

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

7(143)  
2008

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ  
Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### СЕТИ И СИСТЕМЫ СВЯЗИ

- Алгулиев Р. М., Алыгулиев Р. М., Гусейнова А. А. Повышение эффективности корпоративных сетей с применением CDN-технологии . . . . . 2  
Самойлин Е. А. Синтез оптимальных алгоритмов оценивания импульсных помех при восстановлении сигналов в цифровых радиосистемах . . . . . 9  
Домнин А. О. Аналитическая модель клиент-серверного приложения . . . . . 13

### ОПТИМИЗАЦИЯ И СТРУКТУРНЫЙ СИНТЕЗ

- Мухачева Э. А., Бухарбаева Л. Я., Филиппов Д. В., Карипов У. А. Оптимизационные проблемы транспортной логистики: оперативное размещение контейнеров при транспортировке грузов 17  
Галимьянова Н. Н. Экспериментальное исследование параллельных реализаций комбинаторных алгоритмов решения задачи о ранце . . . . . 22

### НЕЙРОСЕТИ И НЕЙРОКОМПЬЮТЕРЫ

- Субботин С. А. Метод синтеза классифицирующих нейронечетких сетей с учетом значимости термов признаков . . . . . 31  
Жуков И. С. Применение нейронной сети для идентификации сигналов, передаваемых по командным радиолиниям космических аппаратов . . . . . 33

### ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Тарасов Г. В., Харитонов Д. И. Построение системы визуализации параллельных программ на основе сетей Петри . . . . . 35  
Рыжков Е. А., Карпов А. Н. Подходы к верификации и тестированию 64-битных приложений 41

### ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

- Бакаев В. В. Система информационно-технической поддержки заказчика . . . . . 46  
Алексеев А. А., Куликов А. И., Старушкин В. Г. Визуальный построитель геологических сред "Seismic Model Manager" . . . . . 48

### ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ

- Мальхина Г. Ф., Меркушева А. В. Совместное использование нейронной сети и вейвлет-преобразования при анализе нестационарного сигнала . . . . . 51  
Кирсанов А. П. Комбинаторный метод описания и контроля процессов . . . . . 57  
Асадов Х. Г., Набиев Н. А. Многокритериальная оптимизация четких и нечетких информационно-измерительных систем нестационарного энергетического состояния . . . . . 61

### ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИОЛОГИИ И МЕДИЦИНЕ

- Чумаченко Е. Н., Шашмурина В. Р. Прогнозирование взаимодействия структур костной ткани беззубой нижней челюсти с полными съемными протезами . . . . . 66  
Короткова М. А., Рекубрятский В. А., Коротков А. Е., Шульга О. А., Скрябин К. Г. Динамическая модель генетического контроля развития растения на примере *Arabidopsis thaliana* . . . . . 71

### ДИСКУССИОННЫЙ КЛУБ

- Шемакин Ю. И. Системантика и информационные нанотехнологии . . . . . 76

### ИНФОРМАЦИЯ

- Пленарное заседание ИСО/МЭК/СТК1/ПК36 в Корее . . . . . 78  
Contents . . . . . 79

- Приложение. Машкина И. В., Гузаиров М. Б. Интеллектуальная поддержка принятия решений по управлению защитой информации в критически важных сегментах информационных систем

Главный редактор  
НОРЕНКОВ И. П.

Зам. гл. редактора  
ФИЛИМОНОВ Н. Б.

Редакционная  
коллегия:

АВДОШИН С. М.  
АНТОНОВ Б. И.  
БАТИЩЕВ Д. И.  
БАРСКИЙ А. Б.  
БОЖКО А. Н.  
ВАСЕНИН В. А.  
ГАЛУШКИН А. И.  
ГЛОРИОЗОВ Е. Л.  
ГОРБАТОВ В. А.  
ДОМРАЧЕВ В. Г.  
ЗАГИДУЛЛИН Р. Ш.  
ЗАЛЕЩАНСКИЙ Б. Д.  
ЗАРУБИН В. С.  
ИВАННИКОВ А. Д.  
ИСАЕНКО Р. О.  
КОЛИН К. К.  
КУЛАГИН В. П.  
КУРЕЙЧИК В. М.  
ЛЬВОВИЧ Я. Е.  
МАЛЬЦЕВ П. П.  
МЕДВЕДЕВ Н. В.  
МИХАЙЛОВ Б. М.  
МУХТАРУЛИН В. С.  
НАРИНЬЯНИ А. С.  
НЕЧАЕВ В. В.  
ПАВЛОВ В. В.  
ПУЗАНКОВ Д. В.  
РЯБОВ Г. Г.  
СТЕМПКОВСКИЙ А. Л.  
УСКОВ В. Л.  
ЧЕРМОШЕНЦЕВ С. Ф.  
ШИЛОВ В. В.

Редакция:

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

Аннотации статей размещены на сайте журнала по адресу <http://www.informika.ru/text/magaz/it/> или <http://novtex.ru/IT>.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора наук.

УДК 004.7

**Р. М. Алгулиев**, чл.-корр. НАН Азербайджана,  
д-р техн. наук, директор,

**Р. М. Алыгулиев**, канд. физ.-мат. наук,  
зав. отделом,

**А. А. Гусейнова**, научный сотрудник  
Институт информационных технологий  
НАН Азербайджана

e-mail: rasim@science.az, a.ramiz@science.az

## Повышение эффективности корпоративных сетей с применением CDN-технологии

*Предлагается метод повышения эффективности корпоративных сетей, который основывается на CDN-технологии. Эффективность его оценивается минимизацией расходов доставки web-контентов к конечным потребителям. Для минимизации расходов предложена математическая модель выбора оптимальных точек размещения серверов и оптимальная репликация web-контентов. Модель описана задачей дискретного программирования. Для решения задачи дискретного программирования применен метод Particle Swarm Optimization.*

### Введение

Корпоративная сеть, как правило, является территориально распределенной, объединяющей офисы, подразделения и другие структуры, находящиеся на значительном удалении друг от друга. Узлы корпоративной сети часто оказываются расположенными в различных городах, а иногда и странах. Принципы построения корпоративных сетей достаточно сильно отличаются от тех, что используются при создании локальной сети, даже охватывающей несколько зданий. Основное отличие состоит в том, что корпоративные сети используют арендованные каналы связи. Если при создании локальной сети основные затраты приходятся на закупку оборудования и прокладку кабеля, то в корпоративных сетях наиболее существенный расход вызывает арендная плата за использование каналов [1]. Это ограничение является принципиальным, таким образом, при эксплуатации корпоративной сети предпринимаются все меры, чтобы минимизировать расходы передачи данных.

Пути решения минимизации расходов условно можно разделить на две группы: экстенсивный и интенсивный. Экстенсивный путь решения требует создания собственной сетевой инфраструктуры. Ясно, что такой путь требует больших капитальных затрат и, следовательно, неэффективен. Таким образом, для повышения эффективности корпоративных сетей следует искать интенсивные пути решения. К интенсивным путям решения можно отнести технологию CDN (*Content Delivery Networks*). По сути CDN является наложенной сетью, которая оперирует на верхних уровнях сети *Internet*, позволяя отказаться от создания собственной дорогостоящей сетевой инфраструктуры [2, 3]. Благодаря такому преимуществу с появлением первого CDN-провайдера началось широкое применение технологии CDN [3].

Технология CDN появилась как новое решение для улучшения показателей QoS (Quality of Service) пользователей. В последние годы причиной ухудшения QoS-показателей пользователей является быстрый рост числа пользователей сети. С ростом числа пользователей популярные web-сервисы накапливаются и становятся узкими местами. Такой сценарий вызывает неуправляемый уровень потока трафика. Для улучшения работы и масштабируемости сети репликация того же web-контента и сервиса на нескольких зеркальных серверах, стратегически размещенных на различных местах, является широко используемым методом. При этом пользователь переадресуется ближайшему серверу, в результате чего уменьшается время отклика на запрос пользователя. Через распределение web-контентов по периферийным серверам, ближе к пользователям, сеть CDN обеспечивает быструю и надежную доставку web-контентов [3].

Сеть CDN содержит в себе три ключевых компонента: контент-провайдер; CDN-провайдер и конечные пользователи. Контент-провайдер для распределения делегирует свои web-контенты CDN-провайдерам. Серверы-источники служат для хранения web-контентов. CDN-провайдер — это владелец организации или компании, который для своевременной и надежной доставки web-контентов конечным пользователям предоставляет контент-провайдерам гибкую инфраструктуру. Конечные пользователи или просто клиенты — это те, кто получает доступ к web-контентам контент-провайдеров [3].

Сеть CDN комбинирует в себе инфраструктуру доставки web-контента, адресацию запроса, распределение web-контентов и учет. Инфраструктура доставки web-контента состоит из множества периферийных серверов (серверов-суррогатов), доставляющих контент-реплики конечным пользователям. Инфраструктура адресации запроса направляет запрос пользователя подходящему серверу-суррогату. Для обеспечения своевременного просмотра web-контентов, хранимых в серверах-суррогатах, она взаимодействует с инфраструктурой распределения web-контентов. Инфраструктура распределения web-контентов перемещает web-контенты из серверов-источников в серверы-суррогаты и гарантирует постоянство в кэшах. Инфраструктура учета хранит журналы доступа клиентов и записи использования CDN-серверов. Эта информация используется для учета трафика и биллинга. Абонентами сетей CDN обычно являются вещательные компании, ISP, торговцы on-line музыкой, мобильные операторы и т. д. Каждый из этих абонентов хочет публиковать и надежно и своевременно доставлять свои web-контенты конечным пользователям [2–5].

В общем, сеть CDN — это набор сетевых элементов, предназначенных для обеспечения более эффективной доставки web-контентов конечным пользователям. Сети CDN имеют разную форму и структуру. Они могут иметь централизованную иерархическую инфраструктуру или быть полностью децентрализованной системой. Общий принцип проектирования сети CDN задан в работе [3]. Функциональная зависимость элементов типичных сетей CDN включает следующие сервисы:

- *сервис адресации запроса и доставки web-контента*, который направляет запрос "наилучшему" ближайшему серверу-суррогату;
- *сервис распределения*, реплицирующий и/или кэширующий web-контент по распределенным серверам-суррогатам;
- *сервис уполномочия web-контента*, который собирает информацию о специфических потребностях каждого пользователя (или групп пользователей);
- *сервис управления*, осуществляющий управление сетевых компонентов, он также обрабатывает учетную информацию, проводит мониторинг и составляет отчет об использовании web-контентов.

В последние годы технология CDN исследована с разных точек зрения. Всесторонний анализ и принципы проектирования сетей CDN описаны в работах [3–5]. Значение этих работ заключается в том, что в них достаточно хорошо проанализированы таксономии и состояние существующих сетей CDN. В работе [3] даются сравнительный анализ таксономии сетей CDN и их классификация

по назначению. Общий принцип работы функциональных элементов сетей CDN изложен в работе [5].

## 1. Формулировка задачи минимизации расходов доставки web-контентов

Применение технологии CDN эффективно в тех случаях, когда пользователи корпоративной сети независимо друг от друга часто обращаются к одним и тем же web-контентам. Такое повторное обращение, очевидно, приведет к дополнительной нагрузке каналов связи и лишним расходам. Таким образом, возникает вопрос уменьшения таких нагрузок и, следовательно, лишних расходов. Этот вопрос в данной статье попытаемся решить через применение такой технологии CDN, где часто запрашиваемые web-контенты перемещаются из серверов-источников в периферийные серверы, ближе к пользователям.

При проектировании сетей CDN требуется согласованное решение нескольких непростых задач, среди которых можно выделить следующие: оптимальный выбор точек размещения серверов в узлах корпоративной сети; оптимальная репликация web-контентов в CDN-серверах; оптимальная адресация запросов по CDN-серверам. Интегрированное решение этих задач обеспечивает минимизацию расходов нагрузки на опорную сеть и доставки web-контентов конечным пользователям.

Перечисленные задачи решены многими исследователями. Например, в работе [6] для оптимального выбора точек размещения CDN-серверов предложен метод, который учитывает степень значимости web-контентов на каждом узле корпоративной сети. Для выбора стратегических точек размещения CDN-серверов в узлах глобальной сети в работе [7] предложена модель, основанная на кластеризации узлов по интересам. В работе [8] предложен интегрированный подход к решению задач: определение числа и точек размещения CDN-серверов; оптимальная репликация web-контентов в CDN-серверах и адресация запросов по CDN-серверам. Этот подход сведен к задаче нелинейного целочисленного программирования. В работе [9] предложен алгоритм репликации web-контентов в сетях CDN с кольцеобразной топологией. Математическая модель распределения запросов в сетях CDN с многоуровневой иерархической архитектурой дана в работе [10].

Настоящая работа посвящена проблеме минимизации расходов доставки web-контентов конечным потребителям. С этой целью в работе предлагается математическая модель, которая на основе априорных знаний об интересах узлов корпоративной сети позволяет определить оптимальные точки размещения серверов и провести оптималь-

ную репликацию web-контентов в CDN-серверах и адресацию запросов по CDN-серверам. Предложенная модель описана задачей математического программирования, которая формулируется следующим образом.

Пусть задана корпоративная сеть, которая состоит из  $n$  узлов, и заданы CDN-серверы с разными пропускными способностями и памятью в количестве  $s$ . Цель задачи заключается в том, что из множества серверов следует выбрать те, которые при их оптимальном размещении в потенциальных узлах и оптимальной репликации web-контентов в них обеспечивают минимальность расходов.

Данная задача будет решена при следующих ограничениях:

- число запросов, адресованных серверам, не должно превышать их способность отвечать на запросы;
- объем реплицированных в серверах web-контентов не должен превышать выделенного объема памяти серверов;
- все обращения узлов должны быть удовлетворены.

При оптимальном определении точек размещения серверов в узлах корпоративной сети большой размерности целесообразно использовать список кандидатов-узлов, который представляет собой небольшой список потенциальных узлов. Поэтому предполагается, что для размещения CDN-серверов заранее известны потенциальные узлы. Список кандидатов формируется на основе априорных знаний о потребностях пользователей каждого узла. Он позволяет исключить заведомо неперспективные варианты и расположить серверы на наиболее обещающих узлах, тем самым существенно сократить область поиска. Предполагается, что число серверов меньше числа потенциальных узлов.

## 2. Сведение минимизации расходов к задаче дискретного программирования

Прежде чем описывать математическую формулировку задачи, введем некоторые обозначения.

Данные задачи:

$I = \{1, 2, \dots, n\}$  — множество узлов корпоративной сети;  $J = \{1, 2, \dots, m\} \subset I$  — множество потенциальных узлов для размещения серверов;  $P = \{1, 2, \dots, k\}$  — множество web-контентов;  $Q = \{1, 2, \dots, s\}$  — множество серверов;  $c_{ij}$  — стоимость передачи единичной информации между узлами  $i \in I$  и  $j \in J$ ,  $c_{ii} = 0$ ;  $v_p$  — объем web-контента  $p \in P$ ;  $d_{ip}$  — ожидаемое число обращений узла  $i \in I$  к web-контенту  $p \in P$ ;  $d_{ijp}$  — число обращений узла  $i \in I$  к реплицированному в потенциальном узле  $j \in J$  web-контенту  $p \in P$ ;  $C_q$  — способность сервера  $q \in Q$  отвечать на запросы;  $V_q$  — выделен-

ный объем памяти сервера  $q \in Q$  для репликации web-контентов;  $F_q$  — стоимость сервера  $q \in Q$ ;  $r_{pj}$  — расходы, связанные с обслуживанием (хранение, обновление) web-контента  $p \in K$  в потенциальном узле  $j \in J$ .

Переменные задачи:

$x_{ij}$  — бинарная переменная, равная единице, если узел  $i \in I$  прикреплен к узлу  $j \in J$ , и равная нулю, в противном случае;

$y_{qj}$  — бинарная переменная, равная единице, если сервер  $q \in S$  размещен в потенциальном узле  $j \in J$ , и равная нулю, в противном случае;

$z_{pj}$  — бинарная переменная, равная единице, если web-контент  $p \in K$  реплицирован в потенциальном узле  $j \in J$ , и равная нулю, в противном случае.

Формулировка задачи минимизации расходов в корпоративной сети как задачи математического программирования имеет следующий вид:

$$f(x, y, z) = \sum_{i \in I} \sum_{j \in J} \sum_{p \in P} c_{ij} v_p d_{ijp} x_{ij} + \sum_{j \in J} \sum_{q \in S} F_q y_{qj} + \sum_{p \in P} \sum_{j \in J} r_{pj} z_{pj} \rightarrow \min. \quad (1)$$

Первый член в целевой функции (1) соответствует стоимости доставки web-контентов, второй член — стоимости размещения серверов, а третий член — расходу обслуживания web-контентов в серверах.

На основе сделанных предположений переменные задачи должны удовлетворять следующим ограничениям.

Каждый узел должен прикрепиться как минимум к одному потенциальному узлу:

$$1 \leq \sum_{j \in J} x_{ij} \leq \sum_{q \in Q} \sum_{j \in J} y_{qj}, \quad \forall i \in I. \quad (2)$$

Число запросов, адресованных серверам, не должно превышать их способности отвечать на запросы:

$$\sum_{i \in I} \sum_{p \in P} d_{ijp} x_{ij} \leq y_{qj} C_q, \quad \forall j \in J, \forall q \in Q. \quad (3)$$

Память серверов не должна быть перегружена:

$$\sum_{p \in P} v_p z_{pj} \leq y_{qj} V_q, \quad \forall j \in J, \forall q \in Q. \quad (4)$$

Суммарное число обращений каждого узла к репликам web-контента должно быть меньше либо равно ожидаемому числу обращений узла к web-контенту:

$$\sum_{j \in J} d_{ijp} z_{pj} \leq d_{ip}, \quad \forall i \in I, \forall p \in P. \quad (5)$$

Каждый web-контент должен реплицироваться как минимум в одном сервере:

$$1 \leq \sum_{j \in J} z_{pj} \leq \sum_{q \in Q} \sum_{j \in J} y_{qj}, \quad \forall p \in P. \quad (6)$$

На каждом потенциальном узле должен размещаться максимум один сервер:

$$\sum_{q \in Q} y_{qj} \leq 1, \forall j \in J. \quad (7)$$

Число серверов, размещенных в потенциальных узлах, не должно превышать общее число серверов:

$$\sum_{q \in Q} \sum_{j \in J} y_{qj} \leq s. \quad (8)$$

Следующие условия непосредственно вытекают из постановки задачи:

$$x_{ij} \leq y_{qj}, \forall i \in I, \forall j \in J, \forall q \in Q, \quad (9)$$

$$z_{jp} \leq y_{qj}, \forall p \in P, \forall j \in J, \forall q \in Q. \quad (10)$$

Условие (9) гарантирует, что если в узле не размещен сервер, то этому узлу невозможно адресовать запрос. Условие (10) гарантирует, что web-контент может реплицироваться только в тех узлах, в которых размещен сервер.

Наконец, следующие условия обеспечивают бинарности переменных:

$$z_{pj} \in \{0, 1\}, \forall p \in P, \forall j \in J, \quad (11)$$

$$z_{qi} \in \{0, 1\}, \forall q \in Q, \forall i \in I, \quad (12)$$

$$x_{ij} \in \{0, 1\}, \forall i \in I, \forall j \in J. \quad (13)$$

Теперь рассмотрим тот случай, когда для создания CDN выделен определенный бюджет. В этом случае задача формулируется таким образом. Из множества серверов следует выбрать те, в которых их стоимость и затраты, связанные с обслуживанием web-контентов в этих серверах, не превышали выделенный бюджет, и при оптимальном их размещении в потенциальных узлах, оптимальной репликации web-контентов в них и оптимальном назначении узлов по серверам была обеспечена минимальность стоимости доставки web-контентов конечным пользователям. Пусть  $B$  — выделенный бюджет. Тогда задача будет сведена к минимизации стоимости доставки web-контентов:

$$f(x) = \sum_{i \in I} \sum_{j \in J} \sum_{p \in P} c_{ij} v_p d_{ij} x_{ij} \rightarrow \min, \quad (14)$$

при тех же ограничениях (2)–(13), но с добавлением нового ограничения:

$$\sum_{j \in J} \left( \sum_{q \in Q} F_q y_{qj} + \sum_{p \in P} f_{pj} z_{pj} \right) \leq B. \quad (15)$$

Несмотря на то, что задачи (1)–(13) и (14), (2)–(13), (15) относятся к задачам линейного дискретного программирования, их решение при большой размерности требует огромных вычислительных ресурсов. Как правило, применение известных точных методов оптимизации к решению таких задач оказывается невозможным. По-

этому в последние годы находят применение приближенные методы оптимизации, среди которых перспективными зарекомендовали себя метаэвристические методы. Метаэвристический метод решения задачи будет описан в следующем разделе.

### 3. Эволюционный метод решения задачи дискретного программирования

В настоящее время для решения задач непрерывной и дискретной оптимизации активно изучаются и развиваются три группы эволюционных методов — генетические алгоритмы (*Genetics Algorithms*), алгоритм PSO (*Particle Swarm Optimization*) и муравьиные алгоритмы (*Ant Algorithms*). Эволюционные методы имеют много общего. Во-первых, они применимы для решения недифференцируемых задач оптимизации, так как не требуют существования градиента целевых функций. И во-вторых, в отличие от последовательных методов поиска они применяют параллельный метод поиска.

В этом разделе для решения задачи дискретной оптимизации будем применять алгоритм PSO.

#### 3.1. Базовый алгоритм PSO

Алгоритм PSO [11] относится к метаэвристическим методам оптимизации, использующим элементы случайности и ориентированным на поиск глобальных экстремумов. Подобно генетическим алгоритмам в алгоритмах PSO тоже сначала случайно инициализируется множество решений, которые в соответствии с принятой в генетических алгоритмах терминологией называют хромосомами (другие возможные названия — агенты). Алгоритм PSO имитирует поведение множества агентов, стремящихся согласовать свое состояние с состоянием наилучшего агента. Каждый агент в популяции описывается такими характеристиками: текущее положение, текущая скорость и лучшее персональное положение.

Текущее положение  $i$ -го агента описывается вектором  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ ,  $i = 1, 2, \dots, N$ , где  $x_{ij}$  —  $j$ -й параметр  $i$ -го агента,  $N$  — число агентов в популяции. Хромосома состоит из генов  $x_{ij}$ . Значения генов называют аллелями, а позиций генов в хромосоме — локусами. Положение  $i$ -го агента оценивается значением функции пригодности  $f(X_i)$ .

На итерации  $t$  каждый агент  $X_i(t) = (x_{i1}(t), \dots, x_{in}(t))$  стремится улучшить свое положение. Это стремление выражается в движении агента к лучшему положению со "скоростью"  $V_i(t) = (v_{i1}(t), \dots, v_{in}(t))$ . Ориентиром лучшего положения являются, во-первых, положение  $X_{swarm}^{best}$  самого лучшего агента популяции, во-вторых, лучшее положение  $X_i^{best}(t) = (x_{i1}^{best}(t), \dots, x_{in}^{best}(t))$  самого агента в процессе его движения.

Принцип работы алгоритма PSO заключается в итерационном вычислении значений  $X_i$ . В настоящее время существуют некоторые версии алгоритма PSO, среди которых наиболее популярной является модель *g-best (global best)*. Согласно этой модели на каждой итерации положение агента определяется по следующим правилам [11]:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1). \quad (16)$$

Скорость изменения положения агента определяется формулой

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_1(x_{ij}^{best}(t) - x_{ij}(t)) + c_2r_2(x_{swarm,j}^{best}(t) - x_{ij}(t)). \quad (17)$$

В формуле (17) коэффициент  $w$  представляет инерцию движения агента (в работе [12] рекомендовано  $w \in [0,4; 0,9]$ );  $x_{ij}^{best}(t)$  —  $j$ -й компонент лучшего положения агента  $X_i$  до  $t$ -й итерации;  $x_{ij}^{best}(t)$  —  $j$ -й компонент положения лучшего агента популяции на  $t$ -й итерации; параметры  $r_1$  и  $r_2$  — случайные числа, равномерно распределенные в интервале  $[0, 1]$ ; коэффициенты ускорения  $c_1$  и  $c_2$  характеризуют силы притяжения агента к лучшим положениям.

Формула (17) содержит три компонента, а именно [11, 13]:

- вес инертности  $w$ , который служит как память предыдущих скоростей. Коэффициент инертности контролирует влияние скорости предыдущей итерации. Большое значение веса инертности благоприятствует процессу предварительного исследования, а малое значение — процессу эксплуатации;
- когнитивный компонент  $(x_{ij}^{best}(t) - x_{ij}(t))$ , который представляет собой поведение агента для нахождения лучшего собственного положения (решения);
- социальный компонент  $(x_{swarm,j}^{best}(t) - x_{ij}(t))$ , который представляет собой поведение всей популяции для нахождения лучшего решения.

Для гарантированной сходимости алгоритма PSO в работе [14] установлено, что коэффициенты ускорения и инертности должны удовлетворять следующему неравенству:

$$\frac{1}{2}(c_1 + c_2) - 1 < w < 1. \quad (18)$$

Лучшее положение агента  $X_i$  на итерации  $(t+1)$  определяется таким образом:

$$X_i^{best}(t+1) = \begin{cases} X_i^{best}(t), & \text{если } f(X_i(t+1)) \geq f(X_i^{best}(t)), \\ X_i(t+1), & \text{если } f(X_i(t+1)) < f(X_i^{best}(t)). \end{cases} \quad (19)$$

На итерации  $t$  положение самого лучшего агента популяции, т. е. агента, имеющего лучшее минимальное *fitness*-значение, определяется таким образом:

$$X_{swarm}^{best}(t) = \min\{f(X_1(t)), \dots, f(X_N(t))\}. \quad (20)$$

Для сравнения с градиентным методом ниже предлагается видоизмененный вариант алгоритма PSO. После некоторых преобразований из формул (16) и (17) получается следующее выражение [12]:

$$x_{ij}(t+1) = x_{ij}(t) + wv_{ij}(t) + \alpha(\beta(t) - x_{ij}(t)), \quad (21)$$

где  $\alpha$  и  $\beta$  определяются формулами

$$\alpha = c_1r_1 + c_2r_2; \quad (22)$$

$$\beta(t) = \frac{c_1r_1x_{ij}^{best}(t) + c_2r_2x_{swarm,j}^{best}(t)}{c_1r_1 + c_2r_2}. \quad (23)$$

Подобно градиентному методу в формуле (21)  $(\beta(t) - x_{ij}(t))$  представляет собой направление поиска, а  $\alpha$  принимается как стохастический шаг, нижнее и верхнее значения которого равны 0 и  $c_1 + c_2$  соответственно, а среднее значение равно  $(c_1 + c_2)/2$ .

Базовый алгоритм PSO решает задачи непрерывной оптимизации. Для решения задач дискретной оптимизации базовый алгоритм PSO не применим. Это непосредственно видно из правил обновления положения и скорости (16), (17). Для решения задач дискретной оптимизации в дальнейших исследованиях предложены дискретные версии алгоритма PSO — Discrete PSO (DPSO). Дискретная версия алгоритма — бинарный PSO — впервые предложена в работе [15]. В бинарном алгоритме PSO положение агента принимает значение из множества  $\{0, 1\}$ . В зависимости от решаемых задач в работах [12, 13, 16] предложены разные версии дискретного алгоритма PSO. В следующем подразделе нами предлагается модификация алгоритма DPSO.

### 3.2. Алгоритм Max-Min DPSO

Каждый алгоритм начинается с кодирования агентов. В зависимости от решаемых задач агенты кодируются по-разному. В нашем случае агенты будем кодировать таким образом:

$$X_l(t) = (u^1, u^2, \dots, u^n)_l, \quad (24)$$

где каждый элемент представляется в виде  $u^i = (x_{i1}, x_{i2}, \dots, x_{im})$ ,  $x_{ij} \in \{0, 1\}$ ,  $i = 1, 2, \dots, n$ ,  $l = 1, 2, \dots, N$ . Наша задача относится к задачам с огра-

ничениями, и поэтому агенты будут генерироваться с соблюдением условий задачи:

$$1 \leq \sum_{j=1}^m x_{ij} \leq s, \forall i \in I, \quad (25)$$

$$\sum_{i \in I} \sum_{p \in P} d_{ijp} x_{ij} \leq y_{jq} C_q, \forall j \in J, \forall q \in Q. \quad (26)$$

Из правила (24) видно, что длина агента равна  $m \cdot n$ . В начале алгоритма значение генов генерируется по такому правилу:

$$x_{ij}(0) = \begin{cases} 0, & \text{если } \pi \geq \pi_0, \\ 1, & \text{если } \pi < \pi_0, \end{cases} \quad (27)$$

где  $\pi_0 \in [0, 1]$  — заранее определенная вероятность, а  $\pi$  — случайное число, равномерно распределенное в интервале  $[0, 1]$ , которое для каждого гена генерируется заново.

В случае дискретной оптимизации правило изменения скорости агента остается в том же виде (17), а формула (16) преобразуется в следующий вид [11, 15, 16]:

$$x_{ij}(t+1) = \begin{cases} 1, & \text{если } pr < \text{sig}(v_{ij}(t+1)), \\ 0, & \text{если } pr \geq \text{sig}(v_{ij}(t+1)), \end{cases} \quad (28)$$

где  $pr$  — случайное число, равномерно распределенное в интервале  $[0, 1]$ , а  $\text{sig}(\theta)$  — сигмоидальная функция:

$$\text{sig}(\theta) = \frac{1}{1 + \exp(-\theta)}. \quad (29)$$

Значение функции (29) стремится к 0,5 при значениях параметра  $\theta$ , близких нулю, стремится к 1,0 при больших положительных значениях параметра  $\theta$ , со сглаженным переходом между теми и другими.

Аналогично муравьиному алгоритму Max-Min [17] во избежание стагнации (преждевременной сходимости к локальному минимуму) правило изменения скорости (17) будем модифицировать следующим образом:

$$v_{ij}(t+1) = \text{MaxMin}(wv_{ij}(t) + c_1 r_1 (x_{ij}^{best}(t) - x_{ij}(t)) + c_2 r_2 (x_{swarm,j}^{best}(t) - x_{ij}(t))). \quad (30)$$

Функция  $\text{MaxMin}(u)$  определяется таким образом:

$$\text{MaxMin}(u) = \begin{cases} v_{\min}, & \text{если } u < v_{\min}, \\ u, & \text{если } v_{\min} \leq u \leq v_{\max}, \\ v_{\max}, & \text{если } u > v_{\max}. \end{cases} \quad (31)$$

Алгоритм Max-Min DPSO отличается от базового дискретного алгоритма следующими правилами: скорость изменения положения агентов ог-

раничивается диапазоном  $v \in [v_{\min}, v_{\max}]$ ; в начале работы алгоритма скорость каждого агента принимается равной  $v_{\max}$ . Ограничение на значение скорости позволяет избежать стагнации.

По результатам сказанного выше предложенный дискретный алгоритм PSO конкретизируется так:

**Шаг 1.** Задать параметры алгоритма: вес инертности  $w$ , коэффициенты ускорения  $c_1$  и  $c_2$ , параметры  $r_1$  и  $r_2$ , максимальное число итераций  $t_{\max}$ , число агентов  $N$ .

**Шаг 2.** Установить  $t = 0$  и по правилам (24)–(27) инициализировать начальную популяцию, скорости агентов присваивать начальное значение, равное  $v_{\max}$ .

**Шаг 3.** По правилу (14) вычислять значение полезности каждого агента, т. е. оценивать их положение.

**Шаг 4.** По правилам (19) и (20) найти лучшее положение агента и лучшее положение популяции.

**Шаг 5.** Для каждого агента применить правила обновления скорости (30) и (31).

**Шаг 6.** Проверить выполнение условий (25) и (26).

**Шаг 7.** Определить положение агента по правилу (28).

**Шаг 8.** Увеличить число итераций  $t = t + 1$ .

**Шаг 9.** Если  $t \leq t_{\max}$ , перейти к шагу 3. В противном случае завершить алгоритм и за оптимальное решение задачи принять  $X_{swarm}^{best}(t)$ .

#### 4. Обновление web-контентов в серверах-суррогатах

В сетях CDN одной из острых проблем является своевременное обновление информации в серверах-суррогатах. Относительно других видов информации (аудио, видео), как правило, чаще всех изменяется текстовая информация. Для определения критерия обновления текстовой информации ниже предлагается мера, основанная на метрике Левенштейна.

Степень изменения документа будем вычислять расстоянием Левенштейна (редактирование). Первоначально расстояние Левенштейна было введено для определения расстояния между словами. Расстояние Левенштейна равно минимальному числу элементарных операций редактирования, необходимых для преобразования одной строки в другую. Набор элементарных операций состоит из операций замены, вставки и удаления одной буквы. Для вычисления расстояния Левенштейна каждое слово должно представляться в виде набора символов. Расстояние Левенштейна между словами  $a = (a_1, \dots, a_N)$  и  $b = (b_1, \dots, b_M)$  вычисляется следующей рекурсивной формулой [18]:

$$L((a_1, \dots, a_N), (b_1, \dots, b_M)) = \begin{cases} N, & \text{если } M = 0, \\ M, & \text{если } N = 0, \\ \min\{L((a_1, \dots, a_{N-1}), (b_1, \dots, b_M)) + 1, \\ L((a_1, \dots, a_N), (b_1, \dots, b_{M-1})) + 1, & \text{в других случаях} \\ L((a_1, \dots, a_{N-1}), (b_1, \dots, b_{M-1})) + \bar{\delta}(a_n, b_M)\}. \end{cases} \quad (32)$$

В формуле (32)  $\bar{\delta}(x, y)$  является отрицанием символа Кронекера:

$$\bar{\delta}(x, y) = \begin{cases} 0, & \text{если } x = y, \\ 1, & \text{если } x \neq y. \end{cases}$$

Для каждого расстояния  $L((a_1, \dots, a_i), (b_1, \dots, b_j))$ ,  $i = 1, \dots, N, j = 1, \dots, M$ , используем сокращенную запись  $L_{ij}$ . В частности, имеем  $L(a, b) = L_{NM}$ . Для вычисления  $L_{NM}$  должны вычисляться все расстояния  $L_{ij}$ ,  $i = 0, 1, \dots, N, j = 0, 1, \dots, M, (i, j) \neq (N, M)$ . Достаточно трудоемко вычислять расстояние Левенштейна по рекурсивной формуле (32). Сложность такого вычисления растет экспоненциально с размером слов. Вместо рекурсивного вычисления (31) расстояние  $L_{ij}$  можно вычислить итеративно. В итеративном алгоритме сначала инициализируются расстояния  $L_{i0} = i, i = 1, \dots, N$  и  $L_{0j} = j, j = 1, \dots, M$ . Потом по значениям  $L_{(i-1)j}$ ,  $L_{i(j-1)}$  и  $L_{(i-1)(j-1)}$  последовательно вычисляется расстояние  $L_{ij}$ ,  $i = 1, \dots, N, j = 1, \dots, M$ .

Для вычисления близости двух текстовых документов расстояние Левенштейна не является адекватной мерой близости. Дело в том, что перестановка отдельных слов в предложении или предложений в документе не воспринимается человеком как существенное изменение, поэтому при определении близости двух текстовых документов из набора элементарных операций будем исключать операции замены.

Пусть  $L(D, D^*)$  — минимальное число элементарных операций удаления или вставки слов, необходимых для преобразования одного документа в другой. Тогда расстояние похожести между документом  $D$  и его копиями  $D^*$  будем определять так:

$$\text{dist}(D, D^*) = \frac{L(D, D^*)}{\lambda + \lambda^*}, \quad (33)$$

где  $\lambda$  и  $\lambda^*$  — длина (в словах) документов  $D$  и  $D^*$  соответственно.

Из определения видно, что  $0 \leq \text{dist}(D, D^*) \leq 1$ . Между идентичными документами расстояние (33) равно нулю, а между различными документами оно равно единице.

### Заключение

Сеть Internet за короткое время достигла значительных размеров, однако методы доставки ин-

формации, которые легли в основу Web, не в полной мере соответствуют его нынешнему состоянию, поэтому в последние годы возникла необходимость в разработке новой технологии. Сети CDN являются одной из таких технологий, позволяющих улучшить QoS-показатели пользователей и эффективнее использовать ресурсы сети Internet. CDN является технической основой для предоставления одноименной услуги в обход сети Internet. Данная технология обеспечивает пользователя высококачественной информацией, одновременно уменьшая трафик на опорной сети Internet. CDN — это выделенная сеть серверов, размещенных в сети Internet, которую контент-провайдеры могут использовать для распределения своего web-контента. По сути CDN представляет собой географически распределенную наложенную сеть, обеспечивающую быстрый и надежный поиск информации из любого месторасположения пользователя. Сети CDN используют технологии, относящиеся к инфраструктуре и требующие решения непростых задач, что позволяет существенно улучшить работоспособность опорной сети, в частности корпоративной.

Исходя из сказанного, в работе исследуется вопрос применения CDN-технологии как интенсивный путь решения повышения работоспособности корпоративных сетей. В частности, рассматривается вопрос минимизации расходов при эксплуатации корпоративных сетей, где получены следующие результаты.

1. Для улучшения работы корпоративных сетей предложена математическая модель, позволяющая определить оптимальные точки размещения серверов CDN в узлах корпоративных сетей и оптимальную репликацию web-контентов в этих серверах. Другое преимущество предложенной модели заключается в том, что она описывается задачей линейного дискретного программирования.

2. В работе разработан алгоритм решения задачи дискретного программирования, основанный на метаэвристическом методе оптимизации PSO. Для улучшения работоспособности в работе предложена модификация алгоритма DPSO.

3. В работе предложена метрика для обнаружения изменения в контент-репликах, которая может быть полезной при определении меры близости текстовых документов.

