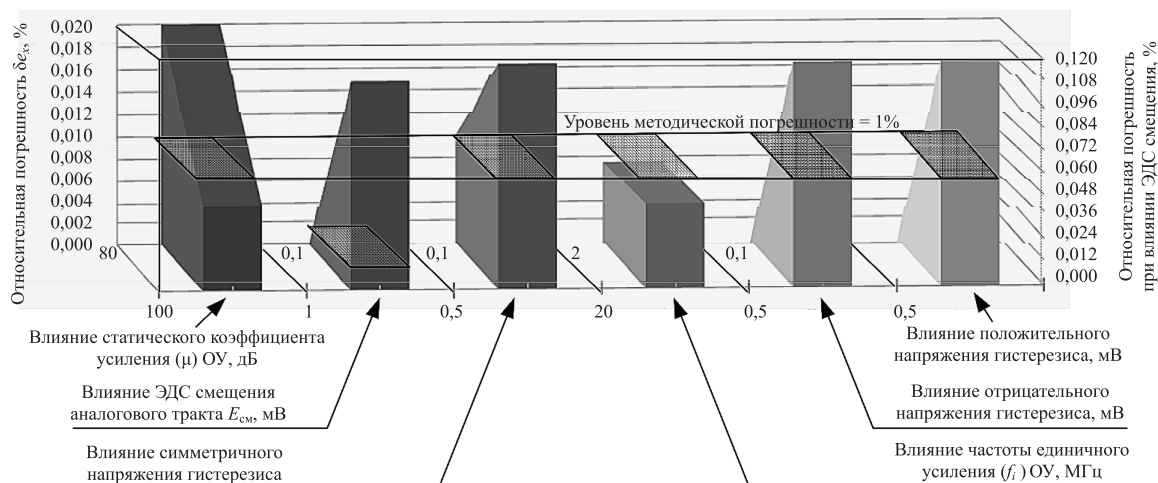


Рис. 4. Диаграмма влияния дестабилизирующих факторов на точность АЦ-преобразования

Таблица 1

Результаты моделирования АЦП с учетом влияния дестабилизирующих факторов

e_{x0} , В	$T_{и}$, мс	$T_{и-расч}$, мс	t_1 , мс	$T_{п}$, мс	$T_{п-расч}$, мс	e_x , В	δe_x , %	e_{x0} , В	e_x , В	δe_{x0} , %
0,238693	10,001	10	9,52	8,938	8,934	0,2379	0,3500	25,088	0,000	-
0,682557	10,001	10	27,715	7,457	7,456	0,6823	0,0360	24,448	24,427	0,086
1,065514	10,001	10	44,711	6,525	6,524	1,0654	0,0069	22,575	22,542	0,145
1,385876	10,001	10	60,921	5,908	5,907	1,3856	0,0214	19,797	19,749	0,243
1,641772	10,001	10	76,63	5,492	5,492	1,6420	0,0154	16,316	16,325	0,052
1,830381	10,001	10	91,99	5,223	5,221	1,8296	0,0427	12,298	12,212	0,707
1,949812	10,001	10	107,135	5,065	5,064	1,9491	0,0385	7,898	7,888	0,125
1,998735	10,001	10	122,17	5,003	5,002	1,9980	0,0367	3,259	3,255	0,117
1,976607	10,001	10	137,183	5,031	5,029	1,9758	0,0434	-1,476	-1,482	0,404
1,88361	10,001	10	152,282	5,1515	5,150	1,8828	0,0456	-6,168	-6,159	0,149
1,720895	10,001	10	167,546	5,3755	5,375	1,7210	0,0035	10,676	10,600	0,722
1,49019	10,001	10	183,101	5,731	5,730	1,4901	0,0033	14,855	14,839	0,112
1,193794	10,001	10	199,1	6,263	6,262	1,1937	0,0098	18,557	18,530	0,146
0,834191	10,001	10	215,762	7,058	7,057	0,8339	0,0293	21,622	21,590	0,148
0,41325	10,001	10	233,438	8,289	8,288	0,4131	0,0417	23,863	23,810	0,223

не менее 0,5 мкс позволяет обеспечить методическую точность преобразования. Так, при использовании ОУ со статическим коэффициентом усиления 100 дБ и частотой единичного усиления 2 МГц, аналогового компаратора с напряжением гистерезиса 25 мкВ, и ЭДС смещения аналогового тракта 25 мкВ и с учетом влияния широкополосного шума с частотой 2 кГц и амплитудой 2 мВ, погрешность измерения носит методический характер. Результаты моделирования этого АЦП при $e_{x0} = 2\sin(4\pi \cdot t)$ В приведены в табл. 1.

Из таблицы видно, что относительная погрешность измеряемой величины имеет максимальное значение (0,05 %) при минимальной величине производной измеряемой величины, что объясняется соответствующими функциями чувствительностей:

$$S_{T_u}^{e_x} = \frac{T_u}{T_u - T_n}; S_{T_n}^{e_x} = \omega T_n \operatorname{ctg}(\omega t), \quad (6)$$

которые имеют максимальное значение при $t < 60$ мс.

Применение указанной методики позволило создать на основе компонентов радиационно-стойкого аналогового базового матричного кристалла (АБМК) [5] 14-разрядный АЦП, воспроизводящий структуру, показанную на рис. 3.

Эта разрядность определялась архитектурой и параметрами микроэлектронной системы в корпусе для задач автоматического управления и технической диагностики. Совокупность технологических ограничений потребовала ограничить $t_u \geq 1$ мкс. Результаты моделирования этого АЦП в системе PSpice приведены в табл. 2.

Следует отметить, что согласно архитектуре SiP (см. рис. 1) в реальных системах используется многоканальный ввод данных без мультиплексирования, поэтому указанное в табл. 2 время

Таблица 2

Основные характеристики АЦП на базе компонентов радиационно-стойкого АБМК

Разрядность n	t_u , мкс	Диапазон рабочих частот, кГц	Максимальное время выборки, мс	Опорное напряжение, В	Нелинейность δ	Потребляемая мощность, мВт
14	1	1	20	2	$1/2^{n+1}$	130

выборки определяет ее реальную производительность.

Полученные результаты позволяют сделать ряд важных выводов для развития современных микроэлектронных средств систем управления и технической диагностики:

повышение точности преобразования измеряемых величин достигается за счет суммирования $e_x(t)$ и E_{on} , что требует применение в ОЗУ МК реверсивного счетчика;

возможность параллельного многоканального ввода в область оперативной памяти позволяет повысить как быстродействие СнК в целом, так и ее производительность;

указанные свойства структуры АЦП не требуют остановки работы МК на интервале пре-

образования и применения в структуре аналогового интерфейса прецизионных ограничителей спектра.

Кроме этого, рассмотренная структура позволяет минимизировать воздействие широкополосного шума на точность преобразования. Так, при $n = 16$ и воздействии помехи с минимальной частотой 2 кГц и амплитудой менее 4 мВ погрешность преобразования не превышает свое методическое значение ($2t_u$). Детальное исследование показывает, что в случае использования АБМК также существует возможность увеличения разрядной сетки АЦП до 20 при минимальной допустимой длительности тактовых импульсов ($t_{u\min}$) 10 нс.

Статья подготовлена в рамках госконтракта № П507 ФЦП «Научные и научно-педагогические кадры инновационной России (2009-2013 гг.)».

СПИСОК ЛИТЕРАТУРЫ

1. **Крутчинский, С.Г.** Смешанные системы на кристалле для систем автоматического управления и технической диагностики: сб. трудов [Текст] / С.Г. Крутчинский. – МЭС-2006. – С. 217–222.

2. **Крутчинский, С.Г.** Аналого-цифровые интерфейсы микроконтроллерных адаптивных регуляторов циклического типа для объектов электроэнергетики [Текст]/С.Г. Крутчинский//Изв. РАН. Сер. Автоматика и телемеханика. –2006. –№ 5. –С. 163–174.

3. **Крутчинский, С.Г.** Аналого-цифровые интерфейсы смешанных систем на кристалле [Текст]/С.Г. Крутчинский, Р.Г. Баранов//Изв. ЮФУ. –2008. –С. 163–174.

4. **Крутчинский, С.Г.** Мультидифференциальные операционные усилители и прецизионная микросхемотехника [Текст] / С.Г. Крутчинский, Е.И. Старченко // Электроника и связь. –2004. –№ 20. –С. 37–45.

5. Каталог разработок Российско-Белорусского центра аналоговой микросхемотехники [Текст]/Под ред. С.Г. Крутчинского. –Шахты, 2006. –87 с.

УДК 621.37

Н.Н. Прокопенко, А.И. Серебряков, В.Г. Манжула

КОМПЕНСАЦИЯ НАПРЯЖЕНИЯ СМЕЩЕНИЯ НУЛЯ ОПЕРАЦИОННЫХ УСИЛИТЕЛЕЙ С НЕСИММЕТРИЧНЫМ ВКЛЮЧЕНИЕМ АКТИВНОЙ НАГРУЗКИ

Дифференциальные операционные усилители (ОУ) с активной нагрузкой, обеспечивающей непосредственное управление двухтактным буферным усилителем [1] имеют одноканальную структуру передачи сигнала по цепи общей отрицательной обратной связи [2] и характеризуются меньшими фазовыми искажениями сигнала, более высокими показателями, характеризующими устойчивость ОУ [3].

Существенный недостаток ОУ данного под-

класса состоит в том, что они имеют повышенное значение систематической составляющей напряжения смещения нуля ($U_{см}$), связанной с несимметрией его архитектуры. Цель настоящей статьи – исследование возможностей предлагаемой архитектуры операционного усилителя [4], в которой создаются специальные условия для взаимной компенсации влияния абсолютных значений токов базы дополнительных транзисторов, а также их температурных и радиационных изменений

на систематическую составляющую напряжения смещения нуля ($U_{см}$).

На рис. 1 представлена схема несимметричного ОУ с малым $U_{см}$ [4]. Рассмотрим факторы, определяющие систематическую составляющую напряжения смещения нуля $U_{см}$, т. е. зависящие от схемотехники ОУ.

Если ток источника тока I_1 равен величине $2I_0$, а I_2 и I_3 – величине I_0 , то токи эмиттеров и коллекторов транзисторов схемы:

$$I_{э3} = 2I_0 + 2I_{б,р}, I_{к3} = 2I_0, I_{к1} = I_{к2} = I_0 - I_{б,р}, \quad (1)$$

$$I_{э7} = I_{э8} = I_0, I_{к7} = I_{к8} = I_0 - I_{б,р}, I_{к4} = I_0 - 2I_{б,р} = I_{к5}, \quad (2)$$

$$I_{вх,пт1} = I_{вых,пт1} = I_0 - I_{б,р}, \quad (3)$$

где $I_{б,i} = I_{э,i} / \beta_i$ – ток базы $n-p-n$ транзисторов при эмиттерном токе $I_{э,i} = I_0$; $I_{вх,пт1} = I_{вых,пт1}$ – входной и выходной токи повторителя тока ПТ1 (так называемого «токового зеркала»); β_i – коэффициент усиления по току базы транзисторов схемы при $I_{э,i} = I_0$.

Поэтому разность токов в узле «А» при его замыкании на некоторый эквивалентный источник напряжения E_A , равный статическому напряжению U_A в схеме ОУ с обратной связью (рис. 1) соответствует:

$$I_p = I_{вых,пт1} - I_{б,6} - I_{к4} = 0, \quad (4)$$

где $I_{б,6} = I_{б,р}$ – ток базы транзистора VT6.

Как следствие, это уменьшает $U_{см}$, т. к. разностный ток I_p в узле «А» создает $U_{см}$, зависящее от крутизны (S) преобразования входного дифференциального напряжения $u_{вх}$ в выходной ток узла «А»:

$$S = \frac{i_p}{u_{вх}} = \frac{1}{r_{э1} + r_{э2}}, \quad (5)$$

где $r_{э1} = r_{э2}$ – сопротивления эмиттерных переходов входных транзисторов VT1 и VT2 входного дифференциального каскада ДК1.

Поэтому

$$U_{см} \approx I_p (r_{э1} + r_{э2}) = I_p \left(\frac{2\phi_T}{I_0} \right) \approx 0, \quad (6)$$

где $\phi_T = 26$ мВ – температурный потенциал.

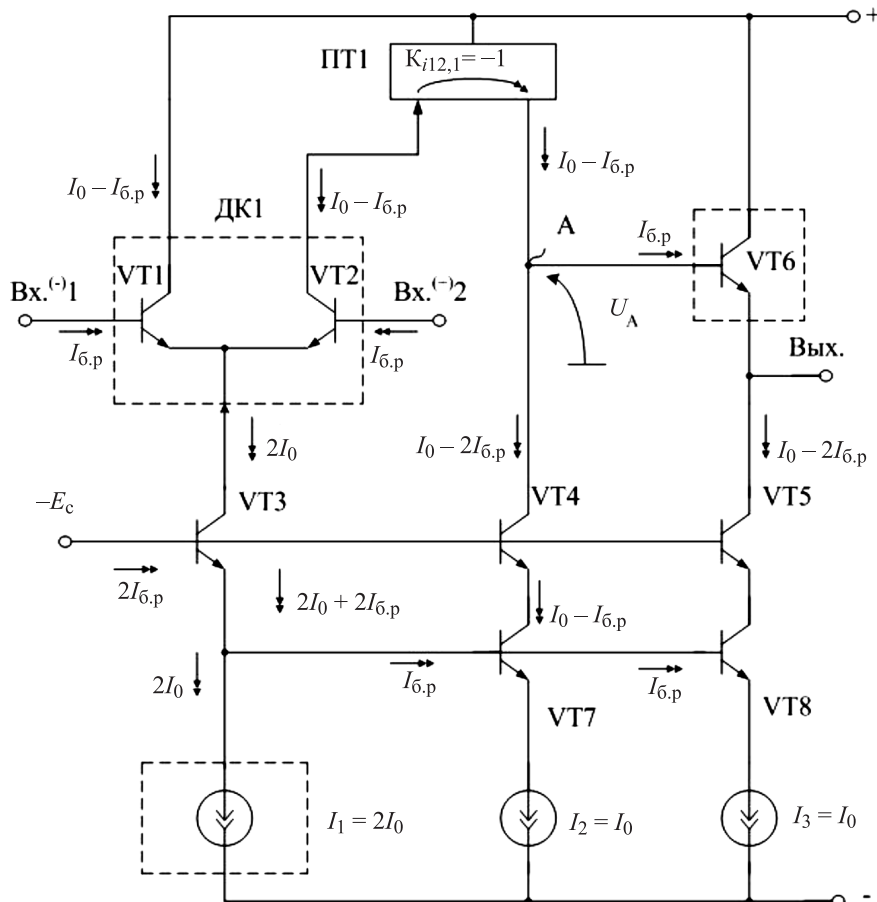


Рис. 1. Архитектура ОУ с малым напряжением смещения нуля

Физически это означает, что абсолютные значения токов базы транзисторов, приведенные к высокоимпедансному узлу «А», а также их приращения, которые в интегральной микросхеме при одинаковых токах эмиттера достаточно идентичны, взаимно компенсируются.

Таким образом, в предлагаемом ОУ при выполнении условия (4) уменьшается систематическая составляющая $U_{см}$, обусловленная конечной величиной β транзисторов и его радиационной (или температурной) зависимостью.

В классическом ОУ [1] разностный ток $I_p \neq 0$, поэтому здесь систематическая составляющая $U_{см}$ получается как минимум на порядок больше, чем в предлагаемой схеме.

На рис. 2 показана модифицированная схема ОУ, в которую введен дополнительный транзистор VT9. Это обеспечивает температурную и

радиационную компенсацию дрейфа токов через $p-n$ переходы VD2, VD3, VD9, VD4, VD7 на подложку транзисторов VT4, VT7 и VT2, VT3, VT9.