## Список литературы

1. Куроуз Д. Ф., Росс К. В. Компьютерные сети (2-е изд.). СПб.: Питер, 2004. 765 с.
2. Таненбаум Э., ван Стен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.
3. Pathan A. M. K., Buaya R. A taxonomy and survey of content delivery networks // Technical Report GRIDS-TR-2007-4. Grid Computing and Distributed Systems Laboratory. University of Melbourne. Australia. 2007. 44 p.
4. Peng G. CDN: content distribution network // Technical Report TR-125 of Experimental Computer Systems Laboratory Department of Computer Science. State University of New York. Stony Brook. NY. USA. 2003. Nov. 26 p.
5. Pallis G., Vakali A. Insight and perspectives for content delivery networks // Communications of the ACM. 2006. Vol. 49. № 1. P. 101–106.
6. Алгулиев Р. М., Алыгулиев Р. М., Шарифов М. Г. Модель оптимального размещения серверов и web-контентов в Content delivery networks // Автоматика и вычислительная техника. 2006. Vol. 40. № 4. С. 42–50.
7. Алгулиев Р. М., Алыгулиев Р. М., Шарифов М. Г. Подход к оптимальному расположению серверов CDN в узлах глобальных сетей // Информационные технологии. 2006. № 11. С. 20–26.
8. Bektas T., Oguz O., Ouveysi I. Designing cost-effective content distribution networks // Computers and Operations Research. 2007. Vol. 34. № 8. P. 2436–2449.
9. Wauters T., Coppens J., De Turck F., Dhoedt B., Demeester P. Replica placement in ring based content delivery networks // Computer Communications. 2006. Vol. 29. № 16. P. 3313–3326.
10. Шарифов М. Г. Минимизация стоимости доставки в Content delivery networks с многоуровневой иерархической архитектурой // Автоматика и вычислительная техника. 2006. Vol. 40. № 6. С. 75–80.
11. Kennedy J., Eberhart R. C. Swarm intelligence. San Francisco: Morgan Kaufmann, 2001. 512 p.
12. Kitayama S., Arakawa M., Yamazaki K. Penalty function approach for the mixed discrete nonlinear problems by particle swarm optimization // Structural and Multidisciplinary Optimization. 2006. Vol. 32. № 3. P. 191–202.
13. Omran M. G. H., Salman A., Engelbrecht A. P. Dynamic clustering using particle swarm optimization with application in image segmentation // Pattern Analysis and Applications. 2006. Vol. 8. № 4. P. 332–344.
14. Van den Bergh F. An analysis of particle swarm optimizers. PhD thesis. Department of Computer Science, University of Pretoria. 2002. 300 p.
15. Kennedy J., Eberhart R. C. A discrete binary version of the particle swarm algorithm // Proc. of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Orlando. Florida. USA. 12–15 October 1997. Vol. 5. P. 4104–4108.
16. Correa E. S., Freitas A. A., Johnson C. G. A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set // Proc. of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06). 8–12 July 2006. Seattle. Washington. USA. P. 35–42.
17. Stützle T., Hoos H. H. Max-Min ant system // Future Generation Computer Systems. 2000. Vol. 16. № 8. P. 889–914.
18. Runkler T. A., Bezdek J. C. Web mining with relational clustering // International Journal of Approximate Reasoning. 2003. Vol. 32. № 2–3. P. 217–236.

УДК 621.396.2

**Е. А. Самойлин**, канд. техн. наук,  
преподаватель,

Ростовский военный институт ракетных войск  
им. Главного маршала артиллерии  
М. И. Неделина  
e-mail: sea@ctsnet.ru

## Синтез оптимальных алгоритмов оценивания импульсных помех при восстановлении сигналов в цифровых радиосистемах

*Синтезированы оптимальные по критерию Неймана—Пирсона алгоритмы оценивания импульсных помех, возникающих в приемных устройствах цифровых радиосистем вследствие многих условий радиовещания и преобразования цифровой информации. Алгоритмы позволяют получить оптимальную оценку помех, используемую в двухэтапных процедурах восстановления цифровых информационных процессов.*

**Ключевые слова:** импульсные помехи, оптимальное оценивание, восстановление сигналов, цифровые радиосистемы.

В настоящее время стремительно развиваются и внедряются цифровые технологии радиосвязи, обладающие несомненными преимуществами перед

традиционными аналоговыми. Между тем условия функционирования многих цифровых радиоэлектронных средств, а также сами принципы цифрового преобразования информации являются в некоторых случаях причиной появления в принятом информационном процессе импульсных помех [1–3]. Импульсные помехи (ИП) представляют собой замену полезных дискретов на помеховые, не коррелированные по сигнальной и пространственной областям. Необходимо отметить, что случай ИП, по сравнению с аддитивными или мультипликативными, является наиболее сложным для синтеза алгоритмов восстановления исходной информации.

В настоящее время известно [1–4] достаточно много процедур восстановления информационных процессов, искаженных ИП, наиболее приемлемой из которых является медианная обработка, предложенная в [4]. Между тем, как известно [5], медианная фильтрация представляет собой инвариантную процедуру (в данном случае по времени), что неизбежно вызывает дополнительные искажения восстанавливаемого сигнала. Снизить указанные искажения позволяет использование не инвариантной по времени обработки, развитой в работе [5] и представляющей собой двухэтапную процедуру, на первом из которых происходит оценивание ИП, а на втором — восстановление исключительно помеховых дискретов. Между тем предложенный в работе [5] алгоритм оценивания

ИП является неоптимальным, так как порог обнаружения устанавливается экспериментально.

Таким образом, задача оптимизации процедуры оценивания ИП при восстановлении информации в приемных устройствах цифровых радиоэлектронных систем является актуальной.

Цель предлагаемой работы — синтез оптимальных алгоритмов оценивания ИП при восстановлении информации в цифровых радиоэлектронных системах.

### Постановка задачи

Модель дискретного информационного сигнала  $\lambda(t)$ , формируемого в передающих радиосистемах, имеет вид [5]

$$\lambda(t) = \Lambda(t\delta T), \quad (1)$$

где  $\Lambda(t\delta T)$  — непрерывный сигнал, соответствующий  $\lambda(t)$ ;  $\delta T$  — интервал дискретизации  $t$ .

Частота квантования звука сигнала (1) может лежать на интервале [5]

$$\lambda(t) \in [0, \dots, 2^N - 1], \quad N \in [1, \dots, 16], \quad (2)$$

где  $N$  — число битов квантования звука.

При радиопередаче каждому значению  $\lambda(t)$  ставится в соответствие определенный код, который модулирует высокочастотный сигнал, излучаемый антенной.

В приемных цифровых радиосистемах происходят демодуляция и обратное преобразование кода в информационный сигнал. При этом вследствие различных ошибок оцифровки сигнала, модуляции и демодуляции и т. д. принятое входное наблюдение содержит импульсные искажения сигнала, описываемые выражением [5]

$$x_t = \begin{cases} \lambda_t \text{ с вероятностью } p(\lambda); \\ h_t \text{ с вероятностью } p(h) = 1 - p(\lambda), \end{cases} \quad (3)$$

где  $p(\lambda)$  — вероятность появления полезного сигнала  $\lambda_t$  в дискрете  $t$ ;  $h_t$  — значения искаженных дискретов, которые являются независимыми случайными величинами с равномерным распределением (rnd), соответствующие интервалу квантования (2);  $p(h)$  — вероятность появления выброса ИП  $h_t$  в дискрете  $t$ , которая не коррелирует ни с шумом в других дискретах, ни с исходным информационным процессом, т. е.  $p(h) = \text{const} \forall t \in 0, \dots, n$ . Появления  $\lambda_t$  и  $h_t$  образуют полную группу несовместных событий для каждого дискрета  $t$ .

Для оценивания ИП  $h_t$  во входном наблюдении  $x_t$  можно воспользоваться выражением [6], согласно которому дискрет принимается за помеху, если его уровень не принадлежит допустимому сигнальному интервалу:

$$\tilde{h}_t = \begin{cases} 1, & x_t \notin [x_{\min}, x_{\max}]; \\ 0, & x_t \in [x_{\min}, x_{\max}], \end{cases} \quad (4)$$

где  $x_{\min}, x_{\max}$  — соответственно минимальное и максимальное значения сигнального интервала, устанавливаемые на основе распределения сигналов смежных дискретов, попавших в скользящее линейное окно или апертуру.

В соответствии с (4) задачу оценивания помех во входном наблюдении можно сформулировать в следующем виде. Необходимо синтезировать алгоритм получения оптимальных пороговых значений  $x_{\min}, x_{\max}$  (4), позволяющий сформировать бинарную оценку  $\hat{h}_t \in [0, 1]$  помех во входном наблюдении  $x_t$  (3), наиболее близкую к ее истинным значениям:

$$[x_{\min}, x_{\max}] = \arg \min_{x_{\min}, x_{\max} \in [0, (2^N - 1)]} \{ \|\tilde{h}_t - \hat{h}_t\| \}, \quad (5)$$

где  $\hat{h}_t$  бинарное помеховое наблюдение

$$\hat{h}_t = \begin{cases} 1, & x_t = h_t; \\ 0, & x_t = \lambda_t \end{cases}$$

### Синтез оптимальных по критерию

#### Неймана—Пирсона алгоритмов оценивания помех

Обозначим через  $f(x/x = \lambda)$  плотность распределения сигнала  $\lambda_r$ , а через  $f(x/x = h)$  — плотность распределения помехи в наблюдении  $x_r$ . Тогда условная вероятность ложного обнаружения по (4) помехи будет иметь вид

$$P_{\text{ло}} = \frac{\sum_{x=0}^{x_{\min}} f(x/x = \lambda) + \sum_{x=x_{\max}}^{2^N - 1} f(x/x = \lambda)}{\sum_{x=0}^{2^N - 1} f(x/x = \lambda)}. \quad (6)$$

Условная вероятность принятия правильного решения об отсутствии помехи будет описываться выражением

$$P_{\text{пр.отс}} = \frac{\sum_{x=x_{\min}}^{2^N - 1} f(x/x = \lambda)}{\sum_{x=0}^{2^N - 1} f(x/x = \lambda)}, \quad (7)$$

причем  $P_{\text{ло}} + P_{\text{пр.отс}} = 1$ .

Условные вероятности пропуска и принятия правильного решения об обнаружении помехи будут иметь соответственно следующий вид:

$$P_{\text{проп}} = \frac{\sum_{x=x_{\min}}^{x_{\max}} f(x/x = h)}{\sum_{x=0}^{2^N - 1} f(x/x = h)}; \quad (8)$$

$$P_{\text{пр.обн}} = \frac{\sum_{x=0}^{x_{\min}} f(x/x=h) + \sum_{x=x_{\max}}^{2^N-1} f(x/x=h)}{\sum_{x=0}^{2^N-1} f(x/x=h)}, \quad (9)$$

причем  $P_{\text{проп}} + P_{\text{пр.обн}} = 1$ .

С учетом  $p(h)$  и условных вероятностей (6)–(9) запишем выражения для безусловных вероятностей ошибок и правильных решений при обнаружении помех. Вероятность ложных тревог будет иметь вид

$$P_{\text{лт}} = (1 - p(h))P_{\text{ло}} = (1 - p(h)) \times \left[ \frac{\sum_{x=0}^{x_{\min}} f(x/x=\lambda) + \sum_{x=x_{\max}}^{2^N-1} f(x/x=\lambda)}{\sum_{x=0}^{2^N-1} f(x/x=\lambda)} \right]. \quad (10)$$

Вероятность принятия правильного решения об отсутствии помехи

$$P_{\text{по}} = (1 - p(h))P_{\text{пр.отс}} = (1 - p(h)) \left[ \frac{\sum_{x=x_{\min}}^{x_{\max}} f(x/x=\lambda)}{\sum_{x=0}^{2^N-1} f(x/x=\lambda)} \right]. \quad (11)$$

Вероятность пропуска помехи будет определяться выражением

$$P_{\text{пр}} = p(h)P_{\text{проп}} = p(h) \left[ \frac{\sum_{x=x_{\min}}^{x_{\max}} f(x/x=h)}{\sum_{x=0}^{2^N-1} f(x/x=h)} \right]. \quad (12)$$

Вероятность принятия правильного решения о наличии помехи

$$P_{\text{пн}} = p(h)P_{\text{пр.обн}} = p(h) \left[ \frac{\sum_{x=0}^{x_{\min}} f(x/x=h) + \sum_{x=x_{\max}}^{2^N-1} f(x/x=h)}{\sum_{x=0}^{2^N-1} f(x/x=h)} \right]. \quad (13)$$

Очевидно, что  $P_{\text{лт}} + P_{\text{по}} + P_{\text{пр}} + P_{\text{пн}} = 1$ .

**Минимизация вероятности пропуска при заданной вероятности ложной тревоги.** С учетом (10) и (12) выражение для искомым границ интервала (4), минимизирующих вероятность пропуска при за-

данной вероятности ложной тревоги, будет выглядеть следующим образом:

$$[x_{\min}, x_{\max}] = \arg \min \left\{ p(h) \left[ \frac{\sum_{x=x_{\min}}^{x_{\max}} f(x/x=h)}{\sum_{x=0}^{2^N-1} f(x/x=h)} \right] \right\} \quad (14)$$

при

$$P_{\text{лт}} \geq (1 - p(h)) \left[ \frac{\sum_{x=0}^{x_{\min}} f(x/x=\lambda) + \sum_{x=x_{\max}}^{2^N-1} f(x/x=\lambda)}{\sum_{x=0}^{2^N-1} f(x/x=\lambda)} \right]. \quad (15)$$

Как видно из (14), для минимизации его правой части необходимо  $x_{\min} \rightarrow \max$ ,  $x_{\max} \rightarrow \min$ , однако это вызовет рост  $P_{\text{лт}}$  (15). Поэтому значения  $x_{\min}$ ,  $x_{\max}$  найдем из ограничения (15). Так как распределение  $f(x/x=\lambda)$  на интервале  $0 \dots (2^N - 1)$  равно единице, перепишем (15) в виде

$$P_{\text{лт}} \geq (1 - p(h)) \left[ 1 - \frac{\sum_{x=x_{\min}}^{x_{\max}} f(x/x=\lambda)}{\sum_{x=0}^{2^N-1} f(x/x=\lambda)} \right]. \quad (16)$$

Поскольку распределение  $f(x/x=\lambda)$  в линейной апертуре, скользящей по дискретам наблюдения, лежит в границах от минимального  $x_{\min}^A$  до максимального  $x_{\max}^A$  значения сигнала, выражение (16) можно записать в виде

$$P_{\text{лт}} \geq (1 - p(h)) \left[ 1 - \frac{x_{\max} - x_{\min}}{x_{\max}^A - x_{\min}^A} \right]. \quad (17)$$

Искомые оптимальные величины  $x_{\min}$ ,  $x_{\max}$  равноудалены на  $\Delta$  от некоторого среднего значения, определяемого как среднее текущей апертуры  $\bar{x}_A$ . В соответствии с этим (17) приобретает вид

$$P_{\text{лт}} \geq (1 - p(h)) \left[ 1 - \frac{(\bar{x}_A + \Delta) - (\bar{x}_A - \Delta)}{x_{\max}^A - x_{\min}^A} \right], \quad (18)$$

или

$$P_{\text{лт}} \geq (1 - p(h)) \left[ 1 - \frac{2\Delta}{x_{\max}^A - x_{\min}^A} \right]. \quad (19)$$

Определим из выражения (19) оптимальный параметр  $\Delta$ :

$$\Delta \geq \frac{(x_{\max}^A - x_{\min}^A) \left( \frac{P_{\text{лт}}}{p(h) - 1} + 1 \right)}{2}. \quad (20)$$

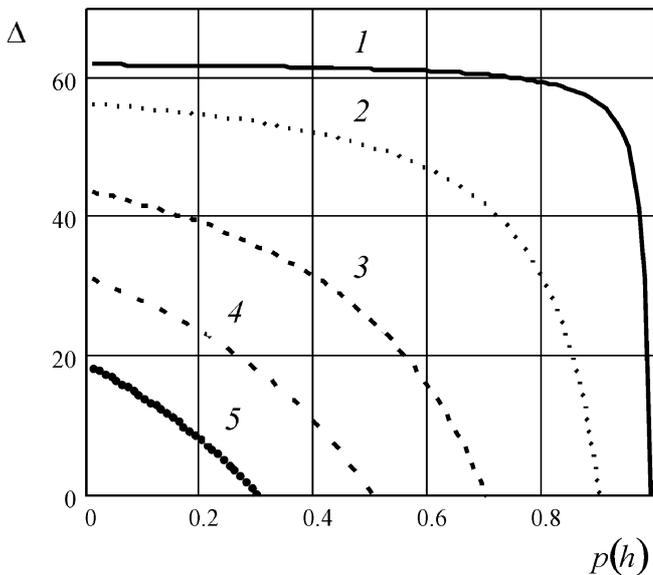


Рис. 1. Зависимость оптимального по (21) значения  $\Delta$  от  $p(h)$  и  $P_{лт}$  при:  
 1 —  $P_{лт} = 0,01$ ; 2 —  $P_{лт} = 0,1$ ; 3 —  $P_{лт} = 0,3$ ; 4 —  $P_{лт} = 0,5$ ;  
 5 —  $P_{лт} = 0,7$

Так как согласно (14) необходимо  $\Delta \rightarrow \min$ , а согласно (20) —  $\Delta \rightarrow \max$ , то оптимальное значение  $\Delta$  будет определяться выражением

$$\Delta = \frac{(x_{\max}^A - x_{\min}^A) \left( \frac{P_{лт}}{p(h) - 1} + 1 \right)}{2}. \quad (21)$$

Таким образом, оптимальное удаление значений  $x_{\min}$ ,  $x_{\max}$  от среднего по апертуре  $\bar{x}_A$

$$\text{составит } \begin{cases} x_{\min} = \bar{x}_A - \Delta; \\ x_{\max} = \bar{x}_A + \Delta. \end{cases}$$

Рассмотрим частный случай, когда, например, интервал от  $x_{\min}^A$  до  $x_{\max}^A$  составляет 50 % от полного интервала квантования уровня (2) при  $N = 8$ , т. е. около 125. При этом зависимость  $\Delta$  от  $p(h)$  и  $P_{лт}$  будет иметь вид в соответствии с рис. 1.

**Минимизация вероятности ложной тревоги при заданной вероятности пропуска.** С учетом (10) и (12) выражение для искомых границ интервала (4), минимизирующих вероятность ложной тревоги при заданной вероятности пропуска, будет иметь вид

$$[x_{\min}, x_{\max}] = \arg \min \times \left\{ (1 - p(h)) \left[ \frac{\sum_{x=0}^{x_{\min}} f(x/x = \lambda) + \sum_{x=x_{\max}}^{2^N-1} f(x/x = \lambda)}{\sum_{x=0}^{2^N-1} f(x/x = \lambda)} \right] \right\} \quad (22)$$

$x_{\min}, x_{\max} \in [0, (2^N - 1)]$

при

$$P_{пр} \geq p(h) \left[ \frac{\sum_{x=x_{\min}}^{x_{\max}} f(x/x = h)}{\sum_{x=0}^{2^N-1} f(x/x = h)} \right]. \quad (23)$$

Как видно из (22), для минимизации его правой части необходимо  $x_{\min} \rightarrow 0$ ,  $x_{\max} \rightarrow 2^N - 1$ , однако это вызовет повышение  $P_{пр}$  (23). Поэтому значения  $x_{\min}$ ,  $x_{\max}$  найдем из ограничения (23). Так как распределение помехи  $f(x/x = h)$  равномерное на интервале  $0 \dots (2^N - 1)$ , перепишем (23) в виде

$$P_{пр} \geq p(h) \left[ \frac{x_{\max} - x_{\min}}{(2^N - 1) - 0} \right]. \quad (24)$$

По аналогии с (18) искомые величины  $x_{\min}$ ,  $x_{\max}$  равноудалены на  $\Delta$  от среднего текущей апертуры  $\bar{x}_A$ . В соответствии с этим (24) примет следующий вид:

$$P_{пр} \geq p(h) \left[ \frac{2\Delta}{2^N - 1} \right], \quad (25)$$

откуда имеем:

$$\Delta \leq \frac{P_{пр}(2^N - 1)}{2p(h)}. \quad (26)$$

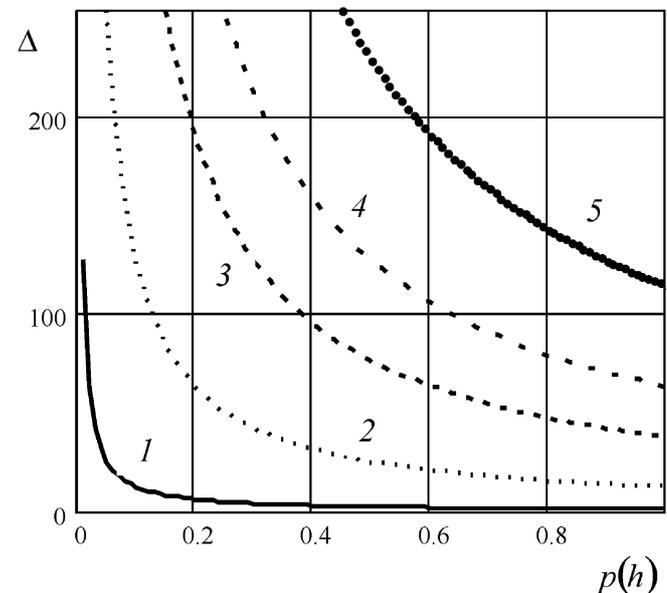


Рис. 2. Зависимость оптимального по (27) значения  $\Delta$  от  $p(h)$  и  $P_{пр}$  при:  
 1 —  $P_{пр} = 0,01$ ; 2 —  $P_{пр} = 0,1$ ; 3 —  $P_{пр} = 0,3$ ; 4 —  $P_{пр} = 0,5$ ;  
 5 —  $P_{пр} = 0,9$

Так как согласно (22) необходимо  $\Delta \rightarrow \max$ , а согласно (26) —  $\Delta \rightarrow \min$ , то оптимальное значение  $\Delta$  будет следовать из выражения (26):

$$\Delta = \frac{P_{\text{пр}}(2^N - 1)}{2p(h)}. \quad (27)$$

Таким образом, оптимальный в соответствии с (22)—(27) интервал значений  $x_{\min}$ ,  $x_{\max}$  также составит

$$\begin{cases} x_{\min} = \bar{x}_A - \Delta; \\ x_{\max} = \bar{x}_A + \Delta. \end{cases}$$

Рассмотрим в качестве примера частный случай 8-битного квантования сигнала  $x_p$ , т. е.  $N = 8$ . При этом зависимость  $\Delta$  от  $p(h)$  и  $P_{\text{пр}}$  (27) будет иметь вид в соответствии с рис. 2.

Таким образом, оптимальные по критерию Неймана—Пирсона границы допустимого интервала при обнаружении ИП в информационном процессе для случая минимизации вероятности пропуска при заданной вероятности ложной тревоги определяются выражением (21), а для случая минимизации вероятности ложной тревоги при заданной вероятности пропуска — выражением (27).

### Заключение

Предложенные алгоритмы оценивания ИП в информационных процессах являются опти-

мальными по критерию Неймана—Пирсона и заключаются в минимизации вероятности пропуска при заданной вероятности ложной тревоги либо минимизации вероятности ложной тревоги при заданной вероятности пропуска. Рассмотренные алгоритмы достаточно легко реализуемы программно и аппаратно и могут использоваться в приемных устройствах различных радиоэлектронных систем цифровой связи при восстановлении информационных процессов.

*Работа выполнена при государственной финансовой поддержке по гранту Президента Российской Федерации № МК-3603.2007.10.*

### Список литературы

1. Ратынский М. В. Основы сотовой связи / Под ред. Д. Е. Зимина. М.: Радио и связь, 1998. 248 с.
2. Финк Л. М. Теория передачи дискретных сообщений. М.: Сов. радио, 1970. 728 с.
3. Акимов П. С., Евстратов Ф. Ф., Захаров С. И. и др. Обнаружение радиосигналов / Под ред. А. А. Колосова. М.: Радио и связь, 1989. 288.
4. Радченко Ю. С. Эффективность приема сигналов на фоне комбинированной помехи с дополнительной обработкой в медианном фильтре // Журнал радиоэлектроники. 2001. № 7. <http://jre.cplire.ru/jre.jul01/index.html>
5. Самойлин Е. А. Алгоритмы неинвариантно-временного восстановления сигналов при цифровом радиоприеме // Информационные технологии. 2006. № 8. С. 59—62.
6. Адаптивные методы обработки изображений: Сб. научных тр. / Под ред. В. И. Сифорова, Л. П. Ярославского. М.: Наука, 1988. 245 с.

УДК 004.451.83

А. О. Домнин, аспирант МИИТ,  
e-mail: retray2000@mail.ru

## Аналитическая модель клиент-серверного приложения

*Предлагается математическая модель обслуживания запроса в сетевом приложении, представленная замкнутой сетью систем массового обслуживания.*

**Ключевые слова:** аналитическая модель, теория массового обслуживания, теория очередей, модель клиент-сервер, производительность распределенных систем.

### Введение

Клиент-серверные приложения — программная база многих современных предприятий [1]. Актуальным является определение производительности этих приложений до ввода в эксплуатацию, что возможно путем аналитического моделирования.

К настоящему моменту уже разработано несколько моделей сетевых приложений [2, 3]. Но

эти модели либо не анализируют производительность, либо представляют структуру сервера очень упрощенно, в основном, как комбинацию устройств процессор и жесткий диск, что не позволяет предсказать "узкие места" в работе системы, связанные с оперативной памятью и кэш.

Обычно производительность распределенных приложений определяется временем отклика — интервалом между регистрацией конца передачи сообщения запроса и началом получения сообщения ответа к станции, порождающей запрос.

Время отклика определяется из известного соотношения [4]:

$$t_{\text{откл}} = T_p + S + W,$$

где  $T_p$ ,  $S$  и  $W$ , соответственно, время распространения, время обслуживания и время ожидания в очереди к ресурсам. При больших нагрузках основной вклад дает время ожидания в очередях ( $W$ ), поэтому именно данный параметр нуждается в расчетах.

## Модель

Корпоративную сеть с работающим клиент-серверным приложением можно представить сетью систем массового обслуживания (рис. 1).

На рис. 1 отдельные СМО это:

СМО1 — источник запросов, пользователи; СМО2 — сетевая карта; СМО3 — оперативная память; СМО4 — процессор; СМО5 — кэш; СМО6 — жесткий диск.

Дуги показывают направление движения запросов и вероятности передач. Отправленный пользователем запрос передается через сеть и принимается сетевой картой сервера, команды запроса загружаются в оперативную память. Далее происходит цикл обработки запроса — несколько последовательных обращений к процессору и жесткому диску для чтения/записи информации, при этом с вероятностью  $P_{36}$  данные будут считаны из кэш. Обмен данными происходит через оперативную память. Вероятности передач  $P_{xx}$  определяются настройками программного обеспечения серверной части и размерами считываемых/записываемых запросом данных.

Соответствующая графу матрица вероятностей передач:

	С	1	2	3	4	5	6
$P =$	МО						
	1	0	1	0	0	0	0
	2	$P_{21}$	0	$P_{23}$	0	0	0
	3	0	$P_{32}$	0	$P_{34}$	$P_{35}$	$P_{36}$
	4	0	0	1	0	0	0
	5	0	0	1	0	0	0
	6	0	0	1	0	0	0

Сеть замкнутая, число запросов, циркулирующих в сети, равно числу пользователей  $N$ . Каждая СМО рассматривается как независимо функционирующая типа  $M/M/m_i/\infty$ . На вход каждой СМО поступает пуассоновский поток запросов с интенсивностью  $\lambda_i$ , которые обрабатываются с интен-

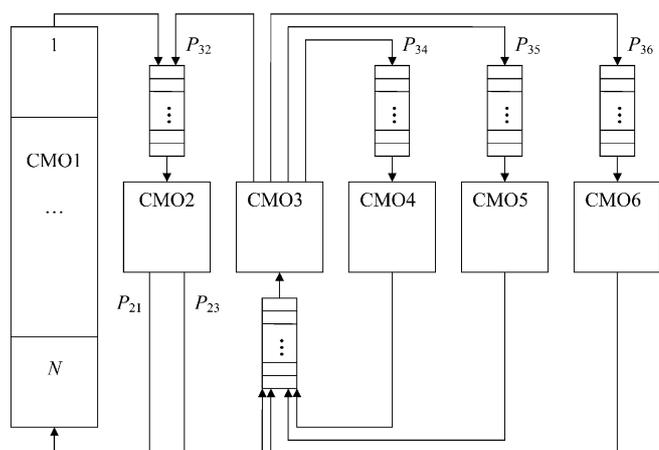


Рис. 1. Замкнутая сеть СМО

сивностью  $\mu_i$  в течение времени, распределенного по экспоненциальному закону.

Интенсивность потока запросов  $\lambda_i$ , поступающих на вход СМО $_i$ , равна интенсивности выходного потока обработанных запросов, исходя из этого получаем систему уравнений:

$$\begin{cases} \lambda_1 = P_{21}\lambda_2, \\ \lambda_2 = \lambda_1 + P_{32}\lambda_3, \\ \lambda_3 = P_{23}\lambda_2 + \lambda_4 + \lambda_5 + \lambda_6, \\ \lambda_4 = P_{34}\lambda_3, \\ \lambda_5 = P_{35}\lambda_3, \\ \lambda_6 = P_{36}\lambda_3. \end{cases}$$

Если просуммировать последние четыре уравнения и сделать необходимые преобразования, можно получить первое:

$$\begin{aligned} \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 &= \lambda_1 + P_{32}\lambda_3 + P_{23}\lambda_2 + \\ &+ \lambda_4 + \lambda_5 + \lambda_6 + P_{34}\lambda_3 + P_{35}\lambda_3 + P_{36}\lambda_3; \\ \lambda_1 &= \lambda_2 + \lambda_3 - P_{32}\lambda_3 - P_{23}\lambda_2 - \\ &- P_{34}\lambda_3 - P_{35}\lambda_3 - P_{36}\lambda_3; \\ \lambda_1 &= \lambda_2(1 - P_{23}) - \lambda_3(1 - P_{32} - \\ &- P_{34} - P_{35} - P_{36}) = \lambda_2 P_{21}. \end{aligned}$$

Следовательно, ранг системы уравнений равен пяти, и первое уравнение можно отбросить. Вводим в рассмотрение величину  $\beta_i = \lambda_i/\lambda_1$  — "приведенную" частоту попадания запросов в СМО $_i$  относительно СМО1:

$$\begin{cases} \beta_2 = 1 + P_{32}\beta_3, \\ \beta_3 = P_{23}\beta_2 + \beta_4 + \beta_5 + \beta_6, \\ \beta_4 = P_{34}\beta_3, \\ \beta_5 = P_{35}\beta_3, \\ \beta_6 = P_{36}\beta_3. \end{cases}$$

Решаем полученную систему уравнений:

$$\begin{cases} \beta_3 = P_{23}(1 + P_{32}\beta_3) + P_{34}\beta_3 + P_{35}\beta_3 + P_{36}\beta_3; \\ \beta_3 - P_{23}P_{32}\beta_3 - P_{34}\beta_3 - P_{35}\beta_3 - P_{36}\beta_3 = P_{23}; \\ \beta_3 = \frac{P_{23}}{1 - P_{23}P_{32} - P_{34} - P_{35} - P_{36}} = \\ = \frac{P_{23}}{P_{32} - P_{23}P_{32}} = \frac{P_{23}}{P_{32}(1 - P_{23})}; \\ \beta_2 = 1 + \frac{P_{23}}{1 - P_{23}}, \\ \beta_3 = \frac{P_{23}}{P_{32}(1 - P_{23})}, \\ \beta_4 = \frac{P_{34}P_{23}}{P_{32}(1 - P_{23})}, \end{cases}$$

$$\left\{ \begin{aligned} \beta_5 &= \frac{P_{35}P_{23}}{P_{32}(1-P_{23})}, \\ \beta_6 &= \frac{P_{36}P_{23}}{P_{32}(1-P_{23})}. \end{aligned} \right.$$

Вычислим вероятности нахождения в состоянии  $\mathbf{l} = (l_1, l_2, \dots, l_6)$  для всех СМО  $i$  сети. Здесь  $\mathbf{l}$  — вектор, показывающий, что на обработке в  $i$ -й СМО находится ровно  $l_i$  запросов. Вероятность нахождения в состоянии  $\mathbf{l}_i$  для СМО1 без очереди [5]:

$$p(l_i) = \frac{1}{l_i!} \left( \frac{\lambda_i}{\mu_i} \right)^{l_i} e^{-\left( \frac{\lambda_i}{\mu_i} \right)}, \quad i = 1.$$

Вероятность нахождения в состоянии  $\mathbf{l}$  для одноканальных СМО2—СМО6:

$$p(l_i) = \left( 1 - \frac{\lambda_i}{\mu_i} \right) \left( \frac{\lambda_i}{\mu_i} \right)^{l_i}, \quad i = 2, 3, 4, 5, 6.$$

Подставляем вероятности состояний каждой из СМО в формулу Гордона—Ньювелла [7]:

$$\begin{aligned} p_{\mathbf{l}} &= \frac{\prod_{i=1}^6 p(l_i)}{\sum_{\mathbf{l} \in S(6, N)} \prod_{i=1}^6 p(l_i)} = \\ &= \frac{\frac{1}{l_1!} \left( \frac{\lambda_1}{\mu_1} \right)^{l_1} e^{-\left( \frac{\lambda_1}{\mu_1} \right)} \left( 1 - \frac{\lambda_2}{\mu_2} \right) \times}{\sum_{\mathbf{l} \in S(6, N)} \frac{1}{l_1!} \left( \frac{\lambda_1}{\mu_1} \right)^{l_1} e^{-\left( \frac{\lambda_1}{\mu_1} \right)} \left( 1 - \frac{\lambda_2}{\mu_2} \right) \times} \\ &\quad \times \left( \frac{\lambda_2}{\mu_2} \right)^{l_2} \dots \left( 1 - \frac{\lambda_6}{\mu_6} \right) \left( \frac{\lambda_6}{\mu_6} \right)^{l_6} \\ &\quad \rightarrow \frac{\left( \frac{\lambda_1}{\mu_1} \right)^{l_1} e^{-\left( \frac{\lambda_1}{\mu_1} \right)} \left( 1 - \frac{\lambda_2}{\mu_2} \right) \times}{\left( \frac{\lambda_2}{\mu_2} \right)^{l_2} \dots \left( 1 - \frac{\lambda_6}{\mu_6} \right) \left( \frac{\lambda_6}{\mu_6} \right)^{l_6}}, \end{aligned}$$

где  $S(6, N)$  — множество всех допустимых состояний сети, таких что

$$\sum_{i=1}^6 l_i = N.$$

Выносим за знак суммы в знаменателе множителя, не зависящие от числа запросов в СМО, и проводим сокращение:

$$\begin{aligned} p_{\mathbf{l}} &= \\ &= \frac{\frac{1}{l_1!} \left( \frac{\lambda_1}{\mu_1} \right)^{l_1} \left( \frac{\lambda_2}{\mu_2} \right)^{l_2} \left( \frac{\lambda_3}{\mu_3} \right)^{l_3} \left( \frac{\lambda_4}{\mu_4} \right)^{l_4} \left( \frac{\lambda_5}{\mu_5} \right)^{l_5} \left( \frac{\lambda_6}{\mu_6} \right)^{l_6}}{\sum_{\mathbf{l} \in S(6, N)} \frac{1}{l_1!} \left( \frac{\lambda_1}{\mu_1} \right)^{l_1} \left( \frac{\lambda_2}{\mu_2} \right)^{l_2} \left( \frac{\lambda_3}{\mu_3} \right)^{l_3} \left( \frac{\lambda_4}{\mu_4} \right)^{l_4} \left( \frac{\lambda_5}{\mu_5} \right)^{l_5} \left( \frac{\lambda_6}{\mu_6} \right)^{l_6}}. \end{aligned}$$

Для любого состояния сети обслуживания сумма запросов в каждой из СМО равна общему числу циркулирующих в сети запросов, т. е. общая ин-

тенсивность потока запросов складывается из интенсивностей потока запросов на каждую из СМО:

$$\lambda_1^{-N} = \lambda_1^{-l_1} \lambda_1^{-l_2} \lambda_1^{-l_3} \lambda_1^{-l_4} \lambda_1^{-l_5} \lambda_1^{-l_6}.$$

Умножаем числитель и знаменатель полученного для вероятностей выражения на  $\lambda_1^{-N}$  и заме-

няем величины  $\frac{\lambda_i}{\lambda_1}$  на  $\beta_i$ :

$$\begin{aligned} p_{\mathbf{l}} &= \\ &= \frac{\frac{1}{l_1!} \left( \frac{\beta_1}{\mu_1} \right)^{l_1} \left( \frac{\beta_2}{\mu_2} \right)^{l_2} \left( \frac{\beta_3}{\mu_3} \right)^{l_3} \left( \frac{\beta_4}{\mu_4} \right)^{l_4} \left( \frac{\beta_5}{\mu_5} \right)^{l_5} \left( \frac{\beta_6}{\mu_6} \right)^{l_6}}{\sum_{\mathbf{l} \in S(6, N)} \frac{1}{l_1!} \left( \frac{\beta_1}{\mu_1} \right)^{l_1} \left( \frac{\beta_2}{\mu_2} \right)^{l_2} \left( \frac{\beta_3}{\mu_3} \right)^{l_3} \left( \frac{\beta_4}{\mu_4} \right)^{l_4} \left( \frac{\beta_5}{\mu_5} \right)^{l_5} \left( \frac{\beta_6}{\mu_6} \right)^{l_6}}. \end{aligned}$$

Получив вероятности всех состояний сети, можно вычислить основные характеристики очередей. Среднее число запросов в каждой из СМО для установившегося режима работы сети [6]

$$\bar{n}_i = \sum_{k=1}^N k p(l_i = k), \quad i = 1 \dots 6.$$

Среднее число запросов в очереди на обслуживание к каждой из СМО

$$\bar{w}_i = \sum_{k=m_i+1}^N (k - m_i) p_i(k), \quad i = 1 \dots 6.$$

Среднее время ожидания в очереди к СМО

$$\bar{T}_{wi} = \frac{\bar{w}_i}{(\bar{w}_i - \bar{n}_i) \mu_i}, \quad i = 1 \dots 6.$$

Среднее время отклика системы (время пребывания запроса вне первой СМО)

$$t_{\text{откл}} = \sum_{j=2}^6 \left( \bar{w}_j + \frac{1}{\mu_j} \right) \beta_j.$$

Как видно, для расчетов по аналитической модели необходимо знать вероятности  $P_{ij}$  в матрице передач. Эти вероятности определяются по собранной статистике числа обращений к устройствам  $N_i$  исходя из того, что суммы вероятностей передач по строкам матрицы равны 1:

$$\sum_{j=1}^6 P_{ij} = 1, \quad i = 1, 2, \dots, 6.$$

Для второй строки матрицы передач каждый переданный на обработку запрос возвращается в СМО1:

$$N_2 = 2.$$

Для третьей строки матрицы передач

$$N_3 = n_{\text{cpu}} + n_{\text{hdd}} + n_{\text{cache}} + n_{\text{net}},$$

где  $n_{cpu}$  — число обращений к процессору;  $n_{hdd}$  — число обращений к жесткому диску;  $n_{cache}$  — число обращений в кэш;  $n_{net}$  — число обращений к сетевой карте;

$$P_{21} = P_{23} = \frac{1}{N_2} = 0,5,$$

$$P_{32} = \frac{1}{N_3},$$

$$P_{34} = \frac{n_{cpu}}{N_3},$$

$$P_{35} = \frac{n_{hdd} P_{cache}}{N_3},$$

$$P_{36} = \frac{n_{hdd} - P_{cache} n_{hdd}}{N_3}.$$

Интенсивности обслуживания запросов в СМО определяются на основе полученной статистики времен обслуживания:

$$\mu_i = 1/T_i$$

где  $T_i$  — время обслуживания в  $i$ -й СМО.