В схеме на рис. 1 температурные и радиационные изменения токов через изолирующие $p-n$ переходы транзисторов VT2 и VT4 на подложку компенсируют друг друга в высокоимпедансном узле «А» благодаря токовому зеркалу ПТ1. В то же время приращения аналогичных токов на подложку транзисторов VT3 и VT7 приводятся к узлу «А» с неодинаковыми коэффициентами передачи. Это связано с делением тока $p-n$ перехода на подложку транзистора VT3 между эмиттерами транзисторов VT1 и VT2. Введение VT9 (рис. 2) создает условия, при которых все токи $p-n$ переходов на подложку (VD3, VD2, VD9, VD4, VD7) компенсируют друг друга в узле «А», что повы-

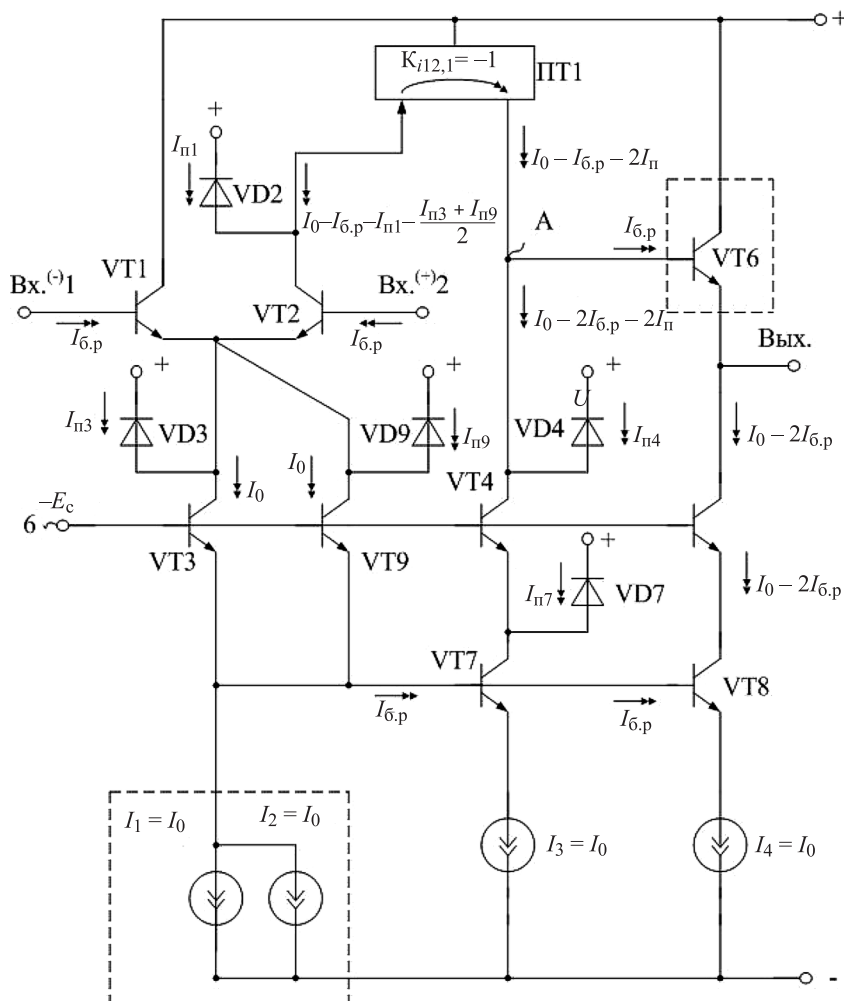
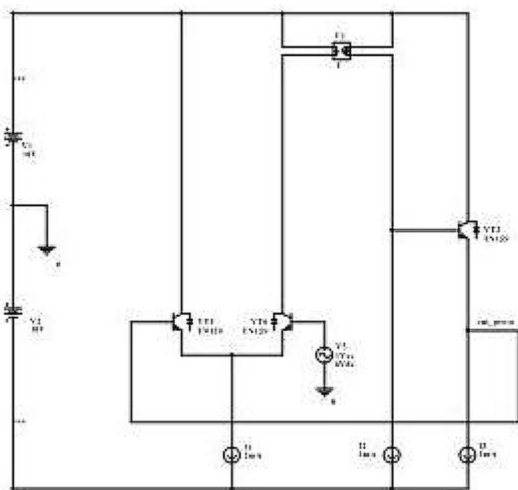


Рис. 2. Операционный усилитель с цепью терморadiационной компенсации токов $p-n$ переходов на подложку транзисторов VT4, VT7 и VT2, VT3, VT9

а)



б)

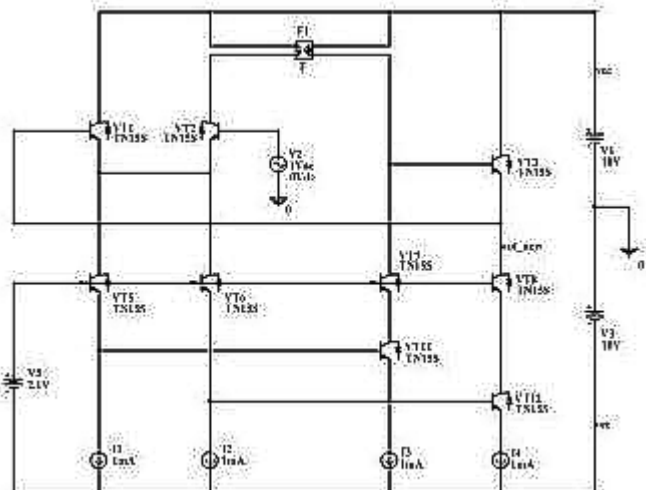


Рис. 3. Схемы классического (а) ($U_{cm} = 2202$ мкВ) и предлагаемого (б) ($U_{cm} = 281$ мкВ) ОУ в среде компьютерного моделирования PSpice

шает общую стабильность U_{cm} при высоких температурах (более 70°), а также при радиационном воздействии.

На рис. 3 показаны схемы классического и предлагаемого ОУ в среде компьютерного моделирования PSpice на моделях интегральных транзисторов ФГУП НПП «Пульсар» при идентичных токах двухполосников $I_1 = I_2 = I_3 = I_4 = 1$ мА.

На рис. 4 приведены результаты компьютер-

ного моделирования U_{cm} сравниваемых схем, которые подтверждают теоретические выводы.

Несмотря на существенное уменьшение β транзисторов вследствие радиационных воздействий, предлагаемый ОУ и в этих условиях имеет меньшее напряжение смещения нуля, чем классический ОУ.

Таким образом, рассмотренный способ компенсации систематической составляющей U_{cm}

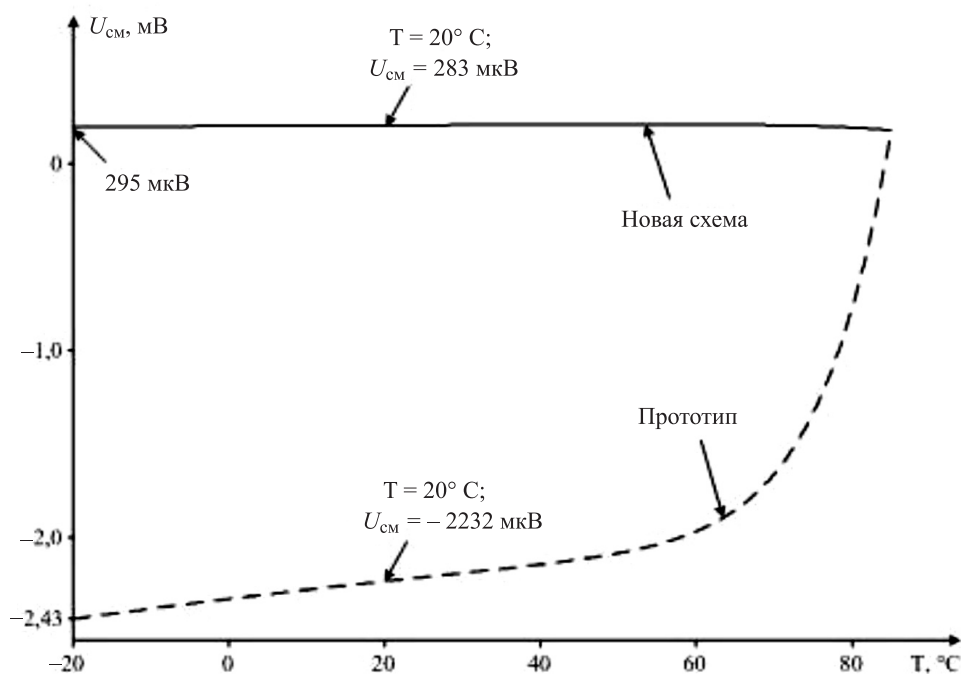


Рис. 4. Температурные зависимости сравниваемых схем ОУ

обеспечивает существенное снижение статической ошибки усиления сигналов постоянного тока ОУ и может использоваться в схемах различных ИР-модулей современных систем на кристалле.

Статья подготовлена в рамках госконтракта П507 от 05.08.2009 г. ФЦП «Научные и научно-педагогические кадры инновационной России (2009-2013 гг.)».

СПИСОК ЛИТЕРАТУРЫ

1. Патент US 4.415.868, fig.3. Current U.S. Class: 330/294 ; 330/107 One and one half pole audio power amplifier frequency compensation / Gross; William H.: Assignee National Semiconductor Corporation (Santa Clara, CA). –Nov. 15, 1983.

2. Прокопенко, Н.Н. Архитектура и схемотехника быстродействующих операционных усилителей: монография [Текст] / Н.Н.Прокопенко, А.С.Будяков. – Шахты: Изд-во ЮРГУЭС, 2006. – 231 с.

3. Полонников, Д.Е. Операционные усилители: Принципы построения, теория, схемотехника: монография [Текст] / Д.Е.Полонников. –М., 1983.–216 с.

4. Дифференциальный операционный усилитель: заявка на патент РФ, МПК8 Н 03 F 3/45, 3/34 [Текст] / Н.Н. Прокопенко, Д.С. Федяшов, А.И. Серебряков. –№ 2009133746/09; заявл. 08.09.2009 (242).

УДК 621.372

С.Г. Крутчинский, А.Е. Титов

МУЛЬТИДИФФЕРЕНЦИАЛЬНЫЙ ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ В РЕЖИМЕ ИНСТРУМЕНТАЛЬНОГО УСИЛИТЕЛЯ

Создание аналоговых и аналого-цифровых интерфейсов смешанных систем на кристалле (СнК), ориентированных на взаимодействие с чувствительными элементами мостового типа всегда предполагает применение инструментальных усилителей (ИУ), выполняющих функции подавления синфазного сигнала и усиление дифференциального напряжения. Как правило, такой ИУ реализуется на базе классической схемы, состоящей из трех операционных усилителей и семи прецизионных резисторов. Поэтому даже при использовании строго идентичных операционных усилителей (ОУ) предельное значение коэффициента ослабления синфазного сигнала определяется точностью реализации резистивных элементов. Так, для прецизионных технологий ($\Theta_R = 0,01\%$ дБ) $K_{осн} = -54$ дБ, что явно недостаточно для построения даже неprecизионных датчиков. При производстве соответствующих сложнофункциональных (СФ) блоков СнК в вариантах система в корпусе (SiP) и система на подложке (SoP) применяется специальная функциональная настройка, направленная на достижение требуе-

мых качественных показателей ($K_{осн} = -54$ дБ). Кроме того, потребляемая мощность таких ИУ достаточно велика, что препятствует использованию в этих системах.

В связи с этим поиск альтернативных вариантов решения аналогичной задачи для смешанных СнК в любом из вариантов их технологической реализации приобретает важное практическое значение.

Для решения данной проблемы с помощью эффективных схемотехнических решений, основанных на введении дополнительных функциональных обратных связей, направленных на минимизацию $K_{сн}$ [1], создан относительно новый класс активных элементов – мультидифференциальных ОУ (МОУ), – которые и могут стать основой схемотехники таких ИУ. Следует отметить, что коэффициент ослабления синфазного сигнала разработанных МОУ практически не зависит от точности реализации резистивных элементов.

Структура и условное обозначение МОУ показаны на рис. 1. Этот активный элемент состоит из двух дифференциальных (ДК), одного проме-

Таблица 1

Параметры МОУ

Пара- метры	$K_{оссн}$	$f_{гр_сн}$	$U_{гр_сн}$	$U_{гр_д}$	μ	f_1	$E_{см}$	$E_{п1}$	I_1	$E_{п2}$	I_2
Схема	дБ	кГц	В	В	дБ	МГц	мкВ	В	мА	В	мА
МОУ	-104	8,3	-4 5	-1,3 1,2	50	5	0,26	-5	2,8	5	4,5

жуточного (ПК) и одного выходного (ВК) каскадов. Результаты моделирования созданного на базе радиационно-стойкого аналогового базового матричного кристалла (АБМК) [2] МОУ в среде PSpice представлены в табл. 1.

Здесь $K_{оссн}$ – коэффициент ослабления синфазного сигнала; $f_{гр_сн}$ – граничная частота $K_{оссн}$; $U_{гр_сн}$ и $U_{гр_д}$ – граничные напряжения при подаче синфазного и дифференциального сигналов; μ – статический коэффициент передачи усилителя; f_1 – частота единичного усиления; $E_{см}$ – ЭДС смещения; $E_{п1}$ и $E_{п2}$ – напряжения шины питания; I_1 и I_2 – потребляемые токи.

Для построения инструментального усилителя на базе МОУ достаточно ввести глубокую отрицательную обратную связь (рис. 2), поэтому устройство (в отличие от классического аналога) будет характеризоваться небольшим потребляемым током. Предельное значение коэффициента передачи синфазного напряжения

$$K_{сн} = K_{оссн} \cdot K_{д} \tag{1}$$

в таком ИУ определяется реализуемым коэффициентом усиления:

$$K_{д} = 1 + \frac{R}{r}. \tag{2}$$

Напряжение дрейфа нуля ИУ ($U_{др}$) здесь также прямопропорционально реализуемому дифференциальному коэффициенту усиления:

$$U_{др} = E_{см} \cdot K_{д}. \tag{3}$$

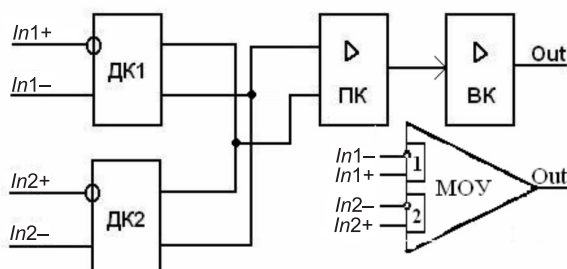


Рис. 1. Структура и условное обозначение МОУ

Эта взаимосвязь параметров и определяет область применения предлагаемого инструментального усилителя. Действительно, в классической схеме влияние дифференциального коэффициента усиления на $U_{др}$ и $K_{сн}$ значительно меньше. Однако наличие в структуре ДК1 компенсирующих обратных связей предварительно обеспечивает глубокое ослабление синфазного напряжения, а взаимосвязь режимов работы динамических нагрузок в структуре МОУ [3] позволяют обеспечить низкое значение $E_{см}$ (табл. 1). Указанные особенности схемотехники МОУ позволяют увеличить достижимый дифференциальный коэффициент усиления при сохранении относительно высоких требований к $K_{сн}$ и $U_{др}$.

Результаты моделирования инструментального усилителя при различных значениях дифференциального коэффициента усиления в среде PSpice представлены в табл. 2.

Здесь $f_{гр_д}$ – граничная частота дифференциального коэффициента усиления; ϑ – скорость нарастания выходного напряжения.

Указаны наихудшие значения полученных параметров ИУ с учетом исследования схемы методом Monte Carlo, принимая во внимание, что погрешность реализации полупроводниковых резисторов находится в пределах 1,5 % в условиях радиационного воздействия.

Для определения влияния структуры инструментального усилителя при различном дифференциальном коэффициенте передачи на величину

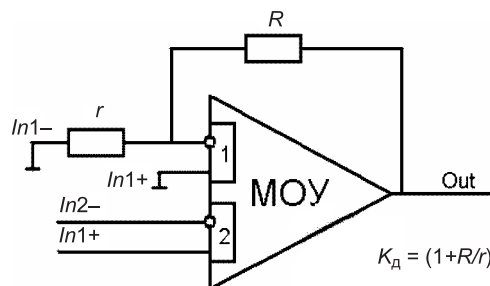


Рис. 2. Инструментальный усилитель на одном МОУ

Таблица 2

Параметры ИУ										
$K_{\text{осн}}$	$f_{\text{гр,сн}}$	$U_{\text{гр,сн}}$	$U_{\text{гр,д}}$	$K_{\text{д}}$	$f_{\text{гр,д}}$	$E_{\text{н1}}$	I_1	$E_{\text{н2}}$	I_2	ϑ
дБ	кГц	В	В		МГц	В	мА	В	мА	В/мкс
-98	8,4	-4	-1,2	2	3	-5	2,8	5	4,5	17
		5	1,4							-2,5
-91	8,3	-3,8	-0,5	5	0,76	-5	2,8	5	4,5	17
		4,9	1,2							-2,5
-84	8,3	-3,8	-0,2	10	0,26	-5	2,8	5	4,5	17
		4,8	0,5							-2,5
-75	8,3	-3,8	-0,1	27	0,1	-5	2,8	5	4,5	17
		4,8	0,2							-2,5

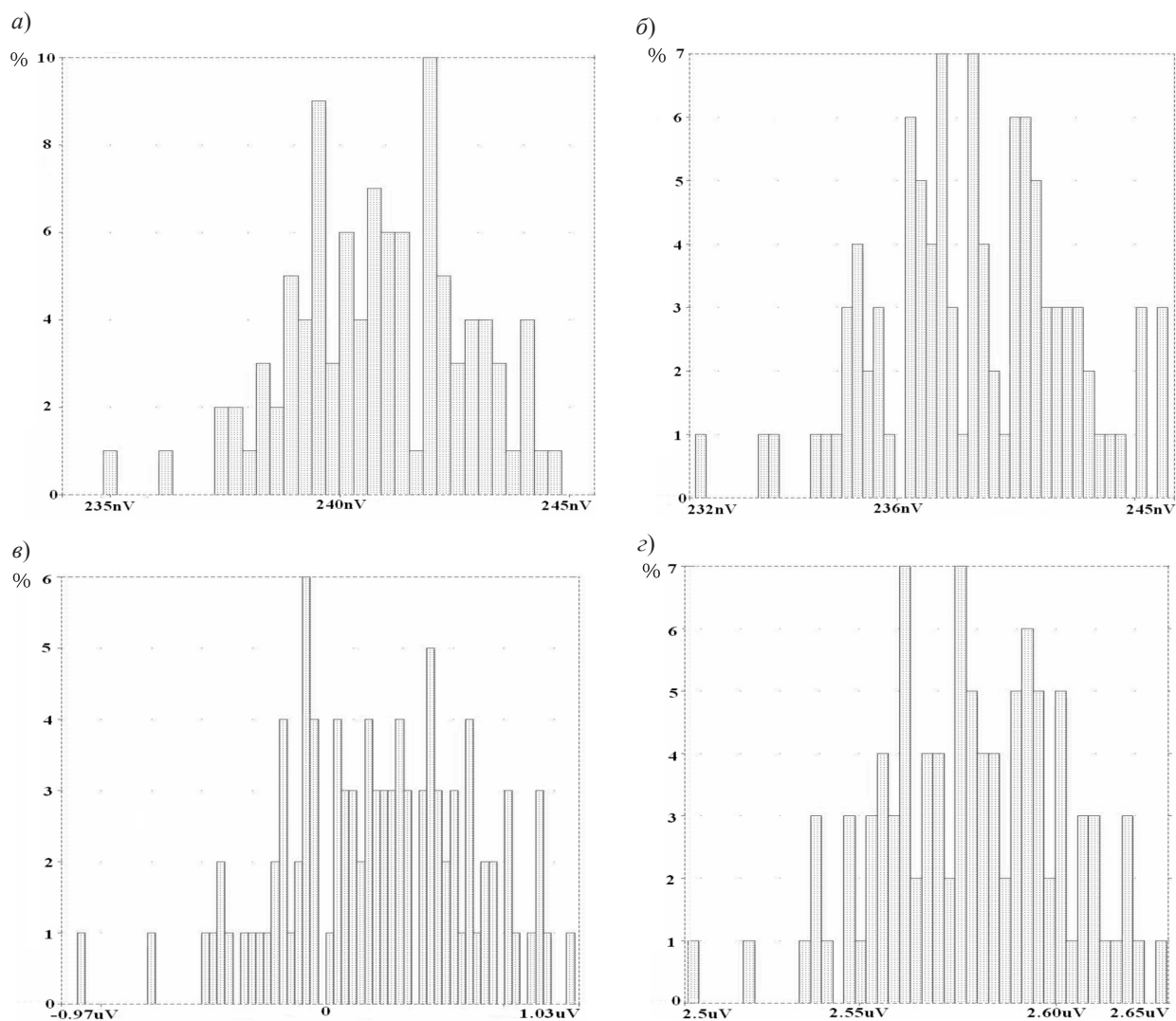


Рис. 3. Дрейф нуля инструментального усилителя:
 а – при $K_d = 2$ (235 – 245 нВ); б – при $K_d = 5$ (232 – 245 нВ); в – при $K_d = 10$ (0,97 – 1,03 мкВ);
 з – при $K_d = 27$ (2,5 – 2,65 мкВ)

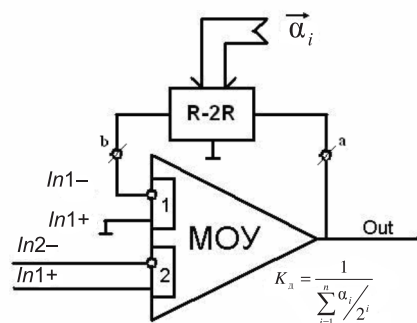


Рис. 4. Инициализируемый позиционным кодом $\alpha_i (i = 1, n)$ ИУ

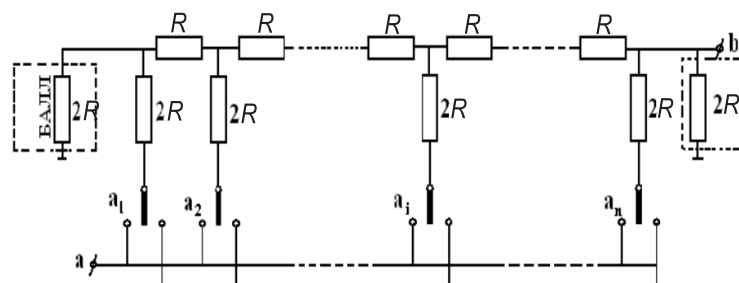


Рис. 5. Лестничная резистивная схема в режиме масштабирования тока

ну дрейфа нуля, проведено исследование схемы методом Monte Carlo. Результаты моделирования в среде PSpice представленные на рис. 3, показывают, что дрейф нуля ИУ при изменении его дифференциального коэффициента усиления сохраняет свое относительно низкое значение.

Например, в случае применения такого ИУ в микроконтроллерных системах с 12-разрядным аналого-цифровым преобразователем с опорным напряжением 2,5 В, указанная величина значительно меньше погрешности метода.

Одним из важных преимуществ анализируемого ИУ является возможность цифрового управления его дифференциальным коэффициентом усиления (рис. 4) посредством высокотехнологичной матрицы R-2R, что обеспечивает его интеграцию в традиционные порты ввода/вывода СнК. При этом, в отличие от традиционных подходов, здесь необходимо использовать резистивные матрицы R-2R в режиме масштабирования тока (рис. 5) [4].

Реализуемый дифференциальный коэффициент усиления определяется соотношением:

$$K_d = \frac{1}{\sum_{i=1}^n \alpha_i / 2^i} \quad (4)$$

Другие параметры схемы сохраняются и определяются соотношениями (1), (3).

Таким образом, полученная схема может быть использована при создании достаточно распространенных датчиков мостового типа вместо традиционных ИУ на трех ОУ, а также в энергоэкономичных портах ввода аналоговых сигналов современных смешанных СнК.

Результаты проектирования инструментального усилителя показывают, что обеспечение его высоких качественных показателей не связано с использованием прецизионных резисторов. Точность резисторов обуславливает только погрешность реализации дифференциального коэффициента усиления и слабо влияет на глубину ослабления синфазного напряжения и дрейф нуля. Именно это свойство предложенных схем позволяет существенно ослабить требования к технологии их изготовления. Кроме этого, использование техники радиационно-стойкого АБМК для реализации специализированных интегральных схем с такими инструментальными усилителями позволяет создать целый класс датчиков, обеспечивающих измерение параметров в условиях спецвоздействий.

Статья подготовлена в рамках госконтракта № П507 ФЦП «Научные и научно-педагогические кадры инновационной России (2009-2013 гг.)».

СПИСОК ЛИТЕРАТУРЫ

1. Крутчинский, С.Г. Структурная оптимизация дифференциальных каскадов [Текст]/С.Г. Крутчинский, А.В. Нефедова//Изв. ЮФУ. Сер. Технические науки. –2008.–№ 7.–С. 41–48.
2. Дворников, О.В. Аналоговый биполярно-полевой БМК с расширенными функциональными возможностями [Текст]/О.В. Дворников, В.А. Чеховский//Chip News. –1999.–№ 2.–С. 21–23.
3. Прокопенко, Н.Н. Оптимизация статических режимов в биполярных ИС [Текст]/Н.Н. Прокопенко//МЭС-2010: Сб. тр.–М., 2010.
4. Крутчинский, С.Г. Структурный синтез аналоговых электронных схем [Текст]/С.Г. Крутчинский // СКНЦ ВШ.–Ростов-на-Дону, 2001.–С. 34–51.

УДК 621.37

Н.Н. Прокопенко, В.Г. Манжула, С.С. Белич

МУЛЬТИДИФФЕРЕНЦИАЛЬНЫЙ ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ С МАЛЫМ НАПРЯЖЕНИЕМ СМЕЩЕНИЯ НУЛЯ В УСЛОВИЯХ ТЕМПЕРАТУРНЫХ И РАДИАЦИОННЫХ ВОЗДЕЙСТВИЙ

Дифференциальные операционные усилители (ОУ) с несимметричным включением активной нагрузки (токового зеркала), обеспечивающей непосредственное управление двухтактным буферным усилителем (БУ) находят широкое применение в современной радиоэлектронной аппаратуре [1, 2]. Такие ОУ обладают важными качествами: создают так называемый, «rail-to-rail выход» («от шины питания до шины питания») и имеют предельно простую структуру.

Существенный недостаток ОУ данного класса [3, 4] состоит в том, что он имеет повышенное значение систематической составляющей напряжения смещения нуля ($U_{см}$), зависящей от погрешности передачи по току (K_i) применяемого токового зеркала. Особенно существенной эта погрешность получается при использовании в ка-

честве токового зеркала простейших, но наиболее высокочастотных, схемотехнических решений, для которых $K_i \neq 1$.

Предлагаемое техническое решение [5] (рис.1) обеспечивает уменьшение абсолютного значения $U_{см}$, а также его температурного и радиационного дрейфа при использовании в схеме токовых зеркал, имеющих коэффициент передачи по току, не равный единице $K_i \neq 1$. Такое значение K_i характерно для многих классических токовых зеркал. В ряде случаев в предлагаемом ОУ могут использоваться входы второго входного дифференциального каскада (Вх.⁽⁻⁾³, Вх.⁽⁺⁾⁴), что позволяет реализовать на его основе так называемые «мультидифференциальные» ОУ, имеющие большие перспективы применения в микросхемотехнике.

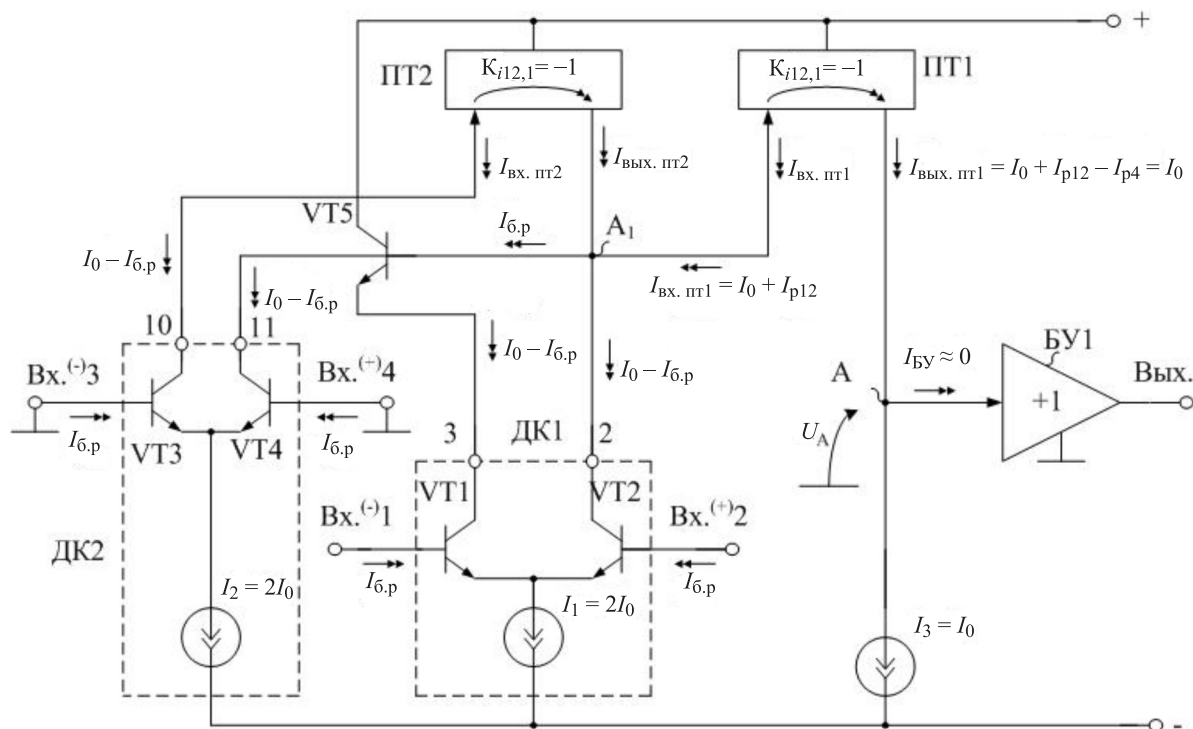


Рис. 1. Структура мультидифференциального ОУ с малым смещением нуля



Рассмотрим факторы, определяющие систематическую составляющую $U_{см}$, т. е. зависящие от схемотехники ОУ (рис. 1). Если величины токов I_1 и I_2 первого (ДК1) и второго (ДК2) входных дифференциальных каскадов равны $2I_0$, а ток источника опорного тока I_3 – величине I_0 , то токи коллекторов транзисторов VT1, VT2, VT3, VT4 первого и второго входных дифференциальных каскадов определяются соотношением:

$$I_{к1} = I_{к2} = I_{к3} = I_{к4} = I_0 - I_{б.р}, \quad (1)$$

где $I_{б.р} = I_0/\beta_i$ – ток базы i -го $n-p-n$ транзистора схемы при токе эмиттера $I_3 = I_0$, β_i – коэффициент усиления по току базы i -го транзистора.

Выходной ток повторителя тока ПТ2 (или так называемого «токового зеркала») определяется как

$$I_{вых. пт2} = I_0 - I_{б.р} - I_{р. пт2}, \quad (2)$$

где $I_{р. пт2}$ – разность токов на выходе и входе токового зеркала ПТ2, зависящая от его схемотехники.

Поэтому на основании закона Кирхгофа находим, что в узле A_1 выполняется следующее токовое соотношение:

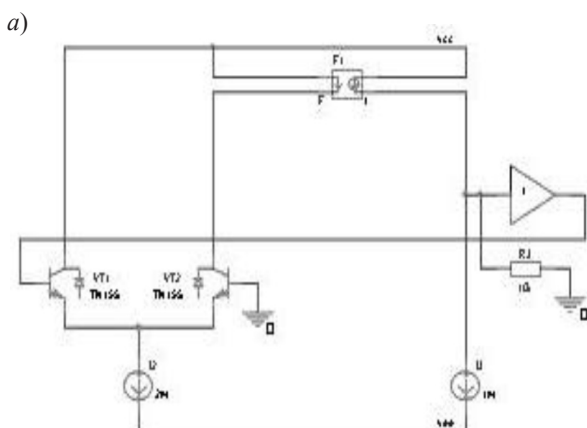
$$I_{вых. пт2} + I_{вх. пт1} = I_{к2} + I_{к4} + I_{б.р}. \quad (3)$$

Как следствие, токи на входе и выходе токового зеркала ПТ1 определяются по формулам:

$$I_{вх. пт1} = I_0 + I_{р. пт2}, \quad (4)$$

$$I_{вых. пт1} = I_0 + I_{р. пт1} - I_{р. пт2}, \quad (5)$$

где $I_{р. пт1}$ – разность токов на входе и выходе токового зеркала ПТ1, зависящая от его схемотехники.



Так как токовые зеркала ПТ1 и ПТ2 выполнены с использованием идентичных схемотехнических решений, то разность их токов ошибки I_p равна нулю. Поэтому разностный ток I_Σ в узле «А» при $I_{бу} \approx 0$, в случае его замыкания на некоторый эквивалентный источник напряжения E_A , равный статическому напряжению U_A в схеме ОУ с обратной связью, будет равен нулю:

$$I_\Sigma = I_{вых. пт1} - I_3 = 0. \quad (6)$$

Таким образом, в предлагаемом ОУ при выполнении условия (6) уменьшается систематическая составляющая $U_{см}$, обусловленная конечной величиной β транзисторов и его радиационной (или температурной) зависимостью. Как следствие, это уменьшает $U_{см}$, т. к. разностный ток I_Σ в узле «А» создает $U_{см}$, зависящее от крутизны S преобразования входного дифференциального напряжения $u_{вх}$ в разностный ток узла «А»:

$$S = \frac{i_\Sigma}{u_{вх}} = \frac{1}{r_{31} + r_{32}}, \quad (7)$$

где $r_{31} = r_{32}$ – сопротивления эмиттерных переходов входных транзисторов VT1 и VT2 входного дифференциального каскада ДК1.

Поэтому для схемы (см. рис. 1) напряжение смещения нуля для входов $Vx1$ и $Vx2$:

$$U_{см} \approx I_\Sigma(r_{31} + r_{32}) = I_\Sigma \left(\frac{2\varphi_T}{I_0} \right) \approx 0, \quad (8)$$

где $\varphi_T = 26$ мВ – температурный потенциал.

В классическом ОУ $I_\Sigma \neq 0$. Поэтому здесь систематическая составляющая $U_{см}$ получается как минимум на порядок больше, чем в предлагаемой схеме.

На рис. 2 показаны схемы классического и предлагаемого ОУ в среде компьютерного моделирования PSpice

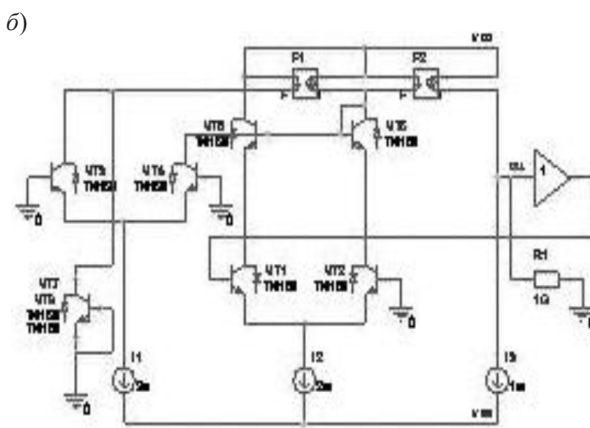


Рис. 2. Схемы классического (а) и предлагаемого (б) ОУ в среде компьютерного моделирования PSpice

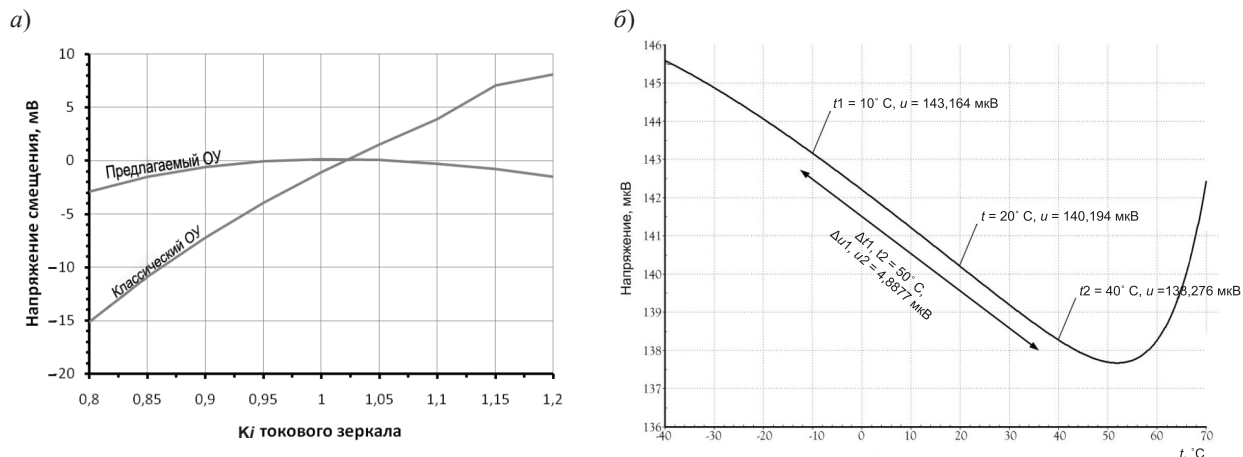


Рис. 3. Зависимости напряжения смещения сравниваемых схем от коэффициента передачи по току K_i токовых зеркал (а) и температурная зависимость напряжения смещения нуля ОУ (б)

лирования PSpice на моделях интегральных транзисторов ФГУП НПП «Пульсар».

На рис. 3, а представлена зависимость напряжения смещения нуля схем (см. рис.2) от коэффициента передачи по току K_i токовых зеркал ПТ1 и ПТ2 ($K_i = K_{i12,1} = K_{i12,2}$), а на рис. 3, б температурная зависимость модуля напряжения смещения нуля этих схем.

Компьютерное моделирование схем показывает, что несмотря на существенное уменьше-

ние β транзисторов вследствие радиационных и тепловых воздействий, предлагаемый мультидифференциальный ОУ и в этих условиях обладает низким напряжением смещения нуля (около 140 мкВ), дрейф которого составляет 5 мкВ в диапазоне 50 $^\circ\text{C}$. Это значительно меньше, чем у классической схемы.

Статья подготовлена в рамках госконтракта П507 от 05.08.2009 г. ФЦП «Научные и научно-педагогические кадры инновационной России (2009–2013 гг.)».