### Числовые результаты

Точность модели была проверена на нескольких сетевых приложениях. Далее представлены результаты, полученные для одного из них. Приложение представительное. Для него характерны большое число одновременно работающих пользователей, низкая интенсивность потока пользовательских запросов и быстрое по времени обслуживание на сервере этих запросов.

Графики на рис. 2 показывают хорошую согласованность времени отклика, предсказанного моделью, и времени отклика, измеренного на приложении, при эмуляции до 600 одновременно работающих пользователей.

Графики на рис. 3 отражают зависимость времени ожидания на обслуживание запроса в очереди к устройствам от числа работающих с приложением пользователей. Видно, что основные очереди могут образовываться у жесткого диска и процессора, что говорит о недостаточной производительности этих устройств.

### Заключение

Предложена математическая модель клиент-серверного приложения в виде сети СМО. Получены аналитические соотношения, которые позволяют рассчитать вероятности всех состояний

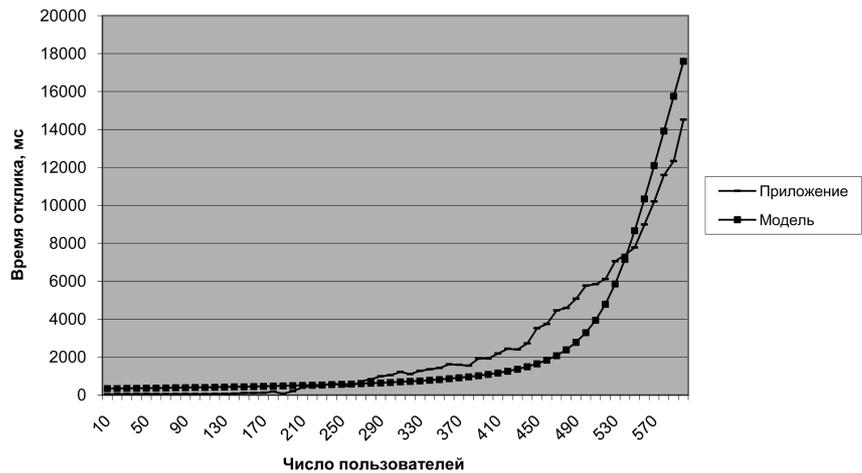


Рис. 2. Среднее время отклика системы

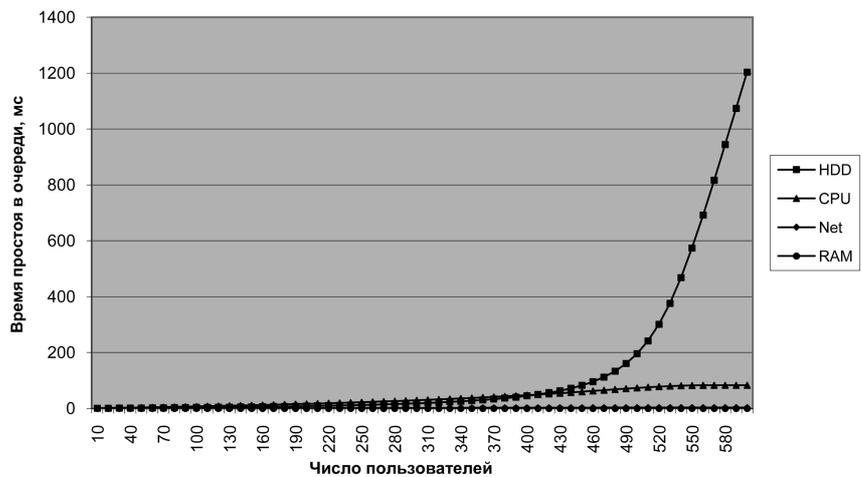


Рис. 3. Среднее время ожидания в очереди к устройствам

сети СМО и ожидаемые характеристики временных задержек при обработке запросов пользователей в основных устройствах сервера. Приведенные на рис. 2 графики позволяют судить о хорошей согласованности времени отклика, предсказанного моделью, с реальным приложением.

### Список литературы

1. Глинских А. Мировой рынок систем электронного документооборота [Электронный ресурс]. — Электрон, текстовые дан. и граф. дан. 2002. Режим доступа: [www.citforum.ru](http://www.citforum.ru). — Загл. с экрана.
2. Komaralingam A., Elleithy K. M. Using a Queuing Model to Analyze the Performance of Web Servers. 2002.
3. Field A. J., Harrison P. G., Parry J. Response times in Client-Server Systems // Lecture Notes in Computer Science. 1998. Vol. 1469. P. 75—86.
4. Клейнрок Л. Теория массового обслуживания: Пер. с англ. / Пер. И. И. Грушко; ред. В. И. Нейман. М.: Машиностроение, 1979. 432 с.
5. Кофман Ф., Крюон Р. Теория массового обслуживания. Теория и приложения. М.: Мир, 1965.
6. Жожикашвили В. А., Вишнеvский В. М. Сети массового обслуживания. Теория и применение к сетям ЭВМ. М.: Радио и связь, 1988. 192 с.
7. Gordon W. J., Newell G. F. Closed queuing systems with exponential servers // Oper. Res. 1967. Vol. 15. Issue 2. P. 254—265.

УДК 004.023

Э. А. Мухачева, д-р техн. наук, проф.,  
Л. Я. Бухарбаева, д-р техн. наук, проф.,  
Уфимский государственный  
авиационный технический университет  
Д. В. Филиппов, директор  
ООО "Матрица Трейд",  
У. А. Карипов, студент УГАТУ  
e-mail: elitamuh@vmk.ugatu.ac.ru

## Оптимизационные проблемы транспортной логистики: оперативное размещение контейнеров при транспортировке грузов

*Приведены логистические схемы, одним из модулей которых является плотное размещение грузов в транспортные средства, обеспечивающее комфортность загрузки складским погрузчиком и разгрузки заказчиком. С этой целью решается оптимизационная задача упаковки с дополнительными ограничениями.*

**Ключевые слова:** транспортно-упаковочный комплекс, маршрутизация, ортогональная упаковка, мультиметодная технология, метаэвристики

### Введение

Процесс глобализации в мировой экономике и усиление экономических связей внутри России и с другими странами приводят к возникновению все более сложных задач в сфере торговли. Увеличение объема грузоперевозок, рост числа экономических агентов, усиление конкуренции между транспортными компаниями обнаруживают необходимость построения новых эффективных логистических систем, отвечающих современным требованиям. Появилась масса литературы в указанной области и, что самое главное, реальные прецеденты использования логистических схем [1]. Соответствующая информационная система должна аккумулировать всю необходимую экономическую информацию — заявки потребителей, наличие товаров на складах, имеющиеся в распоряжении транспортные средства и т. д. — в единый центр принятия решений. Далее система должна учитывать объективные факторы и ограничения. Наконец, система должна быть доступной для пользователей, находящихся на значительном расстоянии друг от друга, обеспечивая

при этом возможность мониторинга протекающих процессов. Это, безусловно, повышает требования к качеству разрабатываемых схем и принимаемых с их помощью решений. Все еще недостаточно внимания уделяется вопросам оптимизации, которые могут возникать в сфере транспортной логистики.

В данной статье представлен проект оптимизационной поддержки логистической системы на условном предприятии, занимающемся международными грузоперевозками автомобильным транспортом. А именно, рассматривается задача рациональной загрузки транспортных средств. Остановимся подробнее на оптимизационных модулях в процессе их взаимодействия в логистической среде.

**Модуль формирования обобщенного заказа.** В данном модуле аккумулируется информация о заявках заказчиков. Под *заявкой* будем понимать множество грузов для доставки одному заказчику одним рейсом к определенному сроку. Заявки поступают в режиме реального времени и могут характеризоваться, например, срочностью, ценностью, важностью и т. д. Под *обобщенным заказом* (о. заказ) будем понимать набор заявок, подлежащих размещению внутри одного транспортного средства (ТС) с последующей транспортировкой одним рейсом. В результате работы модуля должно быть принято решение о включении заявок в о. заказ на транспортировку *очередным рейсом*. О. заказ может быть получен как решение задачи загрузки рюкзака [2], в которой ограничением может являться допустимая грузоподъемность ТС или объем грузового отсека.

**Модуль расчета маршрутов.** Под *маршрутом* транспортировки (МТ) будем понимать последовательность прохождения одним ТС пунктов загрузки/разгрузки. Результат работы модуля зависит от вида поступающей информации. Если на вход поступает информация об *одном* о. заказе, то на выходе следует ожидать *один* маршрут, который может быть получен как решение *задачи коммивояжера* для соответствующих заказу пунктов загрузки/разгрузки [3]. Если на вход поступает информация обо *всех* заявках потребителей, то на выходе следует ожидать *набор* маршрутов, полученных как решение модифицированной *транспортной задачи*, подробно рассмотренной в [4]. Модуль расчета маршрутов обеспечивает минимум затрат, пропорциональных общему пробегу по транспортной сети одного или нескольких выбранных ТС.

**Модуль выбора ТС.** Входные данные определяются набором имеющихся в данный момент ТС,

их геометрическими и техническими параметрами. Работа модуля осложняется в случае переменного парка неоднотипных ТС. В результате работы модуля должно быть принято решение о выборе ТС для выполнения очередного рейса.

**Модуль размещения.** Учитывая выбранное ТС и параметры грузов о. заказа (хрупкость, возможные варианты ориентации в пространстве и др.), необходимо получить карты их размещения. Под *картой размещения* будем понимать графическое изображение рационального способа размещения грузов, а также координаты их установки внутри ТС. Поскольку в одном ТС могут находиться грузы для доставки в различные пункты, следует выполнить *требование комфортности размещения*. Согласно требованию грузы, предназначенные для разгрузки в одном очередном пункте назначения, т. е. составляющие одну заявку, должны быть легко доступны с торцевой стороны ТС (рис. 1).

Требование комфортности размещения может быть выполнено только при заранее известном маршруте движения ТС. Однако обозначенные выше четыре задачи не всегда решаются последовательно в указанном порядке. В следующем параграфе в качестве иллюстрации вышесказанного рассмотрим два примера логистических схем.

## 1. Примеры различных логистических схем

**1.1. Ведущая роль ТС.** Ограничимся задачей поиска характеристик одного очередного рейса с максимизацией прибыли предприятия от его реализации. В результате одного цикла работы схемы на выходе должны быть получены четыре характеристики рейса: о. заказа; маршрут движения ТС; используемое ТС; карта размещения грузов внутри ТС.

Выбор свободного ТС для выполнения рейса в условиях оперативного планирования перевозок часто отягощен множеством субъективных факторов, поэтому предлагается вынести соответствующий модуль на первый этап. На втором этапе в модуле формирования о. заказа решается задача о загрузке рюкзака: проводится выбор заявок, имеющих наибольшую суммарную ценность и не превосходящих по суммарному весу (объему) грузоподъемность (объем грузового отсека) выбранного ранее ТС. Далее информация об о. заказе поступает на модуль расчета маршрутов, где решается *задача коммивояжера* для соответствующих заявкам пунктов загрузки/разгрузки. Таким образом, становится известен маршрут движения ТС, позволяющий выполнить на следующем этапе тре-

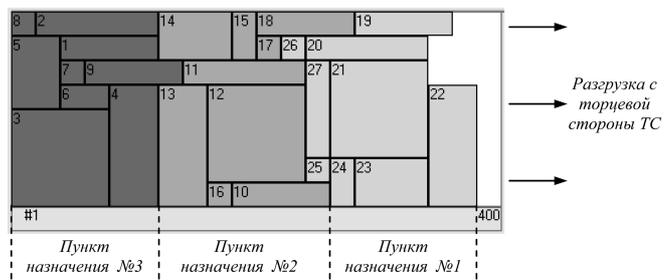


Рис. 1. Пример карты размещения, выполняющей требование комфортности размещения

бование комфортности размещения. Наконец, информация о ТС, о заказе и маршруте поступает в модуль размещения для получения карты. В результате, однако, может оказаться, что данный набор грузов не может быть размещен *целиком* внутри данного ТС. Для разрешения ситуации можно либо подобрать иное ТС, которое обеспечило бы возможность размещения всех грузов, либо скорректировать о. заказ. Схема логических связей между модулями изображена на рис. 2.

При построении схемы стала очевидной возможность получения различных решений вследствие вариативности процесса обмена информацией между различными модулями. Поэтому алгоритм может выполняться по эволюционной схеме, в частности — одноточечного алгоритма локального поиска (1 + 1)-EA [5].

В приведенном примере полученный о. заказ принимается в качестве *начального решения*, которое в модуле размещения проверяется на предмет возможности размещения в одном ТС, т. е. на *допустимость*. В случае недопустимости делается попытка скорректировать о. заказ (возврат на модуль формирования заказа) или подобрать иное ТС (возврат на модуль выбора ТС). Если решение оказывается допустимым, то назначается рейс по соответствующему маршруту, заявки текущего о. заказа считаются исполненными.

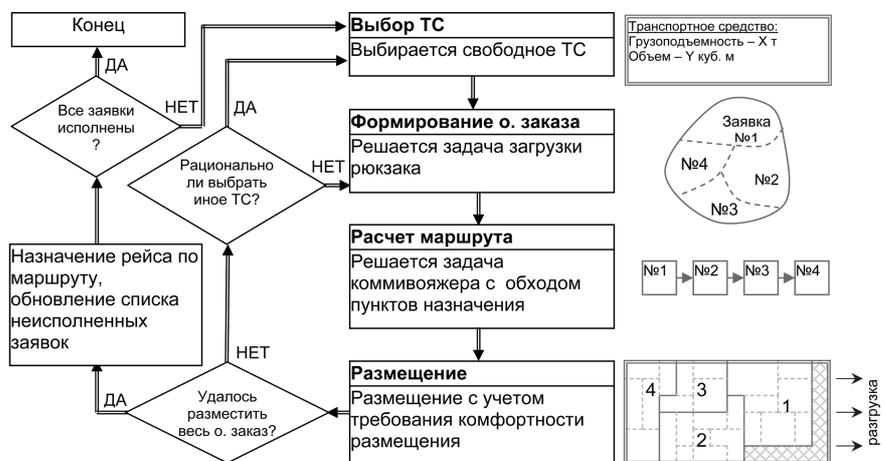


Рис. 2. Схема связей модулей логистической схемы. Ведущая роль ТС

В следующем примере на предварительном этапе принимаются во внимание сразу все заявки потребителей, а результатом работы являются характеристики всех рейсов, необходимых для удовлетворения заявок.

### 1.2. Ведущая роль маршрутов.

Информация обо всех заявках потребителей поступает на модуль расчета маршрутов, где решается обобщенная транспортная задача с учетом многономенклатурности товаров. Получаемый набор маршрутов, во-первых, удовлетворяет все заявки потребителей, а во-вторых, минимизирует затраты по транспортировке. В [4] показано,

что может быть сгенерировано множество таких наборов маршрутов. Далее каждый маршрут из полученного набора рассматривается *отдельно*, и соответствующий ему о. заказ проверяется на предмет возможности размещения в одно выбранное ТС. Допустим, один такой маршрут предписывает прохождение одним ТС нескольких пунктов потребления. Однако после попытки размещения соответствующего о. заказа выясняется, что одного ТС недостаточно. В данной ситуации можно попытаться найти подходящее ТС или, если оно не найдено, отложить заявки данного "неудачного" маршрута до следующего цикла работы схемы. После того, как все маршруты будут обеспечены картами размещения, все заявки потребителей будут удовлетворены.

На рис. 3 приведен пример пяти заявок, для которых получены два маршрута и два соответствующих о. заказа. При этом о. заказ 2 не удалось разместить в одно ТС, и составляющие его заявки вновь пополнили список неисполненных.

В приведенных примерах ведущая роль отдана модулям *выбора ТС* и *МТ*. В свою очередь, они провоцируют оптимизационные задачи упаковки и маршрутизации. Рассмотрим задачу размещения грузов, ориентированную на фиксированный маршрут перевозки.

## 2. Задача размещения контейнеров

**2.1. Описание проблемы.** Предположим, что условное предприятие путем организации регулярных рейсов осуществляет поставки в Россию крупногабаритных грузов промышленного назначения (станки, оборудование) с нескольких складских центров, расположенных в Европе. Очередной рейс заключается в последовательном прохождении ТС задействованных складов для *загрузки* товаров, последним пунктом является

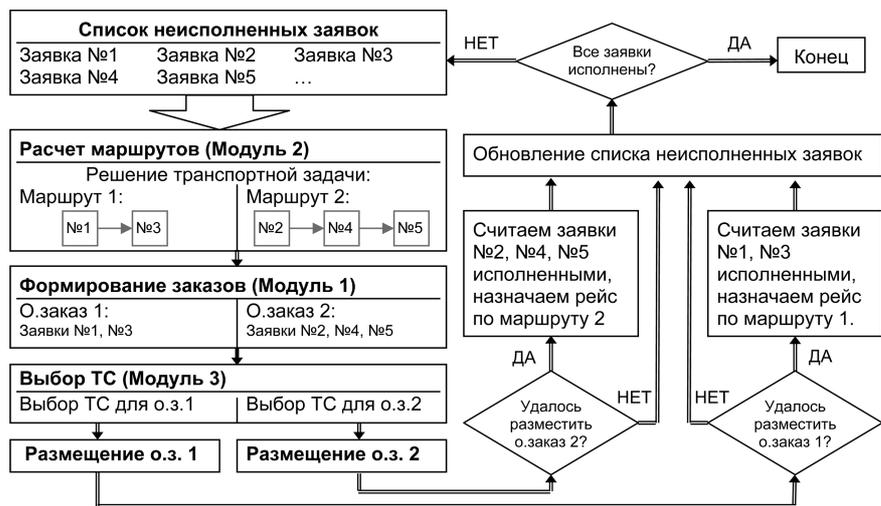


Рис. 3. Схема связей модулей логистической схемы. Ведущая роль маршрутов

склад *разгрузки* в г. Москве. Необходимо составить о. заказ на доставку рейсом, максимизирующим прибыль, которую получит предприятие от его реализации. Данное описание проблемы наиболее близко примеру логистической схемы, описанной п. 1.1. Адаптируя данную схему к конкретным условиям, следует отметить, что компания использует однотипные ТС с фиксированными параметрами грузового отсека. Модуль выбора ТС в данном случае является не значимым, его можно исключить из схемы. Работа модуля расчета маршрута будет заключаться только в решении задачи коммивояжера, т. е. определении порядка прохождения складов, необходимого для соблюдения требования комфортности размещения, и в подсчете затрат, пропорциональных пробегу.

*Основным звеном* схемы становится *модуль размещения*. Будем считать, что каждому грузу соответствует прямоугольный контейнер, имеющий длину, ширину и высоту. В силу специфики поставляемых товаров *кантовка контейнеров запрещена*, т. е. их не разрешается переворачивать и ставить друг на друга. При этом повороты контейнеров в плоскости основания возможны. Высота каждого контейнера не превосходит высоту  $H$  грузового отсека. Данные замечания позволяют представлять контейнеры в виде их двумерных проекций на плоскость оснований, грузовой отсек ТС — в виде прямоугольной области, а всю задачу можно свести к двумерной задаче упаковки контейнеров 2DBP [5]. Важной здесь является интерпретация требования комфортности размещения, проиллюстрированного на рис. 1, так как мы имеем дело не с последовательной разгрузкой ТС, а с его последовательной дозагрузкой, порядок которой определяется порядком прохождения ТС задействованных складских пунктов.

Выбор множества контейнеров для загрузки определяется в модуле формирования заказа как

решение задачи загрузки рюкзака. Исходная информация задается в виде  $\langle S; s; c \rangle$ , где  $S$  — площадь основания грузового отсека ТС (емкость рюкзака),  $s$  — вектор площадей оснований контейнеров (веса предметов),  $c$  — вектор стоимостей. Предлагается использование жадного алгоритма: список предметов упорядочивается по убыванию величины  $c/s$ , после чего выбор предметов выполняется до тех пор, пока  $\sum_i s_i \leq S$ .

**2.2. Постановка задачи.** Имеется единственное ТС, грузовой отсек которого описывается прямоугольной областью ширины  $W$  и длины  $L$ . Имеется информация об о. заказе и маршруте движения ТС, проходящего через  $k$  складов, для каждого из которых известен набор контейнеров, подлежащих загрузке. Контейнеры, подлежащие загрузке на  $j$ -м складе, описываются вектором  $\pi_j = (w_i^{(j)}; l_i^{(j)})$ , где  $w_i^{(j)}$  — значения ширин,  $l_i^{(j)}$  — значения длин,  $i \in \{1, \dots, m_j\}, j \in \{1, \dots, k\}$ , где  $m_j$  — число контейнеров, подлежащих загрузке на  $j$ -м складе. Размеры контейнеров являются положительными целыми числами и удовлетворяют условию

$$(w_i^{(j)} \leq W \vee w_i^{(j)} \leq L) \wedge (l_i^{(j)} \leq W \vee l_i^{(j)} \leq L), \forall i, j.$$

Необходимо найти карту размещения, обеспечивающую размещение контейнеров всех заявок о. заказа внутри имеющегося ТС при выполнении следующих условий:

- ребра контейнеров параллельны ребрам грузового отсека и не пересекаются с ними;
- контейнеры не пересекаются друг с другом;
- все контейнеры о. заказа размещены в одном ТС;
- расположение контейнеров внутри ТС удовлетворяет требованию комфортности размещения;
- допускаются повороты контейнеров на  $90^\circ$ .

Рассматриваемая задача 2DBP является NP-трудной задачей комбинаторной оптимизации. Рациональным представляется привлечение известных метаэвристик. Большую популярность получили эволюционные методы, которые различаются главным образом применяемыми эволюционными стратегиями и алгоритмами конструирования упаковок, называемыми декодерами. Выбор метода должен преследовать как минимум две цели: получение экономного решения; обеспечение требования комфортности размещения. В целях экономии вычислительного времени и рационального размещения грузов в качестве алгоритма локального поиска будем использовать эволюционный (1 + 1)-EA алгоритм. Для достижения второй цели воспользуемся декодером, предложенным И. П. Норенковым [6].

**2.3. Мультиметодная технология И. П. Норенкова.** Основная идея предложенного подхода за-

ключается в том, что на каждом шаге декодер осуществляет случайный выбор простой эвристики из заданного набора алгоритмов. Действие каждой эвристики заключается в детерминированном выборе и установке в свободную область предмета, лучшего в смысле одной из применяемых эвристик. Предмет для установки выбирается из списка  $\pi$ . Предлагается последовательно ограничивать список  $\pi$  предметами, подлежащими загрузке на каком-либо одном складе. Пока не размещены все предметы  $j$ -го склада,  $\pi = \pi_j$ . Если  $\pi = \emptyset$ , то  $\pi = \pi_{j+1}, j \in \{1, \dots, k\}$ . Таким образом, начиная от дальней стенки грузового отсека и далее к торцу, будут загружены предметы 1-го склада, затем 2-го и т. д. Преимущество предложенной технологии в том, что данный принцип попеременного ограничения списка  $\pi$  позволяет существенно снизить размерность для длинных маршрутов, поскольку перебор грузов реализуется для каждого склада в отдельности. При этом обеспечивается выполнение требования комфортности размещения.

Введем обозначения (рис. 4) и опишем список эвристик, уточнив некоторые правила расположения выбранного контейнера.

S1. Объект с параметрами  $(w_i, l_i)$  должен обеспечивать минимальный вертикальный зазор  $gap \geq 0$ . Если деталей с  $gap \geq 0$  нет, то поиск повторяется среди повернутых объектов. Если и в этом случае  $gap < 0$ , то участок шириной  $s$  и длиной  $h$  заполняется фиктивной деталью.

S2. Объект с параметрами  $(w_i, l_i)$  должен обеспечивать минимальный по абсолютной величине горизонтальный зазор  $z = \min\{|z_1|, |z_2|\}$  при условии  $gap \geq 0$ . Если деталей с  $gap > 0$  нет, то поиск повторяется среди повернутых объектов. Если и в этом случае  $gap < 0$ , то участок шириной  $s$  и длиной  $h$  заполняется фиктивной деталью.

S3. Объект с параметрами  $(w_i, l_i)$  должен обеспечивать максимальную площадь  $area = w_i \times l_i$

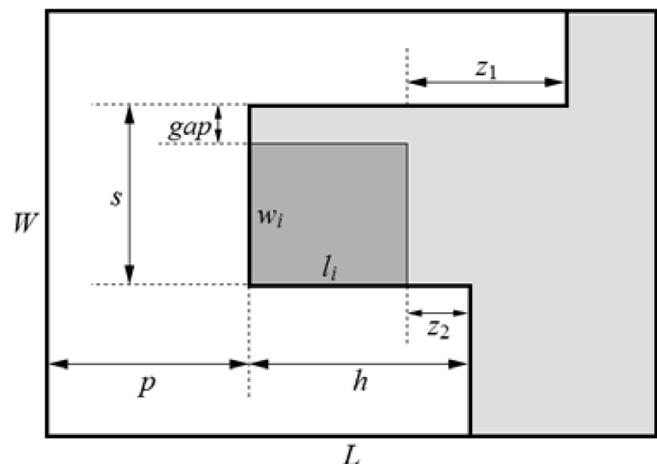


Рис. 4. Иллюстрация мультиметодной технологии

при условии  $gap \geq 0$ . Если деталей с  $gap \geq 0$  нет, то поиск повторяется среди повернутых объектов. Если и в этом случае  $gap < 0$ , то участок шириной  $s$  и длиной  $h$  заполняется фиктивной деталью.

$S4, S5$  и  $S6$  аналогичны соответственно  $S1, S2$  и  $S3$ , но сначала рассматриваются повернутые объекты.

Независимо от выбранной эвристики для очередного размещаемого предмета должны выполняться следующие условия:

$$gap \geq 0; p + l_i \leq L.$$

Если  $gap \geq 0$ , то объект размещается вплотную к тому краю свободной области  $s$ , на котором меньше по абсолютной величине горизонтальный зазор  $z_1$  или  $z_2$ . При  $|z_1| = |z_2|$  выбирается нижний край. Если условие  $p + l_i \leq L$  не выполняется, то участок шириной  $s$  и длиной  $h$  заполняется фиктивной деталью.

### 3. Пример расчета модулей формирования заказа и размещения

Имеется одно ТС, длина грузового отсека 400, ширина 160. Список имеющихся на складах грузов приведен в таблице 1, где  $w, l, c$  — векторы соответственно ширин, длин и стоимостей. Требуется сформировать о. заказ, рассчитать маршрут и получить карту размещения.

Алгоритм поиска решения построим по схеме рис. 1 в соответствии с описанием проблемы п. 2.1.

1. Выбор ТС.  $W = 160, L = 400$ .

Список имеющихся грузов

$w$	$l$	$c$	$s$	$c/s$	
Склад 1					
60	100	5000	6000	0,83	✓
50	50	8000	2500	3,20	✓
40	100	5000	4000	1,25	✓
50	30	6000	1500	4,00	✓
50	90	2000	4500	0,44	
10	40	3000	400	7,50	✓
Склад 2					
100	50	3000	5000	0,60	
60	100	5000	6000	0,83	✓
50	50	8000	2500	3,20	✓
40	100	5000	4000	1,25	✓
Склад 3					
100	50	3000	5000	0,60	✓
60	100	5000	6000	0,83	✓
50	50	8000	2500	3,20	✓
40	100	5000	4000	1,25	✓
50	30	6000	1500	4,00	✓
100	80	6000	8000	0,75	✓
60	100	5000	6000	0,83	✓
50	50	8000	2500	3,20	✓
40	100	5000	4000	1,25	✓
50	30	6000	1500	4,00	✓

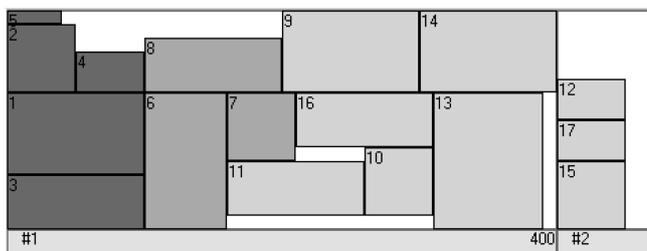


Рис. 5. Начальная карта размещения. Число использованных ТС — 2

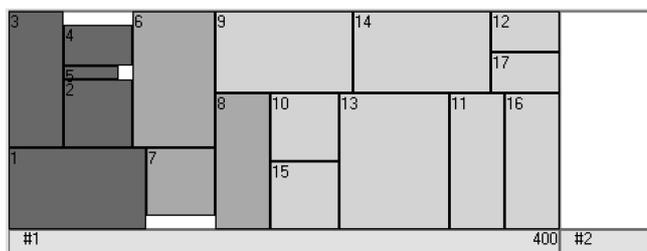


Рис. 6. Конечная карта размещения. Число использованных ТС — 1

2. Формирование о. заказа. Решается задача загрузки рюкзака, входная информация задается в виде  $\langle S; s; c \rangle$ , где  $S = W \times L, s_i = w_i \times l_i, \forall i$ . Каждый груз характеризуется "весом" — стоимостью на единицу площади  $c_i/s_i$ . Отобранные в результате выполнения жадного алгоритма грузы (отмечены в таблице символом ✓) составляют о. заказ,  $k = 3$ . Число контейнеров в о. заказе — 17, общая стоимость — 99000. Заполненность ТС по площади  $\sum_{i=1}^{17} s_i/S = 98,28 \%$ .

3. Расчет маршрута. Допустим, кратчайший маршрут для  $k = 3$  известен: Склад 1 → Склад 2 → Склад 3 → Склад разгрузки.

4. Размещение. Начальная карта размещения приведена на рис. 5. Выбор простых эвристик равновероятный, повороты контейнеров на 90° разрешены. После выполнения серии итераций одноточечного эволюционного алгоритма (1 + 1)-EA все контейнеры о. заказа были размещены в одном ТС (рис. 6).

### Заключение

В статье рассмотрены некоторые подходы к разработке логистических схем. Приведены два примера схем различного типа. В первом характер системы определяет транспортное средство, во втором — маршрут. Этим далеко не исчерпывается разнообразие логистических схем. В рамках схем возникают оптимизационные модули, значение которых часто игнорируется. В данной статье выделены три таких модуля: формирования зака-

за, транспортный и размещения грузов. Специфика первой из приведенных логистических схем такова, что решение модуля формирования заказа сводится к задаче загрузки рюкзака, транспортно-го модуля — к задаче коммивояжера и основная роль отдана модулю размещения. Для решения этой задачи используются односточный эволюционный алгоритм с мультиметодным алгоритмом конструирования упаковки И. П. Норенкова. Показано, что он лучше приспособлен к выполнению условия комфортности загрузки/выгрузки перевозимых грузов. Логистическим схемам, в рамках которых важную роль выполняют транспортные модули, посвящены другие статьи.

#### Список литературы

1. Джонсон Д. С., Вуд Д. Ф. и др. Современная логистика // Издат. дом "Вильямс", 2005. 624 с.

2. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир. 1985. 512 с.

3. Сигал И. Х., Иванова А. П. Введение в прикладное дискретное программирование. М.: Физматлит. 2002. 237 с.

4. Bukharbaeva L. Y., Filippov D. V., Gilmanova N. A., Karipov U. A. Information-Based Approaches to Local Search for Rational Solutions of Transportation and Bin-packing Complex Problems // Proceedings of the Workshop on Computer Science and Information Technologies (CSIT'2007), Krasnounsolsk, Ufa, September 3—16, 2007. Ufa: State Aviation Technical University. 2007. Vol. 2. С. 122-126.

5. Филиппова А. С. Моделирование эволюционных алгоритмов решения задач прямоугольной упаковки на базе блочных структур // Информационные технологии. 2006. № 6. Приложение. 32 с.

6. Норенков И. П. Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. 1999. № 1. С. 2—7.

7. Филиппова А. С., Валиахметова Ю. И. Мультиметодный генетический алгоритм для решения задач ортогональной упаковки // Информационные технологии. 2007. № 12. С. 50—56.

УДК 519.626.2

**Н. Н. Галимьянова**, зам. нач. отдела,  
ОАО "Научно-исследовательский  
и проектно-конструкторский институт  
информатизации и связи  
на железнодорожном транспорте"

## Экспериментальное исследование параллельных реализаций комбинаторных алгоритмов решения задачи о ранце\*

*Рассматриваются параллельные алгоритмы для задачи о булевом ранце с одним ограничением. Описаны параллельные алгоритмы ветвей и границ и динамического программирования. Для динамического программирования построена упрощенная математическая модель процесса вычислений. Представлены результаты вычислительных экспериментов и проведено сравнение алгоритмов.*

**Ключевые слова:** задача о ранце, параллельный алгоритм, параллельный алгоритм ветвей и границ, параллельный алгоритм динамического программирования, комбинированный алгоритм, результаты эксперимента.

### Введение

Внедрение GRID-технологий [2, 4] требует разработки пакета прикладных программ. Средствами (центрами) GRID-технологий являются па-

раллельные и распределенные суперсистемы, погруженные в локальную вычислительную сеть, таким образом, задачи пакета прикладных программ должны быть реализованы на параллельных или распределенных (второе — более общее) системах. Это требует разработки и исследования параллельных универсальных технологий решения оптимизационных задач "большой размерности" [1]. Оптимизация решения таких задач может достигаться по ряду критериев:

- по максимальной полезной производительности выделяемых вычислительных средств;
- по минимуму времени решения;
- по максимальному среднему коэффициенту полезной загрузки оборудования;
- по отношению времени непосредственно решения к общему времени решения и т. д.

### Общие сведения об алгоритмах решения задачи

Данная статья является продолжением работ [6, 12, 14, 15]. По сравнению с ними вычислительный эксперимент проведен для нескольких новых типов задач и исследованы, наряду с рассмотренными ранее, новые показатели процесса решения: загруженность рабочего процессора (отношение времени непосредственно вычисления к общему времени решения); общее число ветвлений для алгоритма ветвей и границ. Предложен также комбинированный алгоритм, в котором на программном уровне сочетаются элементы алгоритмов ветвей и границ и динамического программирования.

\* Работа выполнена при финансовой поддержке РФФИ (код проекта 05-01-00495).

Задачи дискретной оптимизации являются математическими моделями многих проблем, встречающихся в приложениях. С ростом размерности задач при нахождении точного или приближенного (с оценкой точности) решения возникают вычислительные трудности принципиального характера. На последовательных вычислительных системах решение задач рассматриваемого типа, особенно большой размерности [1], не может быть найдено с применением ресурсов, которые могут быть выделены для их решения. Отсюда естественно возникает необходимость применения алгоритмов параллельных вычислений для решения задач дискретной оптимизации.

Для исследования и проведения вычислительного эксперимента была выбрана задача о ранце с одним ограничением. Эта задача является NP-полной и хорошо изучена. Несмотря на ее относительную простоту вычислительные трудности, возникающие при ее решении, являются характерными для задач дискретной оптимизации более общего типа.

Основные методы, применяемые при решении задач дискретной оптимизации, — это динамическое программирование и метод ветвей и границ. Целью работы является исследование параллельных реализаций этих методов для задачи о ранце.

Экспериментальное исследование проводилось на вычислительных комплексах MVS 15000 VM и Hewlett Packard SuperDome (64 CPU), установленных в Межведомственном суперкомпьютерном центре РАН (МСЦ РАН) [17]. В качестве средств параллельного программирования была использована библиотека MPI для языка C++.

**Постановка задачи.** Пусть имеется  $n$  предметов, каждый из которых имеет ценность  $c_j > 0$  и вес  $a_j > 0, j = 1, 2, \dots, n$ . Имеется ранец (рюкзак), грузоподъемность которого равна  $R$ . Необходимо положить в ранец набор предметов с максимальной суммарной ценностью. При этом предполагается, что

$\sum_{j=1}^n a_j > R, 0 < a_j \leq R$ , все значения  $a_j, c_j$  и  $R$  — целые,  $j = 1, 2, \dots, n$ . Для описания задачи на языке целочисленного линейного программирования введем булевы переменные  $x_j$  (0, если предмет с номером  $j$  не кладется в ранец, и 1, если кладется).

Получена следующая задача целочисленного (булевого) линейного программирования:

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \max; \quad (1)$$

$$\sum_{j=1}^n a_j x_j \leq R; \quad (2)$$

$$x_j \in \{0, 1\}, j = 1, 2, \dots, n. \quad (3)$$

Подробное описание задачи и алгоритмов ее решения содержится в [1].

**Метод динамического программирования.** Метод динамического программирования основан на утверждении, получившем название принципа оптимальности Беллмана. Смысл этого принципа состоит в том, что оптимальная стратегия при любом первоначальном состоянии и любом первоначальном решении предполагает, что последующие решения должны быть оптимальны относительно состояния, полученного в результате предыдущего решения (более подробное описание дано в [1]).

Для метода динамического программирования в [6] приведена модель процесса вычислений, из которой следует, что ускорение реализации алгоритма на  $n_p$  параллельных процессорах (далее по тексту просто ускорение) составляет:

$$\frac{T_1}{T_{n_p}} = \frac{n_p}{1 + \frac{\alpha}{2}}, \quad (4)$$

где  $T_1$  — общее время счета на одном процессоре;  $T_{n_p}$  — общее время счета на  $n_p$  процессорах;  $\alpha$  — отношение времени на пересылку одного элемента таблицы (см. [1]) к времени расчета этого элемента.