СПИСОК ЛИТЕРАТУРЫ

1. Патент US № 4.366.442 fig. 2. Current U.S. Class: 330/51 ; 330/254; 330/255; 330/265; 330/267. Amplifier with muting circuit / Yamada; Hisashi : Assignee Tokyo Shibaura Denki Kabushiki Kaisha. –Dec. 28, 1982
2. Патентная заявка US 20070152753 fig. 5c. Current U.S. Class: 330/261. System and method for clamping a differential amplifier / Swanson; Leland Scott; Assignee: Texas Instruments Incorporated. –July 5, 2007
3. Прокопенко, Н.Н. Архитектура и схемотехника быстродействующих операционных уси-

лителей: монография [Текст] / Н.Н.Прокопенко, А.С. Будяков. –Шахты: Изд-во ЮРГУЭС, 2006. –231 с.

4. Полонников, Д.Е. Операционные усилители: Принципы построения, теория, схемотехника: монография [Текст] / Д.Е. Полонников. –М., 1983.–216 с.

5. Дифференциальный операционный усилитель с малым напряжением смещения нуля: заявка на патент РФ, МПК: H03F 3/45, H03F 3/34 [Текст] / Прокопенко Н.Н., Манжула В.Г., Морозов С.А. –№ 2009142660/09; заявл. 18.11.2009.

УДК 621.37

А.С. Будяков, Н.Н. Прокопенко, В.Г. Манжула

АНАЛИЗ ЧАСТОТНЫХ ХАРАКТЕРИСТИК ВХОДНЫХ КАСКАДОВ СВЧ ОПЕРАЦИОННЫХ УСИЛИТЕЛЕЙ ДЛЯ СЛОЖНЫХ ФУНКЦИОНАЛЬНЫХ БЛОКОВ РАДИАЦИОННО-СТОЙКОЙ АППАРАТУРЫ НОВОГО ПОКОЛЕНИЯ

Операционные усилители (ОУ) в последнее время стали часто использоваться при построении высококачественных сложных функциональных (СФ) блоков систем связи с гигагерцовыми рабочими частотами [1]. Возможности современных технологий с транзисторами на основе гетеропереходов кремний–германий (SiGe ГБТ) позволяют строить СВЧ ОУ с частотами единичного усиления единицы–десятки ГГц [2]. Однако СВЧ ОУ не получили достаточно широкого распространения вследствие повышенного влияния паразитных параметров корпуса и высокой стоимости изделий. Построение СВЧ ОУ как СФ блока, системы на кристалле [3] создает предпосылки обеспечения многофункциональной элементной базой систем связи диапазона частот в несколько ГГц при существующем на сегодняшний день уровне развития SiGe технологий.

Рассмотрим три основные схемы (рис. 1), как базовые варианты для построения однока-

скадного ОУ с предельной частотой единичного усиления. Для обеспечения устойчивости ОУ в схеме со 100-процентной обратной связью необходимо, чтобы частота второго полюса f_2 была выше частоты единичного усиления ОУ f_{cp} (для случая двухполюсной частотной характеристики разомкнутого коэффициента усиления ОУ). Так, например, для формирования частотной характеристики петлевого усиления разомкнутого ОУ по Баттерворту с запасом по фазе 60° необходимо обеспечить $f_2/f_{cp} \approx 2$. Отсюда следует, что для сравнения частотных характеристик схем корректно провести анализ расположения полюсов схем ОУ.

Частота расположения первого полюса для схем на рис. 1 б и в определяется одним и тем же выражением:

$$\omega_{1,2} = \omega_{1,3} = \frac{1}{R_{\text{вых}} C_{\text{вых}}}, \quad (1)$$

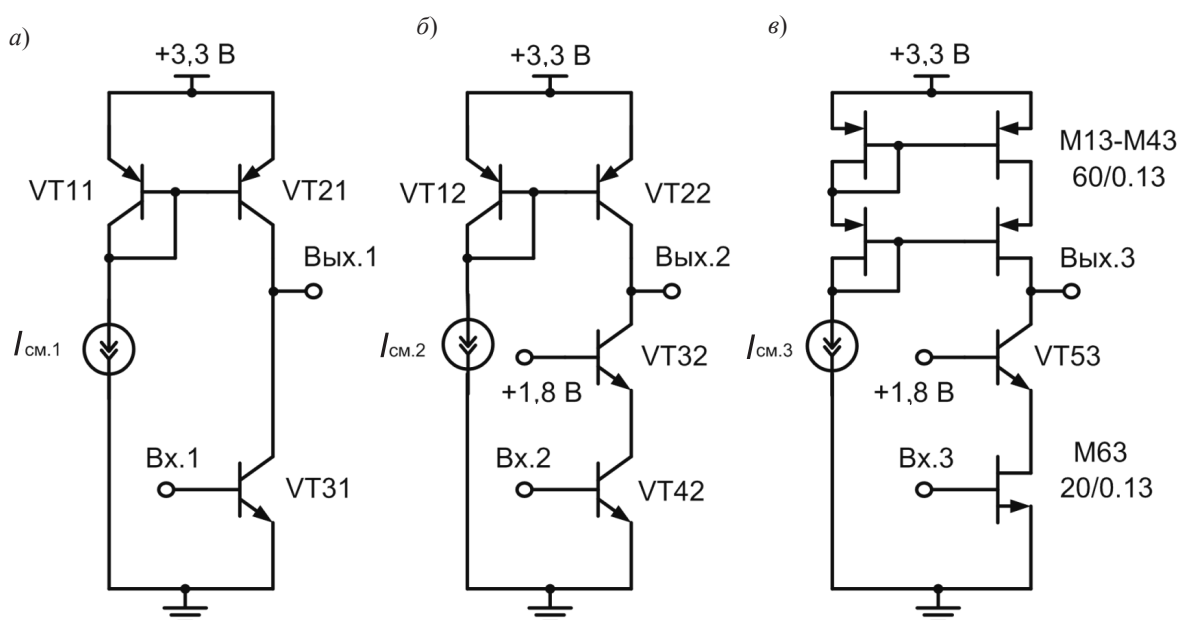


Рис. 1. Упрощенные схемы перспективных ОУ

где $\omega_{1,2}$, $\omega_{1,3}$ – частота первого полюса для схем на рисунках б и в, соответственно; $R_{\text{вых}}$ – эквивалентное активное сопротивление на выходе соответствующей схемы; $C_{\text{вых}}$ – эквивалентная выходная емкость на выходе соответствующей схемы. Для схемы на рис. а частота первого полюса определяется также постоянной времени в цепи базы входного транзистора:

$$\omega_{1,1} = \frac{1}{R_c [C_{\text{эб31}} + C_{\text{кб31}} (1 + K_{y,1})] + R_{\text{вых}} C_{\text{вых}}}, \quad (2)$$

где $K_{y,1} = g_{m31} R_{\text{вых}}$ – коэффициент усиления на постоянном токе схемы; R_c – выходное активное сопротивление источника сигнала; $C_{\text{эб}}$, $C_{\text{кб}}$, $C_{\text{кп}}$ – паразитные емкости между соответствующими выводами транзистора; g_m – малосигнальная крутизна соответствующего транзистора.

Второй ближайший полюс схемы на рис. а образован выходным сопротивлением источника сигнала (R_c), паразитными сопротивлениями транзистора (r_6 – объемное сопротивление базы транзистора) и паразитной эквивалентной входной емкостью схемы:

$$\omega_{2,1} = \frac{1}{(R_c + r_6) \cdot C_{\text{вх.1}} + \frac{C_{\text{эб31}} \cdot C_{\text{кб31}} + (C_{\text{эб31}} + C_{\text{кб31}}) \cdot C_{\text{вых}}}{g_{m31} \cdot C_{\text{кб31}}}}. \quad (3)$$

Если постоянная времени на входе схемы имеет достаточно малое значение, то преобладать будет постоянная времени, обусловленная крутизной g_{m31} и соответствующими емкостями в схеме.

В схеме на рис. б существенное влияние оказывает также постоянная времени, образованная паразитной емкостью в цепи эмиттера VT32:

$$\omega_{2,2} = \frac{1}{(R_c + r_6) \cdot C_{\text{вх.2}} + \frac{1}{g_{m32}} (C_{\text{кб42}} + C_{\text{кп42}} + C_{\text{эб32}})}. \quad (4)$$

Частота второго полюса в схеме на рис. в определяется как

$$\omega_{2,3} = \frac{1}{(R_c + r_{\text{эб3}}) \cdot C_{\text{вх.3}}}, \quad (5)$$

где $C_{\text{вх.1}} = C_{\text{эб31}} + [1 + K_y(p)] \cdot C_{\text{кб31}}$; $C_{\text{вх.2}} = C_{\text{эб41}} + C_{\text{кб42}} \left(\frac{g_{m42}}{g_{m32}} \right)$; $C_{\text{вх.3}} = C_{\text{эб3}} + C_{\text{эб3}}$ – эквивалентные входные емкости схем на рис. 1, а, б и в соответственно; $r_{\text{эб3}}$ – сопротивление затвора M63.

Для минимизации постоянной времени второго полюса в схемах на рис. 1 следует уменьшать

сопротивления на входе и паразитные входные емкости.

Проведем сравнение входной емкости схем рис. 1 вблизи частоты единичного усиления ОУ. Эта емкость определяет расположение второго полюса в схемах, а значит и запас по фазе.

В схеме на рис. а входная паразитная емкость складывается из емкости эмиттер-база и эквивалентной емкости коллектор-база VT31:

$$C_{\text{вх.1}} = C_{\text{эб31}} + [1 + K_y(p)] \cdot C_{\text{кб31}}, \quad (6)$$

где K_y – коэффициент усиления замкнутого ОУ. Емкость эмиттер-база биполярного транзистора состоит из диффузионной емкости (C_{dif}) и емкости обедненной области p - n перехода эмиттер-база (C_{dep}):

$$C_{\text{эб31}} = C_{\text{dif}} + C_{\text{dep}}. \quad (7)$$

Как известно, если транзистор работает не в микрорежиме, то практически соблюдается соотношение:

$$C_{\text{dif}} \gg C_{\text{dep}}. \quad (8)$$

При этом емкость эмиттер-база биполярного транзистора определяется приближенным выражением:

$$C_{\text{эб}} \approx C_{\text{dif}} = \tau_6 \frac{I_{\text{э}}}{\Phi_T}, \quad (9)$$

где $I_{\text{э}}$ – статический ток эмиттера; $\Phi_T \approx 26$ мВ – температурный потенциал.

Заметим, что величина $|K_y(j\omega)|$ на частоте второго полюса обычно достаточно мала. Поэтому основная составляющая входной емкости – это диффузионная емкость эмиттер-база входного транзистора C_{dif} и в первом приближении можно считать:

$$C_{\text{вх.1}} \approx C_{\text{эб31}} \approx C_{\text{dif}} = \tau_6 \frac{I_{\text{э.31}}}{\Phi_T}. \quad (10)$$

В схеме на рис. б входная емкость определяется аналогично (10), т. к. эффект Миллера ослаблен применением каскодной схемы и доминирующей емкостью является C_{dif} :

$$C_{\text{вх.2}} \approx \tau_6 \frac{I_{\text{э.42}}}{\Phi_T}. \quad (11)$$

В схеме на рис. в эффектом Миллера в транзисторе M63 можно пренебречь, поэтому входная емкость определяется выражением:

$$C_{\text{вх.3}} = C_{\text{зи63}} + C_{\text{зс63}}. \quad (12)$$

В результате проведенного анализа можно сделать следующие выводы: частота второго полюса в схеме на рис. *а* в первую очередь определяется постоянной времени на входе схемы. Повышать частоту второго полюса предпочтительнее за счет уменьшения входной емкости схемы. Входная емкость схем на рис. *а* и *б* определяется диффузионной емкостью эмиттер-база и зависит от тока эмиттера входного транзистора. Входная емкость схемы на рис. *в* не зависит от режима и определяется паразитными емкостями входного МОП транзистора.

Устойчивость рассматриваемых схем ОУ определяется, в основном, входной эквивалентной емкостью. В схемах на рис. *а* и *б* следует также учитывать эффекты более высоких порядков. Значительное влияние эффектов высших порядков в схемах (см. рис. 1) затрудняет их анализ, поэтому предпочтительнее провести моделирование частотных характеристик. Необходимо оценить входную емкость биполярного и полевого транзистора, чтобы понять, какой тип транзисторов следует использовать во входном каскаде. Исходя из результатов моделирования, можно более точно определить зависимость запаса по фазе от параметров схемы.

Для определения входной емкости на частоте второго полюса было проведено моделирование

с помощью схемы на рис. 2 (элементы C0-L0, C1-L1 предназначены для установления статического режима и не влияют на динамические параметры схемы). Сравним эквивалентную входную емкость биполярного и полевого МОП транзисторов (рис. 3). Размеры и статический ток стока МОП транзистора M0 были выбраны в соответствии с рекомендациями [2].

Плотность тока стока на мкм ширины затвора МОП составляет 0,25мА/мкм. Успех применения рекомендаций [2] заключается в минимизации входной емкости за счет выбора малых размеров МОП транзистора, за что, однако, приходится платить повышенными значениями статического тока для получения достаточного коэффициента усиления на постоянном токе и частоты единичного усиления (UGB). Биполярный транзистор Q0 имеет максимум частоты единичного усиления $f_t = 170$ ГГц при токе коллектора 4 мА.

Представляет интерес найти ток коллектора биполярного транзистора, при котором его входная емкость равна входной емкости МОП транзистора. Входная емкость транзисторов в схеме на рис. 2 была промоделирована как

$$C_{\text{вх}} = \frac{1}{2\pi f \cdot \Im(Z_{\text{вх}})},$$

где $\Im(Z_{\text{вх}})$ – мнимая часть входного импеданса транзистора; f – частота источника сигнала.

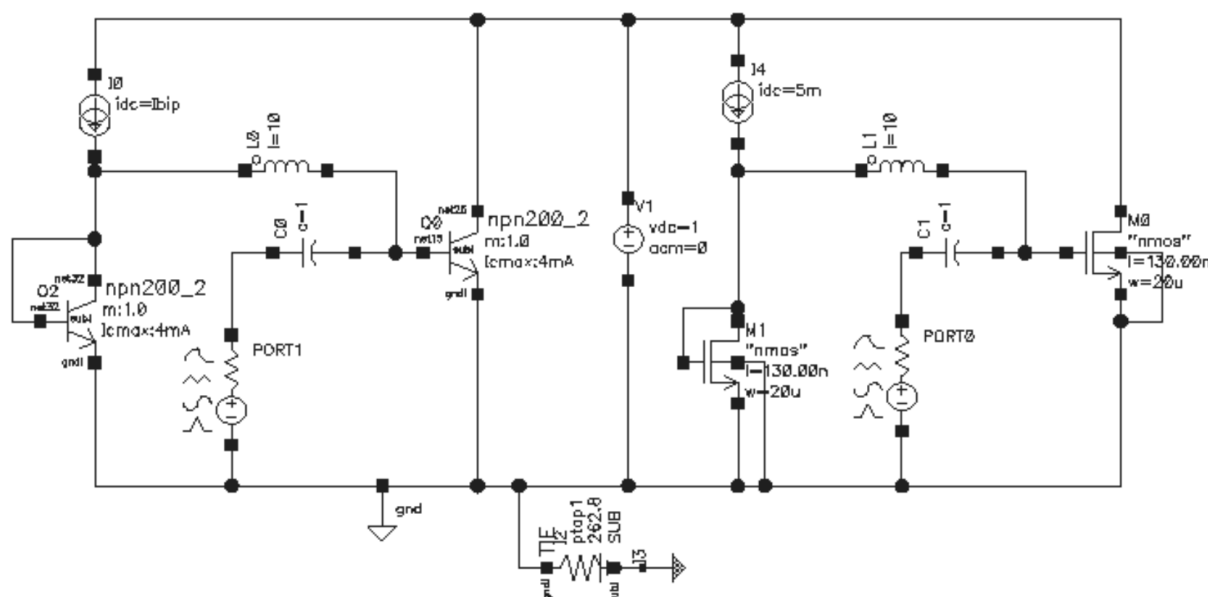


Рис. 2. Схема измерения входной емкости биполярного и полевого транзисторов

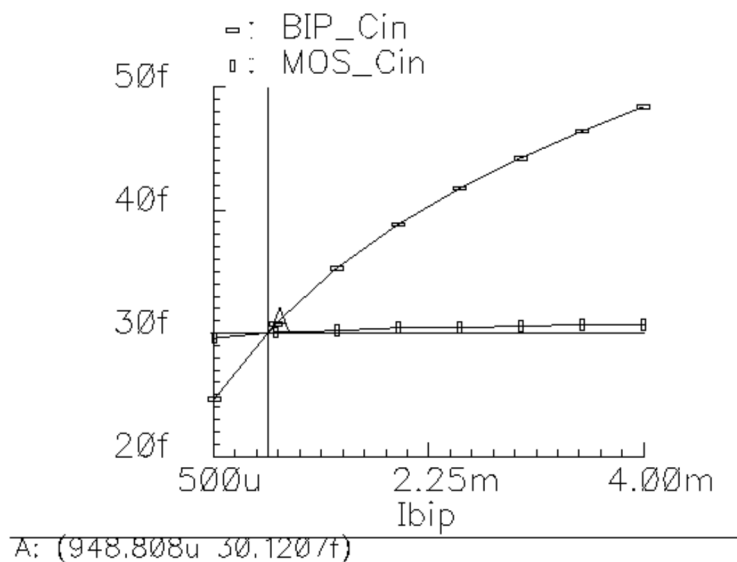
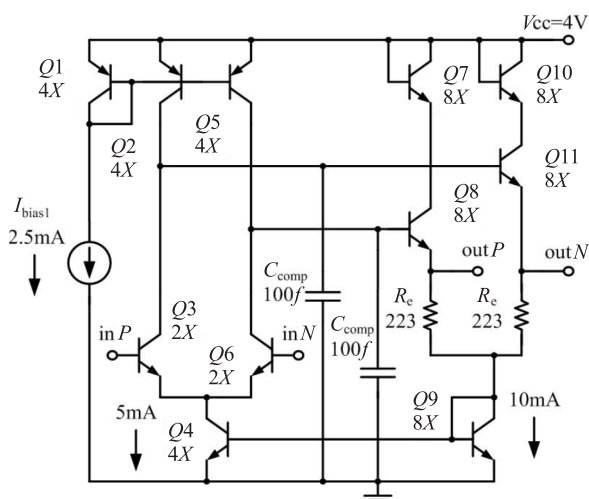


Рис. 3. Зависимость входной емкости биполярного и МОП транзисторов от статического тока коллектора (стока)

Как показывают результаты моделирования схемы на рис. 2, при токе коллектора 1 мА входная емкость биполярного транзистора равна входной емкости МОП транзистора (рис. 3). Отсюда следует, что при токе коллектора менее 1 мА схемы на рис. 1, а и б могут иметь более высокую частоту второго полюса, чем схема на рис. 1, в. Кроме того, схема на рис. 1, а может иметь более высокое значение запаса по фазе за счет меньшей «электрической длины».