**Метод ветвей и границ.** Под методом ветвей и границ понимается метод, имеющий древовидную структуру поиска оптимального решения и использующий результаты решения промежуточных оценочных задач, ускоряющих поиск. Древовидная структура называется обычно деревом ветвления. Подробное описание метода ветвей и границ содержится в [1]. Для этого метода в [7] приведена верхняя оценка ускорения при параллельной реализации.

**Комбинированный метод.** Оба рассмотренные выше метода нахождения точного решения задачи (1)—(3) основываются на идеях последовательного анализа вариантов и используют различные способы отсева подмножеств множества допустимых решений, не содержащих оптимального решения. В методе ветвей и границ отсев проводится по верхней оценке для каждого подмножества, в методе динамического программирования — на основании принципа оптимальности, где отсеиваются все доминируемые множества. Учитывая достоинства и недостатки каждого метода, представляется целесообразным совместное применение обеих процедур (правил) отсева. Предлагается следующий способ совместного применения этих процедур.

За основу берется метод динамического программирования, однако отсев выполняется по обоим правилам, присущим двум рассматриваемым методам.

Обозначим

$$g_k(y) = \max \left\{ \sum_{j=1}^k c_j x_j \mid \sum_{j=1}^k a_j x_j = y \right\}.$$

Решение задачи (1)–(3) методом динамического программирования сводится к расчетам по рекуррентным соотношениям:

$$g_k(y) = \max_{x_k \in \{0, 1\}} \{g_{k-1}(y - a_k x_k) + c_k x_k\}, \quad (5)$$

$$0 \leq y = \sum_{j=1}^k a_j x_j \leq R, \quad k = 2, \dots, n.$$

Совместное применение правил отсева по методу динамического программирования и по методу ветвей и границ реализуется путем исключения из последовательности (5) тех членов  $g_k(y)$ , для которых выполняется условие

$$g_k(y) + [g_k^*(y)] \leq g^0, \quad (6)$$

где  $g^0$  — значение целевой функции для наилучшего из полученных целочисленных решений (рекорд);  $g_k^*(y)$  — верхняя оценка суммы

$$\sum_{j=k+1}^n c_j x_j \text{ при ограничениях}$$

$$\sum_{j=k+1}^n a_j x_j \leq R - y, \quad (7)$$

получаемая по правилу Данцига [1].

Величина  $g_k(y) + g_k^*(y)$  — верхняя оценка, соответствующая данному члену последовательности (5). Поэтому стандартная процедура динамического программирования может рассматриваться как своеобразное правило ветвления подмножеств решений, обеспечивающее исключение бесперспективных вариантов.

### Комбинированный алгоритм

1. Находим начальное приближение  $g^0$  (например, по двум правилам последовательного назначения единиц: по  $c_j$  и  $\lambda_j = c_j/a_j$  с применением локальной оптимизации).

2. Определяем верхнюю оценку  $g_0^*(0)$  ( $k = 0$ ).

3. Если  $[g_0^*(0)] \leq g^0$ , то задача решена, иначе переход к п. 4.

4. Увеличиваем значение  $k$  на единицу.

5. Исходя из значений  $y_{k-1}$  определяем два значения  $y_k = y_{k-1}$  и  $y_k = y_{k-1} + a_k$ , которые со-

ответствуют значениям  $x_k = 0$  и  $x_k = 1$ , далее исключаем повторяющиеся элементы.

6. Для всех  $y_k$  определяем значения  $g_k(y)$  по формуле (5),  $g_k^*(y)$  по формуле (7) и соответствующее значение  $g^0$ .

7. Для всех  $y_k$  проверяем условие (6). Если условие выполняется, то вершина отсеивается.

8. Если для всех  $y_k$  все вершины отсеяны, то задача решена и вектор, которому соответствует рекордное значение  $g^0$ , является решением; если  $k = n$ , то задача решена и решение соответствует максимальному значению  $g_n(y)$ ; если  $k < n$ , то переходим к п. 4.

### Вычислительный эксперимент

Эксперимент проводился на системах с двумя видами памяти (описание см. [18]): с распределенной и общей (см. [2, 4]).

Цель вычислительного эксперимента состоит в сравнении результатов решения задачи в зависимости от числа процессоров, участвующих в вычислениях, для систем с различными видами памяти по следующим показателям:

- ускорению времени решения задачи [4];
- отношению временных ресурсов, затраченных на непосредственные вычисления, к общему времени решения задачи;
- числу ветвлений для метода ветвей и границ.

**Формирование параметров задач для проведения вычислительного эксперимента.** Вычислительный эксперимент проводился для трех серий из  $N = 10$  задач размерности  $n = 1000$ . Для всех се-

рий  $R$  формировалось как  $\gamma \sum_{j=1}^n a_j$ , где  $\gamma = 0,1,$

$0,2, \dots, 0,9$  — параметр, характеризующий вместимость ранца. Изменение параметра  $\gamma$  позволяет исследовать зависимость результата решения от грузоподъемности ранца.

Способ формирования параметров и названия серий задач заимствованы из [12, 13].

*Серия 1.* Слабо связанные параметры:  $c_j$  формировались как целые случайные, равномерно распределенные на отрезке  $[1, 1000]$ ,  $a_j$  формировались как целые случайные, равномерно распределенные на отрезке  $[c_j, c_j + 100]$ .

*Серия 2.* Независимые параметры:  $a_j$  и  $c_j$  формировались как целые случайные, равномерно распределенные на отрезке  $[1, 1000]$ .

*Серия 3.* Сильно связанные параметры:  $c_j$  формировалось как целые случайные, равномерно распределенные на отрезке  $[1, 1000]$ ,  $a_j$  формировались как  $c_j + 100$ .

### Результаты вычислительного эксперимента.

Для всех таблиц  $n_p$  — это число процессоров, на которых проводился счет, столбцу с обозначени-

ем "о" соответствуют результаты для машины с общей памятью, а "р" — распределенной.

**Результаты вычислительного эксперимента для метода динамического программирования.** Время решения задачи методом динамического программирования на распределенных системах можно разбить на две основные составляющие: время на непосредственный счет, время на обмен.

Время на непосредственный счет включает временные затраты на реализацию следующих операций:

- решение уравнения Беллмана;
- создание структур данных;
- поиск данных;
- другие накладные расходы, обусловленные организацией процесса вычислений.

Время на обмен включает время на выполнение следующих операций:

- подготовку информации к отправке;
- передачу информации по сети;
- обработку принятой информации.

На машине с распределенной памятью были решены все задачи всех серий, для машины с общей памятью — для серий 1,3 лишь задачи с параметром  $\gamma \leq 0,2$ , серии 2 с лишь параметром  $\gamma \leq 0,3$ . Это обусловлено тем, что на данной машине не хватает объема памяти для хранения таблицы [1], необходимой для расчетов. В табл. 1 представлено отношение времени непосредственного счета ко всему времени решения задач для всех серий. С увеличением числа процессоров при  $\gamma \leq 0,2$  на машине с общей памятью наблюдается монотонно убывающая зависимость этого отношения. Для машины с распределенной памятью монотонность не наблюдается ни для одного принятого значения  $\gamma$ .

Как и следовало ожидать (см. табл. 1), для одинаковых классов задач доля "чистого" времени, затраченного на счет, по отношению к общему времени решения задач для машины с общей памятью больше, чем для машины с распределенной памятью, хотя для серии 1 наблюдается отклоне-

Таблица 1

**Отношение времени непосредственного счета ко всему времени решения методом динамического программирования для машины с общей памятью Hewlett Packard SuperDome (64CPU) (о) и распределенной памятью MVS 15000 VM (р)**

n <sub>p</sub>	$\gamma$																			
	0,1		0,2		0,3		0,4		0,5		0,6		0,7		0,8		0,9			
	о	р	о	р	о	р	о	р	о	р	о	р	о	р	о	р	о	р		
Серия 1																				
2	0,93	0,82	0,94	0,82	—	0,82	—	0,82	—	0,83	—	0,82	—	0,83	—	0,83	—	0,83	—	0,83
3	0,89	0,32	0,92	0,32	—	0,33	—	0,33	—	0,34	—	0,31	—	0,32	—	0,33	—	0,33	—	0,33
4	0,88	0,41	0,90	0,41	—	0,44	—	0,44	—	0,43	—	0,43	—	0,40	—	0,41	—	0,41	—	0,41
5	0,84	0,41	0,85	0,42	—	0,43	—	0,43	—	0,42	—	0,43	—	0,41	—	0,37	—	0,37	—	0,39
6	0,83	0,40	0,85	0,39	—	0,40	—	0,39	—	0,40	—	0,41	—	0,40	—	0,35	—	0,35	—	0,35
7	0,75	0,70	0,81	0,41	—	0,38	—	0,40	—	0,39	—	0,41	—	0,39	—	0,41	—	0,37	—	0,37
8	0,73	0,73	0,75	0,38	—	0,38	—	0,38	—	0,37	—	0,38	—	0,38	—	0,38	—	0,38	—	0,40
9	0,69	0,71	0,75	0,35	—	0,34	—	0,38	—	0,36	—	0,36	—	0,39	—	0,37	—	0,37	—	0,38
10	0,67	0,67	0,71	0,41	—	0,41	—	0,41	—	0,42	—	0,42	—	0,42	—	0,43	—	0,43	—	0,44
15	0,47	0,72	0,55	0,72	—	0,40	—	0,40	—	0,41	—	0,45	—	0,40	—	0,42	—	0,42	—	0,39
20	0,41	0,75	0,47	0,70	—	0,45	—	0,36	—	0,39	—	0,41	—	0,41	—	0,42	—	0,42	—	0,40
Серия 2																				
2	0,95	0,83	0,94	0,83	0,95	0,81	—	0,82	—	0,83	—	0,83	—	0,83	—	0,83	—	0,83	—	0,84
3	0,91	0,32	0,93	0,33	0,92	0,33	—	0,34	—	0,34	—	0,34	—	0,34	—	0,34	—	0,34	—	0,35
4	0,88	0,39	0,89	0,39	0,89	0,41	—	0,40	—	0,41	—	0,41	—	0,41	—	0,41	—	0,41	—	0,39
5	0,85	0,25	0,86	0,27	0,88	0,28	—	0,29	—	0,28	—	0,28	—	0,28	—	0,28	—	0,28	—	0,28
6	0,81	0,27	0,85	0,27	0,83	0,26	—	0,27	—	0,27	—	0,27	—	0,27	—	0,27	—	0,27	—	0,29
7	0,77	0,51	0,81	0,20	0,84	0,20	—	0,20	—	0,20	—	0,21	—	0,21	—	0,24	—	0,24	—	0,21
8	0,76	0,55	0,76	0,19	0,79	0,19	—	0,20	—	0,20	—	0,19	—	0,19	—	0,20	—	0,20	—	0,20
9	0,68	0,41	0,73	0,15	0,73	0,17	—	0,17	—	0,15	—	0,16	—	0,16	—	0,16	—	0,16	—	0,17
10	0,67	0,49	0,71	0,14	0,75	0,15	—	0,16	—	0,15	—	0,15	—	0,16	—	0,16	—	0,16	—	0,16
15	0,52	0,46	0,58	0,37	0,61	0,09	—	0,09	—	0,09	—	0,10	—	0,10	—	0,09	—	0,09	—	0,10
20	0,45	0,34	0,52	0,33	0,53	0,41	—	0,08	—	0,08	—	0,08	—	0,08	—	0,08	—	0,08	—	0,08
Серия 3																				
2	0,93	0,83	0,95	0,83	—	0,81	—	0,83	—	0,82	—	0,83	—	0,83	—	0,83	—	0,83	—	0,83
3	0,87	0,33	0,92	0,34	—	0,34	—	0,34	—	0,34	—	0,35	—	0,34	—	0,35	—	0,35	—	0,35
4	0,88	0,38	0,87	0,40	—	0,40	—	0,41	—	0,41	—	0,41	—	0,41	—	0,41	—	0,41	—	0,42
5	0,83	0,28	0,86	0,27	—	0,28	—	0,28	—	0,28	—	0,28	—	0,28	—	0,28	—	0,28	—	0,27
6	0,78	0,23	0,82	0,25	—	0,26	—	0,25	—	0,25	—	0,25	—	0,26	—	0,25	—	0,25	—	0,25
7	0,77	0,19	0,80	0,19	—	0,19	—	0,23	—	0,21	—	0,20	—	0,21	—	0,20	—	0,20	—	0,21
8	0,68	0,55	0,72	0,18	—	0,19	—	0,19	—	0,20	—	0,19	—	0,19	—	0,19	—	0,19	—	0,20
9	0,71	0,45	0,72	0,15	—	0,17	—	0,16	—	0,17	—	0,16	—	0,16	—	0,16	—	0,16	—	0,17
10	0,63	0,44	0,69	0,15	—	0,15	—	0,15	—	0,15	—	0,15	—	0,16	—	0,16	—	0,16	—	0,16
15	0,51	0,45	0,57	0,42	—	0,09	—	0,10	—	0,09	—	0,10	—	0,10	—	0,10	—	0,10	—	0,10
20	0,44	0,36	0,47	0,36	—	0,06	—	0,07	—	0,07	—	0,08	—	0,07	—	0,07	—	0,07	—	0,07

ние от этой тенденции для числа процессоров, большего девяти.

Для машины с распределенной памятью для всех серий для числа процессоров не более 7 отношение "чистого" времени, затраченного на счет, к общему времени решения постоянно и не зависит от параметра  $\gamma$ . Для большего числа процессоров и для малых значений параметра  $\gamma$  (менее 0,3) отношение времени счета к общему времени решения задачи значительно больше, а при больших  $\gamma$  это значение становится постоянным. Для двух процессоров отношение времени счета к общему времени решения задачи — самое большое и составляет порядка 0,8. Это связано, прежде всего, с особенностями архитектуры машины. Все процессоры объединены в пары, и скорость обмена между процессорами одной пары больше, чем между процессорами из различных пар.

Для серий задач 2 и 3 при расчетах с распределенной памятью с увеличением числа процессоров отношение времени непосредственного счета

к общему времени решения задачи уменьшается при фиксированном значении параметра  $\gamma$ . Для машины с общей памятью такие выводы сделать не удастся, так как получены результаты не для всех значений параметра  $\gamma$ .

В табл. 2 представлены коэффициенты ускорения для всех серий. Для серии 1, 2, 3 для машины с распределенной памятью максимальное ускорение (13,07; 7,16; 6,58) достигается соответственно для  $\gamma = 0,2; 0,3; 0,2$  на 20 процессорах, а для машины с общей памятью — максимальное ускорение (8,74; 9,71; 8,53) при  $\gamma = 0,2, 0,3, 0,2$  на 20 процессорах.

Для всех серий задач на машине с распределенной памятью при увеличении параметра  $\gamma$  от 0,1 до 0,6 наблюдается тенденция неубывания коэффициента ускорения (см. табл. 2) при фиксированном числе процессоров. Для значений  $\gamma > 0,6$  это ускорение практически постоянно с резким изменением для  $\gamma = 0,9$ .

Таблица 2

**Коэффициент ускорения метода динамического программирования для задач и для машины с общей памятью Hewlett Packard SuperDome (64CPU) (о) и с распределенной памятью MVS 15000 VM (р)**

$n_p$	$\gamma$																	
	0,1		0,2		0,3		0,4		0,5		0,6		0,7		0,8		0,9	
	о	р	о	р	о	р	о	р	о	р	о	р	о	р	о	р	о	р
Серия 1																		
2	1,82	1,62	1,93	1,61	—	1,60	—	1,61	—	1,62	—	1,62	—	1,62	—	1,41	—	1,62
3	2,54	0,92	2,74	0,94	—	0,96	—	0,97	—	0,98	—	0,90	—	0,92	—	0,84	—	0,59
4	3,32	1,58	3,56	1,63	—	1,65	—	1,68	—	1,64	—	1,66	—	1,53	—	1,36	—	1,66
5	3,89	1,89	4,19	2,01	—	2,05	—	2,08	—	2,06	—	2,05	—	1,97	—	1,54	—	1,20
6	4,47	2,23	4,79	2,17	—	2,23	—	2,23	—	2,31	—	2,32	—	2,30	—	1,67	—	1,06
7	4,68	4,62	5,27	2,51	—	2,36	—	2,54	—	2,55	—	2,55	—	2,55	—	2,32	—	1,21
8	5,41	5,25	5,80	2,57	—	2,62	—	2,80	—	2,69	—	2,77	—	2,75	—	2,36	—	1,13
9	5,60	4,80	6,44	2,63	—	2,67	—	2,99	—	2,94	—	2,96	—	3,18	—	2,51	—	1,11
10	5,80	6,06	6,56	3,68	—	3,76	—	3,93	—	3,95	—	3,97	—	3,93	—	3,50	—	1,32
15	5,85	9,54	7,37	9,81	—	5,33	—	5,70	—	5,75	—	5,88	—	5,65	—	5,13	—	1,38
20	7,41	12,45	8,74	13,07	—	7,87	—	6,35	—	6,69	—	7,15	—	6,79	—	6,19	—	1,31
Серия 2																		
2	1,86	1,60	1,90	1,60	1,93	1,59	—	1,59	—	1,61	—	1,61	—	1,62	—	1,62	—	1,62
3	2,71	0,93	2,81	0,96	2,76	0,98	—	0,99	—	1,00	—	1,00	—	1,00	—	1,00	—	0,63
4	3,45	1,48	3,56	1,52	3,43	1,55	—	1,56	—	1,59	—	1,58	—	1,58	—	1,59	—	1,57
5	3,91	1,22	4,21	1,30	4,33	1,33	—	1,36	—	1,37	—	1,38	—	1,37	—	1,35	—	0,84
6	4,63	1,51	5,06	1,47	5,01	1,51	—	1,53	—	1,55	—	1,56	—	1,56	—	1,57	—	1,16
7	4,95	3,46	5,69	1,16	5,63	1,16	—	1,20	—	1,22	—	1,34	—	1,26	—	1,26	—	0,83
8	5,48	3,93	5,85	1,13	5,97	1,28	—	1,28	—	1,29	—	1,28	—	1,36	—	1,37	—	1,09
9	5,76	3,21	6,34	1,18	6,44	1,18	—	1,21	—	1,18	—	1,20	—	1,25	—	1,26	—	0,89
10	5,85	3,37	6,88	1,21	7,06	1,28	—	1,27	—	1,30	—	1,26	—	1,33	—	1,30	—	1,08
15	6,92	5,71	8,11	5,23	8,81	1,02	—	1,12	—	1,09	—	1,15	—	1,14	—	1,18	—	0,88
20	7,28	5,60	9,03	5,76	9,71	7,16	—	1,28	—	1,35	—	1,37	—	1,37	—	1,38	—	1,18
Серия 3																		
2	1,84	1,61	1,91	1,61	—	1,60	—	1,62	—	1,61	—	1,59	—	1,62	—	1,62	—	1,63
3	2,47	0,95	2,72	0,97	—	0,99	—	1,00	—	1,00	—	1,00	—	1,01	—	1,02	—	0,63
4	3,20	1,52	3,41	1,55	—	1,59	—	1,58	—	1,57	—	1,58	—	1,60	—	1,60	—	1,58
5	3,79	1,25	4,14	1,32	—	1,35	—	1,38	—	1,37	—	1,37	—	1,35	—	1,34	—	0,82
6	4,31	1,24	4,79	1,29	—	1,34	—	1,39	—	1,37	—	1,39	—	1,39	—	1,37	—	1,04
7	5,08	1,12	5,27	1,24	—	1,22	—	1,24	—	1,28	—	1,28	—	1,30	—	1,29	—	0,89
8	4,79	4,08	5,64	1,31	—	1,34	—	1,30	—	1,34	—	1,35	—	1,35	—	1,33	—	1,11
9	5,40	3,50	6,35	1,14	—	1,22	—	1,22	—	1,29	—	1,22	—	1,24	—	1,28	—	0,93
10	5,27	3,61	6,46	1,20	—	1,28	—	1,28	—	1,31	—	1,33	—	1,31	—	1,30	—	1,06
15	6,38	6,06	7,94	5,47	—	1,07	—	1,17	—	1,10	—	1,19	—	1,23	—	1,24	—	0,90
20	7,03	6,63	8,53	6,58	—	1,15	—	1,01	—	1,10	—	1,12	—	1,12	—	1,16	—	0,97

Отношение времени непосредственного счета ко всему времени решения для задач методом ветвей и границ для машины с общей памятью Hewlett Packard SuperDome (64CPU) (о) и распределенной памятью MVS 15000 VM (р) для рабочих процессоров

$n_p$	$\gamma$																	
	0,1		0,2		0,3		0,4		0,5		0,6		0,7		0,8		0,9	
	о	р	о	р	о	р	о	р	о	р	о	р	о	р	о	р	о	р
Серия 1																		
2	0,74	0,53	0,91	0,69	0,92	0,66	0,96	0,81	0,94	0,62	0,98	0,86	—	0,83	0,88	0,93	0,97	0,96
3	0,71	0,44	0,89	0,64	0,90	0,61	0,95	0,77	0,94	0,60	0,98	0,84	0,98	0,82	0,91	0,91	0,93	0,95
4	0,67	0,32	0,88	0,56	0,88	0,49	0,93	0,72	0,93	0,47	0,97	0,83	0,97	0,81	0,94	0,91	0,95	0,95
5	0,60	0,27	0,86	0,47	0,88	0,43	0,92	0,70	0,93	0,38	0,97	0,81	0,97	0,78	0,96	0,91	0,96	0,94
6	0,54	0,21	0,85	0,40	0,84	0,34	0,91	0,65	0,92	0,32	0,97	0,79	0,97	0,74	0,95	0,89	0,97	0,94
7	0,48	0,19	0,83	0,36	0,83	0,31	0,90	0,61	0,72	0,28	0,97	0,76	0,97	0,68	0,97	0,89	0,95	0,93
8	0,41	0,16	0,81	0,31	0,81	0,29	0,89	0,60	0,91	0,24	0,96	0,72	0,96	0,62	0,96	0,89	0,95	0,93
9	0,39	0,14	0,79	0,29	0,80	0,26	0,89	0,54	0,90	0,21	0,95	0,67	0,96	0,57	0,95	0,87	0,94	0,94
10	0,35	0,13	0,78	0,28	0,80	0,24	0,88	0,53	0,89	0,19	0,96	0,65	0,96	0,52	0,95	0,84	0,95	0,93
15	0,25	0,11	0,70	0,20	0,71	0,21	0,85	0,45	0,81	0,13	0,94	0,46	0,95	0,37	0,93	0,73	0,93	0,93
20	0,19	0,10	0,59	0,18	0,63	0,17	0,82	0,44	0,73	0,11	0,93	0,36	0,94	0,29	0,92	0,61	0,92	0,93
Серия 2																		
2	0,97	0,90	0,99	0,93	0,99	0,93	0,99	0,95	0,85	0,96	0,97	0,97	0,98	0,97	0,98	0,97	0,92	0,97
3	0,95	0,84	0,98	0,91	0,98	0,90	0,98	0,93	0,91	0,94	0,90	0,95	0,92	0,94	0,94	0,95	0,92	0,94
4	0,94	0,84	0,97	0,89	0,97	0,88	0,97	0,93	0,94	0,92	0,94	0,94	0,92	0,92	0,95	0,93	0,93	0,92
5	0,94	0,83	0,96	0,87	0,97	0,87	0,96	0,92	0,94	0,91	0,95	0,93	0,91	0,90	0,95	0,91	0,91	0,91
6	0,93	0,82	0,96	0,86	0,96	0,88	0,95	0,91	0,93	0,89	0,94	0,92	0,91	0,90	0,94	0,90	0,90	0,91
7	0,93	0,83	0,96	0,86	0,96	0,87	0,95	0,91	0,93	0,89	0,93	0,92	0,91	0,89	0,94	0,89	0,89	0,90
8	0,93	0,83	0,95	0,85	0,95	0,87	0,95	0,90	0,92	0,88	0,93	0,91	0,90	0,89	0,93	0,89	0,89	0,90
9	0,93	0,80	0,94	0,85	0,95	0,85	0,95	0,90	0,91	0,89	0,92	0,91	0,89	0,88	0,93	0,89	0,89	0,90
10	0,93	0,81	0,95	0,85	0,95	0,84	0,94	0,89	0,91	0,89	0,92	0,91	0,89	0,87	0,92	0,88	0,89	0,90
15	0,92	0,80	0,95	0,84	0,95	0,82	0,94	0,89	0,91	0,88	0,91	0,91	0,88	0,88	0,92	0,88	0,89	0,90
20	0,92	0,80	0,95	0,84	0,95	0,82	0,93	0,89	0,90	0,89	0,92	0,90	0,88	0,88	0,91	0,88	0,89	0,89

При фиксированном значении параметра  $\gamma$  с увеличением числа процессоров коэффициент ускорения имеет тенденцию к увеличению. Для машины с распределенной памятью этот процесс происходит резкими скачками, а для машины с общей памятью таких скачков не наблюдается.

**Результаты вычислительного эксперимента для метода ветвей и границ.** Время решения задачи методом ветвей и границ на распределенных системах можно разбить на две основные составляющие: время на непосредственный счет, время на обмен.

Время на непосредственный счет включает в себя временные затраты на реализацию следующих операций:

- ветвление;
- решение оценочных задач;
- создание структур данных;
- поиск данных;
- другие накладные расходы, обусловленные организацией процесса вычислений.

Время на обмен включает в себя временные затраты на реализацию следующих операций:

- подготовку информации к отправке;
- передачу информации по сети;
- обработку принятой информации.

Задачи из серии 3 не были решены, так как в процессе решения возникает достаточно большое число неотсеянных вершин и происходит переполнение памяти. В табл. 3 представлено отно-

шение времени непосредственного счета ко всему времени решения задач для серий 1, 2. Как и следовало ожидать, для одинаковых классов задач доля времени на непосредственные вычисления от общего времени для рабочего процессора машины с общей памятью больше, чем для машины с распределенной памятью.

Для серии 1 (табл. 3) с увеличением параметра  $\gamma$  при фиксированном числе процессоров наблюдается тенденция увеличения отношения времени на вычисления к общему времени для рабочего процессора, для машины с общей памятью условия монотонности практически не нарушаются, а для машины с распределенной памятью наблюдаются нарушения. При фиксированном значении  $\gamma$  с увеличением числа процессоров наблюдается практически монотонное изменение этого отношения.

Как видно из табл. 3, для серии 2 значение отношения практически не зависит ни от числа процессоров, ни от значения параметра  $\gamma$ , ни от вида памяти.

На рис. 1—4 представлены зависимости от числа процессоров суммарного по всем процессорам числа ветвлений для различных  $\gamma$  в зависимости от числа процессоров. На графиках видно, что суммарное число ветвлений с увеличением числа процессоров существенно не изменяется.

В табл. 4 представлены коэффициенты ускорения для серий 1 и 2.

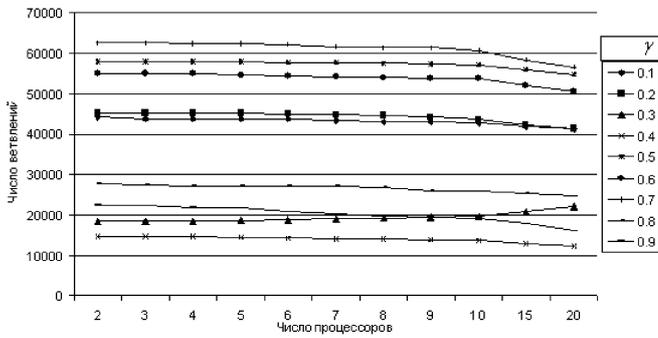


Рис. 1. Общее число ветвлений для задач серии 1 для машины с общей памятью Hewlett Packard SuperDome (64CPU)

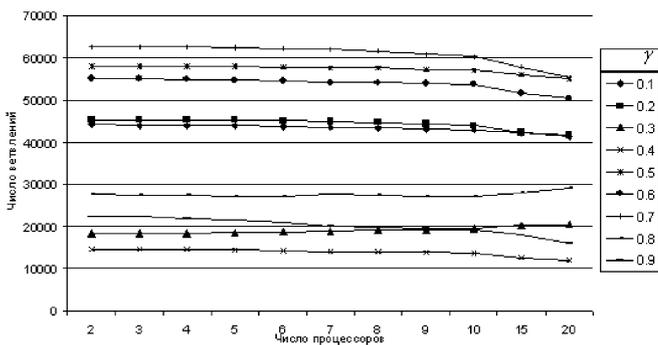


Рис. 2. Общее число ветвлений для задач серии 1 для машины с распределенной памятью MVS 15000 VM

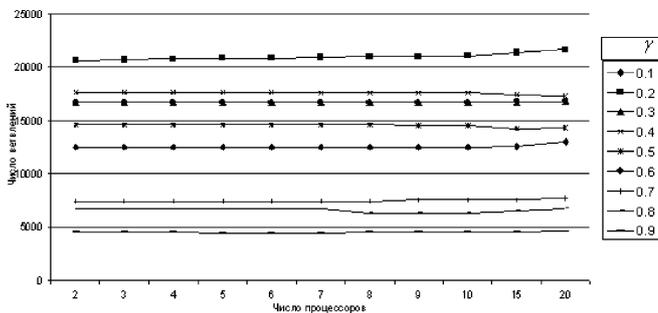


Рис. 3. Общее число ветвлений для задач серии 2 для машины с общей памятью Hewlett Packard SuperDome (64CPU)

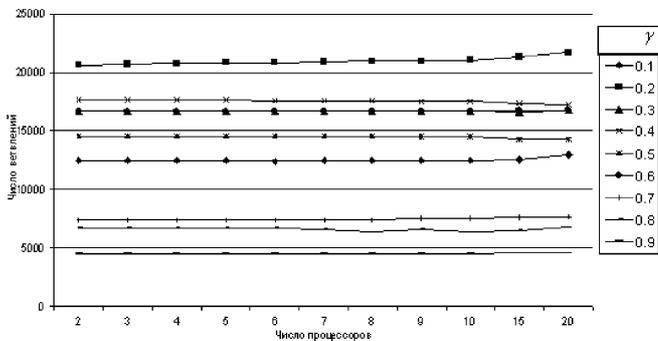


Рис. 4. Общее число ветвлений для задач серии 2 для машины с распределенной памятью MVS 15000 VM

Для *серии 1* (табл. 4) для машины с распределенной памятью максимальное ускорение (9,8) достигается для  $\gamma = 0,9$  на 20 процессорах, а для машины с общей памятью — максимальное ускорение (16,85) при  $\gamma = 0,6$  на 20 процессорах.

При увеличении параметра  $\gamma$  при фиксированном числе процессоров наблюдается, как правило, тенденция увеличения коэффициента ускорения машины с распределенной памятью для любого числа процессоров, а для машины с общей памятью для числа процессоров 15 и 20 наблюдается сначала увеличение  $\gamma$ , а затем уменьшение.

Для задач *серии 2* для различных машин тенденции поведения коэффициента ускорения схожи. Максимальное ускорение достигается для  $\gamma = 0,6$  на 20 процессорах.

При фиксированном числе процессоров, как правило, наблюдается тенденция увеличения коэффициента ускорения при увеличении параметра  $\gamma$  от 0,1 до 0,7, а затем его уменьшение для фиксированного числа процессоров.

Из табл. 4 видно, что для одного и того же числа процессоров ускорение для машины с общей памятью больше по сравнению с ускорением для распределенной памяти для всех значений параметра  $\gamma$ . Такой результат представляется вполне логичным.

**Сравнение с другими работами.** В данной работе рассмотрено несколько серий задач о ранце, размерность которых больше по сравнению с задачами из других работ. Исследован средний коэффициент загрузки процессоров (отношение времени непосредственного счета ко всему времени решения задачи), такое исследование не проводилось в других работах.

В работе [8] дано сравнение параллельных алгоритмов ветвей и границ (поиск в глубину и наилучшая оценка) на GRID-технологиях [2, 4]. Приведены ускорения для задачи о ранце размерности 35. Для поиска в глубину рассматривались два варианта:  $R = 22$ ,  $R = 12$ . Способ генерации параметров  $a_j$  и  $c_j$  не указан. Результаты приведены в табл. 5. Для числа процессоров из первого столбца указаны ускорения в столбцах 2 и 3.

Из столбца для  $R = 22$  видно, что увеличение числа процессоров в 2 раза ускорение возрастает также в 2 раза — линейный рост. Монотонный рост ускорения наблюдается для обоих значений  $R$ .

В статье [8] предложена идея иерархического управляющего-рабочего, которая эффективно использует гранулярную вычислительную среду (GRID).

В работе [9] проводится сравнение параллельных алгоритмов ветвей и границ для двух стратегий ветвления: поиск в глубину (одностороннее ветвление) и поиск по наилучшей верхней оценке (фронтальное ветвление). Здесь находится  $\epsilon$ -при-

Коэффициент ускорения метода ветвей и границ для задач для машины с общей памятью Hewlett Packard SuperDome (64CPU) (о) и распределенной памятью MVS 15000 VM (р)

$n_p$	$\gamma$																	
	0,1		0,2		0,3		0,4		0,5		0,6		0,7		0,8		0,9	
	о	р	о	р	о	р	о	р	о	р	о	р	о	р	о	р	о	р
Серия 1																		
2	0,74	0,53	0,91	0,69	0,92	0,66	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	1,41	0,88	1,79	1,28	1,78	1,19	1,96	1,85	2,00	1,89	2,01	1,95	2,02	1,94	2,08	1,98	1,92	1,98
4	1,94	0,98	2,66	1,68	2,62	1,48	2,83	2,54	2,96	2,26	3,01	2,88	3,03	2,87	3,21	2,95	2,92	2,95
5	2,21	1,02	3,43	1,86	3,35	1,61	3,52	3,06	3,93	2,44	3,99	3,74	4,00	3,69	4,35	3,83	3,91	3,91
6	2,37	1,03	4,12	1,95	3,97	1,66	4,18	3,38	4,87	2,53	4,95	4,56	4,95	4,34	5,33	4,64	4,86	4,84
7	2,44	1,04	4,69	1,97	4,49	1,70	4,68	3,60	4,56	2,62	5,91	5,16	5,89	4,79	6,31	5,36	5,72	5,68
8	2,49	1,05	5,18	1,98	5,00	1,72	5,21	3,78	6,66	2,59	6,87	5,67	6,78	5,07	7,22	6,03	6,47	6,36
9	2,54	1,05	5,62	1,98	5,49	1,70	5,63	3,91	7,51	2,60	7,78	5,98	7,70	5,33	8,09	6,62	7,12	6,98
10	2,54	1,07	6,03	2,03	5,81	1,74	6,03	4,06	8,35	2,63	8,68	6,29	8,58	5,52	8,97	7,04	7,74	7,56
15	2,57	1,07	6,89	2,05	6,88	1,83	7,46	4,42	11,32	2,62	13,00	6,96	12,84	5,90	11,78	7,41	9,91	9,37
20	2,36	1,08	7,10	2,10	7,38	1,87	8,23	4,54	12,80	2,63	16,85	7,29	16,68	6,09	13,61	6,89	10,81	9,80
Серия 2																		
2	0,97	0,90	0,99	0,93	0,99	0,93	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	1,70	1,45	1,95	1,79	1,94	1,77	1,96	1,92	2,14	1,95	1,87	1,96	1,85	1,91	1,89	1,93	1,99	1,91
4	2,07	1,73	2,80	2,51	2,83	2,54	2,68	2,60	3,27	2,85	2,85	2,81	2,68	2,70	2,74	2,73	2,82	2,66
5	2,26	1,87	3,47	3,06	3,50	3,08	3,21	3,14	4,26	3,60	3,67	3,55	3,35	3,34	3,42	3,36	3,24	2,99
6	2,37	1,95	3,96	3,40	3,96	3,46	3,67	3,57	5,04	4,25	4,35	4,14	3,90	3,82	3,96	3,86	3,29	3,06
7	2,45	2,02	4,28	3,71	4,34	3,76	4,05	3,90	5,56	4,69	4,89	4,61	4,15	4,05	4,39	4,26	3,35	3,13
8	2,49	2,05	4,54	3,87	4,63	4,00	4,35	4,19	5,95	4,97	5,25	4,93	4,35	4,22	4,80	4,62	3,40	3,14
9	2,53	2,04	4,68	4,01	4,80	4,13	4,59	4,40	6,20	5,23	5,53	5,17	4,48	4,36	5,03	4,74	3,43	3,17
10	2,56	2,05	4,87	4,14	4,92	4,20	4,80	4,60	6,33	5,38	5,79	5,40	4,61	4,46	5,12	4,84	3,44	3,16
15	2,62	2,07	5,32	4,42	5,30	4,44	5,25	4,97	6,69	5,58	6,38	5,96	4,74	4,63	5,30	5,03	3,48	3,23
20	2,64	2,06	5,471	4,50	5,57	4,61	5,37	5,07	6,72	5,65	6,82	6,23	4,74	4,63	5,35	5,09	3,48	3,23

ближенное решение, но значение  $\varepsilon$  не указано. В табл. 6 приведены ускорения для 10 задач о ранце размерности 35. Параметры  $a_j$  формировались как случайные между 0 и 100,  $c_j = a_j + 10$ , способ генерации  $R$  не указан. Расчет выполнялся на машине с распределенной памятью.

Из табл. 6 видно, что ускорение для фронтального ветвления для любого числа процессоров больше.

В работе [10] проведено сравнение метода ветвей и границ и динамического метода для задачи о ранце на различных машинах. В трех типах задач  $a_j$  и  $c_j$  формировались как целые случайные на отрезках [1, 64], [1, 512], [1, 1024]. Для всех типов задач рассматривались размерности  $n = 100$ ,  $n = 200$ ,  $n = 300$ . Самое большое возможное значение для метода динамического программирования

составило  $R = 4095$ , а для метода ветвей и границ  $R = 4200$ .

Описание параллельных реализаций методов не было приведено. В параллельном варианте был реализован только метод динамического программирования, для которого достигается ускорение порядка 8 на 10 процессорах, а далее с увеличением числа процессоров ускорение не увеличивается. Для метода ветвей и границ приведены коэффициенты среднего параллелизма и было сделано предположение, что распараллеливание будет эффективно.

Полученный коэффициент в работе [10] выше, чем в данной, прежде всего потому, что размерность в [10] на порядок меньше, что является существенным фактором для параллельной реализации метода динамического программирования.

Таблица 5

Число процессоров	$R = 22$	$R = 12$
1	1	1
2	1,987	1,887
4	3,923	3,393
8	7,968	5,669
16	15,872	8,471
32	31,504	11,271
64	61,934	13,571
128	119,578	30,055
256	224,019	73,133

Таблица 6

Число процессоров	Одностороннее ветвление	Фронтальное ветвление
1	1	1
2	1,91	1,95
4	3,32	3,73
8	5,92	6,96
16	9,85	13,00
32	15,63	23,16
64	24,82	39,40
128	45,25	70,20
256	92,63	108,89

В статье [11] реализован алгоритм динамического программирования для размерности  $n = 1000$ ,  $R = 1\ 500\ 000$ , где  $a_j$  и  $c_j$  формировались как случайные на отрезке  $[10, 1000]$ , на восьми процессорах ускорение достигалось порядка 2,5. Способ формирования начальных данных незначительно отличается от серии 2 при  $\gamma = 0,3...0,4$ , для которой коэффициент ускорения составляет 5,97 на машине с общей памятью и 1,28 с распределенной, откуда следует, что коэффициент ускорения существенно зависит от вычислительной системы.

### Заключение

Рассмотрены и реализованы параллельные алгоритмы методов динамического программирования и метода ветвей и границ. Был проведен обширный вычислительный эксперимент, в котором представлены зависимости в табличном виде от числа процессоров и отношения времени счета к общему времени решения. Также для рассмотренных серий задач изучена зависимость ускорения от числа процессоров. Представлены графические зависимости числа ветвлений от числа процессоров для метода ветвей и границ.

Экспериментально подтвержден уже известный факт, что для одного и того же числа процессоров ускорение для машины с общей памятью больше по сравнению с ускорением для распределенной памяти для всех значений параметра  $\gamma$ .

Для обоих методов получено уменьшение доли времени, требуемого для счета, от общего времени решения с ростом числа процессоров. Для метода ветвей и границ серии 2 на машине с распределенной памятью это отношение постоянно.

Хотя в данной работе не сравниваются алгоритмы методов и ветвей и границ и динамического программирования, отметим что в среднем алгоритм динамического программирования работает медленнее, но имеет больший коэффициент ускорения. Для числа процессоров 20 и больше время решения для обоих методов практически совпадают.

Предложен комбинированный алгоритм динамического программирования и ветвей и границ.

Проведено сравнение с другими работами, в которых исследовались параллельные алгоритмы для задачи о ранце.

*Автор выражает благодарность А. Б. Барскому и И. Х. Сигалу за внимание к работе и обсуждение результатов.*

### Список литературы

1. Сигал И. Х., Иванова А. П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: Учеб. изд. 2-е, доп. и испр. М.: Физматлит, 2007.
2. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2004.
3. Martello S., Toth P. Knapsack problems: algorithms and computer implementations. Chichester: John Wiley&Sons Ltd. 1990.
4. Барский А. Б. Параллельные информационные технологии: Учебн. пос. М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2007. 503 с.
5. Kellerer H., Pfershy U., Pisinger D. Knapsack Problems. Berlin: Springer Verlag, 2004.
6. Галимьянова Н. Н. Сравнение эффективности параллельных алгоритмов ветвей и границ и динамического программирования для решения задачи об одномерном булевом ранце // Информационные технологии. № 10. 2006. С. 33—39.
7. Посыпкин М. А., Сигал И. Х., Галимьянова Н. Н. Параллельные алгоритмы в задачах дискретной оптимизации: вычислительные модели, библиотека, результаты экспериментов. М.: ВЦ РАН, 2006.
8. Aida K., Futakata Y., Osumi T. Parallel Branch and Bound Algorithm with the Hierarchical Master-Worker Paradigm on the Grid // IPSJ Trans. on Advanced Computing Systems. Sep. 2006. Vol. 47. No. SIG 12 (ACS 15).
9. Li G. J., Wah B. W. How to Cope with Anomalies in Parallel Approximate Branch-and-Bound Algorithms, Proc. National Conf. on Artificial Intelligence. AAAI, Aug. 1894. P. 212—215.
10. Kindervater G. A. P., Trienekens H. W. J. M. Experiments with parallel algorithms for combinatorial problems European Journal of Operation Research. 1988. Vol. 33. N 1. P. 65—81. North-Holland.
11. Andonov R., Raimbault F., Qumton P. Dynamic Programming Parallel Implementations for the Knapsack Problem. Technical Report PI-740, IRISA, Campus de Beaulieu, Rennes, France, 1993.
12. Посыпкин М. А., Сигал И. Х., Галимьянова Н. Н. Алгоритмы параллельных вычислений для решения некоторых классов задач дискретной оптимизации. М.: ВЦ РАН, 2005.
13. Посыпкин М. А., Сигал И. Х. Исследование алгоритмов параллельных вычислений в задачах дискретной оптимизации ранцевого типа // ЖВМ и МФ. 2005. Т. 45. № 10. С. 1801—1809.
14. Sigal I. Kh., Galimyanova N. N. Parallel algorithms of the decision of a knapsack problem. VI International congress on mathematical modeling / Book of abstracts / September 20—26, 2004. Nizhny Novgorod. P. 60.
15. Посыпкин М. А., Сигал И. Х., Галимьянова Н. Н. Алгоритмы параллельных вычислений для решения некоторых классов задач дискретной оптимизации. Информационные модели экономики // Сб. тр. III Всероссийской научно-практ. конф. (Москва, 20 декабря 2005 г. — 20 января 2006 г.). С. 91—92.
16. Тимошевская Н. Е. Параллельные методы обхода дерева // Матем. моделирование. 2004. Т. 16. № 4. С. 105—114.
17. Межведомственный суперкомпьютерный центр РАН. <http://www.jssc.ru>.
18. Информационно-аналитический центр по параллельным вычислениям. <http://www.parallel.ru>.

# НЕЙРОСЕТИ И НЕЙРОКОМПЬЮТЕРЫ

УДК 004.93

**С. А. Субботин**, канд. техн. наук, доц.  
Запорожский национальный  
технический университет, Украина

## Метод синтеза классифицирующих нейронечетких сетей с учетом значимости термов признаков

*Предложен неитеративный метод синтеза классифицирующих нейронечетких сетей, основанный на разбиении диапазонов значений признаков на интервалы равной длины и определении принадлежности распознаваемого экземпляра к классам по принадлежностям к перекрывающимся интервалам значений признаков с учетом их значимостей.*

**Ключевые слова:** нейронечеткие сети, термы признаков, распознавание признаков, синтез.

### Введение

Нейронечеткие сети являются общепризнанным и весьма эффективным инструментом для решения широкого класса задач, связанных с распознаванием образов и автоматизацией принятия решений в системах управления и диагностики.

Классические методы синтеза нейронечетких моделей [1, 2], как правило, предполагают выполнение настройки их весовых коэффициентов и параметров функций принадлежности на основе итеративной оптимизации, реализуемой на основе модификаций метода обратного распространения ошибки.

Существенными недостатками классических методов синтеза нейронечетких моделей являются низкая скорость обучения, обусловленная итеративной коррекцией параметров, чрезвычайно высокая зависимость качества и времени обучения от выбора начальной точки поиска в пространстве оптимизируемых переменных, возможность попадания в локальные минимумы целевой функции обучения.

В качестве альтернативы классическим методам итеративной настройки нейронечетких моделей ранее автором были предложены неитеративные методы [3–8], основанные на выделении для каждого признака интервалов значений, в которых номер класса остается неизменным. При этом число выделяемых интервалов для признаков оказывается разным, а сами интервалы имеют различную ширину.

Достоинствами данного подхода являются высокий уровень изоморфности нейронечеткой сети топологии обучающего множества и легкость последующего анализа человеком. Однако аппаратная реализация нейронечетких сетей, настраиваемых с помощью данных методов, является несколько затруднительной, поскольку число интервалов для признаков априорно неизвестно.

Целью данной работы является создание нового метода, позволяющего в неитеративном режиме синтезировать нейронечеткие модели, легко реализуемые аппаратно и программно и в то же время удобные для восприятия и анализа человеком.

### Постановка задачи синтеза нейронечеткой модели

Пусть мы имеем обучающую выборку  $\langle x, y \rangle$ , состоящую из множества экземпляров  $\{x^s\} = \{x_j^s\}$ ,  $s = 1, 2, \dots, S$ ,  $j = 1, 2, \dots, N$ , где  $S$  — число экземпляров в выборке;  $N$  — число признаков, характеризующих экземпляры;  $x_j^s$  — значение  $j$ -го признака, характеризующего  $s$ -й экземпляр выборки  $x^s$ , и набора значений целевого (выходного) признака  $y = \{y^s\}$ , поставленных в соответствие каждому экземпляру выборки, где  $y^s$  — значение целевого признака для экземпляра  $x^s$ ,  $y^s \in \{1, 2, K\}$ ,  $K$  — число классов.

Тогда задача синтеза модели зависимости  $y(x)$  по выборке  $\langle x, y \rangle$  на основе нейронечеткой сети заключается в определении числа нейроэлементов в сети, задании их дискриминантных (постсинаптических, весовых) и активационных функций, определении топологии сети и расчете весовых коэффициентов межнейронных связей.

### Формирование нечетких термов признаков и определение их значимостей для определения классов

Нечеткие термы будем формировать для равных по ширине интервалов значений каждого из признаков. При этом функции принадлежности будем задавать таким образом, чтобы, с одной стороны, учитывать положение распознаваемого экземпляра относительно центра интервала термина, а с другой стороны, обеспечить частичное перекрытие термов для учета нелинейностей границ классов. После формирования нечетких термов представляется целесообразным определить оценки значимости (важности, информативности) соответствующих термов для определения принадлежности распознаваемого экземпляра к каждому классу.

Для формирования нечетких термов и оценивания их значимости предлагается выполнять последовательность шагов 1–5.

**Шаг 1.** Инициализация. Задать обучающую выборку  $\langle x, y \rangle$ . Задать число уровней квантования значений признаков  $q$  и масштабирующий коэффициент ширины окна активации нечетких термов  $\delta$ ,  $0 \leq \delta < 0,5$ .

**Шаг 2.** Разбить диапазон значений каждого признака на  $q$  равных интервалов, шириной  $q_i = (\max(x_i) - \min(x_i))/q$ .

**Шаг 3.** Сформировать нечеткие термы для каждого признака. Для чего задать функции принадлежности:

$$\mu_{i,j}(x_i^s) = \begin{cases} 0, & x_i^s \leq q_i(j - \delta - 1); \\ 0, & x_i^s \geq q_i(j + \delta); \\ \frac{x_i^s - q_i(j - \delta - 1)}{q_i(0,5 - j + \delta)}, & q_i(j - \delta - 1) < x_i^s \leq q_i(2j - 1,5); \\ \frac{x_i^s - q_i(j + \delta)}{q_i(j - 1,5 - \delta)}, & q_i(2j - 1,5) < x_i^s \leq q_i(j + \delta), \end{cases}$$

где  $\mu_{i,j}(x_i^s)$  — функция принадлежности распознаваемого  $s$ -го экземпляра выборки  $x^s$  по  $i$ -му признаку к  $j$ -му терму  $i$ -го признака,  $i = 1, 2, \dots, N$ ;  $j = 1, 2, \dots, q$ .

**Шаг 4.** Найти для каждого признака каждого экземпляра обучающей выборки значения функций принадлежности экземпляра к нечетким термам

$$\mu_{i,j}(x_i^s), \quad s = 1, 2, \dots, S, \quad i = 1, 2, \dots, N; \\ j = 1, 2, \dots, q.$$

**Шаг 5.** Определить значения коэффициентов, характеризующих априорную принадлежность  $j$ -го терма  $i$ -го признака к  $k$ -му классу:

$$\alpha_{i,j,k} = \frac{\sum_{s=1}^S \mu_{i,j}(x_i^s), y^s = k}{\sum_{s=1}^S \mu_{i,j}(x_i^s)}.$$

### Синтез нейронечеткой сети

На основе информации о выделенных нечетких термах возможно синтезировать трехслойную нейронечеткую сеть, схема которой изображена на рисунке.

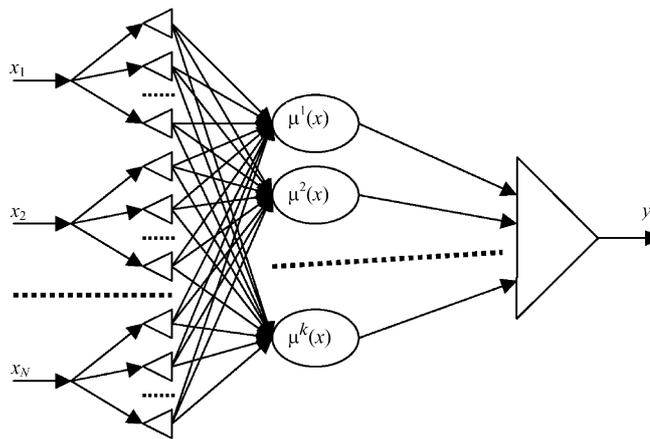


Схема нейронечеткой сети

На входы нейронечеткой сети поступают значения соответствующих признаков распознаваемого экземпляра.

Нейроны первого слоя сети осуществляют вычисление значений функций принадлежности распознаваемого экземпляра к нечетким термам признаков.

Нейроны второго слоя сети объединяют принадлежностей распознаваемого экземпляра к термам признаков в принадлежностей к термам классов с учетом значимости термов признаков для определения соответствующего класса (на рисунке обозначены как  $\mu^k(x)$ ).

Единственный нейрон последнего слоя сети осуществляет объединение принадлежностей к классам и дефаззификацию результата.

Дискриминантные функции нейронов нейронечеткой сети будут задаваться формулами

$$\varphi^{(2,i)}(w_j^{(2,i)}, x_j^{(2,i)}) = w_j^{(2,i)} x_j^{(2,i)}, \quad i = 1, 2, \dots, K; \\ j = 1, 2, \dots, Nq,$$

$$\varphi^{(3,1)}(w_j^{(3,1)}, x_j^{(3,1)}) = \frac{\sum_j w_j^{(2,i)} x_j^{(2,i)}}{\sum_j x_j^{(2,i)}}, \\ j = 1, 2, \dots, K,$$

где  $\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)})$  — дискриминантная (пост-синаптическая, весовая) функция  $i$ -го нейрона  $\eta$ -го слоя;  $w^{(\eta,i)}$  — набор весовых коэффициентов  $i$ -го нейрона  $\eta$ -го слоя;  $w^{(\eta,i)} = \{w_j^{(\eta,i)}\}$ ,  $w_j^{(\eta,i)}$  — весовой коэффициент  $j$ -го входа  $i$ -го нейрона  $\eta$ -го слоя;  $x^{(\eta,i)}$  — набор входов  $i$ -го нейрона  $\eta$ -го слоя,  $x^{(\eta,i)} = \{x_j^{(\eta,i)}\}$ ,  $x_j^{(\eta,i)}$  — значение на  $j$ -м входе  $i$ -го нейрона  $\eta$ -го слоя сети.

Функции активации нейронов нейронечеткой сети будут задаваться формулами

$$\psi^{(2,i)}(\varphi^{(2,i)}) = \max_j \{\varphi_j^{(2,i)}\}$$

$$\text{или } \psi^{(2, i)}(\varphi^{(2, i)}) = \frac{1}{Nq} \sum_j \varphi_j^{(2, i)},$$

$$i = 1, 2, \dots, K; j = 1, 2, \dots, Nq,$$

$$\psi^{(3, 1)}(\varphi^{(3, 1)}) = \text{round}(\varphi^{(3, 1)}),$$

где  $\psi^{(\eta, i)}(x)$  — функция активации  $i$ -го нейрона  $\eta$ -го слоя сети.

Весовые коэффициенты нейронов нейронечеткой сети будут определяться по формуле

$$w_j^{(\eta, i)} = \begin{cases} \alpha_{i, j, k}, \eta = 2, i = 1, 2, \dots, K; \\ j = (p - 1)N + r, p = 1, 2, \dots, N, \\ r = 1, 2, \dots, q; \\ j, \eta = 3, i = 1, j = 1, 2, \dots, K. \end{cases}$$

### Заключение

Предложен новый метод синтеза нейронечетких классифицирующих моделей, позволяющий в неитеративном режиме настраивать параметры нейронечеткой модели. Нейронечеткие сети, синтезируемые на основе предложенного метода, являются логически прозрачными (удобны для последующего анализа и восприятия человеком), а также легко реализуются аппаратно и программно, что позволяет рекомендовать предложенный метод для широкого применения на практике при

решении задач неразрушающей диагностики и контроля качества промышленных изделий, биомедицинской диагностики, распознавания изображений и в других областях.

### Список литературы

1. Борисов В. В., Круглов В. В., Федулов А. С. Нечеткие модели и сети. М.: Горячая линия-Телеком, 2007. 284 с.
2. Abraham A. Neuro-Fuzzy Systems: State-of-the-Art Modeling Techniques // Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence / Eds.: J. Mira and A. Prieto. — Granada: Springer-Verlag, 2001. P. 269—276.
3. Субботин С. А. Синтез распознающих нейронечетких моделей с учетом информативности признаков // Нейрокомпьютеры: разработка, применение. 2006. № 10. С. 50—56.
4. Субботин С. А. Неитеративный синтез и редукция нейронечетких моделей // Искусственный интеллект. 2006. № 3. С. 323—330.
5. Субботин С. А. Методы синтеза нейронечетких классификаторов для случая нескольких классов // Информационные технологии. 2006. № 11. С. 31—36.
6. Субботин С. А. Метод формирования баз знаний для нейронечетких моделей // Нейроинформатика и ее приложения: Материалы XIV Всероссийского семинара, 6—8 октября 2006 г. / Под ред. А. Н. Горбаня, Е. М. Миркеса. Красноярск: ИВМ СО РАН, 2006. С. 116—118.
7. Субботин С. А. Синтез нейронечетких сетей с группировкой признаков // Программные продукты и системы. 2006. № 4. С. 3—7.
8. Субботин С. А. Синтез нейронечетких моделей для выделения и распознавания объектов на сложном фоне по двумерному изображению // Компьютерное моделирование та інтелектуальні системи: Збірник наукових праць / За ред. Д. М. Пізи, С. О. Субботіна. Запоріжжя: ЗНТУ, 2007. С. 68—91.

УДК 621.396.93

**Жуков И. С.,**

Ростовский военный институт ракетных войск  
им. Главного маршала артиллерии  
М. И. Неделина

## Применение нейронной сети для идентификации сигналов, передаваемых по командным радиолиниям космических аппаратов

*Рассмотрен вопрос применения нейронной сети для идентификации информации, циркулирующей в командных радиолиниях космического аппарата и подверженной искажению из-за помех, как в самой аппаратуре, так и при прохождении радиосигнала в атмосфере.*

**Ключевые слова:** помехи, помехоустойчивое кодирование, нейронная сеть.

Импульсные помехи на борту, паразитные наводки, излучения от других бортовых систем и от самой аппаратуры командных радиолиний (КРЛ) могут привести к ошибкам в приеме (передаче) отдельных

символов командных сообщений. Статистика показывает, что на космическом аппарате (КА) с большим количеством электро- и радиоэлектронной аппаратуры получить вероятность ошибки символа менее чем  $10^{-4} \dots 10^{-5}$  для КРЛ с большой пропускной способностью становится затруднительно [1].

Применение существующего метода помехоустойчивого кодирования, такого как избыточное кодирование, приводит к существенному увеличению кодовой посылки. Минимальное число избыточных символов для исправления одиночной ошибки определяется из уравнения  $2^r = n$ , где  $r$  — число избыточных символов. Если возможна ситуация, когда ошибка не происходит, число избыточных символов нужно увеличить, чтобы одна из  $2^r$  комбинаций обозначала отсутствие ошибок. В свою очередь, увеличение длины кодовой посылки приводит к увеличению времени на ее передачу. Если КА находится, например, на расстоянии 1500 км, то на однократное прохождение радиоволнами этого расстояния потребуется 5 мс, что при большом объеме команд или передаче программы полета может оказаться недопустимым [2]. В связи с этим предлагается не увеличивать помехозащищенность радиолинии методом избыточного коди-

рования, а распознавать такое сообщение, принятое с ошибками, с помощью нейронной сети. Применение нейронной сети позволяет распознавать не только сообщение, принятое с ошибкой, но и практически искаженное либо принятое не полностью, когда космический аппарат при передаче (приеме) информации находился, например, на границе видимости, так как обученной нейронной сети достаточно 30% полезного сигнала для восстановления полной информации.

Нейронная сеть представляет собой совокупность нейроподобных элементов, определенным образом соединенных друг с другом и с внешней средой с помощью связей, определяемых весовыми коэффициентами. В зависимости от функций, выполняемых нейронами в сети, можно выделить три их типа [3]:

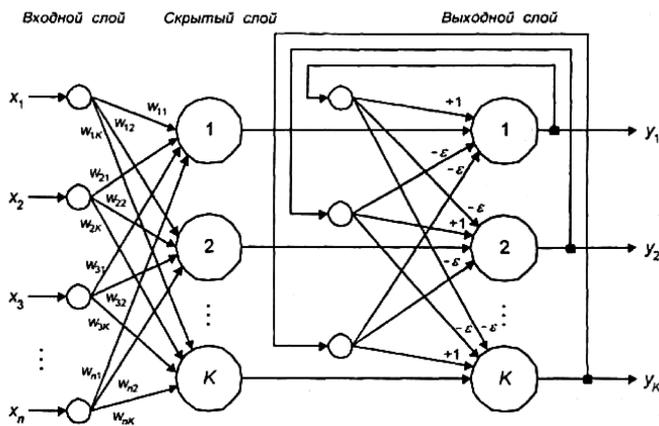
- *входные нейроны*, на которые подается вектор, кодирующий входное воздействие;
- *выходные нейроны*, выходные значения которых представляют выходы нейронной сети;
- *промежуточные нейроны*, составляющие основу нейронной сети.

Нейронная сеть строится в два этапа: 1) выбор типа (архитектуры) сети; 2) подбор весов (обучение) сети.

На первом этапе нет необходимости придумывать нейронную сеть "с нуля", так как существуют несколько десятков различных нейросетевых архитектур, причем эффективность многих из них доказана математически.

На втором этапе проводится обучение выбранной сети посредством настройки ее весов. Количество весов может быть велико, поэтому обучение представляет собой сложный и длительный процесс. Для многих архитектур разработаны специальные алгоритмы обучения, наиболее популярный из которых — это алгоритм обратного распространения ошибки.

Для распознавания информации передаваемой по КРЛ предлагается применить нейронную сеть Хэмминга (см. рисунок). Нейронная сеть Хэмминга состоит из входного, скрытого и выходного слоев нейронов. Скрытый и выходной слои содержат по  $K$  нейронов, где  $K$  — число эталонов (команд, передаваемых по КРЛ). Число входов сети Хэмминга будет соответствовать разрядности передаваемого сообщения в КРЛ, а число выходов — числу команд из строго ограниченного перечня. Каждый разряд передаваемого кодового слова представляется в виде нечеткого числа и подается на вход нечеткой нейронной сети Хэмминга. Причем число команд  $\ll 2^n$ , где  $n$  — разрядность передаваемого сообщения по КРЛ. Нейроны скрытого слоя  $n$  синапсами соединены с выходами нейронов входного слоя сети. Выходы нейронов выходного слоя связаны с входами остальных нейронов этого слоя отрицательными обратными (ингибиторными) связями. Единственная поло-



Структура нейронной сети Хэмминга

жительная обратная связь подается с выхода для каждого нейрона выходного слоя на его же вход.

Сеть выбирает эталон с минимальным хэмминговым расстоянием от предъявленного входного вектора путем активации только одного выхода сети (нейрона выходного слоя), соответствующего этому эталону (команде).

Хэммингово расстояние представляет собой пример меры сходства или, вернее, различия, первоначально введенной для бинарных функций в диадном пространстве. Эта мера применима для сравнения любых упорядоченных наборов, принимающих дискретные значения, и, вероятно, является наилучшей из известных мер сходства между цифровыми кодами. Для бинарных последовательностей  $x = (x_1, \dots, x_n)$  и  $x' = (x'_1, \dots, x'_n)$  хэммингово расстояние можно определить по формуле

$$\rho(x, x') = bc\{(\bar{x}_i \wedge x'_i) \vee (x_i \wedge \bar{x}'_i) | i = 1..n\}. \quad (1)$$

Здесь функция  $bc\{\bullet\}$  определяется как число элементов набора  $\{\bullet\}$ , принимающих значение логической "1".

На этапе настройки сети Хэмминга устанавливаются следующие значения весов нейронов скрытого слоя и порога их активационной функции:

$$w_{ik} = \frac{x_i^k}{2}, \theta_k = n/2, \quad (2)$$

где  $x_i^k$  —  $i$ -й компонент  $k$ -го эталона;  $i = 1..n$ ,  $k = 1..K$ .

Коэффициенты отрицательных обратных связей нейронов выходного слоя задают равными некоторому значению из интервала  $0 < \varepsilon < 1/K$ , а коэффициенты положительной обратной связи, равными +1.

Рассмотрим алгоритм функционирования сети Хэмминга.

**Шаг 1.** На нейроны входного слоя подается вектор  $X = \{x_i\}$ ,  $i = 1..n$ . На их выходах формируются следующие значения (верхний индекс указывает номер слоя):

$$y_k^{(1)} = s_k^{(1)} = \sum_{i=1}^n w_{ik} x_i + \theta_j, k = 1 \dots K. \quad (3)$$

В соответствии с этим устанавливаются значения на выходах нейронов выходного слоя:

$$y_k^{(2)} = y_k^{(1)}, k = 1 \dots K. \quad (4)$$

**Шаг 2.** В результате новой  $(t + 1)$ -й итерации определяются новые состояния нейронов выходного слоя:

$$s_k^{(2)}(t+1) = y_k^{(2)}(t) - \varepsilon \sum_{\substack{j=1 \\ j \neq k}}^n y_j^{(2)}(t), k = 1 \dots K, \quad (5)$$

$$y_k^{(2)}(t+1) = f[s_k^{(2)}(t+1)], k = 1 \dots K, \quad (6)$$

где активационная функция  $f$  имеет вид порога, причем "ступенька" должна быть достаточно большой, чтобы возможные значения  $s_k^{(2)}$  не привели к насыщению.

**Шаг 3.** Проверка изменения состояний нейронов выходного слоя за последнюю итерацию и переход к шагу 2 в случае, если наблюдались изменения. Иначе — окончание процедуры.

Роль нейронов входного слоя весьма условна: воспользовавшись один раз на шаге 1 значениями его весовых коэффициентов, сеть больше не обращается к нему, поэтому этот слой может быть вообще исключен из сети (заменен на матрицу весовых коэффициентов).

В заключение можно сделать следующее обобщение: сеть Хэмминга позволяет просто и эффективно решить задачу автоассоциативной памяти — воссоздания образов по неполной и искаженной информации. Предложенная модель оптимизирующей нейронной сети, реализующая восстановление по искаженному (зашумленному) образу ближайший к нему эталонный, позволяет снять ограничения на различия и ограничения длин сравниваемых векторов с сохранением высокой достоверности распознавания. Применение предложенной нейронной сети Хэмминга в КРЛ передачи информации приводит к увеличению вероятности достоверного приема кодового сообщения, а также к уменьшению разрядности кодового сообщения, что, в свою очередь, уменьшает время на его передачу и позволяет своевременно реагировать на возникающие ситуации. Невысокая емкость сетей (число запоминаемых образов) объясняется тем, что сети не просто запоминают образы, а позволяют проводить их обобщение, например, с помощью сети Хэмминга возможна классификация по критерию максимального правдоподобия.

#### Список литературы.

1. **Воздушно-космическая** телеметрия: Пер. с англ. М.: Воениздат, 1968.
2. **Гуткин Л. С., Пестряков В. Б., Типугин В. Н.** Радиоуправление. М.: Сов. радио, 1970.
3. **Круглов В. В., Борисов В. В., Харитонов Е. В.** Нейронные сети: конфигурации, обучение, применение. Смоленск: Изд-во Моск. энерг. ин-та, 1998.

## ПРОГРАММНАЯ ИНЖЕНЕРИЯ

УДК 004.4'233

**Г. В. Тарасов**, научн. сотр., **Д. И. Харитонов**, канд. техн. наук, научн. сотр.,  
Институт автоматизации и процессов управления ДВО РАН, г. Владивосток

### Построение системы визуализации параллельных программ на основе сетей Петри

*Описывается архитектура системы визуализации параллельных программ на основе сетей Петри. Рассмотрены принципы моделирования структуры последовательного процесса в терминах сетей Петри. Предложен новый способ графического отображения взаимодействия параллельных процессов, суть которого заключается в использовании вспомогательной модели для описания поведения среды взаимодействия.*

**Ключевые слова:** теория сетей Петри, параллельное программирование, отладка программ.

#### Введение

Последние годы в научной и научно-популярной литературе все чаще говорят о кризисе параллельного программного обеспечения. Эту тенденцию отмечают многие авторы. В частности, в работе [1] говорится, что технологии и язы-

ки программирования, широко используемые на последовательных компьютерах и суперкомпьютерах с малым числом процессоров, на больших системах могут применяться лишь частично. Действительно, в настоящее время наблюдается существенное отставание методов и средств раз-

работки параллельных программ от возможностей аппаратного обеспечения. Наиболее остро отсутствие гибких средств ощущается на этапе отладки, когда наряду с традиционными последовательными ошибками проявляются специфические параллельные ошибки, вызванные нарушением логики взаимодействия процессов, например, тупиковые состояния, взаимные блокировки, условия гонок и др. Поиск и исправление таких ошибок последовательным методом отладки, перезапуск по контрольным точкам для больших параллельных программ зачастую оказывается невозможным, так как с увеличением числа процессов разработчику все сложнее представлять логику работы программы и удерживать в голове множество состояний всех ее процессов.

Визуализация состояния параллельной программы отчасти упрощает процесс ее отладки. Например, такие популярные пакеты, как XMPi, Jumpshot, AIMS, TotalView, Intel Thread Profiler, предлагают ряд графических форм представления программы: временные диаграммы взаимодействия процессов, матрицы и очереди сообщений, граф вызовов процедур и функций. Несмотря на архитектурные различия и перечень исполняемых функций они разработаны по однотипной схеме, которая включает три стадии: анализ исходного кода и модификация его отладочной информацией; обеспечение сбора и накопления отладочной информации от исполняемой программы; графическое отображение накопленной информации.

Первая стадия позволяет сохранить информацию о текущем состоянии программы или совершаемом действии. Например, модифицированный MPI-вызов приема-передачи информации сохраняет информацию о конверте сообщения, о типе и размере передаваемых данных. Принципом модификации исходного кода является вставка вызовов служебных функций, передающих информацию отладчику, осуществляющему мониторинг исполнения параллельной программы. Включение и выключение режимов мониторинга определяется соответствующими параметрами компиляции или вызовом специальной функции в начале программы.

Вторая стадия подготовки программы к визуализации состоит в сборе и накоплении отладочной информации от исполняемой программы в процессе ее исполнения. Чаще всего для этих целей используется группа файлов, в которую помещается отладочная информация всех исследуемых процессов. По окончании выполнения программы или завершения процедуры профилирования файлы проходят дополнительную обработку для получения целостной истории работы параллельной программы.

На третьей стадии происходит графическое отображение накопленной отладочной информа-

ции. Наиболее популярным способом отображения истории выполнения программы является линейная временная диаграмма. Для более наглядного представления участки диаграммы раскрашивают в разные цвета в соответствии с текущим состоянием процессов, а линиями на диаграммах отображают факты взаимодействия процессов. Помимо временных диаграмм применяются вспомогательные формы представления: иерархии процессов; графы и маршруты передачи сообщений; матрицы очередей сообщений.

Указанные способы, несомненно, повышают наглядность и удобство представления исполнения программ, но при более глубоком рассмотрении и при использовании параллельных программ, включающих большое число процессов со сложной логикой взаимодействия, применение этих средств оказывается малоэффективным. Так, линейная временная диаграмма взаимодействия большого числа процессов будет представлять собой "паутину" линий, в которой тяжело разобраться, какая пара или группа процессов взаимодействовала в данный момент. Использование других форм представления с увеличением числа процессов также оказывается ограниченным. Возникновение перечисленных проблем вызвано отсутствием гибких средств модификации программ и адекватных графических примитивов, отображающих реальное взаимодействие параллельных процессов [2].

В данной статье авторы описывают архитектуру программной системы для визуализации состояния параллельных программ, основанную на использовании формальной модели программы в терминах сетей Петри. Использование сетей Петри позволяет математически точно зафиксировать и проследить ожидаемое поведение параллельной программы и отобразить взаимодействие ее процессов. В первой части статьи описывается способ представления структуры и взаимодействия последовательных процессов в терминах сетей Петри. Во второй части статьи рассматривается архитектура программного средства визуализации параллельных программ на основе сетей Петри. В заключении дается короткий анализ возможностей сетей Петри для визуализации, предлагаются дальнейшие направления исследования. Развитие этого подхода направлено на совершенствование инструментальных средств отладки сложных параллельных программ.

### **Представление программ в сетях Петри**

Традиционно вычисления в параллельном программировании описываются в виде последовательных взаимодействующих процессов. В каждом процессе можно выделить две группы дейст-

вий: выполнение локальных (последовательных) вычислений и взаимодействие с другими процессами. Фундаментом последовательных вычислений являются объектно-ориентированные и структурные методы программирования, включающие такие понятия, как классы, методы, функции, а также условия, циклы и т. п. Взаимодействие процессов реализуется с использованием сторонних библиотек, организующих среду взаимодействия между процессами. Так, например, в параллельных вычислениях стандартом является MPI, в котором определены наборы функций для организации передачи сообщений между двумя или более процессами.

Модель последовательного процесса удобно и наглядно записывается в терминах сетей Петри. В основе построения модели лежит отображение потока команд процесса на структурные элементы сетей Петри. Принципы отображения приведены в следующем списке:

- *сеть Петри* — модель последовательного процесса или функции, представленная в виде графа и описывающая поток управления заданного процесса (функции); граф имеет четко выраженную последовательную структуру с единственной начальной вершиной и одной или более конечными вершинами; граф представляет собой последовательную связь мест и переходов, каждое из которых описывает следующее;
- *места сети Петри* — моделируют состояния последовательного процесса, в котором он может существовать; будем считать, что состояние процесса есть образ его памяти плюс дополнительная информация, определяющая его существование в операционной системе и среде взаимодействия;
- *переходы сети Петри* — моделируют выполнение действий последовательным процессом (выражения, операторы, блоки выражений и т. п.); будем считать, что переход может отражать выполнение как одного, так и нескольких последовательных действий, если промежуточные действия являются малозначимыми и "невидимыми" с точки зрения визуализации всей программы в целом;
- *структура переход—место—переход* — иногда имеет специальное значение и моделирует выполнение вызова функции; первый переход моделирует вход в тело функции и начало ее выполнения, место отражает состояние ожидания завершения функции, второй переход моделирует выход из функции и завершение ее выполнения;
- *сеть Петри, начинающаяся одним переходом и заканчивающаяся одним или несколькими переходами*, — может моделировать тело функции; внутренняя структура данной сети представля-

ет собой модель потока управления функции, построенную по правилам, определенным для процесса;

- *токен* — проекция состояния процесса на модель в сетях Петри; перемещение токена по графу определяется классическими правилами срабатывания переходов в базовой теории сетей Петри;
- *токен возврата* — является вспомогательным элементом и используется при моделировании ожидания завершения функции.

Графическая интерпретация всех перечисленных понятий представлена на рис. 1. Места и переходы отображаются в виде окружностей и прямоугольников, соединенных направленными дугами. Токены отображаются в виде небольших сплошных кружков внутри мест. Токены возврата отображаются небольшими кружками с крестиком внутри. Последовательная цепочка операторов, выражений и других конструкций программы отображается последовательностью переходов (рис. 1, а). На рис. 1, б—д показаны шаблоны стандартных алгоритмических конструкций: условие, выбор, цикл-после и цикл-до. Два перехода и место, моделирующие вызов функции, отображаются выделенным цветом (рис. 1, е, верхняя сеть). Далее в примерах, когда необходимо точно показать границы этой структуры, будем дополнительно выделять эти элементы областью, ограниченной штриховой линией. Тело функции отображается отдельной сетью Петри (рис. 1, е, нижняя сеть) и связывается с вызовом функции по правилам параллельной композиции.

Операция параллельной композиции определяет способ синхронизации сетей по переходам и позволяет описывать ситуации, когда срабатывание перехода(-ов) в одной сети приводит к сраба-

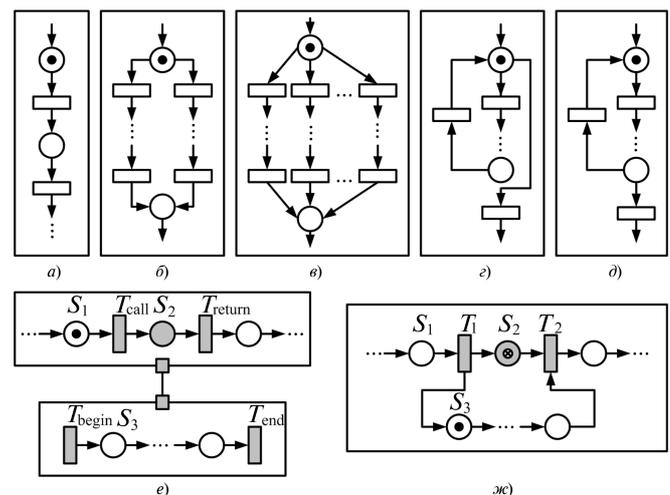


Рис. 1. Графическое представление последовательности выражений (а), условия (б), выбора (с), цикла-до (д), вызова функции (е) и тела функции (ж)

тыванию перехода(-ов) в другой сети. Формальное определение операции приведено, например, в работе [3] и строится на понятии точки доступа. Точка доступа определяет множество переходов, срабатывание которых в сети является видимым извне событием. Для случая моделирования вызова функции и передачи управления в тело функции операция композиции определяется четырьмя переходами: в вызывающей сети — переход вызова функции и переход ожидания ее завершения; в вызываемой сети (тело функции) — переход начала исполнения функции и переход завершения исполнения. Пример описания вызова функции и результат выполнения операции параллельной композиции приведены на рис. 1, *е, ж*. Переходы  $T_{call}$  и  $T_{begin}$  сливаются в переход  $T_1$ , а переходы  $T_{return}$  и  $T_{end}$  — в переход  $T_2$ . В процессе моделирования исполнения вызова функции токен из места  $S_1$  по правилам срабатывания переходов, принимая во внимание участие перехода в операции композиции, преобразуется в два токена в местах  $S_2$  и  $S_3$ . Отметим, что место  $S_2$ , согласно описанной выше конструкции вызова функции, будет содержать токен возврата, обозначающий выполнение функции. В процессе моделирования наличие токенов возврата в общей модели последовательного процесса позволяет получить информацию о текущем стеке вызванных функций.