Разработанные выше рекомендации и методики проектирования каскадов были положены в основу построения практических схем СВЧ ОУ (рис. 4) на базе топологии блока кристаллов 0_T214_bench_WW, изготовленных на фабрике Института ИНР (Германия) по технологии SG25H2. К схемотехническим особенностям ОУ можно отнести построение отрицательной обратной связи по синфазному сигналу на основе токового зеркала Q4, Q9, которая обеспечивает

а)



б)

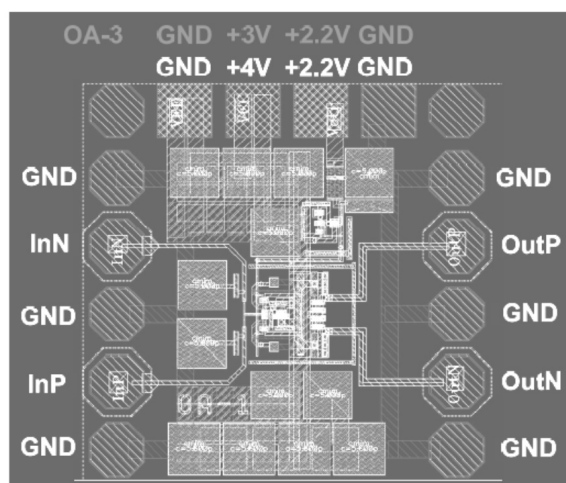


Рис. 4. Схема (а) и топология (б) кристаллов СВЧ ОУ

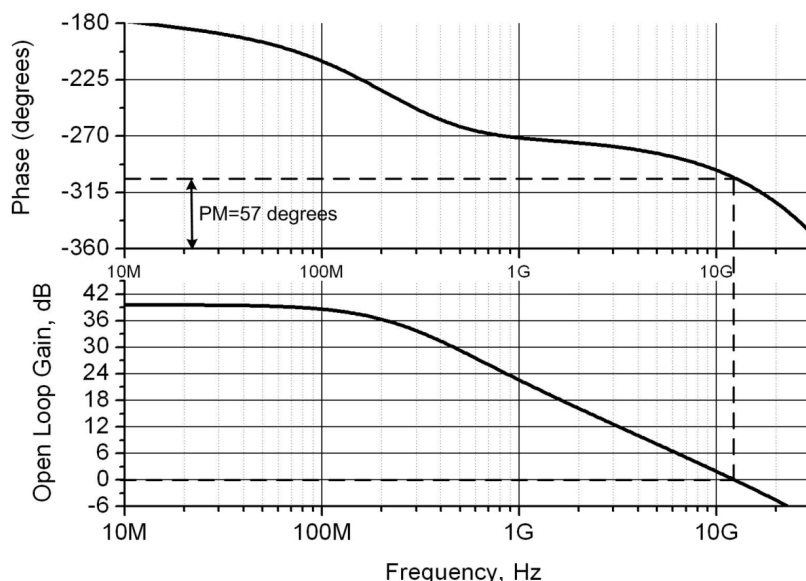


Рис. 5. Частотные характеристики СВЧ ОУ

стабилизацию статического режима и минимизирует влияние выходной проводимости Q4 на коэффициент ослабления входной синфазной составляющей.

Схема ОУ (рис. 4, а) обладает запасом по фазе 57° , частотой единичного усиления по однофазному выходу – 12,3 ГГц, тогда как дифференциальное значение соответствует 23 ГГц. Площадь топологии (рис. 4, б) составляет 110 мкм × 130 мкм без контактных площадок. Частотные характеристики данного ОУ приведены на рис. 5.

В таблице представлены основные параметры ОУ, построенного на основе схемотехнического и топологического решения, приведенного на рис. 4.

Таким образом, проведенный анализ частотных характеристик вариантов входных каскадов позволил сформировать рекомендации и разработать методики проектирования СВЧ ОУ для гигагерцового диапазона на основе SiGe технологий.

Статья подготовлена в рамках госконтракта № П507 ФЦП «Научные и научно-педагогические кадры инновационной России (2009-2013 гг.)».

Параметры СВЧ ОУ по одному выходу

Название СВЧ ОУ	Один выход, $f_{ср}$, ГГц	Дифференциальный выход, $f_{ср}$, ГГц	K_y (разомкнутый), дБ	Запас по фазе, градусы	Входная точка компрессии, дБм	Выходная точка компрессии, дБм	Коэффициент шума, дБ	Напряжение питания, В	Потребляемый ток, мА
Условия измерения	Уровень 0 дБ	Уровень -6 дБ	$R_H=50$ Ом	Один выход	$f_c=2$ ГГц	$f_c=2$ ГГц	$f_c=2$ ГГц		
ccfbiproa_v2 (OA-1)	12,3	23	38,4	57	-19,2	-3,8	19	4	17,5

СПИСОК ЛИТЕРАТУРЫ

1. **Budyakov, A.** Design of Fully Differential OpAmps for GHz Range Applications [Текст]/А. Budyakov, K. Schmalz, N. Prokopenko [et al.]//Проблемы современной аналоговой микросхемотехники : сб. матер. Междунар. науч.-практ. семинара. Ч. 1 / Шахты: Изд-во ЮРГУЭС, 2007.–С. 106–110.

2. **Voinigescu, S.P.** Design Methodology and Applications of SiGe BiCMOS Cascode Opamps with up to 37-GHz Unity Gain Bandwidth [Текст]/ S.P. Voinigescu [et al.]//IEEE CSICS, Techn. Digest. –Nov. 2005. –Р. 283–286.

3. **Немудров, В.Г.** Системы на кристалле. Проектирование и развитие [Текст]/ В.Г. Немудров, Г. Мартин. – М.: Техносфера, 2004. – 216 с.

НАУЧНЫЙ СЕМИНАР СПбГПУ

В 2010 г. после длительного перерыва возобновил работу научный семинар факультета технической кибернетики. Семинар предполагается проводить регулярно, 3-4 раза в семестр, с публикацией в журнале «Научно-технические ведомости СПбГПУ» тезисов или кратких аннотаций выступлений. На одном из первых заседаний был представлен и обсужден доклад профессора Г.Н. Черкесова и аспиранта В.В. Чуркина. Краткая аннотация доклада приводится ниже.

Очередное заседание семинара назначено на сентябрь 2010 г. Планируется выступление профессора Котлярова. Следите за объявлениями на сайте факультета (<http://ftk.spbstu.ru/>).

Г.Н. Черкесов, В.В. Чуркин

О ПРОБЛЕМЕ РАСЧЕТА НАДЕЖНОСТИ ВОССТАНАВЛИВАЕМЫХ СИСТЕМ ПРИ НАЛИЧИИ ЗАПАСНЫХ ЭЛЕМЕНТОВ

В докладе рассматривается проблема расчета надежности систем в условиях, когда альтернативой ремонту является использование комплектов запасного имущества и принадлежностей (ЗИП), и восстановление работоспособности происходит путем замены отказавшего модуля (сменной части) на работоспособную запасную часть (ЗЧ).

Используемые в настоящее время в промышленности методики расчета надежности восстанавливаемых систем при наличии ЗИП являются приближенными, имеющими знакопеременную погрешность при оценке показателей надежности, значение которой не известно как по знаку, так и по абсолютной величине.

Оценить погрешность можно только с помощью точных формул, разработанных в статьях авторов доклада. Выборочная проверка точности показывает, что имеется область применения приближенных методик, в которой ошибка является приемлемой. Например, погрешность приемлема при оценке любых ПН нерезервированных систем, при оценке показателей безотказности однородных дублированных систем при периодическом пополнении с экстренными поставками (модели П2 и ПЭД2).

Вместе с тем, имеются области применения, где погрешность достигает сотен и тысяч процентов: системы с групповым дублированием (моде-

ли П4), дублированные системы (модели П2, П4, ПЭД2 при оценке показателей готовности).

Существуют области, вообще не охваченные приближенными методиками: неоднородные резервированные структуры, многоуровневые однородные структуры последовательно-параллельного типа, системы со сложной структурой («мостиковые» структуры), многофункциональные структуры в полном объеме.

Широкое распространение ЗИП как способа восстановления работоспособности, большая практическая значимость оценок надежности систем с ЗИП, тяжелые последствия оценок показателей надежности с погрешностью, исчисляемой десятками, сотнями и даже тысячами процентов, требуют разработки нового раздела в теории надежности: теории надежности систем с восстановлением при помощи ЗИП.

Идея построения этого раздела состоит в прямом включении ресурсов ЗИП в модели надежности. Полученные в настоящее время результаты доказывают возможность такой разработки. Однако эта работа находится в начальной стадии и требует привлечения значительных научных сил. Цель доклада – обратить внимание слушателей на данную проблему, что может создать благоприятные условия для ускоренной разработки теории надежности систем при наличии ЗИП.



СВЕДЕНИЯ ОБ АВТОРАХ

КОНТАКТНЫЕ ДАННЫЕ

АРТАМОНОВ Александр Витальевич – сотрудник научно-образовательного центра «LG-Политехник» Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29

АТИК Сафван – адъюнкт Военной академии связи.

194064, Санкт-Петербург, Тихорецкий пр., д. 3, Тел. (812) 556-94-92, e-mail: arien@bk.ru

БАБКОВА Елена Васильевна – доцент кафедры вычислительной математики и кибернетики Уфимского государственного авиационного технического университета, кандидат технических наук.

450000, г. Уфа, ул. Карла Маркса, д. 12. Тел. (8-347) 273-79-67, e-mail: evressim@yandex.ru

БЕЛИЧ Сергей Сергеевич – студент кафедры информационных систем и радиотехники Южно-Российского государственного университета экономики и сервиса.

346500, г. Шахты Ростовской обл., ул. Шевченко, д. 147. Тел. (8636) 22-61-30

БЕЛЯЕВ Алексей Борисович – студент кафедры распределенных вычислений и компьютерных сетей Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 21. E-mail: belyaevab@gmail.com

БЛЕКАНОВ Иван Станиславович – ассистент кафедры технологии программирования Санкт-Петербургского государственного университета, аспирант.

198504, Санкт-Петербург, Петергоф, Университетский пр., д. 35. E-mail: a_vanes@mail.ru

БОНДАРЕНКО Дмитрий Сергеевич – выпускник Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29

БУДЯКОВ Алексей Сергеевич – старший научный сотрудник ФГУП НПП «ПУЛЬСАР», кандидат технических наук.

346500, г. Шахты Ростовской обл., ул. Шевченко, д. 147. Тел. (8636) 22-61-30

ВАСИЛЬЕВ Алексей Евгеньевич – доцент кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Санкт-Петербург, ул. Политехническая, д. 29. Тел. (812) 297-42-18, e-mail: avasil@aivt.ftk.spbstu.ru

ВАСИЛЬЕВ Павел Николаевич – студент кафедры информационной безопасности компьютерных систем Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29. E-mail: vipn@mail.ru

ВЛАДИМИРОВ Сергей Сергеевич – аспирант кафедры обработки и передачи дискретных сообщений Санкт-Петербургского государственного университета телекоммуникаций.

191186, Санкт-Петербург, наб. р. Мойки, д. 61. Тел. (812)-585-71-22, e-mail: vss.vlcom@gmail.com

ГЕТМАНСКИЙ Виктор Викторович – аспирант кафедры высшей математики Волгоградского государственного технического университета.

400131, г. Волгоград, пр. Ленина, д. 28. Тел. (8442) 23-00-76, e-mail: efim.sergeev@gmail.com

ГЛУХИХ Михаил Игоревич – доцент кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

194021, Санкт-Петербург, ул. Политехническая, д. 21. Тел. (812) 297-22-38

ГОРОБЦОВ Александр Сергеевич – профессор кафедры высшей математики Волгоградского государственного технического университета, доктор технических наук.

400131, г. Волгоград, пр. Ленина, д. 28. Тел. (8442) 23-00-76, e-mail: gorobtsov@avtlg.ru

ЕЛИНА Татьяна Николаевна – доцент кафедры информационных систем и технологий Норильского индустриального института, кандидат экономических наук.

663310, г. Норильск, ул. 50-лет Октября, д. 7. Тел. (3919) 42-17-39, e-mail: mva_etn@mail.ru

ЕЛИСЕЕВ Дмитрий Сергеевич – студент кафедры прикладной математики Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29

ЕРЕМЕЕВ Игорь Юрьевич – начальник кафедры комплексного радиоэлектронного контроля космических комплексов Военно-космической академии имени А.Ф. Можайского, кандидат технических наук, доцент.

197082, Санкт-Петербург, ул. Ждановская, д. 13. Тел. (812) 347-95-57

ЖУРАВЛЕВА Любовь Михайловна – доцент кафедры радиотехники и электросвязи Московского государственного университета путей сообщения.

127994, Москва, ул. Образцова, д. 9, стр. 9. Тел. (495) 684-24-39

ЗАМАРИН Александр Иванович – профессор кафедры комплексного радиоэлектронного контроля космических комплексов Военно-космической академии имени А.Ф. Можайского, доктор технических наук.

197082, Санкт-Петербург, ул. Ждановская, д. 13. Тел. (812) 347-95-57.

ЗУБКОВ Александр Фёдорович – декан Института промышленной экономики, информатики и сервиса Пензенской государственной технологической академии, кандидат технических наук, профессор.

440605, г. Пенза, проезд Байдукова/ул. Гагарина, д. 1а/11. Тел. (8412)-49-59-80, e-mail: zubkov@pgta.ru

ИВАНКОВ Алексей Александрович – доцент кафедры прикладной математики Санкт-Петербургского государственного политехнического университета, кандидат физико-математических наук.

195251, Санкт-Петербург, ул. Политехническая, д. 29

ИЦЫКСОН Владимир Михайлович – доцент кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

194021, Санкт-Петербург, ул. Политехническая, д. 21. Тел. (812) 297-22-38, e-mail: vlad@ftk.spbstu.ru, itsykson@mail.ru

КОГНОВИЦКИЙ Олег Станиславович – профессор кафедры обработки и передачи дискретных сообщений Санкт-Петербургского государственного университета телекоммуникаций.

191186, Санкт-Петербург, наб. р. Мойки, д. 61. Тел. (812)-585-71-22, e-mail: kogn@yandex.ru

КОЧКАРОВ Азрет Ахматович – декан факультета информационных технологий Института гуманитарного образования и информационных технологий Карачаево-Черкесской государственной технологической академии, кандидат физико-математических наук.

369000, г. Черкесск, ул. Ставропольская, д. 36, корп. 5. Тел. (878)20-21-90, e-mail: Azret_kochkarov@mail.ru

КОЧКАРОВ Расул Ахматович – научный сотрудник Финансовой академии при Правительстве РФ.

369000, г. Черкесск, ул. Ставропольская, д. 36, корп. 5. Тел. (878)20-21-90

КРУТЧИНСКИЙ Сергей Георгиевич – профессор кафедры систем автоматического управления, доктор технических наук, профессор Таганрогского технологического института Южного федерального университета.

347928, г. Таганрог Ростовской обл., пер. Некрасовский, д. 44. Тел. (8634) 37-16-89, e-mail: sgkrutch@mail.ru

ЛИТВИНЧУК Александр Владимирович – аспирант кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29. Тел. (812) 297-42-18

МАКАРОВ Сергей Борисович – заведующий кафедрой радиоэлектронных средств защиты информации Санкт-Петербургского государственного политехнического университета, доктор технических наук, профессор.

195251, Санкт-Петербург, ул. Политехническая, д. 29. E-mail: makarov@cee.spbstu.ru

МАЛЬЧЕВСКИЙ Сергей Алексеевич – программист Норильского индустриального института.

663310, г. Норильск, ул. 50-лет Октября, д. 7. Тел. (3919) 47-39-13, e-mail: slam-norilsk@mail.ru

МАНЖУЛА Владимир Гаврилович – доцент кафедры информационных систем и радиотехники Южно-Российского государственного университета экономики и сервиса, кандидат технических наук.

346500, г. Шахты Ростовской обл., ул. Шевченко, д. 147. Тел. (8636) 22-61-30, manjula@sssu.ru



МАТВЕЕВ Владимир Александрович – доцент кафедры алгебры и геометрии Псковского государственного педагогического университета, кандидат физико-математических наук.

180000, г. Псков, пл. Ленина, д. 2. Тел. (8112) 25-23-18, факс (8112) 16-00-39, e-mail: matveev176@rambler.ru

МАХОВЕНКО Елена Борисовна – доцент кафедры информационной безопасности компьютерных систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Санкт-Петербург, ул. Политехническая, д. 29. Тел. (812) 552-76-32, e-mail: helen@ssl.stu.neva.ru

МЕДВЕДЕВ Андрей Викторович – доцент кафедры радиофизики Санкт-Петербургского государственного политехнического университета, кандидат физико-математических наук.

195251, Санкт-Петербург, ул. Политехническая, д. 29

МЕСЯГУТОВ Марат Артурович – аспирант кафедры математики Уфимского государственного авиационного технического университета.

450000, г. Уфа, ул. К. Маркса, д. 12. E-mail: marat.mesyagutov@googlemail.com

МОИСЕЕВ Михаил Юрьевич – старший преподаватель кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 21, ауд. 306. Тел. (812) 297-22-38, e-mail: mmickle@mail.ru

МОЛЕДУ Маурисио Филипе – аспирант кафедры распределенных интеллектуальных систем Санкт-Петербургского государственного политехнического университета.