Приведенный выше пример позволяет описать вызов произвольного числа функций, заданных в программе. Решение о включении или исключении функции из модели определяется из условий требуемого уровня детализации. В случае, если необходимая функция является внешней по отношению к программе, например, разработчик использует некоторую стороннюю библиотеку вместо описания реального тела функции, можно представить некоторую его модель, опираясь на общие принципы моделирования в сетях Петри. Проиллюстрируем этот подход на примере реальной задачи.

Одной из классических задач параллельного программирования является задача разделения множеств двумя процессами. Ее спецификация и анализ неоднократно обсуждались в литературе [4, 5]. Суть задачи состоит в следующем. Дано два процесса, в которых задано по некоторому произвольному числовому множеству. Необходимо организовать схему взаимодействия процессов и построить алгоритм, в результате выполнения которого в одном процессе собираются все минимальные

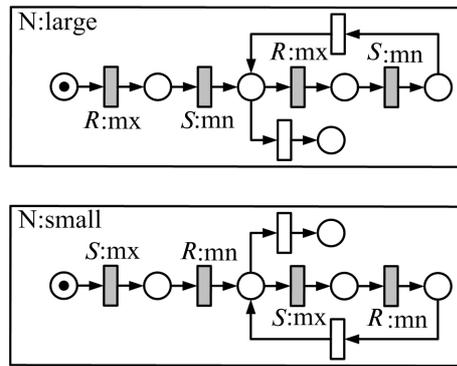


Рис. 2. Модель программы разделения множеств

элементы (*small*-процесс), в другом, соответственно, все максимальные (*large*-процесс). При этом мощности множеств не изменяются. Элементарная модель алгоритма в сетях Петри, решающая эту задачу, приведена на рис. 2.

В модели выделены четыре основных перехода, отображающие моменты взаимодействия процессов. Пара переходов образует пару элементарных действий — прием—передача сообщения. За каждый шаг взаимодействия процессы обмениваются локальными максимальным и минимальными элементами. Обмен продолжается до тех пор, пока максимальный элемент *small*-процесса превышает минимальный элемент *large*-процесса. Для реализации данного алгоритма использовали MPI и стандартный набор функций обмена MPI\_Send и MPI\_Recv. Для краткости описания в данной задаче вызов функций MPI поместили в функции-обертки (*wrapper-function*), которые принимают только необходимые параметры из основной программы. Остальные параметры, определенные в MPI (тип данных, объем данных, коммутатор и тэг сообщения), задали стандартными по умолчанию.

На рис. 3 отображен фрагмент реализации алгоритма и модели параллельной программы. Первый шаг, который должны сделать оба процесса, — это определить свои максимальный и минимальный элементы, которыми они далее должны обменять-

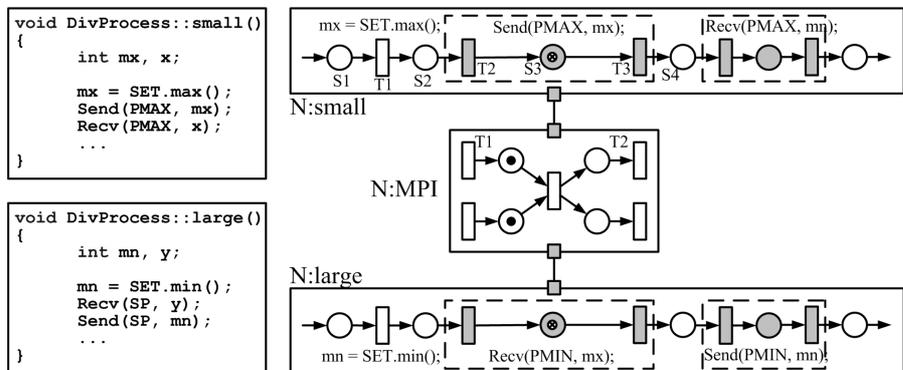


Рис. 3. Визуализация взаимодействия процессов

ся. В модели программы выделены вызовы функций обмена сообщениями, реализация которых представлена на модели в центре рисунка (MPI). Опуская для краткости тонкости реализации этих функций [6], описываем их выполнение в виде небольшой сети, в которой центральный переход сработает только тогда, когда будут вызваны соответствующие пары MPI\_Send и MPI\_Recv.

Таким образом, модель параллельной программы представляет собой композиционную сеть Петри, составленную из подсетей-функций, заданных в программе, и подсетей, моделирующих логику взаимодействия процессов. Токены в модели представляют собой процессы, их общее распределение на сети отображает текущее состояние параллельной программы. Рассмотренный подход можно использовать не только для моделирования взаимодействия процессов с помощью передачи сообщений через MPI. Соответствующие модели могут быть построены, например, для подсистемы синхронизации потоков (IPC), библиотеки передачи данных через сокеты и других средств взаимодействия процессов.

### Прототип системы визуализации состояния параллельной программы

Общая технология визуализации состояния параллельной программы была сформулирована в начале статьи: модификация исходного кода, сбор и анализ данных, отображение данных. Рассмотрим реализацию этих шагов применительно к модели программы в терминах сетей Петри.

В системе разработан следующий набор отладочных функций, обеспечивающих связь между исполняемым представлением и моделью параллельной программы:

- *функция инициализации* — вызывается в начале каждого процесса параллельной программы и выполняет настройку общих параметров: инициализация временного буфера для хранения отладочной информации, инициализация канала связи для передачи во внешнюю среду содержимого буфера и т. д.;
- *функция состояния* — вызывается из программы в такие моменты времени, когда необходимо отразить факт достижения процессом некоторого состояния; параметром функции служит полное название места в модели, которое данная функция записывает во временный буфер; дополнительными параметрами данной функции могут быть значения

```
// функция инициализации
int Initialize(int* argc, char*** argv);
// функция состояния
int Checkpoint(const char* S, ...);
// функции выполнения перехода
int BeginFire(const char* T, ...);
int EndFire(const char* T, ...);
// функции выполнения вызова
int BeginCall(const char* Tcall,
              const char* S,
              const char* Tbegin, ...);
int EndCall(const char* S,
            const char* Treturn,
            const char* Tend, ...);
// функции завершения
int Finalize();
```

```
void DivProcess::small()
{
    int mx, x;

    Checkpoint("small::S1");

    BeginFire("small::T1");
    mx = SET.max();
    EndFire("small::T1");

    Checkpoint("small::S2");

    BeginCall("small::T2",
              "small::S3", "MPI::T1");
    Send(PMAX, mx);
    EndCall("small::S3",
            "small::T3", "MPI::T2");

    Checkpoint("small::S4");

    Recv(PMAX, x);
    ...
}
```

Рис. 4. Отладочные функции и пример модификации программы

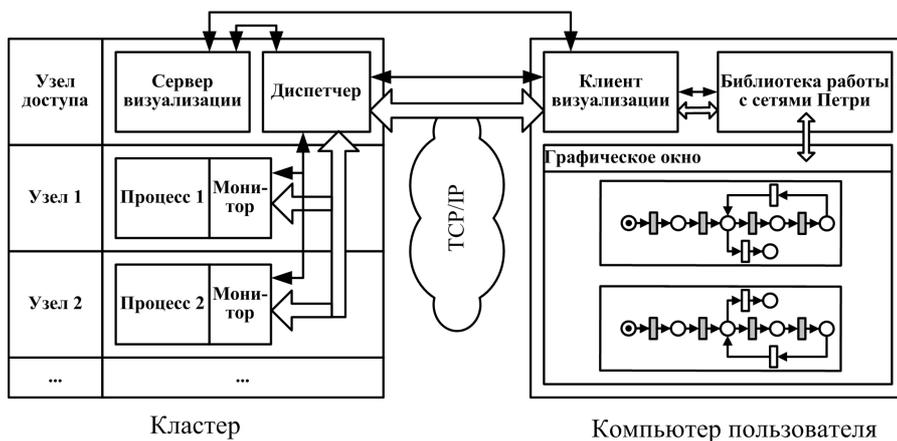


Рис. 5. Архитектура системы визуализации

вершением отображения информации на модели или командой пользователя на продолжение выполнения. Интерактивный режим работы существенно расширяет возможности системы визуализации параллельной программы, добавляя некоторые функции отладчиков. В этом режиме модель определяет спецификацию параллельной программы и определяет правильный ход ее исполнения.

Архитектура системы визуализации исполнения параллельных программ с использованием сетей Петри показана на рис. 5. В системе следует выделить следующий набор основных структурных компонентов.

*Монитор* — библиотека, реализующая функции обмена отладочной информацией, описанные в предыдущем разделе. При сборке отладочной версии пользовательской программы данная библиотека статически линкуется с программой и исполняется в ее адресном пространстве.

*Диспетчер* — самостоятельное приложение, выполняющее сбор информации о работе программы со всех процессов. При запуске параллельной программы каждому процессу передается информация о том, как установить связь с диспетчером. Эту работу выполняет функция инициализации.

*Сервер визуализации* — это сервисное приложение, работающее в фоновом режиме. В его задачи входит прием запросов от клиентов визуализации на запуск параллельной программы и инициализация диспетчера для организации двухсторонней связи между клиентским приложением на компьютере пользователя и процессами параллельной программы, которую в данный момент собираемся визуализировать. Таким образом, в системе на одну параллельную программу отводится ровно один диспетчер.

*Клиент визуализации* — это графическое приложение, поддерживающее протоколы связи с сер-

вером визуализации и диспетчером для двухсторонней связи с процессами параллельной программы. В случае неинтерактивного способа визуализации параллельных программ данное приложение сохраняет всю полученную от диспетчера информацию в файлах, откуда в дальнейшем они могут быть считаны и обработаны библиотекой работы с сетями Петри. В случае интерактивного способа работы клиент может напрямую обращаться к библиотеке для выполнения необходимых задач рисования модели программы, анализа срабаты-

вания переходов и перемещения токенов. Все протоколы связи, реализуемые в рамках данного инструментального средства, ориентированы на передачу текстовой информации на базе транспортной сети TCP/IP. На рис. 5 тонкими стрелками отображается взаимодействие компонентов на уровне команд и запросов, толстыми стрелками отображается взаимодействие компонентов на уровне передачи отладочной информации в виде потоков имен мест и переходов.

## Заключение

В работе рассмотрены принципы построения системы визуализации параллельных программ с использованием теории сетей Петри. Описаны механизм отображения исходного кода параллельной программы в модельное представление и архитектура системы визуализации параллельных программ.

Использование сетей Петри в качестве графического языка отображения предоставляет дополнительные преимущества по сравнению с существующими средствами и позволяет более формально подойти к вопросу отображения состояния параллельной программы. Следует выделить три наиболее существенных факта.

Во-первых, анализ работы параллельной программы зачастую требует различных ракурсов ее представления с произвольной степенью детализации. Универсальность базовых понятий сетей Петри позволяет построить как максимально детальное, так и максимально общее представление параллельной программы.

Оптимальный вариант модели, в которую разработчик может включить только заданные участки программы, позволяет более эффективно сгенерировать отладочную информацию и минимизировать степень ее влияния на ход выполнения программы.

Во-вторых, информация о состоянии среды взаимодействия процессов зачастую является бо-

лее важной с точки зрения поиска ошибок в параллельных программах, чем информация о самих процессах, и может оказать существенную помощь на этапе отладки. Вовлечение в процесс визуализации в сетях Петри среды взаимодействия достигается применением операции параллельной композиции к моделям процессов и модели среды взаимодействия. Последняя может быть построена вручную и отражать заданные аспекты передачи информации от одного процесса к другому. В результате разработчик получает полную картину как о внутреннем состоянии процессов программы, так и о событиях, происходящих в среде взаимодействия.

В-третьих, нахождение причин ошибки и ее исправление зачастую зависит от того, какая была предыстория выполнения процессов до момента проявления ошибки. В сетях Петри история работы модели выражается последовательностью сработавших переходов, по которой можно восстановить порядок наступления состояний. Анализируя каждое из состояний, предшествовавших ошибке, можно принять решение о возможных причинах некорректного исполнения программы.

Несмотря на очевидные преимущества сети Петри являются незнакомым для рядового пользователя аппаратом, и их практическое применение для визуализации и отладки параллельных

программ оказывается непривычным и поэтому неудобным. Этот недостаток чисто практического характера, он может быть исправлен реализацией удобного пользовательского интерфейса, в частности, предоставляющего уже известные формы отображения состояния параллельной программы.

*Работа выполнена при финансовой поддержке грантов ДВО (раздел 1) и программы Президиума РАН № 14 (раздел 2).*

#### Список литературы

1. **Воеводин В. В.** Математические проблемы параллельных вычислений // Труды Всероссийской научной конференции "Научный сервис в сети Интернет: технологии распределенных вычислений". 2005. С. 3—8.
2. **Авербух В. Л., Байдалин А. Ю., Исмаилов А. Ю., Казанцев А. Ю.** Состояние дел в визуализации программного обеспечения параллельных вычислений // 15-я Международная конференция по компьютерной графике и ее приложениям (труды конференции). 2005. С. 179—186.
3. **Анисимов Н. А., Голенков Е. А., Харитонов Д. И.** Композиционный подход к разработке параллельных и распределенных систем на основе сетей Петри // Программирование. 2001. № 6. С. 30—43.
4. **Малышкин В. Э.** Основы параллельных вычислений: Учебное пособие. Ч. 1. Новосибирск: Изд-во НГТУ, 1998. 60 с.
5. **Карпов Ю. Г.** Анализ корректности параллельной программы разделения множеств // Программирование. 1996. № 6. С. 27—33.
6. **Message Passing Interface Forum.** MPI: A Message-Passing Interface Standard. (URL: <http://www.mpi-forum.org/>).

УДК 004.415.53

**Е. А. Рыжков**, аспирант, ассистент кафедры АТМ,  
Тульский государственный университет,  
**А. Н. Карпов**, младший научный сотрудник,  
Объединенный институт высоких температур РАН,

## Подходы к верификации и тестированию 64-битных приложений

*В результате появления на рынке персональных компьютеров 64-битных процессоров перед разработчиками программ возникает задача переноса старых 32-битных приложений на новую платформу. После переноса кода приложение может вести себя некорректно. Рассмотрены вопросы, связанные с верификацией и тестированием программного обеспечения. Обозначены сложности, с которыми может столкнуться разработчик 64-битных приложений, и пути их преодоления.*

**Ключевые слова:** статический анализ, анализ исходного кода, верификация программ, тестирование программ.

### Введение

Появление 64-битных процессоров — это очередной этап в эволюции вычислительной техники. Но получить преимущества от использования нового аппаратного обеспечения можно только после перекомпиляции программ для поддержки новых наборов инструкций и регистров. Изменение соотношений между типами данных, проис-

ходящее при смене архитектуры процессора, может привести к новым ошибкам в существующих программах. Проблемы при переносе кода обнаруживаются в приложениях, разработанных с использованием низкоуровневых языков программирования типа Си или Си++. В языках с четко структурированной системой типов (например, языки .NET Framework), как правило, таких проблем не возникает.

В чем же проблема именно с низкоуровневыми языками? Дело в том, что даже все высокоуровневые конструкции и библиотеки Си++ в конечном итоге реализованы с использованием низкоуровневых типов данных, таких как указатель, машинное слово и т. п. Поскольку при изменении архитектуры эти типы данных меняются, то и поведение программ также может измениться.

Поставим задачу следующим образом. Необходимо убедиться в том, что 64-битное приложение после перекомпиляции имеет такое же поведение, что и 32-битное (за исключением очевидных архитектурных изменений). Процесс проверки работоспособности 64-битной версии программы и будем называть верификацией.

В первой части статьи описаны основные методы тестирования и верификации приложений. Те, кто хорошо знаком с этими методами, могут пропустить эту часть и сразу же перейти к следующей части, в которой рассматриваются особенности применения методик тестирования для 64-битных систем.

### Подходы к тестированию приложений

Существуют различные подходы к обеспечению корректности кода приложений, как поддающиеся автоматизации, так и нет. Среди неподдающихся автоматизации методов выделяют просмотр кода вручную, тестирование методом белого ящика и ручное тестирование. К автоматизируемым методам относятся статические анализаторы кода и тестирование методом черного ящика. Рассмотрим эти методы подробнее.

**Просмотр кода.** Самым старым, проверенным и надежным подходом к поиску дефектов является совместный просмотр кода (англ. *code review*) [1]. Эта методика основана на совместном чтении кода с выполнением ряда правил и рекомендаций [2]. К сожалению, эта практика неприменима для крупномасштабной проверки современных программных систем в силу их большого объема. Хотя этот способ дает наилучшие результаты, он не всегда используется в условиях современных жизненных циклов разработки программного обеспечения, где немаловажным моментом является срок разработки и время выхода продукта на рынок. Поэтому просмотр кода чаще всего сводится к нечастым встречам, целью которых ставится обучение новых и менее опытных сотрудников написанию качественного кода, нежели чем для проверки работоспособности ряда модулей. Это очень хороший способ повышения квалификации программистов, но его нельзя рассматривать как полноценное средство контроля качества разрабатываемой программы.

**Статические анализаторы кода.** На помощь разработчикам, которые осознают необходимость регулярного просмотра кода, но не имеют достаточного количества времени, приходят средства статического анализа кода [3]. Их основной задачей является сокращение объема кода, требующего внимания человека, и тем самым сокращение времени его просмотра. К статическим анализаторам кода относится достаточно большой класс программ, реализованных для различных языков программирования и имеющих разнообразный набор функций от простейшего контроля выравнивания кода до сложного анализа потенциально опасных мест. Систематизированное использование статических анализаторов позволяет существенно повысить качество кода и найти многие ошибки. У подхода, основанного на статическом анализе, много поклонников, и ему посвящено много интересных работ [4, 5]. Преимущество данного подхода заключается в том, что он применим вне зависимости от сложности и размера разрабатываемого программного решения.

**Метод белого ящика.** Под тестированием методом белого ящика [6] принято понимать выполнение максимально доступного числа различных веток кода с использованием отладчика или иных средств. Чем большее покрытие кода было достигнуто, тем более полно выполнено тестирование. Под тестированием по методу белого ящика также иногда понимают простую отладку приложения в целях поиска известной ошибки. Полноценное тестирование методом белого ящика всего кода программ уже давно стало невозможным в силу огромного объема кода современных программ. Сейчас тестирование по методу белого ящика удобно применять на этапе, когда ошибка найдена и необходимо понять причину ее возникновения. У тестирования методом белого ящика существуют оппоненты, отрицающие полезность отладки программ в реальном времени. Основной мотив заключается в том, что возможность наблюдать ход работы программы и при этом вносить изменения в ее состояние порождает недопустимый подход в программировании, основанный на большом числе исправлений кода методом проб и ошибок. Мы не будем касаться данных споров, но заметим, что тестирование по методу белого ящика в любом случае — очень дорогой способ повышения качества больших и сложных программных систем.

**Метод черного ящика.** Намного лучше себя зарекомендовал метод черного ящика [7]. Сюда же можно отнести юнит-тестирование (англ. *unit test*) [8]. Основная идея метода заключается в написании набора тестов для отдельных модулей и функций, проверяющего все основные режимы их работы. Ряд источников относят юнит-тестирование к методу белого ящика, поскольку оно осно-

выдается на знании устройства программы. Однако функции и модули следует рассматривать как черные ящики, поскольку юнит-тесты не должны учитывать внутреннее устройство функции. Обоснование этому может служить методология разработки, когда тесты разрабатываются до начала написания самих функций, что способствует повышению контроля их функциональности с точки зрения спецификации.

Методу юнит-тестирования посвящено много литературы [9, 10]. Юнит-тестирование хорошо зарекомендовало себя при разработке как простых, так и сложных проектов. Одно из преимуществ юнит-тестирования в том, что можно легко проверить корректность вносимых в программу исправлений прямо в ходе разработки. Стараются делать так, чтобы все тесты проходили в течение нескольких минут, что позволяет разработчику, который внес изменения в код, сразу заметить ошибку и исправить ее. Если прогон всех тестов невозможен, то обычно длительные тесты выносятся отдельно и запускают, например, ночью. Это также способствует оперативному обнаружению ошибок, по крайней мере, на следующее утро.

**Ручное тестирование.** Это, пожалуй, завершающий этап любой разработки, но его не следует рассматривать как хорошую и надежную методику. Ручное тестирование обязательно должно существовать, так как невозможно обнаружить все ошибки в автоматическом режиме или просмотром кода. Однако если программа имеет низкое качество и большое число внутренних дефектов, ее тестирование и исправление могут затянуться на очень продолжительное время, и все равно при этом нельзя обеспечить надлежащее качество программы. Единственный метод получения качественной программы — это качественный код. Поэтому мы также не будем рассматривать ручное тестирование как полноценную методику при разработке больших проектов.

**Выводы по методам тестирования.** Итак, что же заслуживает наибольшего внимания при разработке крупных программных систем? Это статический анализ и юнит-тестирование. Данные подходы способны существенно повысить качество и надежность программного кода, и им следует уделить наибольшее внимание, хотя конечно не стоит забывать и про другие.

### **Особенности тестирования и верификации 64-битных приложений**

Перейдем к вопросу тестирования 64-битных программ, так как применение выбранных нами методик сталкивается с несколькими неприятными затруднениями.

### **Использование статических анализаторов кода.**

Как это ни странно, несмотря на все свои огромные возможности, длительный период разработки и практику использования статические анализаторы оказались плохо подготовлены к поиску ошибок в 64-битных программах. Рассмотрим ситуацию на примере анализа Си++ кода как область, где статические анализаторы нашли наибольшее применение. Многие статические анализаторы поддерживают ряд правил, связанных с поиском кода, имеющего некорректное поведение при переносе его на 64-битные системы. Но реализуют они это весьма разрозненными методами и весьма неполно. Особенно это хорошо проявилось после начала массовой разработки приложений под 64-битную версию операционной системы Windows в среде Microsoft Visual C++ 2005.

Объяснением этого может служить то, что большинство проверок основано на достаточно старых материалах по исследованию проблем переноса программ на 64-битные системы с точки зрения языка Си. В результате ряд конструкций, появившихся в языке Си++, был обделен вниманием с точки зрения контроля переносимости и не нашел своего отражения в анализаторах. Не учтен и ряд других изменений, таких как, например, существенно возросший объем оперативной памяти и использование в разных компиляторах различных моделей данных (LP64, LLP64, ILP64 [11]).

Для наглядности рассмотрим несколько примеров.

```
double *DoubleArray;
unsigned Index = 0;
while (...)
    DoubleArray[Index++] = 1.0f;
```

Получить диагностическое предупреждение на подобный код не удастся даже такими мощными анализаторами, как Parasoft C++test (<http://www.parasoft.com>) и Gimpel Software PC-Lint (<http://gimpel-online.com>). Это не удивительно. Приведенный код не вызывает никаких подозрений у рядового разработчика, привыкшего к практике использовать в качестве индексов массивов переменные типа int или unsigned. К сожалению, приведенный код на 64-битной системе будет неработоспособен, если объем обрабатываемого массива DoubleArray превысит размер в 4 Гбайт элементов. В этом случае произойдет переполнение переменной Index, и результат работы программы будет некорректен. Корректным вариантом будет использование типа size\_t при программировании под Windows x64 (модель данных LLP64) или size\_t/unsigned long при программировании под Linux (модель данных LP64).

Причина, по которой статические анализаторы не могут диагностировать подобный код, пожалуй, скрывается в том, что когда исследовались вопросы переноса под 64-битные системы, то

вряд ли кто-то представлял себе массивы более чем из 4 млрд элементов. А 4 млрд элементов типа `double` — это  $4 \cdot 8 = 32$  Гбайт памяти для одного массива. Огромный объем, тем более, если учесть что это 1993—1995 гг. Именно на это время приходится большинство публикаций и обсуждений, посвященных использованию 64-битных систем.

В результате на возможную некорректную индексацию при использовании типа `int` никто не обратил внимания, а в дальнейшем вопросы переноса исследовались достаточно редко. И практически ни один статический анализатор не выдаст предупреждение на приведенный код. Исключением может служить, пожалуй, только анализатор Viva64 (<http://www.viva64.com>). Он был разработан, для того чтобы закрыть пробелы в диагностике 64-битного Си++ кода другими подобными системами, и основан на вновь проведенных исследованиях. Но у него есть существенный недостаток, заключающийся в том, что это не анализатор общего назначения. Он специализируется только на поиске ошибок, возникающих при переносе кода на 64-битные системы Windows, и поэтому должен быть использован только в сочетании с другими средствами поиска ошибок в программах, чтобы обеспечить надлежащее качество кода.

Рассмотрим другой пример.

```
char *p;
long g = (long)p;
```

С помощью этого простого примера можно проверить, какие модели данных умеет понимать используемый вами статический анализатор. Проблема в том, что большинство из них рассчитаны только на модель данных LP64. Это вновь вызвано историей развития 64-битных систем. Именно модель данных LP64 на начальных этапах развития 64-битных систем получила наибольшую популярность и сейчас широко используется в Unix-мире. В этой модели данных тип `long` имеет размер 8 байт, а значит, такой код полностью корректен. Но в 64-битных системах Windows реализована модель данных LLP64, где размер `long` остался 4-байтовым, и приведенный код будет некорректен. В Windows в таких случаях принято использовать тип `LONG_PTR` или `ptrdiff_t`.

К счастью, приведенный код будет диагностироваться как опасный даже самим компилятором Microsoft Visual C++ 2005. Но всегда следует помнить о подобных подводных камнях при использовании статических анализаторов.

Получилась интересная ситуация. Вопрос переноса программ на 64-битные системы был подробно обсужден, были реализованы различные методики и правила проверки в статических анализаторах, после чего интерес к этой тематике угас. Прошло много лет, многое изменилось, но правила, по которым осуществляется анализ, остаются без из-

менений и модификаций. Чем это вызвано, объяснить сложно. Возможно, просто разработчики не замечают изменений, считая, что вопрос тестирования и проверки 64-битных приложений давно решен. Однако то, что было актуально 10 лет назад, сейчас, возможно, таковым не является, но зато появилось много нового. Используя средства статического анализа, убедитесь, что они совместимы с используемой вами 64-битной моделью данных. Если анализатор не удовлетворяет необходимым условиям, не поленитесь поискать другой или восполнить пробел, используя узконаправленный анализатор. Усилия, потраченные на это, с лихвой окупятся повышением надежности программы, уменьшением сроков отладки и тестирования.

**Использование метода черного ящика.** Теперь поговорим о юнит-тестах. С ними на 64-битных системах разработчиков также ожидает ряд неприятных моментов. Стремясь сократить время выполнения тестов, при их разработке стараются использовать небольшой объем вычислений и объем обрабатываемых данных. Например, при разработке теста на функцию поиска элемента в массиве не имеет большого значения, будет она обрабатывать 100 элементов или 10 000 000. Сто элементов будет достаточно, а вот по сравнению с обработкой 10 000 000 элементов скорость выполнения теста может быть существенно выше. Но если вы хотите разработать полноценные тесты, чтобы проверить эту функцию на 64-битной системе, вам потребуется обработать более 4 млрд элементов! Вам кажется, что если функция работает на 100 элементах, она будет работать и на миллиардах? Нет. Если не верите, то для демонстрации попробуйте приведенный пример на 64-битной системе.

```
boot FooFind(char *Array, char Value,
             size_t Size)
{
    for (unsigned i = 0; i != Size, ++i)
        if (i % 5 == 0 && Array[i] == Value)
            return true;
    return false;
}
#ifdef _WIN64
    const size_t BufSize = 5368709120ui64;
#else
    const size_t BufSize = 5242880;
#endif
int _tmain(int, _TCHAR *) {
    char *Array =
        (char *)calloc(BufSize, sizeof(char));
    if (Array == NULL)
        std::cout << "Error allocate memory";
    if (FooFind (Array, 33, BufSize))
        std::cout << "Find";
    free (Array);
}
```

Как видно из примера, если ваша программа на 64-битной системе начнет обрабатывать больший объем данных, то не стоит рассчитывать на старые наборы юнит-тестов. Следует их обяза-

тельно расширить с учетом обработки больших объемов данных.

Но, к сожалению, написать новые тесты мало. Здесь мы сталкиваемся с проблемой скорости выполнения модифицированного набора тестов, охватывающего обработку больших объемов данных. Первым следствием станет то, что такие тесты нельзя будет добавить в набор тестов, запускаемых программистом в ходе разработки. С внесением их в ночные тесты тоже могут возникнуть сложности. Суммарное время выполнения всех тестов может вырасти на порядок или два, а то и более. В результате тест может не уложиться даже в 24 ч. Следует помнить об этом и подойти к доработке тестов для 64-битной версии программы со всей серьезностью.

Выходом из создавшейся ситуации может стать разбиение всех тестов на несколько групп, выполняемых параллельно на нескольких компьютерах. Также можно использовать многопроцессорные системы. Конечно, это несколько усложнит систему тестирования и потребует дополнительных аппаратных ресурсов, но это будет являться самым правильным и, в итоге, простым шагом к решению задачи построения системы юнит-тестирования.

Естественно, для этого потребуется воспользоваться системой автоматизированного тестирования, которая позволит организовать запуск тестов на нескольких машинах. Примером может служить система автоматизации тестирования Windows-приложений AutomatedQA TestComplete (<http://www.automatedqa.com>). С ее помощью можно выполнять распределенное тестирование приложений на нескольких рабочих станциях, осуществлять синхронизацию и сбор результатов [12].

**Использование метода белого ящика.** В конце хочется еще раз вернуться к вопросу тестирования методом белого ящика, который мы посчитали неприемлемым для больших систем. Нужно добавить, что при отладке 64-битных приложений, обрабатывающих большие массивы, этот способ становится еще более неприменимым. Отладка таких приложений может занимать намного больше времени или быть затруднительной на машинах разработчика. Поэтому стоит заранее обдумать возможность использования систем логирования для отладки приложений или использовать иные методы, например, удаленную отладку в случае использования для разработки нескольких компьютеров.

### Заключение

Подводя итоги, хочется сказать, что не стоит полагаться на единственную отдельную методику. Качественное приложение может быть разработано только с применением нескольких из рассмот-

ренных подходов к тестированию и верификации. Причем задуматься об этих методиках необходимо до начала переноса кода на новую архитектуру, чтобы сразу же можно было контролировать качество приложения.

Резюмируя проблемы разработки и тестирования 64-битных систем, хочется еще раз напомнить ключевые моменты:

- будьте готовы к неожиданностям при разработке и тестировании 64-битных приложений;
- будьте готовы к тому, что отладка 64-битных приложений методом белого ящика может стать невозможной или крайне затруднительной, если обрабатываются большие массивы данных;
- внимательно ознакомьтесь с возможностями вашего статического анализатора и, если он не удовлетворяет всем необходимым требованиям, не поленитесь найти другой или использовать дополнительный специализированный статический анализатор;
- не стоит доверять старым наборам юнит-тестов. Обязательно просмотрите их и дополните новыми тестами, учитывающими особенности 64-битных систем;
- помните о существенном замедлении наборов юнит-тестов и заранее позаботьтесь о новых компьютерах для их запуска;
- используйте систему для автоматизации тестирования, поддерживающую распределенный запуск приложений для обеспечения быстрой проверки приложений;
- наилучший результат можно достичь, только используя сочетание различных методик.

### Список литературы

1. **Wikipedia.** Code review. [http://en.wikipedia.org/wiki/Code\\_review](http://en.wikipedia.org/wiki/Code_review)
2. **McConnell Steve.** Code Complete, 2nd Edition // Microsoft Press, Paperback, 2nd edition, Published June 2004. 914 p.
3. **Wikipedia.** Static code analysis. [http://en.wikipedia.org/wiki/Static\\_code\\_analysis](http://en.wikipedia.org/wiki/Static_code_analysis)
4. **Meyers S., Klaus M.** A First Look at C++ Program Analyzers, 1997, [http://www.aristeia.com/ddjpaperl\\_frames.html](http://www.aristeia.com/ddjpaperl_frames.html)
5. **Schilling W., Alam M.** Integrate Static Analysis Into a Software Development Process. 01.2006. [http://www.ddj.com/dept/debug/193500967?cid=RSSfeed\\_DDJ\\_debugging](http://www.ddj.com/dept/debug/193500967?cid=RSSfeed_DDJ_debugging)
6. **Wikipedia.** White box testing. [http://en.wikipedia.org/wiki/White\\_box\\_testing](http://en.wikipedia.org/wiki/White_box_testing)
7. **Wikipedia.** Black box testing. [http://en.wikipedia.org/wiki/Black\\_box\\_testing](http://en.wikipedia.org/wiki/Black_box_testing)
8. **Wikipedia.** Unit testing. [http://en.wikipedia.org/wiki/Unit\\_test](http://en.wikipedia.org/wiki/Unit_test)
9. **Gehrland J.** 10 Ways to Make Your Code More Testable. July, 2004. <http://www.theserverside.net/it/articles/showarticle.tss?id=10WaysTestableCode>
10. **Hamill P.** Unit Test Frameworks. November 2004. 212 p.
11. **Josey A.** Data Size Neutrality and 64-bit Support. <http://www.usenix.org/publications/login/standards/10.data.html>
12. **AutomatedQA.** TestComplete — Distributed Testing Support. [http://www.automatedqa.com/products/testcomplete/tc\\_distributed\\_testing.asp](http://www.automatedqa.com/products/testcomplete/tc_distributed_testing.asp)

УДК 004.7

**В. В. Бакаев**, канд. техн. наук,  
начальник сектора логистической поддержки  
ОАО "КнААПО", г. Комсомольск-на-Амуре

## Система информационно-технической поддержки заказчика

*Надежная и оперативная система связи между организациями, владеющими изделием в течение его жизненного цикла, является залогом обеспечения сокращения затрат на владение. Одной из проблемных задач является создание системы информационно-технической поддержки заказчика. Современные информационные технологии позволяют организовать этот процесс в режиме 24 часов в сутки, 7 дней в неделю, 365 дней в году. В настоящей статье представлено краткое описание системы, обеспечивающей решение организационных и режимных вопросов в процессе информационно-технического обеспечения заказчиков авиационной техники.*

**Ключевые слова:** *GALS-технологии, интегрированная логистическая поддержка, информационно-техническая поддержка, заказчик, Help Desk, диспетчерская служба, сайт поддержки, проблемы в эксплуатации, поддержка эксплуатации.*

Внедрение системы интегрированной логистической поддержки (ИЛП) на Комсомольском-на-Амуре производственном объединении имени Ю. А. Гагарина (КнААПО) проводится в несколько этапов в рамках концепции холдинга "Сухой" и политики предприятия в области качества.

Под интегрированной логистической поддержкой понимается установленное средство управления стоимостью жизненного цикла изделия на основе использования информационных технологий путем воздействия на процесс его проектирования, производства, эксплуатации и утилизации, выбора технических средств обслуживания и ремонта, систем обучения, снабжения, а также других ресурсов обеспечения эксплуатации.

Одной из задач ИЛП является оперативная и качественная поддержка эксплуатантов в гарантийный и послегарантийные периоды. Для этих целей была разработана система информационной технической поддержки (СИТП), позволяющая вести оперативный обмен информацией с гарантийными бригадами, находящимися на объектах эксплуатации, и самим эксплуатантом. Эта

система позволяет вести диалог с авторизовавшимися на Web-сервере ОАО "КнААПО" пользователями сети Internet.

СИТП предназначена:

а) для обеспечения информационной поддержки специалистами КнААПО внешних пользователей через Internet:

- обеспечения гарантированного ответа на вопросы внешних пользователей в установленное время в соответствии с бизнес-процессами предприятия;
- оперативной передачи информации внешним пользователям завода через Internet;

б) для организации базы знаний (архива) по решению возникающих в ходе эксплуатации проблем:

- обеспечение выборок из базы данных вопросов и ответов (внутренний сервер) с целью определения типовых проблем и методов их решения; выборки проводят по номеру борта, серии изделий и типу изделий;
- предоставление внешним пользователям информации о типовых проблемах и методах их решения, классифицированной по типам и сериям изделий.

В связи с особенностью организации компьютерной корпоративной сети (ККС) предприятия, не имеющей физической связи с сетью Internet, СИТП поделена на две подсистемы: внешнюю (система управления Web-сервером) и внутреннюю (система обработки сообщений). Передача данных между подсистемами осуществляется оператором на стандартном носителе информации (это может быть как диск, так и USB Flash Drive).

База данных СИТП представляет собой логистическую структуру состава изделия по отношениям "тип изделия", "серия изделия", "номер борта". Такая иерархия позволяет конкретизировать информацию вплоть до конкретной детали и агрегата определенного самолета, при этом структурировать общую статистику по типам и сериям изделия.

Внешняя подсистема представляет собой защищенный Web-сервер (Apache + PHP + SSL) и дополнительное программное обеспечение для настройки и сопровождения системы. Подсистема выполняет следующие задачи:

- предоставление внешним пользователям общей информации о СИТП и инструкции по работе с ней;
- предоставление персональной информационно-технической поддержки авторизованным внешним пользователям;

- хранение архива всех сообщений и предоставление авторизованного доступа к нему по соответствующей тематике;
- синхронизация данных с внутренней подсистемой;
- предоставление доступа к базе знаний часто задаваемых вопросов;
- резервирование информации и профилактика базы данных.

Внешними пользователями СИТП могут являться:

- сотрудники гарантийных бригад ОАО "КнААПО";
- заказчики изделий;
- поставщики комплектующих изделий и партнеры ОАО "КнААПО".

Для получения персональной информационно-технической поддержки внешнему пользователю передается персональный USB-ключ, который необходимо установить на клиентской машине, а также сообщается имя учетной записи и пароль для входа в систему. Клиентским ПО служит Microsoft Internet Explorer с встраиваемым специальным модулем для проверки ключа. При желании опция проверки ключа для определенного пользователя (или для всех) может быть отключена на стороне сервера. После прохождения авторизации пользователь получает доступ к информации, предназначенной ему для работы в системе.