195220, Санкт-Петербург, Гражданский пр., д. 28. E-mail: mauledoux@gmail.com

МОРОЗОВ Антон Игоревич – начальник бюро вычислительных технологий ЗФ ОАО «ГМК «Норильский Никель».

663310, г. Норильск, ул. 50-лет Октября, д. 7. Тел. (3919) 47-39-13, e-mail: morozov@mz.nk.nornik.ru

ПАШКОВСКИЙ Александр Владимирович – доцент кафедры информационных систем, электропривода и автоматики Невинномысского технологического института (филиал ГОУ ВПО СевКавГТУ) кандидат технических наук.

357100, г. Невинномысск, Ставропольский край, ул. Гагарина, д. 1. E-mail: alecsandr_607@rambler.ru

ПЕТРОВ Дмитрий Дмитриевич – студент кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29. Тел. (812) 297-42-18

ПИСАРЕВ Александр Иванович – заведующий кафедрой электропривода и автоматизации технологических процессов, кандидат технических наук, доцент.

663310, г. Норильск, ул. 50-лет Октября, д. 7. Тел. (3919) 42-17-39, e-mail: morozov@mz.nk.nornik.ru

ПОПКОВА Алла Алексеевна – старший преподаватель кафедры информационных систем и технологий Норильского индустриального института.

663310, г. Норильск, ул. 50-лет Октября, д. 7. Тел. (3919) 42-17-39, факс (3919) 42-17-41, e-mail: infalla@bk.ru

ПОСОВ Илья Александрович – ассистент кафедры высшей математики № 2 Санкт-Петербургского государственного электротехнического университета «ЛЭТИ».

197376, Санкт-Петербург, ул. Профессора Попова, д. 5. Тел. (812) 234-63-81, e-mail: iposov@gmail.com

ПРОКОПЕНКО Николай Николаевич – заведующий кафедрой информационных систем и радиотехники Южно-Российского государственного университета экономики и сервиса, доктор технических наук, профессор.

346500, г. Шахты Ростовской обл., ул. Шевченко, д. 147. Тел. (8636) 22-61-30

РАШИЧ Андрей Валерьевич – ассистент кафедры радиоэлектронных средств защиты информации Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Санкт-Петербург, ул. Политехническая, д. 29

РОСТОВ Николай Васильевич – доцент кафедры систем автоматического управления Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Санкт-Петербург, ул. Политехническая, д. 29

РУКШИН Сергей Евгеньевич – доцент кафедры математического анализа Российского государственного педагогического университета имени А.И. Герцена, кандидат физико-математических наук.

197376, Санкт-Петербург, ул. Профессора Попова, д. 5. Тел. (812) 234-63-81, e-mail: vliuser@gmail.com

СЕМЕНОВ Иван Александрович – студент Пензенской государственной технологической академии.

440605, г. Пенза, проезд Байдукова/ул. Гагарина, д. 1а/11. Тел. (8412) 49-59-80

СЕМЕНЮК Сергей Сергеевич – адъюнкт кафедры комплексного радиоэлектронного контроля космических комплексов Военно-космической академии имени А.Ф. Можайского.

197082, Санкт-Петербург, ул. Ждановская, д. 13. Тел. (812) 347-95-57

СЕРГЕЕВ Ефим Сергеевич – аспирант кафедры электронно-вычислительных машин и систем Волгоградского государственного технического университета.

400131, г. Волгоград, пр. Ленина, д. 28. Тел. (8442) 23-00-76, e-mail: efim.sergeev@gmail.com

СЕРЕБРЯКОВ Александр Игоревич – магистрант кафедры информационных систем и радиотехники Южно-Российского государственного университета экономики и сервиса.

346500, г. Шахты Ростовской обл., ул. Шевченко, д. 147. Тел. (8636) 22-61-30

ТИМОФЕЕВ Денис Анатольевич – магистрант кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета.

194021, Санкт-Петербург, ул. Политехническая, д. 21. Тел. (812) 297-22-38

ТИТОВ Алексей Евгеньевич – магистрант кафедры систем автоматического управления Таганрогского технологического института Южного федерального университета.

347928, г. Таганрог Ростовской обл., пер. Некрасовский, д. 44. Тел. (8634) 37-16-89, e-mail: sgkrutch@mail.ru

УЗДЕНОВ Ахмат Абдуллахович – ассистент кафедры математики Карачаево-Черкесской государственной технологической академии.

369000, г. Черкесск, ул. Ставропольская, д. 36, корп. 5. Тел. (878)20-21-90

ФОМИЧЕВА Светлана Григорьевна – заведующий кафедрой информационных систем и технологий Норильского индустриального института, кандидат технических наук, доцент.

663310, г. Норильск, ул. 50-лет Октября, д. 7. Тел. (3919) 42-17-39, факс (3919) 42-17-41, e-mail: levikha@rambler.ru

ХАПАЕВА Лёля Халисовна – старший преподаватель кафедры информатики и информационных технологий Карачаево-Черкесской государственной технологической академии.

369000, г. Черкесск, ул. Ставропольская, д. 36, корп. 5. Тел. (878)20-21-90, e-mail: lelia.kazalieva@yandex.ru

ЦЫБИН Михаил Сергеевич – студент кафедры систем автоматического управления Таганрогского технологического института Южного федерального университета.

347928, г. Таганрог Ростовской обл., пер. Некрасовский, д. 44. Тел. (8634) 37-16-89, e-mail: sgkrutch@mail.ru

ЧАПАЛО Иван Евгеньевич – студент кафедры радиофизики Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29

ШЕРСТЮК Сергей Сергеевич – аспирант кафедры вычислительной математики и кибернетики Уфимского государственного авиационного технического университета.

450000, г. Уфа, ул. Карла Маркса, д. 12. Тел. (8-347) 273-79-67, e-mail: evressim@yandex.ru

ЩЕРБАНЬ Анатолий Борисович – заместитель заведующего кафедрой по воспитательной работе, доцент кафедры информационных систем и технологий Пензенской государственной технологической академии, кандидат технических наук.

440605, г. Пенза, проезд Байдукова/ул. Гагарина, д. 1а/11. Тел. (8412)-49-59-80

АННОТАЦИИ

КЛЮЧЕВЫЕ СЛОВА

Васильев А.Е., Литвинчук А.В., Петров Д.Д. ПРОЕКТИРОВАНИЕ СРЕДСТВ И СИСТЕМ НЕЧЕТКОЙ ОБРАБОТКИ ИНФОРМАЦИИ НА ОСНОВЕ СЕТЕЙ ЭЛЕМЕНТАРНЫХ ВЫЧИСЛИТЕЛЕЙ НЕЧЕТКИХ ЛОГИЧЕСКИХ ФУНКЦИЙ.

Рассмотрены проблемы синтеза встраиваемых интеллектуальных систем управления. Изложены новые подходы к их решению, основанные на применении нечетких сетей.

ВСТРАИВАЕМАЯ СИСТЕМА. НЕЧЕТКАЯ ЛОГИКА. МИКРОКОНТРОЛЛЕР. НЕЧЕТКАЯ СЕТЬ.

Когновицкий О.С. АНАЛИЗ ПРЯМЫХ И ИНВЕРСНЫХ РЕКУРРЕНТНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ.

Предложен метод обработки «прямых» и инверсных рекуррентных последовательностей с помощью одного общего устройства, работающего на основе двойственного базиса. Приведены конкретные примеры.

М-ПОСЛЕДОВАТЕЛЬНОСТЬ. ПОСЛЕДОВАТЕЛЬНОСТЬ ГОУЛДА. ЛРД-ПОСЛЕДОВАТЕЛЬНОСТЬ. ДВОЙСТВЕННЫЙ БАЗИС.

Журавлева Л.М. ПОВЫШЕНИЕ ПРОПУСКНОЙ СПОСОБНОСТИ ОПТИЧЕСКИХ ЛИНИЙ СВЯЗИ.

Проанализирована пропускная способность оптических линий связи на примере технологий TDM и WDM. Рассмотрены потенциальные возможности повышения скорости передачи информации. Предложены пути увеличения пропускной способности ВОЛС с помощью нанотехнологий.

ПРОПУСКНАЯ СПОСОБНОСТЬ. ОПТИЧЕСКИЕ ЛИНИИ. НАНОТЕХНОЛОГИИ.

Еремеев И.Ю., Замарин А.И., Семенюк С.С. ТЕХНОЛОГИЯ СТРУКТУРНОГО АНАЛИЗА СИГНАЛОВ С ПСЕВДОСЛУЧАЙНОЙ ПЕРЕСТРОЙКОЙ НЕСУЩЕЙ ЧАСТОТЫ В ЗАДАЧАХ РАДИОМОНИТОРИНГА.

Представлены информативные признаки, используемые для объединения частотных посылок в сигнал со скачкообразным изменением несущей частоты, варианты представления частотно-временной структуры указанного сигнала, а также содержание этапов идентификации недвоичных псевдослучайных последовательностей и решения прикладных задач.

ИДЕНТИФИКАЦИЯ. РАДИОСИГНАЛ СО СКАЧКАМИ ЧАСТОТЫ. НЕДВОИЧНЫЕ ПСЕВДОСЛУЧАЙНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ.

Макаров С.Б., Рашич А.В., Атик С. АЛГОРИТМ ПРИЕМА СПЕКТРАЛЬНО-ЭФФЕКТИВНЫХ СИГНАЛОВ С OFDM.

Предложена методика приема спектрально-эффективных сигналов с OFDM на основе анализа решетки амплитудно-фазовых траекторий подобно алгоритму Витерби. Показано, что энергетические потери при приеме по данному алгоритму составляют от 0,7 до 7,3 дБ для вероятностей ошибки 10^{-4} – 10^{-5} в зависимости от параметров спектрально-эффективных сигналов с OFDM.

СПЕКТРАЛЬНО-ЭФФЕКТИВНЫЕ СИГНАЛЫ. АЛГОРИТМ ВИТЕРБИ. OFDM. ПИК-ФАКТОР.

Ростов Н.В. ПАРАМЕТРИЧЕСКАЯ ОПТИМИЗАЦИЯ ЦИФРОВЫХ МОДАЛЬНЫХ РЕГУЛЯТОРОВ.

Рассмотрены этапы синтеза цифровых модальных регуляторов с наблюдателем методом размещения полюсов. Изложена методика итерационной параметрической оптимизации таких регуляторов в пространстве полюсов по интегральным критериям, совмещенной с процедурами синтеза и анализа динамики цифровой САУ. Приведены примеры такой оптимизации.

ЦИФРОВЫЕ МОДАЛЬНЫЕ РЕГУЛЯТОРЫ. СИНТЕЗ. РАЗМЕЩЕНИЕ ПОЛЮСОВ. ПАРАМЕТРИЧЕСКАЯ ОПТИМИЗАЦИЯ. ИНТЕГРАЛЬНЫЕ КРИТЕРИИ.

Зубков А.Ф., Щербань А.Б., Семенов И.А. СТРУКТУРНО-СИНТАКСИЧЕСКИЙ ПОДХОД К ПОИСКУ АЛЬТЕРНАТИВ УПРАВЛЕНИЯ СЛОЖНЫМИ СИСТЕМАМИ.

Рассмотрена формализация структурно-синтаксического подхода в терминах идентификационно-структурного управления, с использованием структурного представления сложных систем и задач идентификационно-структурного анализа, связанных с поиском изоморфных отображений синтаксических структурных моделей.

СТРУКТУРНЫЙ ПОДХОД. ИДЕНТИФИКАЦИЯ СТРУКТУР. СТРУКТУРНАЯ МОДЕЛЬ.

Посов И.А., Рукшин С.Е. АРХИТЕКТУРА СИСТЕМЫ ПРОВЕДЕНИЯ УДАЛЕННЫХ СОРЕВНОВАНИЙ И ОРГАНИЗАЦИИ РАБОТЫ С МАТЕМАТИЧЕСКИМИ ЗАДАЧАМИ

Представлена структура и технические особенности системы для проведения удаленных соревнований и тестирований, ориентированной на проведение научных олимпиад для школьников, интересующихся естественными и техническими науками. Основная особенность данной системы – способность к расширению методов ввода и проверки решений, поддержка широкого класса типов соревнований и различных методик контроля знаний.

СИСТЕМА ПРОВЕДЕНИЯ УДАЛЕННЫХ СОРЕВНОВАНИЙ. АРХИТЕКТУРА ПРОГРАММНОЙ СИСТЕМЫ. АВТОМАТИЧЕСКАЯ ПРОВЕРКА РЕШЕНИЙ.

Моледу М.Ф. РАСПРЕДЕЛЕННОЕ МНОГОКРИТЕРИАЛЬНОЕ УПРАВЛЕНИЕ ДЛЯ ВЕТРОПАРКОВ.

Рассмотрен подход к децентрализованному принципу управления распределенными объектами и технологическими процессами, основанными на агентах. Особое внимание уделено дизайну и анализу новых критериев оптимизации интеллектуальной системы управления сетью распределенных ветрогенераторов. Разработан интерфейс на C++ для сравнения новой распределенной системы управления с централизованной системой.

МНОГОЦЕЛЕВАЯ ОПТИМИЗАЦИЯ. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ. РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ. МОДЕЛИРОВАНИЕ.

Ицыксон В.М., Глухих М.И. ЯЗЫК СПЕЦИФИКАЦИЙ ПОВЕДЕНИЯ ПРОГРАММНЫХ КОМПОНЕНТОВ.

Рассмотрена проблема статического анализа программ, использующих внешние программные компоненты. Проанализированы существующие подходы к построению спецификаций поведения компонентов. На основании анализа сформированы требования к построению специализированного языка и ограничения к данному языку. Приведено описание языка спецификаций PanLang, используемого в составе системы статического анализа.

ЯЗЫК. СПЕЦИФИКАЦИЯ. АННОТАЦИЯ. ПОВЕДЕНИЕ ВНЕШНИХ ПРОГРАММНЫХ КОМПОНЕНТОВ. СТАТИЧЕСКИЙ АНАЛИЗ. РЕСУРСЫ. УТЕЧКИ. PANLANG.

Васильев П.Н., Артамонов А.В., Маховенко Е.Б. КЛАССИФИКАЦИОННАЯ СХЕМА ГРУППОВЫХ ПОДПИСЕЙ ДЛЯ ПОСТРОЕНИЯ РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЙ.

Рассмотрены некоторые существующие схемы групповой подписи. Предложена классификационная схема групповых подписей, ее работа продемонстрирована на примере групповой подписи BBS. Предложена эффективная реализация схемы BBS на основе библиотеки MIRACL, а также распределенного приложения, демонстрирующего ее возможности.

ГРУППОВАЯ ПОДПИСЬ. СХЕМА BBS. ЭЛЛИПТИЧЕСКАЯ КРИВАЯ. РАСПРЕДЕЛЕННОЕ ПРИЛОЖЕНИЕ.

Иванков А.А., Елисеев Д.С. ДИНАМИЧЕСКОЕ ОТСЛЕЖИВАНИЕ МОДУЛЕМ ИНФОРМАЦИОННО-ПОИСКОВОЙ СИСТЕМЫ ИЗМЕНЕНИЙ В СТРУКТУРЕ ИЛИ ТЕКСТЕ ИНТЕРНЕТ-РЕСУРСА.

Предложен алгоритм отслеживания семантических изменений интернет-ресурсов на основе комбинированного подхода, путем последовательного анализа изменений иерархической структуры и текста документа. Оценка изменений в иерархической структуре вычисляется как редактирующее расстояние между ориентированными деревьями. Изменения в тексте оцениваются с привлечением векторной модели документа. Показано, что одна только иерархия семантик, построенная по HTML-коду документа, не позволяет решить поставленную задачу в тех случаях, когда существенно изменяется структура документа.

ИНФОРМАЦИОННО-ПОИСКОВЫЕ СИСТЕМЫ. ИНФОРМАЦИОННЫЙ ПОИСК. СЕМАНТИЧЕСКИЕ ИЗМЕНЕНИЯ. ИЕРАРХИЧЕСКАЯ СТРУКТУРА ДОКУМЕНТА. HTML-ГРАММАТИКА. ВЕКТОРНАЯ МОДЕЛЬ ДОКУМЕНТА.

Моисеев М.Ю. АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ ДЕФЕКТОВ В МНОГОПОТОЧНЫХ ПРОГРАММАХ МЕТОДАМИ СТАТИЧЕСКОГО АНАЛИЗА.

Рассмотрены вопросы обнаружения дефектов в многопоточных программах. Предложен подход, расширяющий алгоритмы статического анализа последовательных программ на многопоточные программы, в основе которого алгоритм анализа потоков, выполняющий анализ параллельных блоков программы и взаимодействий между конструкциями синхронизации.

МНОГОПОТОЧНАЯ ПРОГРАММА. ОБНАРУЖЕНИЕ ДЕФЕКТОВ. СТАТИЧЕСКИЙ АНАЛИЗ. АНАЛИЗ ПАРАЛЛЕЛЬНЫХ КОНСТРУКЦИЙ.

Сергеев Е.С., Гетманский В.В., Горобцов А.С. ПЕРЕНОС СИСТЕМЫ МНОГОТЕЛЬНОЙ ДИНАМИКИ НА ВЫЧИСЛИТЕЛЬНЫЙ КЛАСТЕР.

Представлены вопросы адаптации системы моделирования многоотельной динамики, пакет Фрунд, к распределенным вычислительным системам. Изложены основные проблемы, с которыми столкнулись авторы в процессе проведения работы, и предложены варианты их решения с подробным обоснованием. Дальнейшее развитие работы направлено на использование вычислительных ресурсов GRID-систем.

САЕ. MBS. ДИНАМИКА СИСТЕМ МНОГИХ ТЕЛ. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ. РАСПРЕДЕЛЕННАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА. ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ. MPI. MPI-I/O.

Ицыксон В.М., Тимофеев Д.А. ТЕХНОЛОГИЯ МОДИФИКАЦИИ ПРОГРАММНОГО КОДА, ОСНОВАННАЯ НА ПАРАМЕТРИЗУЕМЫХ ШАБЛОНАХ.

Предложен подход, упрощающий процесс описания и осуществления модификации программного кода. Рассмотрены основные этапы реализации: создание языка шаблонов, выбор модельного представления программы, использование алгоритма построения дифференциального абстрактного синтаксического дерева для модификации программного кода. Разработан прототип системы модификации программного кода для языка программирования Java.

АВТОМАТИЗИРОВАННАЯ МОДИФИКАЦИЯ ПРОГРАММНОГО КОДА. ЯЗЫК ШАБЛОНОВ. ДИФФЕРЕНЦИАЛЬНОЕ АБСТРАКТНОЕ СИНТАКСИЧЕСКОЕ ДЕРЕВО. ПОИСК ШАБЛОНОВ КОДА.

Бабкова Е.В., Шерстюк С.С. МОДЕЛЬ И АЛГОРИТМ КАЛЕНДАРНОГО РАСПРЕДЕЛЕНИЯ ВЗАИМОЗАМЕНЯЕМЫХ РЕСУРСОВ С УЧЕТОМ ФРОНТА ВЫПОЛНЕНИЯ РАБОТ.

Рассмотрены модель и алгоритм календарного распределения взаимозаменяемых ресурсов с учетом текущего фронта выполнения работ.

ВЗАИМОЗАМЕНЯЕМЫЙ РЕСУРС. ФРОНТ РАБОТ. МОДЕЛЬ. АЛГОРИТМ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ.

Беляев А.Б. ВЕРИФИКАЦИЯ АЛГОРИТМА ПОДДЕРЖКИ ТРАНЗАКЦИОННОЙ ПАМЯТИ.

Представлены результаты автоматической верификации одного из алгоритмов программной реализации транзакционной памяти для многоядерных процессоров. В алгоритме была обнаружена и исправлена тонкая ошибка, которая в некоторых специальных случаях могла привести к серьезным программным сбоям.

ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ. ВЕРИФИКАЦИЯ. MODEL CHECKING. ТЕМПОРАЛЬНАЯ ЛОГИКА. ТРАНЗАКЦИОННАЯ ПАМЯТЬ. OPACITY.

Блеканов И.С., Бондаренко Д.С. ТЕМАТИЧЕСКИЙ КРАУЛИНГ НА ОСНОВЕ АЛГОРИТМА HITS.

Описаны принципы функционирования сфокусированного Веб-краулера, его алгоритм обхода Веб-пространства, в качестве которого используется модифицированный алгоритм Клеинберга HITS. Приведены примеры и оценки эффективности результатов его работы как в локальном Веб-пространстве (в рамках одного домена), так и в региональном (по русскоязычному Веб-пространству). А также его сравнение со стандартным Веб-краулером.

ОБХОД ВЕБ-ГРАФА. ВЕБ-КРАУЛЕР. ТЕМАТИЧЕСКИЙ ВЕБ-КРАУЛЕР. АЛГОРИТМ HITS. ОЦЕНКА КАЧЕСТВА В ИНФОРМАЦИОННОМ ПОИСКЕ. ЭКСПЕРИМЕНТ.

Матвеев В.А. ИССЛЕДОВАНИЕ РАВНОВЕСНЫХ СИТУАЦИЙ В ИГРОВОЙ ЗАДАЧЕ ДВУХ ЛИЦ С ВЕКТОРНЫМИ ВЫИГРЫШАМИ.

Рассмотрена бескоалиционная игра двух лиц с векторными выигрышами и определено конусное равновесие. Такое решение уточнено посредством отношения предпочтения по конусу в пространстве критериев. Приведен пример, показывающий, что предложенная процедура позволяет выделить единственное решение.

ВЕКТОРНЫЙ ВЫИГРЫШ. БЕСКОАЛИЦИОННАЯ ИГРА. РАВНОВЕСИЕ ПО НЭШУ. ОПТИМАЛЬНОСТЬ ПО КОНУСУ.

Месягутов М.А. МОДИФИКАЦИЯ МЕТОДА ВЕТВЕЙ И ГРАНИЦ ДЛЯ ПОИСКА ОПТИМАЛЬНОГО РЕШЕНИЯ ЗАДАЧИ ОДНОМЕРНОЙ УПАКОВКИ ПРЯМОУГОЛЬНО-ОРИЕНТИРОВАННОЙ СТРУКТУРЫ.

Рассмотрена задача прямоугольно-ориентированной одномерной упаковки. Для ее решения предложен метод ветвей и границ с использованием правила «следующий подходящий» и оценок, получаемых с помощью решения вспомогательной задачи линейного программирования.

ПРЯМОУГОЛЬНО-ОРИЕНТИРОВАННАЯ ОДНОМЕРНАЯ УПАКОВКА. МЕТОД ВЕТВЕЙ И ГРАНИЦ. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ.

Кочкаров А.А., Хапаева Л.Х. АЛГОРИТМ РАСПОЗНАВАНИЯ ПРЕДФРАКТАЛЬНОГО ГРАФА С ЧЕРЕДОВАНИЕМ ЗАТРАВОК.

Рассмотрена частная задача распознавания предфрактального графа, порожденного парой полных затравок чередованием. Предложенный алгоритм распознавания решает эту задачу за полиномиальное время.

ПРЕДФРАКТАЛЬНЫЙ ГРАФ. РАСПОЗНАВАНИЕ ГРАФА. СЕТЕВЫЕ СИСТЕМЫ. ЗАТРАВКА.

Пашковский А.В. ПОВЫШЕНИЕ ТОЧНОСТИ РАСЧЕТОВ МАГНИТНОГО ПОЛЯ И СИЛОВЫХ ХАРАКТЕРИСТИК ЭЛЕКТРОМЕХАНИЧЕСКИХ УСТРОЙСТВ КОМБИНИРОВАННЫМ МЕТОДОМ СТАНДАРТНЫХ И КОНЕЧНЫХ ЭЛЕМЕНТОВ

Для расчета магнитного поля и силовых характеристик асинхронного тягово-подъемного модуля использован комбинированный метод стандартных и конечных элементов. Проанализирована точность и возможности метода на основе сравнения с экспериментальными данными и результатами расчета комбинированным методом граничных и конечных элементов.

МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ. МЕТОД СТАНДАРТНЫХ ЭЛЕМЕНТОВ. КРАЕВЫЕ ПОЛЕВЫЕ ЗАДАЧИ. ЧИСЛЕННОЕ РЕШЕНИЕ.

Узденов А.А., Кочкаров Р.А. АЛГОРИТМ ПОИСКА ВНЕШНЕ-ВНУТРЕННЕГО ЦЕНТРА ПРЕДФРАКТАЛЬНОГО ГРАФА С СОХРАНЕНИЕМ СМЕЖНОСТИ СТАРЫХ РЕБЕР.

Рассмотрена задача о внешне-внутреннем центре на предфрактальном графе, смежность старых ребер которого сохраняется. Впервые предложен алгоритм поиска внешне-внутреннего центра. Исследованы вопросы вычислительной сложности алгоритма.

ВНЕШНЕ-ВНУТРЕННИЙ ЦЕНТР. ПРЕДФРАКТАЛЬНЫЙ ГРАФ. СМЕЖНОСТЬ РЕБЕР. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ.

Владимиров С.С. МОДЕЛИРОВАНИЕ ПРОЦЕССОВ МАЖОРИТАРНОГО ДЕКОДИРОВАНИЯ КОМБИНАЦИИ ЭКВИДИСТАНТНОГО КОДА ПО k ЛИНЕЙНО-НЕЗАВИСИМЫМ ЭЛЕМЕНТАМ.

Рассмотрен алгоритм декодирования комбинаций эквидистантного двоичного циклического кода с длиной $n = 2^k - 1$ по k линейно-независимым элементам последовательности. Приведены результаты, полученные при моделировании данного алгоритма. Выполнено сравнение этого алгоритма с другими методами декодирования указанных эквидистантных кодов.

МАЖОРИТАРНОЕ ДЕКОДИРОВАНИЕ. ЭКВИДИСТАНТНЫЙ КОД. ЦИКЛИЧЕСКИЙ КОД. МОДЕЛИРОВАНИЕ.

Медведев А.В., Чапало И.Е. ПРИМЕНЕНИЕ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ В СИСТЕМЕ ОХРАНЫ НА ОСНОВЕ МЕЖМОДОВОЙ ИНТЕРФЕРЕНЦИИ В ВОЛОКОННОМ СВЕТОВОДЕ.

Рассмотрен датчик для охранной сигнализации, использующий межмодовую интерференцию на выходе волоконного световода. Описано разработанное устройство приема и обработки информации.

МЕЖМОДОВАЯ ИНТЕРФЕРЕНЦИЯ. ОХРАННАЯ СИГНАЛИЗАЦИЯ. ФЕДИНГ. ЦИФРОВАЯ ОБРАБОТКА.

Мальчевский С.А. ИСПОЛЬЗОВАНИЕ МЯГКИХ ВЫЧИСЛЕНИЙ ДЛЯ ОПТИМИЗАЦИИ ПАРАМЕТРОВ ДЕВИАЦИИ ПРИЗНАКОВ ПРИ РАСПОЗНАВАНИИ ОБРАЗОВ.

Рассмотрены методы оптимизации параметра девиации признаков в задачах распознавания образов. Предложен наиболее оптимальный подход с применением нечетких нейронных сетей.

РАСПОЗНАВАНИЕ ОБРАЗОВ. НЕЧЕТКИЕ НЕЙРОННЫЕ СЕТИ. НЕЧЕТКИЙ ВЫВОД. НЕЙРОННЫЕ СЕТИ. ДЕВИАЦИЯ ПРИЗНАКОВ.

Морозов А.И., Писарев А.И. МЕТОД СОЗДАНИЯ ЛИНГВИСТИЧЕСКИХ ПЕРЕМЕННЫХ НЕЧЕТКОЙ МОДЕЛИ.

Предложен метод создания лингвистических переменных нечетких моделей объектов и процессов, который исключает участие экспертов при их разработке. Алгоритм имеет широкую область применения, т. к. основан только на обработке статистических данных.

МОДЕЛИРОВАНИЕ. СТАТИСТИЧЕСКИЕ ДАННЫЕ. НЕЧЕТКАЯ ЛОГИКА. АЛГОРИТМ СУГЕНО. БАЗА ПРАВИЛ. ФУНКЦИЯ ПРИНАДЛЕЖНОСТИ. ТЕРМ. АВТОМАТИЗАЦИЯ.

Фомичева С.Г., Елина Т.Н. НЕЙРО-НЕЧЕТКОЕ МОДЕЛИРОВАНИЕ СТРАТЕГИЧЕСКОГО ВЗАИМОДЕЙСТВИЯ УЧАСТНИКОВ РЫНКА ТРУДА МОНОИНДУСТРИАЛЬНОГО РЕГИОНА.

Рассмотрены проблемы управления рынком труда закрытого моноиндустриального региона. Построены статистические модели прогнозирования демографических показателей региона. Описана нейро-нечеткая модель формирования спроса и предложения на региональном рынке труда.

РЫНОК ТРУДА. УПРАВЛЕНИЕ. МОДЕЛИРОВАНИЕ. НЕЧЕТКИЕ МНОЖЕСТВА. НЕЙРОННЫЕ СЕТИ. ПРОГНОЗИРОВАНИЕ. СТАТИСТИКА.

Попкова А.А., Фомичева С.Г. МЕТОДЫ АПРИОРНОЙ ОЦЕНКИ СТЕПЕНИ АГРЕГАЦИИ И ПРОИЗВОДИТЕЛЬНОСТИ ЗАПРОСОВ В ХРАНИЛИЩАХ ДАННЫХ.

Предложен подход к априорной оценке степени агрегации данных многомерного гиперкуба и методика увеличения скорости выполнения аналитических запросов с низкой и средней избирательностью за счет уменьшения вычислительных затрат, связанных с агрегированием обобщенных данных.

ХРАНИЛИЩЕ ДАННЫХ. OLAP-ТЕХНОЛОГИИ. СТЕПЕНЬ АГРЕГАЦИИ ДАННЫХ. ОРИЕНТИРОВАННЫЙ ГРАФ. ВЫЧИСЛИТЕЛЬНЫЕ ЗАТРАТЫ.

Крутчинский С.Г., Цыбин М.С. АНАЛОГО-ЦИФРОВЫЕ ПРЕОБРАЗОВАТЕЛИ ДЛЯ СМЕШАННЫХ СИСТЕМ НА КРИСТАЛЛЕ.

Для многоканальных смешанных СнК выполнена схемотехническая разработка импульсно-потенциального АЦП с минимальным числом прецизионных пассивных и активных компонентов. Рассмотрены основные составляющие погрешности преобразования и доказано, что потенциальная точность в ряде случаев превосходит традиционные АЦП. Показана целесообразность использования в аналоговых интерфейсах мультидифференциальных операционных усилителей. Приведены результаты моделирования принципиальной схемы АЦП на компонентах радиационно-стойкого АБМК.

АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ. АНАЛОГОВЫЙ БАЗОВО-МАТРИЧНЫЙ КРИСТАЛЛ. ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ.

Прокопенко Н.Н., Серебряков А.И., Манжула В.Г. КОМПЕНСАЦИЯ НАПРЯЖЕНИЯ СМЕЩЕНИЯ НУЛЯ ОПЕРАЦИОННЫХ УСИЛИТЕЛЕЙ С НЕСИММЕТРИЧНЫМ ВКЛЮЧЕНИЕМ АКТИВНОЙ НАГРУЗКИ.

Рассмотрен способ компенсации систематической составляющей напряжения смещения операционного усилителя, обеспечивающий существенное снижение статической ошибки усиления сигналов постоянного тока.

ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ. НАПРЯЖЕНИЕ СМЕЩЕНИЯ. МОДЕЛИРОВАНИЕ.

Крутчинский С.Г., Титов А.Е. МУЛЬТИДИФФЕРЕНЦИАЛЬНЫЙ ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ В РЕЖИМЕ ИНСТРУМЕНТАЛЬНОГО УСИЛИТЕЛЯ.

Рассмотрены результаты проектирования структурно оптимальной принципиальной схемы инструментального усилителя на базе мультидифференциального операционного усилителя с низким коэффициентом передачи синфазного напряжения. Показано, что использование такого инструментального усилителя позволяет минимизировать влияние резистивных элементов на коэффициент передачи синфазного сигнала при реализации необходимого дифференциального коэффициента усиления. Приведены результаты моделирования набора принципиальных схем, реализованных на компонентах радиационно-стойкого аналогового базового матричного кристалла.

МУЛЬТИДИФФЕРЕНЦИАЛЬНЫЙ ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ.

Прокопенко Н.Н., Манжула В.Г., Белич С.С. МУЛЬТИДИФФЕРЕНЦИАЛЬНЫЙ ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ С МАЛЫМ НАПРЯЖЕНИЕМ СМЕЩЕНИЯ НУЛЯ В УСЛОВИЯХ ТЕМПЕРАТУРНЫХ И РАДИАЦИОННЫХ ВОЗДЕЙСТВИЙ.

Рассмотрено схемотехническое решение, обеспечивающее уменьшение абсолютного значения напряжения смещения, а также его температурного и радиационного дрейфа при использовании в схеме токовых зеркал, имеющих коэффициент передачи по току, не равный единице.

МУЛЬТИДИФФЕРЕНЦИАЛЬНЫЙ ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ. НАПРЯЖЕНИЕ СМЕЩЕНИЯ. МОДЕЛИРОВАНИЕ.

Будяков А.С., Прокопенко Н.Н., Манжула В.Г. АНАЛИЗ ЧАСТОТНЫХ ХАРАКТЕРИСТИК ПОДСХЕМ ВХОДНЫХ КАСКАДОВ СВЧ ОПЕРАЦИОННЫХ УСИЛИТЕЛЕЙ ДЛЯ СЛОЖНЫХ ФУНКЦИОНАЛЬНЫХ БЛОКОВ РАДИАЦИОННО-СТОЙКОЙ АППАРАТУРЫ.

Рассмотрены особенности построения входных каскадов операционных усилителей СВЧ диапазона для использования в сложных функциональных блоках.

ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ. МОДЕЛИРОВАНИЕ. ВХОДНОЙ КАСКАД.

Черкесов Г.Н., Чуркин В.В. О ПРОБЛЕМЕ РАСЧЕТА НАДЕЖНОСТИ ВОССТАНАВЛИВАЕМЫХ СИСТЕМ ПРИ НАЛИЧИИ ЗАПАСНЫХ ЭЛЕМЕНТОВ.

Представлена аннотация доклада профессора Г.Н. Черкесова и аспиранта В.В. Чуркина на научном семинаре факультета технической кибернетики.

ANNOTATION

KEYWORDS

Vassiliev A.E., Litvinchuk A.V., Petrov D.D. DESIGN of RESOURCES and SYSTEMS of FUZZY-ORIENTED DATA PROCESSING on the BASIS of NETWORKS of ELEMENTARY FUZZY-LOGIC SOLVERS.

A problem of synthesis of intelligent embedded control systems is considered. New approaches of synthesis “fuzzy nets” are discussed.

EMBEDDED SYSTEM. FUZZY LOGIC. MICROCONTROLLER. FUZZY NET.

Kognovitzkiy O.S. ANALYSIS OF DIRECT AND INVERSE RECURRENT SEQUENCES.

Method of direct and inverse recurrent sequences processing with using single common device based on dual basis is introduced. Certain examples are presented.

M-SEQUENCE. GOULD-SEQUENCE. JPL-SEQUENCE. DUAL BASIS.

Zhuravleva L.M. CAPACITY GROWTH OF THE OPTICAL LINKS.

This passage analyses possibilities of capacity growth of an optical links in terms of the TDM and WDM technologies. It examines possibilities of optical fiber transmission system. The author describes the ways of increase information transfer rate by applying an nanotechnologies.

CAPACITY. OPTICAL LINKS. NANOTECHNOLOGIES.

Eremeev I.Y., Zamarin A.I., Semenuk S.S. TECHNOLOGY OF THE STRUCTURAL ANALYSIS OF FREQUENCY HOPPING SIGNALS IN THE TASKS RADIOMONITORING.

In this article the informative attributes used for association of time-frequency burst in a frequency hopping signal, the variants of representation of time-frequency structure of the specified signal, and also the contents of stages of identification of non binary pseudo-random sequences and decisions of applied tasks are presented.

IDENTIFICATION. FREQUENCY HOPPING SIGNAL. NON BINARY PSEUDO-RANDOM SEQUENCES.

Makarov S.B., Rashich A.V., Atik S. ALGORITHM FOR SPECTRALLY EFFICIENT OFDM-SIGNALS RECEPTION.

New method for spectrally efficient OFDM-signals reception is proposed. It is based on analysis of amplitude-phase grid similar to Viterbi decoding algorithm.