Сообщения от внешних пользователей с определенной периодичностью синхронизируются (одним из модулей подсистемы) и физически переносятся оператором в ККС ОАО "КнААПО".

Внутренняя подсистема служит для обработки полученных от пользователей сообщений и представляет собой специализированное ПО, построенное на клиент-серверной архитектуре. Подсистема состоит из сервера баз данных (Oracle) и набора автоматизированных рабочих мест (АРМ) пользователей ККС:

- АРМ администратора подсистемы;
- АРМ диспетчера отдела спец. связи;
- АРМ диспетчерской службы, руководителей и специалистов.

АРМ администратора подсистемы предназначено для системного сопровождения СИТП и решает следующие задачи:

- управление учетными записями внешних и внутренних пользователей СИТП;
- управление данными об изделиях и типах изделий, подлежащих обслуживанию СИТП;
- управление данными о точках обслуживания гарантийными бригадами;
- настройка путей прохождения сообщений в зависимости от типов сообщений;
- резервирование информации и профилактика базы данных.

АРМ диспетчера отдела спецсвязи предназначено для управления синхронизацией внешней и внутренней подсистем СИТП.

АРМ диспетчерской службы, руководителей и специалистов представляет собой клиентский модуль, позволяющий решать разные задачи в зависимости от принадлежности внутреннего пользователя к одной из определенных (настраиваемых) групп:

- диспетчеров подразделения ОАО "КнААПО";
- специалистов;
- руководителей;

АРМ диспетчера подразделения дает возможность:

- проводить анализ поступающих сообщений, вести базу знаний СИТП;
- направлять сообщение определенному подразделению ОАО "КнААПО" или специалисту своего подразделения;
- вносить дополнительную информацию касательно поступившего сообщения;
- инициировать отправку сообщения внешнему пользователю СИТП.

АРМ руководителя позволяет руководителю подразделения:

- назначить специалиста своего подразделения для обработки сообщения с указанием срока исполнения;
- подтвердить легитимность полученного от специалиста ответа;
- вносить дополнительную информацию касательно поступившего сообщения.
- останавливать обработку сообщения с возвратом на предыдущий шаг обработки.

АРМ специалиста позволяет специалисту подразделения (в зависимости от подразделения):

- создавать ответ на полученное сообщение с возможностью прикрепления дополнительных файлов;
- инициировать отправку сообщения внешнему пользователю СИТП;
- проверять сообщения на наличие секретной и конфиденциальной информации;
- останавливать обработку сообщения с возвратом на предыдущий шаг обработки;
- вносить дополнительную информацию касательно поступившего сообщения.

Обработка сообщений во внутренней подсистеме СИТП ведется с помощью настраиваемых маршрутов прохождения сообщений через подразделения предприятия.

Предварительный анализ сообщений ведут диспетчеры, которые направляют сообщение в ответственное за ответ подразделение. Диспетчерская служба подразделения назначает ответственного специалиста. После написания ответа его проверяет начальник соответствующего подразделения, начальник диспетчерской службы, специа-

лист режимно-секретного отдела (PCO), после этого сообщение можно переносить во внешнюю подсистему. Каждый ответственный специалист делает в системе соответствующие отметки о статусе сообщения.

Вся информация о прохождении сообщений остается в базе данных, что позволяет вести учет и анализ работы всей внутренней подсистемы СИТП. С определенной регламентом работы с СИТП периодичностью экспертная группа рассматривает вопросы, поступившие по СИТП, и ответы на них на предмет выявления типовых проблем. Проблемы классифицируются по отношению "тип изделия", "серия изделия", "номер борта". Ответы на часто задаваемые вопросы (ЧЗВ) становятся результатом аналитической обработки базы и доступны по установленной классификации. Каждый эксплуатант изделия имеет доступ к ЧЗВ по своему изделию и использует этот накопленный опыт.

Применение СИТП существенно сокращает время на подготовку поступившего ответа, позво-

ляет обобщать и анализировать поступающую информацию в автоматизированном режиме. В целом экономическая эффективность системы складывается из сокращения трудоемкости на обеспечение информационно-технической поддержки процесса эксплуатации самолетов и повышения привлекательности продукции предприятия на внешнем и внутреннем рынках.

#### Список литературы

1. **Информационное** обеспечение, поддержка и сопровождение жизненного цикла изделия / Под общ. ред. В. В. Бакаева. М.: Машиностроение-1, 2005. 624 с.
2. **Судов Е. В., Левин А. И., Барабанов В. В., Давыдов А. Н.** Концепция развития CALS-технологий в промышленности России. М.: ВИМИ, 2002. 127 с.
3. **CALS** (Continuous Acquisition and Life cycle Support — непрерывная информационная поддержка жизненного цикла продукции) в авиастроении / Научн. ред. А. Г. Братухин. — М.: Изд-во МАИ, 2002. — 676 с.
4. **Информационные** технологии в наукоемком машиностроении: Компьютерное обеспечение индустриального бизнеса / Под общ. ред. А. Г. Братухина. — Киев: Техніка, 2001. — 728 с.

УДК 004.4'234

**А. А. Алексеев**, alex@informos.ru

**А. И. Куликов**, kulikov@nmsf.ssc.ru

**В. Г. Старушкин**, starushkin@mail.ru,

Новосибирский государственный университет,

Институт вычислительной математики и математической геофизики СО РАН,

г. Новосибирск

## Визуальный построитель геологических сред "Seismic Model Manager"

*Описывается разработанный программный комплекс, который позволяет строить непротиворечивую модель геологического разреза, редактировать ее, сохранять и экспортировать ее в различные геофизические форматы. Отличительной особенностью данного инструментария является интерактивный пересчет параметров модели при визуальном редактировании пользователем геометрии модели и верификация внесенных изменений с подсказкой пользователю о возможных действиях. Разработанный программный комплекс позволяет провести расчет волнового поля по созданной геологической модели на удаленном вычислительном сервере по одной или серии расчетных программ. В комплексе имеется различный инструментарий для визуализации результатов численного моделирования. Результаты численного моделирования могут быть использованы специалистом-геофизиком как для модификации геологической модели, так и для дальнейшей ее обработки.*

**Ключевые слова:** геологический разрез, визуализация данных, визуальный редактор, изометрическое сглаживание.

### Введение

Целью работы является создание инструментария, необходимого специалистам-геофизикам для проведения численного моделирования распространения сейсмических волн в сложно построенных моделях упругих сред с помощью различных алгоритмов численного моделирования.

По сути построитель "Seismic Model Manager" и Seismic Model Editir, как его основная часть, является специализированным графическим векторным редактором, позволяющим подготавливать данные для определенных задач математической геофизики. Редактор позволяет создавать непротиворечивую модель, поддерживает вери-

фикацию и сохранение полученной модели в различных форматах. Важным моментом является то, что визуальное редактирование влечет за собой изменение геофизических данных модели [5].

Поскольку специалист-геофизик работает на персональной рабочей станции, а расчет проводится на удаленном высокопроизводительном вычислительном комплексе, важным компонентом продукта является утилита Seismic Model Client, входящая в представленный комплекс, которая позволяет осуществлять удаленный доступ к сибирскому суперкомпьютерному центру для выполнения расчетов моделей с использованием имеющихся алгоритмов.

Потребность в такой инструментарию актуальна как для ученых-теоретиков, так и для практиков. Вообще говоря, подобные продукты и решения, состоящие из модуля создания, редактирования и сохранения модели и "тонкого клиента", позволяющего осуществлять удаленную работу с моделью, используются в различных областях, таких как, например, газовая динамика, 3D-моделирование.

### Проектирование

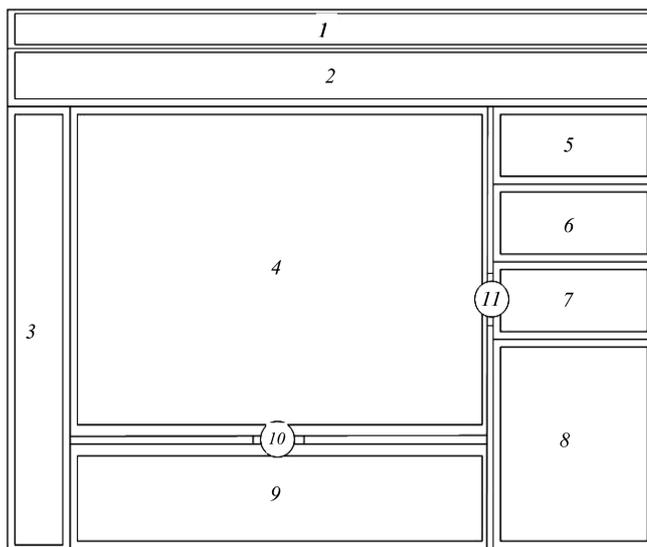
При проектировании системы был рассмотрен ряд специализированных продуктов [1]. В частности, японский построитель моделей, используемый в рамках пакета Nautilus компании Schlumberger и ASPIS, который является российской разработкой, выполненной на базе Института вычислительной математики и математической геофизики (бывший ВЦ) СО РАН, а также некоторые общеизвестные программные продукты [4].

В соответствии с этим обзором был разработан и предложен графический интерфейс (см. рисунок) пользователя с подробным описанием функциональности каждого элемента [2].

### Архитектура

В процессе проектирования командой разработчиков велось активное обсуждение и выяснение требований к программному продукту со стороны специалистов геофизиков. Было составлено техническое задание и зафиксированы все требования. Затем были выбраны технологии и разработана архитектура программного средства. Контроль над проектом велся с помощью MS Project.

Проект было решено реализовывать с использованием технологий .Net. Являясь современной и удобной средой разработки, Microsoft Visual Studio .Net позволяет разработчику с легкостью создавать Desktop приложения.



### Интерфейс приложения:

1 — строка главного меню; 2 — область включаемых тулбаров; 3 — стандартный тулбар редактирования; 4 — рабочая область; 5 — панель "Layers"; 6 — окно точечного задания физических параметров; 7 — панель "Info"; 8 — окно Model Explorer; 9 — окно физических параметров для границы разреза; 10 — кнопка, которая сдвигает (сворачивает) панель (9) вниз; 11 — кнопка, которая сдвигает (сворачивает) панель (11) вправо

При разработке комплекса использовался подход с разделением системы на различные модули. Так как одним из главных требований к комплексу было предоставление возможности дальнейшего расширения и настройка продукта под конкретного специалиста, то при таком архитектурном решении возможна отдельная доработка модели, что необходимо довольно часто. И наоборот, возможно использование самой модели отдельно в других приложениях.

### Реализация

В процессе разработки были выполнены следующие задачи:

1. Создание формализма для описания модели геолого-геофизической среды, ее геометрической структуры и визуальных методов работы с этой моделью. При этом нужно отметить, что речь идет о различных средах — пористой, слоистой, с памятью и т. д.

2. Разработка программного комплекса для визуальной работы с моделью, создание, редактирование и сохранение в специальной базе данных. Комплекс должен поддерживать задание системы возбуждения и регистрации.

3. Создание библиотеки конвертирования, позволяющей разработчикам различных алгоритмов использовать предложенный формат модели. Такой подход позволит унифицировать работу

с программными модулями, реализующими различные алгоритмы.

4. Предоставление возможности для работы геофизика в режиме удаленного доступа к высокопроизводительному многопроцессорному комплексу, например, на базе Сибирского суперкомпьютерного центра.

Математическая модель среды состоит из набора элементов:

- геометрических свойств геологического разреза;
- физических параметров среды, заданных для каждой точки;
- системы возбуждения и регистрации сейсмических полей;

Над элементами модели выполняются следующие операции:

- редактирование;
- генерация данных для того или иного алгоритма расчета сейсмического поля;
- верификация.

При этом учитываются ограничения на данные элементы и их взаимосвязи.

Постоянная верификация модели поддерживает указания пользователю о возможных действиях с моделью [6].

При построении границ слоев модели используется метод изогеометрического сглаживания [7].

Основная часть работы была связана с созданием редактора моделей.

Приложение разделено на два модуля:

- модуль работы с формализмом модели (реализация библиотеки DLL на C++);
- модуль графической надстройки (GUI, C#.NET).

**Модуль работы с моделью.** Для данного модуля разработана иерархическая модель данных, представляющая собой структуру вида: Модель—Слой—Граница—Точка.

**Модуль удаленного доступа.** В архитектуру системы была изначально заложена клиент-серверная модель взаимодействия для того, чтобы пользователь имел возможность отправить модель для расчета на удаленном кластере. Данный модуль также реализован на языке C#.

Клиентская часть поддерживает размещение модели и метаданных к ней (кем была создана модель, для какого геологического профиля и др.) в хранилище моделей и постановку задачи на счет с использованием определенной модели. При этом модель берется из хранилища, выбирается указанный алгоритм и формируется так называемый *make-file* — файл, который будет выполнен на кластере. После того, как задача расчета инициализирована, выдается сообщение сервера о статусе работ. По окончании работы результаты расчета забираются с сервера, и клиенту сообщается о завершении задачи.

Таким образом, система предоставляет возможность создания и редактирования модели, а также позволяет проводить удаленные вычисления.

## Итоги

Представленное программное средство позволяет:

- создавать и визуально редактировать геометрию геологической модели среды;
- задавать необходимые параметры (скорость, плотность, коэффициенты поглощения и т. д.);
- задавать систему возбуждения и наблюдения;
- проводить расчет по выбранной программе на многопроцессорном вычислительном комплексе, в том числе и с удаленного терминала;
- получать и визуализировать результаты численного моделирования.

В рамках данной работы, как одной из задач проекта, студентами и аспирантами, участвующими в проекте, осваиваются технологии и продукты компании Microsoft. Это среда разработки Microsoft Visual Studio .Net, Microsoft Visio, C#. Управление проектом ведется с помощью MS Project [3].

Сейчас реализована и протестирована 2D- и псевдо3D-версия Model Editor. Планируется перейти к разработке 3D-версии. Также в ближайшее время будет проходить внедрение и использование в проектах институтов для работы с реальными геофизическими данными.

## Список литературы

1. **Липаев В. В.** Технико-экономическое обоснование проектов сложных программных средств. М.: СИНТЕГ, 2004. 284 с.
2. **Липаев В. В.** Системное проектирование сложных программных средств для информационных систем. М.: СИНТЕГ, 2002. 268 с.
3. **Лабор В. В., Си Шарп.** Создание приложений для Windows. Минск: Харвест, 2003. 385 с.
4. **Vikiz** method for logging oil and gas boreholes. Novosibirsk: Publishing House of the SB RAS, Branch "Geo", 2002. 110 p.
5. **Пузырев Н. Н.** Методы и объекты сейсмических исследований. Новосибирск: Изд-во СО РАН, НИЦ ОИГГМ, 1997.
6. **Эйнджел Э.** Интерактивная компьютерная графика. М.: Издательский дом "Вильямс", 2001.
7. **Куликов А. И.** Некоторые задачи вычислительной геометрии. Изогеометрическое сглаживание // Труды 15-й Международной конференции по компьютерной графике и ее приложениям "ГрафиКон2005". М.: Изд. Института вычислительной математики и математической геофизики СО РАН. С. 382—385.

## Как получить программный комплекс

Для того чтобы получить демоверсию программного комплекса, необходимо отправить запрос одному из авторов статьи по электронной почте, указанной в статье. Запрос будет рассмотрен и в случае положительного решения ограниченная версия продукта будет выслана по почте. В таком же порядке принимаются заявки на полнофункциональную версию продукта. Условия предоставления такой версии обсуждаются в индивидуальном порядке.

УДК 621.391; 519.21; 519.245

**Г. Ф. Малыгина**, д-р техн. наук, проф.,  
**А. В. Меркушева**, канд. техн. наук,  
Санкт-Петербургский политехнический  
университет

## Совместное использование нейронной сети и вейвлет- преобразования при анализе нестационарного сигнала

*Представлены некоторые приложения на основе совместного использования нейронной сети и вейвлет-преобразования (ВП) в задачах анализа нестационарных сигналов. Первая задача относится к системе мониторинга речевого сигнала, в которой выполнение ВП используется для сжатия речевого сигнала, идентификации распределения вероятности ВП сигнала и вейвлет-фильтрации с адаптивным порогом. Алгоритм нейронной сети используется для распознавания типа фрейма сигнала (речь/пауза) и управления двумя фазами: определения порогов (на фреймах паузы) и вейвлет-фильтрации речевого сигнала на основе этих порогов (на фреймах речи). С помощью такого метода достигается адаптация к изменению уровней шума. Разложение сигнала обеспечивается на основе вейвлет-пакета, частотные субполосы которого аппроксимируют критические полосы перцептуальной модели. Получены новые соотношения для вычисления порогов на основе экспоненциально-степенного распределения, которые являются обобщением выражений Крамера для распределения максимального значения Гауссовской переменной. Это позволяет фильтровать (в вейвлет-области) шумы самого общего типа. Во второй задаче предложен алгоритм для определения уровней свободного газа в трубопроводе коммерческой нефти. Метод основан на бесконтактном радиационном плотномере. Показания плотномера обрабатываются нейронной сетью, которая выполняет идентификацию в потоке нефти фреймов, информационно значимых для вейвлет-преобразований сигнала: временных интервалов с газом и интервалов, не содержащих газа.*

Сложные сигналы, подлежащие контролю, обработке и анализу в измерительной информационной системе (ИИС), как правило, отражают нестационарные процессы. Поэтому основой анализа служит информация о динамике частотной структуры сигнала, получаемая с помощью вре-

мя-частотных преобразований (ВЧП) или вейвлет-преобразований (ВП). Вместе с тем сигналы ИИС часто носят фазовый характер, при котором интервалы сигнала, несущие проблемно значимую информацию, перемежаются с интервалами, отражающими "шум", т. е. не несущими информационный компонент сигнала. Смена этих фаз носит случайный характер и обычно не может адекватно моделироваться. Для отделения интервалов нестационарного сигнала, несущих интересующую информацию, от интервалов с фоновыми характеристиками сигнала целесообразно использовать нейронную сеть. Ниже будут рассмотрены две прикладные задачи, решение которых основано на совместном использовании методов вейвлет-преобразования и нейросетевого алгоритма для идентификации типа фреймов контролируемых сигналов. Одна из задач связана с информационной системой мониторинга речевых сигналов, которые являются наиболее общей формой нестационарного сигнала, не имеющего какой-либо определенной модели динамики. В другой задаче представлен способ бесконтактного измерения параметров газожидкостного потока, основанный на ВП и нейронной сети.

Систематизация ВЧП (линейных — Габора, КПФ, Зака; и билинейных — Вигнера, Вильсона, Бесселя, Бертрена) показала их эффективность для сигналов специального вида, отсутствие быстрых алгоритмов и ограниченность по разрешению [1, 2]. ВП имеет ряд разновидностей, характеристики которых обусловлены типом и свойствами базиса. Выбор вида ВП, лучшим образом соответствующего сигналу и задачам обработки, определяется целевыми установками информационной системы.

### Нейросетевой алгоритм анализа речевого сигнала в области его вейвлет-отображения

Исследование связано с мониторингом речевого сигнала (РС) средствами специализированной информационной системы, обрабатывающей семантику фрагментов РС, эквивалентных элементам предзаданной базы данных; выполняющей контроль, регистрацию и многоканальное распределение информационно значимых интервалов РС с такими фрагментами. Необходимые формы обработки включают: сегментацию на фреймы длительностью 10...20 мс; детектирование вида фрейма (речь/пауза); реализацию ВП; определение распределения мощности шума по субполо-

сам; выбор адаптивного уровня дискриминации шума для коррекции фреймов речевой активности; проведение коррекции (снижение уровня искажений за счет шума); формирование базы данных; передачу сигнала в канал (непосредственно или со скремблированием для защиты информации); восстановление и аудиоконтроль речевого сигнала. Желательными характеристиками ВП являются компактность, гладкость (нулевые моменты), симметрия (линейнофазовость), полное восстановление сигнала, наличие быстрого алгоритма. Кроме того, модификация вейвлет-преобразования в форме вейвлет-пакетов позволяет до определенной степени целенаправленно формировать бинарное дерево разложения сигнала так, чтобы получать его представление в предварительно заданных неравномерных частотных диапазонах. Как показано далее, последнее свойство дает возможность при ВП представлять речевой сигнал и сопутствующие виды шума в частотных диапазонах, эквивалентных "критическим полосам" модели восприятия речи, которая учитывает особенности фильтрации звука слуховым аппаратом человека.

ВП объективно не могут быть ранжированы по предпочтительности совокупности свойств, так как ряд свойств не может присутствовать у базиса одновременно. Это относится к ортогональности, компактности и симметрии. Поэтому на основе компромисса свойств основной группы ВП (базисы Добеши, Вильсона, Вальтера, биортогональные, базисы сплайнов [3–5]) выбраны биортогональные ВП с базисами bior6.8 и bior3.9. Решающим в таком выборе послужило требование симметрии, так как линейно-фазовость необходима для качественного восстановления речевого сигнала.

Предложенный метод детектирования фреймов с речевой активностью преследует две цели:

- за счет нейросетевого алгоритма удаления пауз получить более совершенный способ сжатия сигнала, которое необходимо для баз данных в цифровой телефонии, криптографии, распознавании элементов речи;
- построить схему адаптивной вейвлет-фильтрации речевого сигнала (РС) для снижения уровня искажения за счет шумов различного типа.

Обе задачи решаются на основе ВП. Разложение сигнала осуществляется на основе вейвлет-пакета, структура которого согласована с перцептуальной моделью [6]. Результатом этой процедуры является совокупность наборов вейвлет( $w$ )-коэффициентов, которая соответствует числу субполос и числу критических полос перцептуальной модели. Дисперсия каждого набора  $w$ -коэффициентов характеризует мощность сигнала в субполосе. Вектор дисперсий  $\sigma_1 \dots \sigma_n$  ( $n$  — число субполос разложения) используется как характеристика со-

стояния процесса (речь/пауза), на которой основано детектирование. Критерием качества аппроксимации перцептуальной модели  $w$ -пакетом служит показатель  $Q$  средней квадратичной точности соответствия границ критических полос модели и границ субполос  $w$ -пакета:

$$Q = \frac{1}{n} \sum_{i=1}^n \left\{ \left( \frac{\hat{f}_i^1 - \hat{f}_i^1}{\bar{f}_1} \right)^2 + \left( \frac{\hat{f}_i^2 - \hat{f}_i^2}{\bar{f}_2} \right)^2 \right\}, \quad (1)$$

где  $f_i^1, f_i^2$  — нижняя и верхняя границы частотной полосы в перцептуальной модели;  $\bar{f}_i$  — положение центра полосы;  $\hat{f}_i^1, \hat{f}_i^2$  — то же для субполос сформированного  $w$ -пакета;  $n$  — число частотных диапазонов в модели, для речевого сигнала  $n = 22$  в полосе 8 кГц.

Детектирование, как задача бинарного распознавания, реализуется нейронной сетью в форме двухслойного перцептрона. При обучении перцептрона использовано два способа: на основе алгоритма с обратным распространением ошибки [7] и на основе оптимизации нелинейной функции отображения сети вход—выход [8]. Во втором способе применен метод сопряженных градиентов с модификацией обновления градиента по схемам Флетчера—Ривса и Полака—Рибере [9]. Осуществляется коррекция вектора весов перцептрона  $w(n)$  в направлении  $p(n)$  и одновременно модификация самого вектора  $p(n)$  по соотношениям

$$w(n+1) = w(n) + \eta(n)p(n); \quad (2)$$

$$p(n+1) = -g(n+1) + \beta(n)p(n), \quad (3)$$

где  $g(n)$  — вектор градиента;  $\eta(n)$  — параметр скорости обучения, а параметр  $\beta(n)$  определяется по соотношению (4) в первой схеме и по соотношению (5) — во второй:

$$\beta(n) = \left[ \frac{g^T(n+1)g(n+1)}{g^T(n)g(n)} \right] \quad (\text{Флетчер—Ривс}); \quad (4)$$

$$\beta(n) = \left[ \frac{g^T(n+1)[g(n+1) - g(n)]}{g^T(n)g(n)} \right] \quad (\text{Полак—Рибере}). \quad (5)$$

При непосредственном применении алгоритма обратного распространения использован метод наискорейшего спуска с определением параметра скорости обучения в соответствии с правилом дельта—дельта адаптации или обобщенной дельта—дельта адаптации [7], [10].

Бинарный выход перцептрона — индикатор принадлежности фрейма к речи или паузе (шуму) — используется либо для выработки уровней дискриминации (порогов)  $w$ -фильтрации (на паузах), либо для самой фильтрации (на сегментах речи).



Рис. 1. Блок-схема метода вейвлет-фильтрации речевого сигнала с применением перцептуальной модели и нейросетевого алгоритма

После  $w$ -фильтрации сигнал поступает в канал связи (непосредственно или со скремблированием), в базу данных информационной системы мониторинга или после восстановления — на аудиоконтроль.

Общая схема метода обработки нестационарного сложного сигнала (каким является РС) с использованием нейросетевого алгоритма на вейвлет-преобразовании этого сигнала представлен на рис. 1. В ходе экспериментального исследования предложенного метода для информационной системы речевого мониторинга выполнен анализ записей речевого сигнала и моделирование на компьютере алгоритмов обучения многослойного перцептрона. Как отмечено выше, при применении алгоритма обучения в форме оптимизации нелинейной функции преобразования сети вход—выход с двумя модификациями метода сопряженных градиентов и в базовом алгоритме обучения с обратным распространением ошибки использовалась форма адаптации параметра скорости обучения  $\eta(n)$  на основе метода Джекобса, т. е. правила дельта—дельта и обобщенного правила дельта—дельта.

Из возможных видов шума, сопутствующего речевому сигналу, апробированы белый шум (с равномерным спектром и некоррелированными отсчетами); окрашенный шум с медленным изменением частотных компонентов; узкополосный шум ("трансформаторный") — шум от высокоиндуктивных приборов и электромеханического оборудования; а также музыкальный фон.

Выход перцептрона управляет последующим этапом обработки  $w$ -образа речевого сигнала: сегмент — пауза используется для выработки порогов  $w$ -фильтрации и их запоминания, сегмент речевой активности подвергается  $w$ -фильтрации для снижения уровня шума.

Для Гауссова белого шума в качестве порога дискриминации  $t_j$   $w$ -коэффициентов в субполосе  $j$  предложено соотношение

$$t_j = \sqrt{2 \ln(n)} \sigma_{jn}, \quad (6)$$

где  $n$  — число отсчетов;  $\sigma_{jn}^2$  — дисперсия  $w$ -коэффициентов шума в субполосе  $j$ .  $W$ -фильтрация

реализует преобразование  $w$ -коэффициентов  $\{d_{jk}\}_k$  по гибкой схеме, которая является субоптимальной процедурой [10]:

$$\tilde{d}_{jk} = \text{sign}(d_{jk}) [d_{jk} - t_j]_+, \quad (7)$$

Проведенный анализ показал, что выражение (6) получено в работе [11] с использованием асимптотической (при  $n \rightarrow \infty$ ) оценки распределения максимумов для Гауссова шума. Уточнение [12], связанное с конечностью числа отсчетов на контролируемом сегменте сигнала, определяет значение смещенности порога. Значение смещенности  $\Delta t_j$  определяется соотношением

$$\Delta t_j = \left[ \frac{\ln(\ln(n)) + \ln \xi + \ln(2\pi\sqrt{2})}{\sqrt{2 \ln(n)}} \right] \sigma_j, \quad (8)$$

где  $\xi$  — статистическая компонента с плотностью распределения  $h(\xi) = e^{-\xi}$ , которая не учтена в работе [10]. Для оценки поправки порога  $\Delta t_j$  следует взять значение  $\times$  с желаемой достоверностью в соответствии с ее распределением.

В условиях изменяющегося уровня помеха—шум на интервалах, соответствующих паузам, можно оценить значения  $\{\sigma_{jn}\}$ , которые целесообразно использовать для корректировки порогов дискриминации на следующем интервале, несущем речевой сигнал.

Исследован и другой аспект  $w$ -фильтрации. Речевой сигнал отражает нестационарный процесс так же, как и ряд модификаций шума, поэтому спектр их может быть неравномерным и распределение не Гауссовым. Кроме того, разложение на субполосы при ВП связано с изменением закона распределения сигнала, так как в совокупности  $w$ -коэффициентов более вероятными становятся малые значения. Так что широко распространенное представление о Гауссовом распределении сигнала в вейвлет-области противоречит факту сжатия сигнала. Поэтому адекватность схемы определения уровней дискриминации при ВП может быть обеспечена только путем учета корректного вида статистического распределения  $w$ -коэффициентов.

При анализе вида распределения установлено, что оно соответствует экспоненциально-степенному закону. Семейство этих распределений допускает широкое варьирование формы (от остроговершинного до равномерного, включая распределения Коши и Гаусса) и имеет плотность

$$f(x) = \frac{\alpha}{2\lambda\sigma\Gamma(1/\alpha)} \exp\left(-\left|\frac{x}{\lambda\sigma}\right|^\alpha\right), \quad (9)$$

где  $\lambda\sigma$  и  $\alpha$  — параметры масштаба и формы;  $\Gamma$  — гамма-функция.

Найденная закономерность для не Гауссова шума положена в основу схемы определения порогов дискриминации. Результаты этой схемы можно сформулировать следующим образом [11]:

- максимальные значения шума на контролируемом сегменте объема  $n$  отсчетов имеют функцию распределения  $G(x)$  вида

$$G(x) = \int_{-\infty}^{\frac{x}{\lambda\sigma}} \frac{n\alpha}{2^n \Gamma\left(\frac{1}{\alpha}\right)} \left[ 1 + \frac{\Gamma\left(\zeta; \frac{1}{\alpha}\right)}{\Gamma\left(\frac{1}{\alpha}\right)} \right]^{n-1} \frac{d\zeta}{e^{-\zeta\alpha}}, \quad (10)$$

где  $\Gamma(\zeta; 1/\alpha)$  — неполная гамма-функция;  $\Gamma\left(\zeta; \frac{1}{\alpha}\right) =$

$$= \int_0^{\zeta} \exp(-t) t^{\left(\frac{1}{\alpha}-1\right)} dt;$$

- уровни  $P$ -процентной достоверности  $x_p$  для максимальных значений шума общего вида с экспоненциально-степенным распределением и для сегмента с  $n$  отсчетами определяются выражением

$$x_p = \left\{ F_{\Gamma}^{-1} \left[ \frac{n+2\ln P}{n}; \frac{1}{\alpha} \right] \right\}^{\frac{1}{\alpha}} \lambda\sigma, \quad (11)$$

где  $F_{\Gamma}^{-1}$  — обратная функция от интегральной функции распределения  $F_{\Gamma}$ ;

$$F_{\Gamma}(x, \alpha) = \frac{1}{\Gamma(\alpha)} \int_0^x e^{-t} t^{\alpha-1} dt. \quad (12)$$

Введение нормированного порога  $t = x/\lambda\sigma$  позволяет привести выражение (11) к следующему виду:

$$t = [L]^{\frac{1}{\alpha}} \left[ 1 - \frac{\frac{\alpha-1}{\alpha} \ln(L) + \ln\left(2\Gamma\left(\frac{1}{\alpha}\right)\right)}{\alpha L} + \frac{\ln\left(1 + \left(\frac{1-\alpha}{\alpha}\right)\frac{1}{L}\right)}{\alpha L} \right] - [L]^{\frac{1}{\alpha}} \frac{\ln \xi}{\alpha L}. \quad (13)$$

Применение (13) к нормальному распределению ( $\alpha = 2, \sigma = 1, \lambda = \sqrt{2}$ ) дает формулу Крамера [12] для экстремального уровня отсчетов шума с Гауссовым распределением:

$$t = \sqrt{2L} - \frac{\ln(L) + \ln(4\pi)}{2\sqrt{2L}} - \frac{\xi}{\sqrt{2L}}. \quad (14)$$

(В соотношениях (13) и (14) для компактности введено обозначение  $L \equiv \ln(n)$ ). Поэтому выражение (13) является обобщением соотношения, полу-

ченного Крамером, на класс экспоненциально-степенных распределений. В методе вейвлет-фильтрации речевого сигнала с сопутствующим шумом выражение (13) интерпретируется как основа для получения уровней дискриминации.

Таким образом, вейвлет-фильтрация осуществляется на основе выбора порогов дискриминации в каждой субполосе разложения сигнала. Вектор дисперсий вейвлет-коэффициентов в субполосах служит исходным для детектирования типа сегмента речь/пауза (шум). Детектирование (распознавание) реализуется нейронной сетью в форме двухслойного перцептрона, бинарный выход которого управляет выбором одной из двух фаз общего алгоритма — формирование и запоминание порогов дискриминации на сегменте паузы и реализация фильтрации на сегменте речевой активности.

### Нейросетевой алгоритм на вейвлет-преобразовании сигнала при определении параметров газожидкостного потока

Вторая задача (из упомянутых в вводной части статьи) связана с получением характеристик газожидкостного потока на основе применения нейросетевого алгоритма к вейвлет-отображению сигнала, который снимается с бесконтактного датчика радиационного плотномера в ИИС, осуществляющей внешний контроль нефтегазовой смеси в трубопроводе.

Полноценный учет объема продукции товарной нефти на стадии добычи, транспортировки по трубопроводам и на нефтегазоперекачивающих станциях зависит от возможности регистрации относительной доли свободного газа (ОДСГ). Интервалы потока нефтегазовой смеси, в которых газовая фракция присутствует в форме пузырей, спорадически перемежаются с интервалами чистой нефти. ОДСГ  $\phi$  связана с объемами жидкости и газа  $V_{\text{ж}}$  и  $V_{\text{г}}$ , проходящими через контролируемое сечение трубопровода, соотношением

$$\phi = 1 - \frac{V_{\text{ж}}}{V_{\text{ж}} + V_{\text{г}}}, \quad (15)$$

и может быть выражена через плотность нефти  $\rho_{\text{ж}}$  и плотность нефтегазовой смеси  $\rho_{\text{см}}$ :

$$\phi = 1 - \frac{\rho_{\text{см}}}{\rho_{\text{ж}}}. \quad (16)$$

Контроль плотности лежит в основе бесконтактного метода измерения ОДСГ в ИИС "Фактор", разработанной в ЦНИИ РТК [13]. В методе используется измерение интенсивности гамма-излучения Cs-137 после поглощении его потоком с переменной плотностью, обусловленной нали-

чем интервалов с содержанием свободного газа. Переменная интенсивность гамма-излучения контролируется блоком детектирования (БД) со стинцилляционным кристаллом Na(J) и фотоумножителем.

Ввиду большой скорости потока (до 5 м/с) контроль его проводится непрерывно и дает информацию в форме нестационарного процесса с интенсивностью отсчетов, зависящей от ОДСГ. Сложным этапом обработки и интерпретации этой информации является разделение длительности контроля потока на интервалы с чистой нефтью и с нефтегазовой фракцией, включающей пузыри. В ИИС использованы два способа обработки: метод бегущей выборки с оценкой статистических параметров распределения отсчетов БД, получением отношения правдоподобия и сравнением его с допуском; метод, в котором используется полуэмпирические зависимости автокорреляции отсчетов и результаты градуировки на поверочно-испытательном стенде. Оба метода не вполне совершенны и имеют небольшую зависимость от скорости потока и плотности нефти. Это стимулирует поиск новых форм анализа измерительной информации.

Измерение и обработка сигналов в ИИС связаны со сложным изменением характеристик процессов, которые трудно описать априорными моделями. Основой обработки нестационарного сигнала может служить информация о динамике его частотной структуры, получаемая с помощью время-частотных преобразований (ВЧП) или вейвлет-преобразования (ВП). Анализ группы ВЧП (аналогично первой рассмотренной выше задаче) показал предпочтительность использования ВП. ВП дают неравномерную (логарифмическую) шкалу разрешения по частоте, которая соответствует равномерной относительной ширине частотных интервалов анализа сигнала, реализуется с помощью быстрых алгоритмов на основе квадратурных зеркальных фильтров (КИХ-фильтров с короткими характеристиками и рядом полезных свойств). В связи с этим при разработке нового метода обработки информации в ИИС "Фактор" в качестве основы анализа нестационарного сигнала использовано ВП с биортогональными базисами Добеши *bior6.8* и *bior3.9*. Вторая составляющая метода — применение нейронной сети для определения типов интервалов регистрации (с чистой нефтью или с примесью газообразной фракции).

Таким образом, в целях совершенствования наиболее критического этапа измерения ОДСГ в потоке нефти исследован метод на основе вейвлет-преобразования и нейросетевого алгоритма на многослойном персептроне. ВП позволяет локализовать частотные свойства сигнала во време-

ни, дает сжатую форму его представления, снижает уровень шумов и для сигнала со спорадически чередующимися характерными типами интервалов позволяет применить нейронную сеть для различения типа интервала.

В методе используется вейвлет-преобразование на последовательности сегментов измерительных данных, поступающих с блока детектирования ИИС, вычисление мощности вейвлет-коэффициентов в каждой области разложения, нормирование вектора мощности и использование его в виде входного вектора для нейронной сети, в качестве которой применен персептрон для бинарной классификации типа интервала.

Выполнено исследование, позволившее выбрать структуру многослойного персептрона и обосновать метод обучения, имеющего лучшую скорость сходимости. Для повышения эффективности алгоритма обучения персептрона выбрана его конфигурация с использованием метода наращивания. Сопоставление структур проводилось на двух лучших (для анализируемых данных) алгоритмах обучения, и предпочтение отдано структуре 5-2-1.

Обучение персептрона основано на алгоритме с обратным распространением ошибки. Алгоритм реализуется итеративно с двумя проходами по сети — прямым и обратным, при которых вычисляются функциональные преобразования вектора входного сигнала  $\mathbf{x} = (x_1 \dots x_p)^T$  и корректировка вектора параметров  $\mathbf{w}$ . При обратном проходе на каждой итерации  $n$  локальный градиент  $\delta_j(n)$  определяется, если нейрон в выходном слое, по формуле

$$\delta_j(n) = e_j(n)\phi'(v_j(n)), \quad (17)$$

если в скрытом слое, по формуле

$$\delta_j(n) = y_j(n)[1 - y_j(n)] \sum_k \delta_k w_{jk}, \quad (18)$$

где  $v_j(n)$ ,  $y_j(n)$ ,  $e_j(n)$  — возбуждение, выход и ошибка  $j$ -го нейрона,  $\phi'$  — производная его функции нелинейности;  $w_{jk}$  — параметры веса. Величина  $\delta_j(n)$  используется для корректировки весов с помощью  $\delta$ -правила

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n), \quad (19)$$

или с помощью модифицированного  $\delta$ -правила

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n), \quad (20)$$

где  $\alpha$  — параметр инерции;  $\eta$  — параметр скорости обучения.