SPECTRALLY EFFICIENT SIGNALS. ALGORITHM VITERBI. OFDM. AMPLITUDE FACTOR.

Rostov N.V. PARAMETRICAL OPTIMIZATION OF DIGITAL MODAL CONTROLLERS.

The main stages of digital modal controller synthesis by the pole placement technique are discussed. Some statements of the controller computer-aided optimization by means of iterative methods are considered. Two examples are presented to solve optimization problems using integral criteria.

DIGITAL MODAL CONTROLLERS. STATE OBSERVERS. SYNTHESIS. POLE PLACEMENT TECHNIQUE. COMPUTER-AIDED OPTIMIZATION. INTEGRAL CRITERIA.

Zubkov A.F., Scherban A.B., Semenov I.A. THE STRUCTURAL-SYNTACTIC APPROACH TO SEARCH OF ALTERNATIVES OF MANAGEMENT BY DIFFICULT SYSTEMS.

We consider a formalization of the structural-syntactic approach, in terms of identity-structural control, using the structural representation of complex systems and problems of identity-structural analysis of the search isomorphic mappings of syntactic structural models.

STRUCTURAL APPROACH. THE IDENTIFICATION OF STRUCTURES. STRUCTURAL MODEL.

Posov I.A., Rukshin S.E. ARCHITECTURE OF A SYSTEM FOR DISTANCE CONTESTS EXECUTION AND ORGANIZATION OF WORKING WITH MATHEMATICAL PROBLEMS.

The paper presents the architectural structure and technical features of a system to execute distance tests and contests that is devoted to hold scientific Olympiads for students interested in natural and technical sciences. The main feature of the system is an ability to extend available input and checking methods for solutions, a support for a wide class of contest types and different techniques of knowledge control.

DISTANCE CONTESTS SYSTEM. SOFTWARE ARCHITECTURE. AUTOMATIC SOLUTION CHECK.

Mauledoux M.F. DISTRIBUTED CONTROL BY MULTI-OBJECTIVE OPTIMIZATION FOR WIND PARKS.

The work is devoted to the development of an agent-based approach and decentralized control principle of the distributed objects and technological processes. The task is focused on the design and analysis the new criteria of optimization via Multi-Objective optimal control of energies balance, controlling the wind farm as a team of agents, and not like a single isolated system. To show the advantages was developed an interface in C++ showing the wind simulation, the behavior of the wind farm using the new distributed intelligent control system and the comparison with the centralized system.

MULTI-OBJECTIVE OPTIMIZATION. GENETIC ALGORITHMS. DISTRIBUTED SYSTEMS. MODELLING.

Itsykson V.M., Glukhikh M.I. A PROGRAM COMPONENT BEHAVIOR SPECIFICATION LANGUAGE.

Static analysis of programs with external components is considered in this article. Existing methods of component specification description are analyzed. Requirements and restrictions for special-purpose language are developed using analysis results. Description of the PanLang specification language is presented. This language is used as the part of the static analysis tool.

LANGUAGE. SPECIFICATION. ANNOTATION. EXTERNAL PROGRAM COMPONENTS BEHAVIOR. STATIC ANALYSIS. RESOURCES. LEAKS. PANLANG.

Vasilyev P.N., Artamonov A.V., Makhovenko E.B. CLASSIFICATION SCHEME OF GROUP SIGNATURES FOR DISTRIBUTED APPLICATIONS DEVELOPMENT.

Some existing group signature schemes are considered. The classification scheme of group signatures is proposed, its work is demonstrated by BBS group signature. Efficient implementation of BBS scheme is proposed, as well as implementation of distributed application that illustrates its capabilities.

GROUP SIGNATURE. BBS SCHEME. ELLIPTIC CURVE. DISTRIBUTED APPLICATION.

Ivankov A.A., Eliseev D.S. WEB-PAGE STRUCTURE AND TEXT MONITORING WITH A ROBOT OF SEARCH-ENGINE.

The algorithm for the dynamic monitoring of Web-page structure and text is developed. The algorithm is implemented as a robot of search-engine. Document structure changes are estimated as tree-edit distance. Vector model is in use to estimate the changes into the text. Semantics hierarchy to be obtained from the HTML source code is not an efficient tool for the case the structure is changed significantly.

SEARCH-ENGINE. INFORMATION RETRIEVING. SEMANTIC CHANGES. HIERARCHICAL STRUCTURE OF DOCUMENT. HTML-GRAMMAR. VECTOR MODEL OF DOCUMENT.

Moiseev M.J. STATIC ANALYSIS METHODS FOR AUTOMATIC DEFECT DETECTION IN MULTITHREADED PROGRAMS.

Defect detection in multithreaded programs is considered in the article. An approach of usage static analysis algorithms of sequential programs for multithreaded programs is proposed. This approach is based on the thread analysis algorithm that analyzes parallel blocks and synchronization interactions.

MULTITHREADED PROGRAM. DEFECTS DETECTION. STATIC ANALYSIS. MAY HAPPEN IN PARALLEL ANALYSIS.

Sergeev E.S., Getmanski V.V., Gorobtsov A.S. TRANSFER THE SYSTEM OF MULTIBODIES DYNAMICS TO COMPUTING CLUSTER.

Presents problems of adaptation of system modeling multibodies dynamics package Frund to distributed computing systems. The basic problems which authors in the during of work carrying out have faced are stated and variants of their decision with a detailed substantiation are offered. Further development work is directed to the use of computing resources of GRID-Systems.

CAE. MULTIBODIES SYSTEMS DYNAMICS. MATHEMATICAL MODELING. DISTRIBUTED COMPUTING SYSTEM. PARALLEL PROGRAMMING. MPI. MPI-I/O.

Itsykson V.M., Timofeyev D.A. SOURCE CODE MODIFICATION TECHNOLOGY BASED ON PARAMETRIZED CODE PATTERNS.

Source code modification approach based on parametrized code patterns is considered in this article. Basic implementation stages are presented: pattern language creation, source code model selection, use of differential abstract syntax tree extraction algorithm for source code modification. Prototype of source code modification system based on presented approach has been implemented for Java programming language.

AUTOMATED SOURCE CODE MODIFICATION. CODE PATTERNS LANGUAGE. DIFFERENTIAL ABSTRACT SYNTAX TREE. SOURCE CODE PATTERNS SEARCH.

Babkova E.V., Sherstyuk S.S. CALENDAR ALLOCATION OVERTYPE RESOURCES MODEL AND ALGORITHM WITH FRONT OF WORKS.

The paper regards the calendar allocation overtype resources model and algorithm with active front of works. OVERTYPE RESOURCE. FRONT OF WORKS. THE MODEL. CALENDAR PLANING ALGORITHM.

Belyaev A.B. VERIFICATION THE ALGORITHM OF TRANSACTIONAL MEMORY.

The paper presents the results of the mechanical verification the algorithm of software transactional memory. Verification has found a subtle error in the algorithm, which may lead to a system failure in some special cases. Then it was corrected.

PARALLEL PROGRAMMING. VERIFICATION. MODEL CHECKING. TEMPORAL LOGIC. TRANSACTIONAL MEMORY. OPACITY.

Blekanov I.S., Bondarenko D.S. TOPICAL CRAWLER THE HITS ALGORITHM.

The paper presents the functional principles of operation of a focused Web-crawler, its algorithm of traversal of Web-space, based on a modified version of Kleinberg's HITS algorithm. Examples and estimations of the new Web-crawler performance over both local (within a domain) and regional (Russian) Web-spaces are given along with a detailed comparison with the standard Web-crawler results

GRAPH TRAVERSAL. WEB-CRAWLER. TOPICAL WEB-CRAWLER. HITS ALGORITHM. QUALITY EVALUATION OF INFORMATION. EXPERIMENT.

Matveev V.A. THE RESEARCH OF EQUILIBRIA IN TWO PERSON GAME WITH VECTOR PAYOFFS.

The two persons game with vector payoff is considered. The cone equilibrium is defined. This decision is refined by the cone preference relation in criteria space. The example is presented and shown that the offered procedure allows to share out the unique decision.

VECTOR PAYOFF. NONCOOPERATIVE GAME. NASH EQUILIBRIUM. CONE OPTIMALITY.

Mesyagutov M.A. A MODIFICATION OF BRANCH-AND-BOUND METHOD FOR OPTIMAL SOLUTION OF THE 1D PACKING PROBLEM OF RECTANGULAR-ORIENTED STRUCTURE.

The 1D rectangular-oriented packing problem is considered. For its solution we propose a branch-and-bound method with usage of «next fit» rule and bounds based on the auxiliary linear programming problem.

1D RECTANGULAR-ORIENTED PACKING. BRANCH-AND-BOUND METHOD. LINEAR PROGRAMMING.

Kochkarov A.A., Hapaeva L.H. ALGORITHM OF RECOGNIZING OF PREFRACTAL GRAPH WITH ZAPRAVKA ALTERNATIONS.

The private task of prefractal graph recognizing, arising from pair of full zatravkas by alternations, was examined. The offered algorithm of recognizing decide this task during the polynomial time.

PREFRACTIONAL GRAPE. GRAPH RECOGNIZING. NET SYSTEM. ZATRAVKA.

Pashkovsky A.V. ACCURACY INCREASE OF CALCULATIONS THE MAGNETIC FIELD AND POWER CHARACTERISTICS OF ELECTROMECHANICAL DEVICES THE COMBINED METHOD OF STANDARD AND FINAL ELEMENTS.

The combined method of standard and final elements for calculation of the magnetic field and power characteristics of the asynchronous traction-elevating module is used. Accuracy and opportunities of a method are analyzed on the basis of comparison with experimental data and results of calculation by the combined method of boundary and final elements.

METHOD OF FINAL ELEMENTS. METHOD OF STANDARD ELEMENTS. REGIONAL FIELD PROBLEMS. THE NUMERICAL DECISION.

Uzdenov A.A., Kochkarov R.A. ALGORITHM OF SEARCH OF THE OUTWARDLY-INTERNAL CENTRE OF PRE-FRACTAL GRAPHS WITH PRESERVATION OF CONTIGUITY OF OLD EDGES.

In article the problem about the outwardly-internal centre on pre-fractal graphs is considered, the contiguity of which old edges remains. The algorithm of search of the outwardly-internal centre for the first time is offered. Questions of computing complexity of algorithm are investigated.

OUTWARDLY-INTERNAL CENTRE. PRE-FRACTAL GRAPH. CONTIGUITY OF EDGES. COMPUTING COMPLEXITY.

Vladimirov S.S. MODELLING OF MAJOR DECODING PROCESSES OF EQUIDISTANT CODE COMBINATION WITH K LINEAR-INDEPENDENT ELEMENTS.

In clause decoding algorithm of equidistant code combinations with length $n = 2^k - 1$ with k linear-independent elements is considered. Results of modelling of this algorithm are added in. Comparison with other decoding methods for such equidistant codes is included.

MAJOR DECODING. EQUIDISTANT CODE. CYCLE CODE. MODELLING.

Medvedev A.V., Chapalo I.E. ALARM SYSTEM ON BASIS OF MULTIMODE INTERFERENCE IN OPTICAL FIBER WITH DIGITAL SIGNAL PROCESSING.

Multimode fiber optic sensor for intrusion detection is considered. Signal detection and processing unit is described. MULTIMODE INTERFERENCE. INTRUSION DETECTION. FADING. DIGITAL SIGNAL PROCESSING.

Malchevskiy S.A. SOFT COMPUTING FOR OPTIMIZATION DEVIATION'S PARAMETERS OF FEATURES IN PATTERN RECOGNITION.

In clause considered the optimization methods of deviation parameter for tasks of pattern recognition. The most optimal approach using fuzzy neural networks is offered. An experimental study demonstrates the optimal settings of network by example of field-independent graphical patterns recognition.

PATTERN RECOGNITION. FUZZY NEURAL NETWORKS. FUZZY INFERENCE. NEURAL NETWORKS. DEVIATION OF FEATURES.

Morozov A.I., Pisarev A.I. METHOD OF AUTOMATICALLY CREATING OF A FUZZY MODEL INDUSTRIAL OBJECT.

This paper proposes a method of creating of linguistic variables for the fuzzy models of objects and processes, which excludes the participation of expert in their design. The algorithm has a wide of application, since it is based only on statistical data processing.

MODELING. STATISTICS. «FUZZY» LOGIC. SUGENO ALGORITHM. THE BASE RULES. MEMBERSHIP FUNCTION. A TERM. AUTOMATION.

Fomicheva S.G., Yelina T.N. NEURO-FUZZY MODELING OF STRATEGIC INTERACTION OF PARTICIPANTS OF THE LABOUR MARKET IN MONOINDUSTRIAL REGION.

In article problems of management are considered by a labour market of the closed monoindustrial region. Statistical models of forecasting of demographic indicators of region are constructed. The neuro-fuzzy model of formation of a supply and demand on a regional labour market is described.

LABOUR MARKET. MANAGEMENT. MODELLING. FUZZY SETS. NEURAL NETWORKS. FORECASTING. STATISTICS.

Popkova A.A., Fomicheva S.G. APRIORISTIC METHODS OF THE ESTIMATION FOR AGGREGATION DEGREE AND PRODUCTIVITY OF CALCULATIONS IN DATA WAREHOUSE.

In given article the approach to an aprioristic estimation of data aggregation degree for a multidimensional hypercube and a technique of increase in speed of performance of analytical inquiries with low and average selectivity for the account of reduction for the computing expenses connected with aggregation of the generalised data is offered

DATA WAREHOUSE. OLAP-TECHNOLOGY. AGGREGATION DEGREE OF DATA. THE DIRECTED GRAPH. PRODUCTIVITY OF CALCULATIONS.

Krutchinsky S.G., Tsybin M.S. ANALOG TO DIGITAL CONVERTERS MULTIMODAL SNK.

For multi-channel mixed SoC circuitry design made of pulse-potential ADC with the minimum number of precision of passive and active components. The main components of the conversion error and show that the potential accuracy in some cases exceeds traditional ADC. The expediency of use of analog interfaces multidifferentsialnyh operational amplifiers. The results of modeling concept ADC components of radiation-resistant ABMC.

ANALOG-DIGITAL CONVERTER. ANALOG BASIC-MATRIX KRISTAL. THE OPERATIONAL AMPLIFIER.

Prokopenko N.N., Serebrjakov A.I., Manzhula V.G. WAY OF INDEMNIFICATION OF THE REGULAR COMPONENT OF PRESSURE OF DISPLACEMENT OF ZERO OF OPERATIONAL AMPLIFIERS WITH ASYMMETRICAL INCLUSION OF ACTIVE LOADING.

In article the way of indemnification of a regular component of pressure of displacement of the operational amplifier, providing essential decrease in a static error of strengthening of signals of a direct current is considered.

THE OPERATIONAL AMPLIFIER. PRESSURE OF DISPLACEMENT. MODELLING.

Krutchinsky S.G., Titov A.E. MULTIDIFFERENTIALNY OPERATIONAL AMPLIFIER IN TREATMENT INSTRUMENTATION AMPLIFIERS.

The results of the structural design of the optimal concept instrumentation amplifier based on multidifferentsialnogo operational amplifier with a low coefficient of transmission of common mode voltage. It is shown that the use of this instrumentation amplifier can minimize the influence of resistive elements in the transfer coefficient in-phase signal in the implementation of the required differential gain. Results of modeling a set of concepts, implemented on the components of radiation-resistant analog basic matrix of the crystal.

MULTIDIFFERENTIALNY OPERATIONAL AMPLIFIER.

Prokopenko N.N., Manzhula V.G., Belich S.S. THE MULTIDIFFERENTIAL OPERATIONAL AMPLIFIER WITH SMALL PRESSURE OF DISPLACEMENT OF ZERO IN THE CONDITIONS OF TEMPERATURE AND RADIATING INFLUENCES.

In article it is considered the decision providing reduction of absolute value of pressure of displacement, and also its temperature and radiating drift at use in the scheme current the mirrors having factor of transfer on a current, not equal to unit.

THE MULTIDIFFERENTIAL OPERATIONAL AMPLIFIER. PRESSURE OF DISPLACEMENT. MODELLING.

Budjakov A.S., Prokopenko N.N., Manzhula V.G. THE ANALYSIS OF FREQUENCY CHARACTERISTICS OF SUBCIRCUITS OF ENTRANCE CASCADES FOR GHZ RANGE OF OPERATIONAL AMPLIFIERS FOR DIFFICULT FUNCTIONAL BLOCKS RADIATIONNOSTOJKOJ OF EQUIPMENT.

In article features of construction of entrance cascades of operational amplifiers GHZ of a range for use in difficult functional blocks are considered.

THE OPERATIONAL AMPLIFIER. THE ENTRANCE CASCADE. MODELLING.

Cherkesov G.N., Tchurkin V.V. ABOUT THE PROBLEM OF CALCULATION OF RELIABILITY OF RESTORED SYSTEMS IN THE PRESENCE OF SPARE ELEMENTS.

The summary of the report of the professor of Cherkesov G.N and post-graduate student Tchurkin V.V. at a scientific seminar of faculty of technical cybernetics is presented.

НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ СПбГУ

№ 3 (101) 2010

Учредитель – Санкт-Петербургский государственный политехнический университет

Издание зарегистрировано в Госкомпечати РФ, свидетельство № 013165 от 23.12.94

Редакция журнала

д.э.н., профессор, канд. техн. наук *А.В. Бабкин* – научный редактор
Е.А. Калинина – литературный редактор, корректор
А.А. Родионова – менеджер по работе с клиентами
М.О. Давыденко – ответственный секретарь, выпускающий редактор

Телефон редакции 552-62-16, 297-18-21

E-mail: infocom@spbstu.ru

Компьютерная верстка *А.Н. Смирнов*

Директор Издательства Политехнического университета *А.В. Иванов*

Лицензия ЛР № 020593 от 07.08.97

Подписано в печать 28.06.2010. Формат 60×84 1/8. Бум. тип. № 1.
Печать офсетная. Усл. печ. л. 26,5. Уч.-изд. л. 26,5. Тираж 1000. Заказ

Санкт-Петербургский государственный политехнический университет
Издательство Политехнического университета
член Издательско-полиграфической ассоциации университетов России
Адрес университета и издательства: 195251, Санкт-Петербург, Политехническая ул., д. 29.