Оценка параметра  $\eta(n)$  проводилась методом адаптации на основе правила Джексона.

Другой подход к обучению персептрона для детектирования типа интервала основан на трактовке этой фазы как оптимизации нелинейной функ-

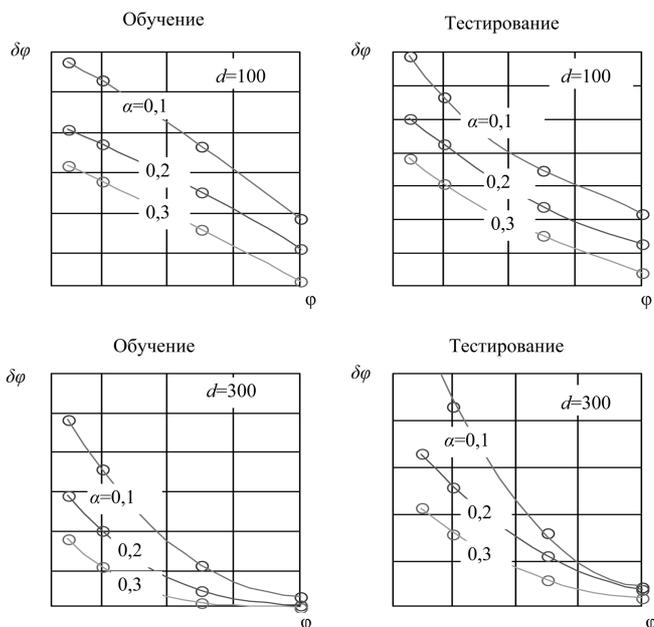


Рис. 2. Ошибка относительного содержания свободного газа для измерений на трубопроводе 100 и 300 мм с анализом информации на основе нейросетевого алгоритма и вейвлет-преобразования сигнала радиационного плотнмера (результаты имитационного моделирования)

ции отображения сети вход—выход. В исследовании применен метод сопряженных градиентов и метод Бройдена—Флетчера, который является модификацией метода Ньютона и связан с аппроксимацией матрицы Гессе с помощью конечных разностей градиента функции ошибок. Проведенный анализ показал, что лучшие результаты дает метод Бройдена—Флетчера. Он использован в моделирующей программе при исследовании характеристик метода обработки информации.

Моделирование процесса регистрации состояния гетерогенного потока нефти аппаратурой ИИС "Фактор" осуществлено с помощью программы, выполненной в среде Microsoft Visual C 6.0. Алгоритм имитирует симметричную геометрию расположения блока гамма-излучения и блока детектирования. Нестационарный характер изменения поглощения потока гамма-квантов за счет переменной плотности нефтегазовой смеси моделируется с помощью модуля "генератора пузырей",

который формирует газосодержание как процесс авторегрессии с автокорреляционной функцией экспоненциального вида. Значение параметра затухания этой функции ( $\alpha$ ) значительно влияет на погрешность определения ОДСГ.

В алгоритме моделирования использована аппроксимация закона Пуассона для распределения отсчетов БД Гауссовым законом, где  $\sigma_{\{N_i\}}^2 = \bar{N}_i$  ( $N_i$  — отсчеты БД,  $\bar{N}_i$  — параметр среднего). Корректность аппроксимации обеспечивает высокая интенсивность отсчетов БД.

Погрешность определения ОДСГ показана на рис. 2. Использование предложенного метода дает лучшие результаты при измерениях в потоках, которые характеризуются высокой степенью затухания автокорреляционной функции, т. е. в потоках, содержащих мелкие газовые включения и высокие скорости перемещения.

#### Список литературы

1. Jawerth B., Sweldens W. Wavelet-based multi-resolution analysis // SIAM Review. 1994. V. 36. N 3. P. 377.
2. Меркушева А. В. Классы преобразований нестационарного сигнала (время-частотные преобразования) // Научное приборостроение. 2002. Т. 12. № 2. С. 59—70.
3. Daubechies I. Wavelet transform and orthogonal wavelet bases // Proc. of Symposium in Applied Mathematics. 1993. V. 47. P. 1—32.
4. Mallat S. Multiresolution signal decomposition: wavelet representation // IEEE Transactions on pattern analysis and machine intellect. 1989. 11. P. 674—693.
5. Воробьев В. И., Грибунин В. Г. Теория и практика вейвлет-преобразования. СПб, 1999. 203 с.
6. Jayant N. S., Shohan Y. Signal compression based on model of human perception / Proc. of IEEE. 1993. V. 81. N 10. P. 1390.
7. Haykin S. Neural networks. A comprehensive foundation, NY: Prentice Yall, 1994. — 680 p.
8. Hornic K., Stinchcombe M., White H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward network // Neural Networks. 1990. V 3. P. 551—560.
9. Химмельблау Р. Прикладное нелинейное программирование. М.: Мир, 1975. 98 с.
10. Jackobs R. A. Increased rates of convergence through learning rate adaptation // Neural Networks, 1989. V. 1. P. 295—307.
11. Donoho D., Jonstone I. Ideal spatial adaptation by wavelet shrinkage // Biometrika. 1994. V. 81, N 3. P. 425—455.
12. Малыгина Г. Ф., Меркушева А. В. Вейвлет-фильтрация нестационарного сигнала с адаптацией на основе нейронной сети // Труды международной конференции по мягким вычислениям и измерениям SCM-2001, СПб: СПбГЭТУ, 2001. Т. 1. С. 239—242.
13. Кратиров В. А., Казаков А. Н. и др. Способ измерения параметров газожидкостного потока. Патент на изобретение N 94007263/25(007607), 1996.

**А. П. Кирсанов**, д-р техн. наук, проф.,  
Государственный университет —  
Высшая школа экономики

## Комбинаторный метод описания и контроля процессов

*Предлагается метод контроля, позволяющий при неточных и неполных измерениях параметров контролируемого дискретного процесса принимать решения о его принадлежности заданному классу процессов. Метод основан на описании процессов и результатов контроля так называемыми структурными параметрами, образующим частично упорядоченное множество. В терминах частичного порядка сформулированы условия принадлежности контролируемого процесса заданному классу процессов. Исследованы вероятностные характеристики предлагаемого метода контроля.*

**Ключевые слова:** метод контроля, структурные параметры, упорядоченное множество.

### Введение

Рассматривается задача контроля выполнения заданных последовательностей событий, именуемых далее процессами в технических системах. Контроль за процессом осуществляется по информации об измеряемых параметрах. Предполагается, что эти параметры могут принимать конечное число значений за весь период контроля, т. е. наблюдаемый процесс является ступенчатым.

Существенной особенностью предлагаемых в работе способа описания процессов, процедур и контроля является то, что вместо значений параметров используются их ранги или номера значений в ранжированном ряду. Такое рассмотрение оправдано при доступности надежной информации о рангах параметров и отсутствии либо сложности получения достоверной информации о точных значениях контролируемых параметров. Другой, более принципиальной причиной такого подхода является то, что правильность процесса может определяться не абсолютными значениями некоторого параметра, а их взаимным положением в ранжированном по величине ряду возможных значений параметра.

Информация, на основе которой принимается решение о правильности процесса, представляет собой последовательность рангов измерений. Число контрольных измерений, как правило, меньше числа значений, принимаемых контролируемым параметром за период контроля. Неполнота измерений может являться следствием сложности и длительности выполнения измерений или

их значительной стоимости. Предполагается, что каждое значение контролируемого параметра измеряется не более одного раза за интервал времени, в течение которого параметр остается неизменным.

В работе предлагается метод контроля, позволяющий в условиях неполноты информации о процессе описанного типа принимать решение о принадлежности контролируемого процесса заданному классу процессов. Метод основан на подходе, предложенном в работе [1], и использовании для описания процессов таких комбинаторных объектов, как перестановки чисел.

### Комбинаторное описание контролируемого процесса и результатов контроля

Будем представлять наблюдаемую часть процесса как последовательность  $G = \{g_1, g_2, \dots, g_n\}$  этапов  $g_i$ , представляющих собой интервалы времени, на которых контролируемые параметры сохраняют постоянные значения. Длительности этапов не являются элементами описания процесса. Для простоты основные идеи метода контроля рассматриваются в статье для процессов, контролируемых по одному параметру.

Для описания процесса используется так называемый структурный параметр (СП), являющийся перестановкой чисел  $1, 2, \dots, n$ . Структурный параметр  $d = (r_1, r_2, \dots, r_n)$  заданного процесса  $G = \{g_1, g_2, \dots, g_n\}$  строится следующим образом. Все значения, которые принимает контролируемый параметр на этапах  $g_i$ , ранжируются по убыванию (меньшие значения получают большие ранги или номера). Если на различных этапах параметр имеет одинаковые значения, то более "позднему" этапу присваивается больший ранг. Соответствующие этапам  $g_i$  ранги  $r_i$  записываются в список  $s$  в порядке следования этапов. Пример построения СП, описывающего ступенчатый процесс, показан на рис. 1. Из этого рисунка, в частности, видно, что СП не определяет однозначно процесс. Двум процессам на рис. 1 соответствует один СП  $s = (41325)$ . Таким образом, СП определяет целый класс процессов, отличающихся друг от друга длительностью этапов и значениями контролируемого параметра, но имеющих одинаковой порядок следования рангов этапов. Далее в качестве описания класса правильных, эталонных, процессов будет использоваться соответствующий СП.

По известному СП можно находить различные характеристики процесса. Например, можно узнать на каком этапе контролируемый параметр принимал  $i$ -е по величине (максимальное, если  $i = 1$ ) значение. Для этого достаточно найти номер позиции,

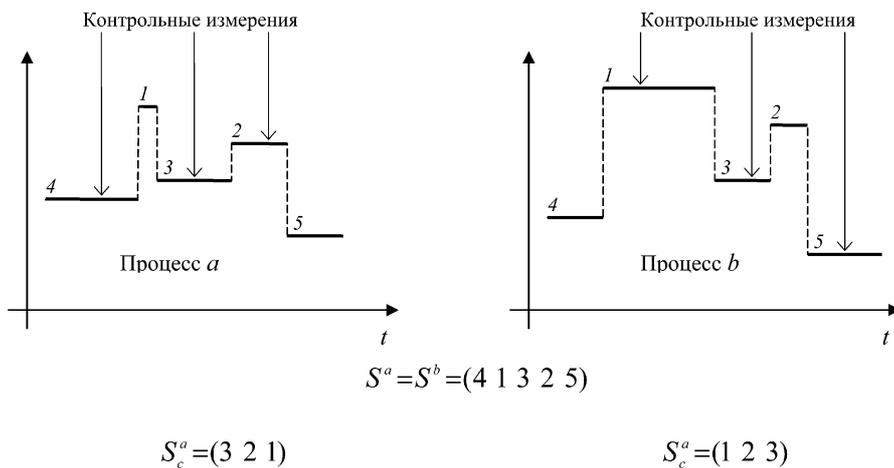


Рис. 1. Описание процесса и результатов контроля структурными параметрами

которую занимает в СП число  $i$ . Номер этой позиции совпадает с номером искомого этапа.

Как отмечалось во введении, контроль процесса осуществляется путем контроля отдельных его этапов. Последовательность этапов, которые были подвергнуты контролю, можно рассматривать как новый процесс с меньшим числом этапов. Такой процесс будем называть подпроцессом исходного процесса. Он таким же способом, как и исходный процесс, может быть описан структурным параметром. Длина СП, описывающего результат контроля, равна числу проведенных измерений. СП, полученный по результатам выборочного контроля, будем называть дочерним структурным параметром (ДСП) для СП, описывающего весь процесс. Таким образом, дочерним структурным параметром для СП, описывающего контролируемый процесс  $G = \{g_1, g_2, \dots, g_n\}$ , является структурный параметр подпроцесса  $H = \{g_{i_1}, g_{i_2}, \dots, g_{i_k}\}$ ,  $k \leq n$ .

СП процесса  $G' = \{g_{i_1}, g_{i_2}, \dots, g_{i_{n-1}}\}$ , содержащего на один этап меньше, чем исходный процесс  $G$ , будем называть *прямым ДСП*.

Каждый подпроцесс  $H$  контролируемого процесса  $G$  описывается некоторым ДСП для СП  $s$ , описывающего процесс  $G$ . Различные подпроцессы  $H = \{g_{i_1}, g_{i_2}, \dots, g_{i_k}\}$ ,  $H_2 = \{g_{j_1}, g_{j_2}, \dots, g_{j_k}\}$  могут описываться одним ДСП. Множество  $U_s$  всех ДСП некоторого СП  $s$  является частично упорядоченным множеством с отношением порядка "быть дочерним". Это отношение обозначим символом  $<$  и будем писать  $s_1 < s_2$ , если СП  $s_1$  является дочерним для  $s_2$ , или  $s_1 = s_2$ . Запись  $s_1 < \cdot s_2$  будет означать, что  $s_1$  является прямым ДСП, для  $s_2$ . Множество всех прямых ДСП для СП  $s$  обозначим через  $D(s)$ . Для наглядности частично упорядоченное множество  $U_s$  удобно изображать в виде диаграммы Хассе [2]. На рис. 2 показана диаграмма Хассе множества  $U_{(3241)}$  (выделена утолщенными линиями) и  $U_{(4312)}$  (выделена штриховыми линиями), соответствующих изображенным на рис. 2 процессам. Эти множества являются подмножествами бесконечного частично упорядоченного множества  $U$  всех СП (перестановок) с отношением порядка  $<$ . Множество  $U$  назовем для краткости структурным универсумом. Оно имеет единственный минимальный элемент  $(\ )$ , но не имеет максимальных. Нижняя часть универсума  $U$  изображена на рис. 2. Универсум  $U$  имеет ярусное строение, каждый ярус состоит из СП одинаковой длины. На рис. 2 составляющие каждый ярус СП расположены в лексикографическом порядке.

СП длины  $n$  могут иметь от 1 до  $n$  прямых ДСП. В работе [1] приведено рекуррентное соотношение, характеризующие распределение всех  $n!$  СП длины  $n$  по числу имеющихся у них ДСП, а также формула для среднего числа прямых ДСП для СП длины  $n$ . Каждая прямая ДСП получается

СП длины  $n$  могут иметь от 1 до  $n$  прямых ДСП. В работе [1] приведено рекуррентное соотношение, характеризующие распределение всех  $n!$  СП длины  $n$  по числу имеющихся у них ДСП, а также формула для среднего числа прямых ДСП для СП длины  $n$ . Каждая прямая ДСП получается

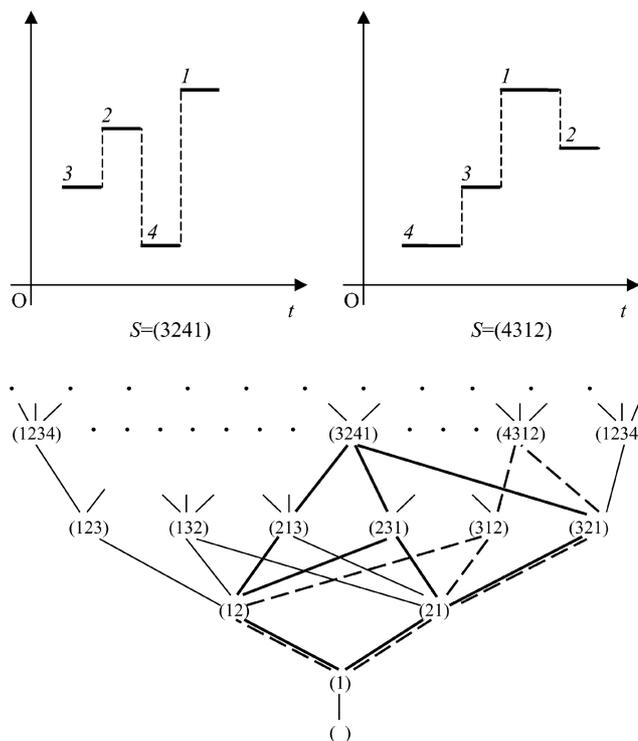


Рис. 2. Процессы и их СП. Структурный универсум

удалением некоторого элемента  $r_i$  из СП  $s$  и уменьшением на единицу всех элементов из  $s$ , превышающих  $r_i$ . Все прямые ДСП для СП  $s$  длины  $n$  получаются повторением данной процедуры для  $i = 1, 2, \dots, n$ . При удалении различных элементов СП могут получиться одинаковые ДСП. Множество всех ДСП длины  $k < n$  получается объединением множеств прямых ДСП для дочерних СП длины  $k + 1 < n$ .

Для контроля процессов наряду с ДСП используются СП длины  $n + 1$ , для которых заданный СП длины  $n$  является прямым ДСП. Такие СП будем называть родительскими СП. Можно показать, что каждый СП длины  $n$  имеет  $n_2 + 1$  родительский СП длины  $n + 1$ . Множество всех таких родительских СП для СП  $s$  обозначим через  $F(s)$ .

### Метод контроля процессов с использованием структурных параметров

Как показано ранее, если реальный процесс  $G = \{g_1, g_2, \dots, g_n\}$ , течение которого контролируется выборочными измерениями параметра, описывается СП  $s$ , то СП  $s_c$ , описывающий результат контроля, является дочерним для СП  $s$ , т. е.

$$s_c \in U_s. \quad (1)$$

Данное включение является необходимым условием правильности процесса, и его проверка составляет основное содержание процедуры контроля процесса с использованием структурных параметров. При нарушении условия (1) контролируемый процесс не соответствует эталонному процессу, заданному СП  $s$ .

Поскольку включение (1) является только необходимым условием, то его выполнение не гарантирует, что контролируемый процесс в точности соответствует эталонному процессу, описываемому СП  $s$ . Гарантия правильности процесса повышается, если условие (1) выполняется не для одного, а для нескольких структурных параметров, описывающих результаты контроля. Различные СП  $s$  и  $s'$  могут иметь общие ДСП. Поэтому, если СП  $s$  соответствует заданному эталонному процессу  $G$ , СП  $s'$  описывает реальный контролируемый процесс  $G'$ , а результат контроля описывается СП  $s_c \in U_s \cup U_{s'} \subset U_s$ , то будет сделан ложный вывод о соответствии контролируемого процесса  $G'$  эталонному  $G$ . Пример такой ситуации показан на рис. 2, где  $s = (3241)$ ,  $s' = (4312)$ ,  $s_c = (321) \subset U_s \cup U_{s'}$ . Ошибки такого типа характерны для задач проверки гипотез и называются ошибками второго рода [3].

Чтобы определить, как часто возможны такие ошибки, необходимо задать некоторую вероятностную модель процедуры контроля процесса. Далее будут использованы следующие предположе-

ния. Длительности этапов являются независимыми, одинаково распределенными случайными величинами  $\tau_i = \tau$ . Время, необходимое для измерения контролируемого параметра на  $i$ -м этапе, является случайной величиной  $v_i$ . Величина  $v_i$  независима и одинаково распределена  $v_i = v$ . Условием успешного измерения контролируемого параметра на  $i$ -м этапе является неравенство  $\tau_i \geq v_i$ , которое выполняется с одинаковой для всех этапов вероятностью  $p = P\{\tau_i - v_i \geq 0\} = P\{\tau \geq v\}$ .

Модель формирования результата контроля аналогична схеме испытаний Бернулли. Каждому этапу  $g_i$  процесса  $G$  соответствует испытание, исходом которого является либо успешное измерение параметра (с вероятностью  $p$ ), либо незавершенное (неудачное) измерение (с вероятностью  $1 - p$ ). Контролю всего процесса соответствует последовательность независимых испытаний. Результатом контроля является СП  $s_c = (r_1, r_2, \dots, r_k)$ , составленный из рангов  $r_i$  успешных измерений.

Булеан  $\mathbf{B}(G)$  множества этапов процесса  $G$  можно рассматривать как конечное вероятностное пространство, в котором каждому  $k$ -элементному подмножеству (элементарному событию)  $H \in \mathbf{B}(G)$  соответствует вероятность  $p^k(1 - p)^{n - k}$  успешных измерений на  $k$  этапах, составляющих  $H$ , из  $n$  этапов процесса  $G$ . В соответствии с предложенным способом описания результатов контроля каждому элементарному событию  $H \in \mathbf{B}(G)$  ставится в соответствие СП, т. е. задано отображение  $\eta: \mathbf{B}(G) \rightarrow U_s$ . Элементарные события, описываемые одним СП  $t$ , составляют событие  $\eta^{-1}(t) \subset \mathbf{B}(G)$ . Вероятность  $P(t|s)$  события  $\eta^{-1}(t)$  при условии, что процесс описывается СП  $s$  есть

$$P(t|s) = m(t, s)p^k(1 - p)^{n - k}, \quad (2)$$

где  $k = |t|$  — длина СП  $t$ ;  $m(t, s) = |\eta^{-1}(t)|$  — число подпроцессов  $H \subset G$ , описываемых СП  $t$ , при условии, что весь процесс описывается СП  $s$ .

При  $t < \cdot s$  величину  $m_{ts} = m(t, s)$  назовем кратностью  $t$  относительно  $s$ . Кратность  $m_{ts}$  представляет собой число ребер диаграммы булеана  $\mathbf{B}(G)$ , соединяющих каждую вершину из  $\eta^{-1}(t)$  с вершинами из  $\eta^{-1}(s)$ . Заметим, что если  $t$  и  $s$  не связаны отношением  $<$ , то  $m(t, s) = 0$ .

Более подробно отображение  $\eta: \mathbf{B}(G) \rightarrow U_s$ , лежащее в основе вероятностной модели контроля процесса, описано в работе [1].

Таким образом, результатом контроля процесса может быть с определенной вероятностью один из дочерних СП для СП, описывающего контролируемый процесс. Рассмотренную модель процедуры контроля (вследствие двух возможных исходов контроля одного этапа) будем называть биномиальной.

## Вероятностные характеристики метода контроля

Вероятностные характеристики предлагаемого метода контроля исследуются при следующих предположениях. Заданный класс процессов описывается СП  $s$ . Результатом наблюдения контролируемого процесса является СП  $s_c$ , описывающий один из его подпроцессов. Решение о принадлежности контролируемого процесса заданному классу принимается при истинности включения  $s_c \in U_s$ . При сделанных предположениях вероятность  $P_2 = P\{U_s|t\}$  выполнения условия  $s_c \in U_s$  в случае, когда реальный процесс описывается СП  $t$  (вероятность ошибки 2-го рода) определяется по формуле

$$P_2 = P\{U_s|t\} = \sum_{u \subset U_s} P(u|t). \quad (3)$$

Условные вероятности  $P(u|t)$  в этом выражении определяются по формуле (2). Основная сложность в определении  $P(u|t)$  заключена в вычислении величины  $m(u, t)$ . Можно показать, что для величин  $m(u, t)$  справедливо соотношение

$$(n - k)m(u, t) = \sum_{q \in F(u)} m_{uq} m(q, t). \quad (4)$$

Умножив (4) на  $p^k(1 - p)^{n - k}$ , получим рекуррентную формулу для условной вероятности

$$P(u|t) = \frac{1 - p}{(n - k)p} \sum_{q \in F(u)} P(q|t) m_{uq}.$$

Эта формула выражает  $P(u|t)$  через кратности  $m_{uq}$  и вероятности  $P(q|t)$ , в которых разность длин СП  $t$  и  $q$  на единицу меньше разности длин СП  $t$  и  $u$ .

Множественно используя формулу (4) для преобразования ее правой части, величину  $m(u, t)$  можно выразить через кратности  $m_{q_1 q_2}, u < q_1 < q_2 < \dots < q_{n-k+1} < t$ , по формуле

$$m(u, t) = \frac{1}{(n - k)!} \sum_{\substack{q_1 \in F(u) \\ q_2 \in F(q_1) \\ \dots \\ q_{n-k+1} \in D(t)}} m_{uq_1} \times \\ \times m_{q_1 q_2} \cdot \dots \cdot m_{q_{n-k+1} t}. \quad (5)$$

Формула (5) имеет комбинаторный смысл. Сумма, стоящая в ней, равна числу максимальных цепей в булеане  $\mathbf{B}(G)$ , соединяющих элемент  $\eta^{-1}(t) = G$  со всеми элементами из  $\eta^{-1}(u)$ .

Формула (2) аналогична формуле для вероятности  $k$  положительных исходов в  $n$  испытаниях в схеме Бернулли. Отличие заключается в том, что вместо биномиального коэффициента  $C_n^k$  в формуле стоит коэффициент  $m(t, s)$ . Если  $C_n^k$  есть число способов выбора  $k$  элементов из  $n$  от-

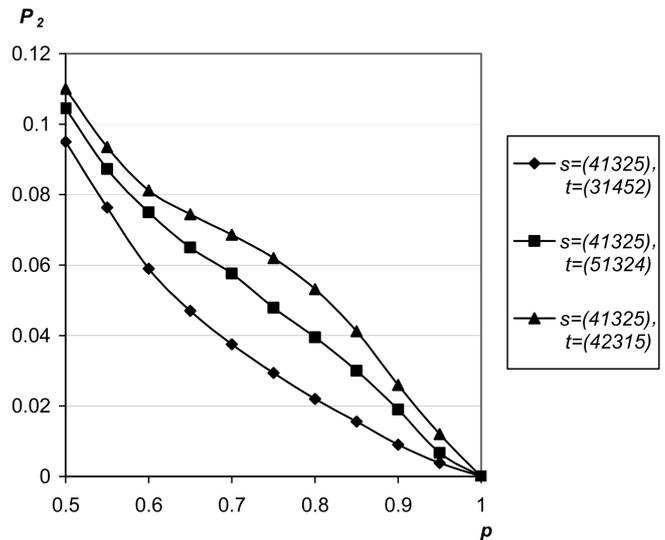


Рис. 3. Вероятности ошибок контроля второго рода

личимых неупорядоченных элементов, то  $m(t, s)$  определяет число способов выбора  $k$  элементов из  $n$  неотличимых, упорядоченных в соответствии с СП  $s$  элементов, так, что выбранные элементы описываются СП  $t$ .

В качестве характеристики качества предлагаемого метода контроля рассмотрим вероятность ошибки второго рода  $P_2 = P\{U_s|t\}$ , которая может быть вычислена по формуле (3). Основная проверяемая гипотеза заключается в том, что наблюдаемый процесс входит в определяемый СП  $s$  класс заданных процессов. Ошибки 2-го рода рассматриваются по отношению к простой альтернативной гипотезе, состоящей в том, что контролируемый процесс описывается СП  $t \neq s$ . Вероятность ошибки зависит от альтернативной гипотезы, характеризуемой СП  $t$ . На рис. 3 представлены зависимости вероятности ошибок 2-го рода  $P_2$  от вероятности  $p = P\{\tau - v \geq 0\}$  осуществления контроля одного этапа процесса для трех различных альтернативных гипотез.

Вероятности ошибок контроля зависят от степени "сходства" структурных параметров, соответствующих основной и альтернативной гипотезам. Чем больше это "сходство", тем больше вероятность ошибки. На рис. 3 показаны зависимости, соответствующие похожим СП  $s$  и  $t$  (отличающимся только перестановкой двух чисел). При более значительных различиях СИ  $s$  и  $t$ , соответствующих основной и альтернативной гипотезам, вероятности ошибок контроля 2-го рода существенно меньше. Исследования показали, что доля пар СП, обладающих большим сходством, уменьшается с ростом числа этапов контролируемых процессов или длин соответствующих СП.

## Заключение

Проведенные исследования показали возможность использования комбинаторного подхода для описания и контроля дискретных процессов некоторых типов. СП процесса позволяет, не имея информации о значениях параметров процесса, компактно описывать важные компоненты динамики процесса. Информации, содержащейся в СП, достаточно для контроля рассмотренных в статье ступенчатых процессов с приемлемыми для ряда приложений вероятностными характеристиками.

Применение СП целесообразно при дистанционном контроле, когда информация о процессе должна передаваться по каналам связи. В этих

случаях использование СП вместо последовательностей значений контролируемых параметров существенно сокращает объем передаваемых о процессе данных, сохраняя при этом необходимую для контроля информацию. Предлагаемый подход к описанию и контролю процессов может использоваться совместно с другими методами.

## Список литературы

1. **Кирсанов А. П.** Комбинаторно-геометрический метод описания и отождествления изображений // Изв. АН СССР. Техническая кибернетика. 1991. № 4. С. 81—90.
2. **Рыбников К. А.** Введение в комбинаторный анализ. М.: Изд-во МГУ, 1985.
3. **Боровков А. А.** Математическая статистика. М.: Наука, 1984.

УДК 519.7

**Х. Г. Асадов**, д-р техн. наук, нач. отдела,  
**Н. А. Набиев**, аспирант,  
Азербайджанское национальное  
аэрокосмическое агентство

## Многокритериальная оптимизация четких и нечетких информационно-измерительных систем нестационарного энергетического состояния

*Рассмотрены вопросы оптимизации информационно-измерительных систем (ИИС) нестационарного энергетического состояния по методу многоцелевой оптимизации. С использованием комбинированного функционала цели, состоящего из линейной комбинации информационных и нечетких функционалов, осуществлена многокритериальная оптимизация данного подкласса ИИС. Выработаны практические рекомендации по построению аналогичных систем.*

**Ключевые слова:** информационно-измерительные системы, оптимизация, критерии оптимизации, нестационарность, многокритериальная оптимизация, нечеткий анализ.

Нестационарность информационно-энергетического состояния является одним из важнейших качественных показателей информационно-измерительных систем (ИИС) и охватывает широкую область конкретных технических приложений [1].

Нестационарность измерительных систем в широком смысле слова может быть рассмотрена

как присутствие измерительной системы в нестационарном энергетическом состоянии. При этом нестационарная энергия, "мешающая" измерениям, может присутствовать как внутри измерительной системы, так и во внешней среде измерителя. Закономерность изменения нестационарной энергии, влияющая на результат измерения, может быть систематической и случайной. Систематическая нестационарность ИИС исследована достаточно подробно (см. например, [2]). Случайная нестационарность ИИС, т. е. работа систем в условиях аддитивных и мультипликативных шумов применительно к энергетическому параметру системы, обладающая рядом замечательных свойств, исследована, на наш взгляд, менее подробно в плане решения оптимизационных задач.

Случайные факторы, воздействующие на систему изнутри и извне, изучаются методами четкого и нечеткого анализа. Аппарат нечеткого анализа применяется для выработки оптимальных решений в будущем, в то время как классический статистический анализ — аппарат теории вероятностей и теории информации — применительно к нестационарным системам скрупулезно анализирует и учитывает предысторию изучаемого явления и, что особенно важно, дает рекомендации, направленные на улучшение "прошлого опыта".

С учетом такого качественного различия между указанными двумя подходами к обработке и анализу нестационарных измерительных систем естественным первичным методом исследования оптимальных режимов работы таких систем должен стать аппарат теории случайных погрешностей, позволяющий создание "компенсирующего" воздействия при измерениях, заключающегося

в том, что в "классических четких" системах эквивалентная результирующая средняя квадратическая погрешность серии из  $n$  измерений оказывается зависимой от  $n$ , т. е.

$$\sigma_{\text{ЭКВ}} = \sqrt{\frac{\sum_{i=1}^n \sigma_i^2}{n}}. \quad (1)$$

Если все существующие в тракте измерения помехи случайного характера приписать к случайной погрешности измерителя, то выражение (1) будет означать, что случайная погрешность может быть уменьшена в сколь угодно раз. Таким образом, всякая нестационарность, (систематическая, случайная) приводящая в конечном счете к уменьшению отношения сигнал/шум в измеряемом сигнале, имеет мощное противодействие в виде фундаментальной возможности "расчищения сигнала" путем многократных измерений.

Посмотрим, к каким последствиям приводит наличие такого компенсирующего механизма. Для дальнейшего описания примем следующие ограничительные условия применительно к множеству групповых измерений определенного набора амплитуды сигнала, находящегося в пределах  $U_i < U_t < U_n$ ;  $i = \overline{1, n}$ , где каждая амплитуда  $U_i$  измеряется  $d_i$  раз при ограничительном условии

$$\sum_{i=1}^n d_i = \text{const}. \quad (2)$$

Здесь условие (2) физически отражает ограниченность ресурса на общее число измерений величин  $U_i$ ,  $i = \overline{1, n}$ .

Ограничение на фаззи-ресурс измерительной системы имеет следующий вид:

$$\mu_{\text{инт}} = \int_{U_{\text{imin}}}^{U_{\text{imax}}} \mu(U) dU = \text{const}. \quad (3)$$

Условие (3) задает ограниченность фаззи-ресурса системы измерений. Физически это условие является отражением закономерности (1), т. е. чем больше  $d_i$  (число измерений параметра  $U_i$ ), тем уже оказывается интервал определения  $\mu(U)$ ,  $U_1 \div U_n$ , и тем больше величина  $\mu(U_i)_{\text{max}}$ . Такая трансформация  $\mu(U_i)$  в зависимости от  $n_i$  показана на рис. 1.

Как видно на рис. 1, максимальное значение функции принадлежности оказывается зависимым от числа измерений  $d_i$ , где  $d_{i_1} < d_{i_2} < d_{i_3}$ .

Покажем, что для фаззи-систем, функции принадлежности которых удовлетворяют ограничению

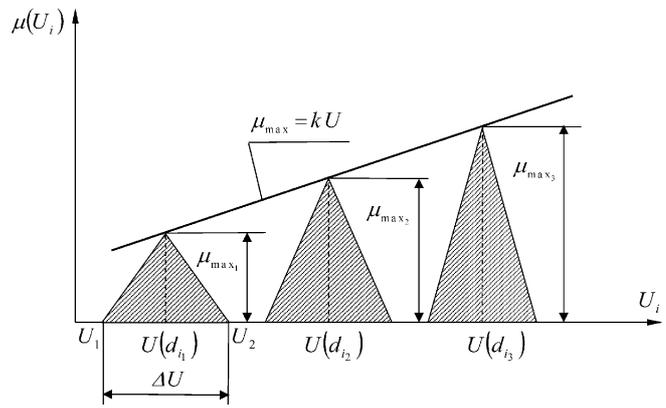


Рис. 1. Трансформация  $\mu(U_i)$  при нестационарности величины  $d$

(3), указанное ограничивающее условие эквивалентно следующему ограничительному условию:

$$\int_0^{H_{\text{max}}} \mu_{\text{max}}(H) dH = \text{const}. \quad (4)$$

Для доказательства (4) с учетом (3) напомним (рис. 1)

$$\int_{U_1}^{U_2} \mu(U) dU = C = \frac{\mu_{\text{max}_1} \Delta U}{2}. \quad (5)$$

Из (5) имеем

$$\mu_{\text{max}} = \frac{2C}{\Delta U}. \quad (6)$$

Допустим, что

$$\mu_{\text{max}} = kU, \quad (7)$$

т. е.  $\mu_{\text{max}}$  является линейной функцией  $U$  (рис. 1).

С учетом (6) и (7) имеем

$$kU = \frac{2C}{\Delta U}$$

или

$$\Delta U = \frac{2C}{kU}. \quad (8)$$

В этом случае с учетом (6) и (8) имеем:

$$\int_0^{U_m} \mu_{\text{max}}(U) dU = \frac{kU_m^2}{2},$$

т. е. ограничение (4) справедливо.

Следовательно, в качестве критерия оптимизации может быть выбрана центрированная суммарная логарифмическая функция, используемая для дефаззификации фаззи-обработанных данных с ограничением типа (3):

$$\Phi = \frac{\sum_{i=1}^n U_i \ln[\mu_{\text{max}}(U_i)]}{\sum \mu_{\text{max}}(U_i)}. \quad (9)$$

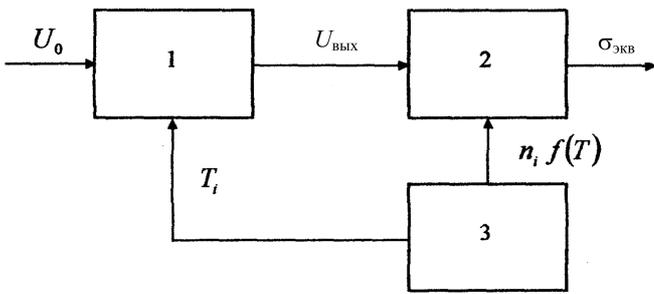


Рис. 2. Общая структурная схема нестационарно-четких ИИС: 1 — объект с нестационарным энергетическим состоянием; 2 — статистический измеритель; 3 — устройство управления

Пример решения подобного типа задач проанализирован в [3] и здесь не приводится.

Подробно рассмотрим две базовые задачи оптимизации применительно к нестационарно-четким ИИС. Общая структурная схема измерений показана на рис. 2.

Будем полагать, что объект выдает на выход сигнал

$$U_{\text{вых}} = U_0 \varphi(T), \quad (10)$$

где  $U_0$  — измеряемая величина;  $\varphi(T)$  — весовая функция, отображающая энергетическую нестационарность системы;  $T$  — параметр нестационарности.

С учетом (10) отношение сигнал/шум на выходе блока 2 определим как

$$\begin{aligned} H(T) &= \frac{U_0 \varphi(T)}{\sigma_{\Sigma}} = \frac{U_0 \sqrt{n} \varphi(T)}{\sigma_{\Sigma}} = \\ &= \sqrt{\frac{U_0 f(T)}{\sigma_{\Sigma}}} \sqrt{\frac{U_0}{\sigma_{\Sigma}}} \varphi(T), \end{aligned} \quad (11)$$

где  $n = f(T)$ ;  $\sigma_{\Sigma}$  — суммарное среднее квадратичное отклонение.

Наложив на функцию  $\varphi(T)$  и  $f(T)$  ограничения

$$\int_0^{T_{\max}} \varphi(T) dT = C_1; \quad (12)$$

$$\int_0^{T_{\max}} f(T) dT = C_2, \quad (13)$$

а также выразив основной информационный функционал оптимизации в интегральной форме

$$H_n(T) = \int_0^{T_{\max}} T \ln \left[ \sqrt{\frac{U_0 f(T)}{\sigma_{\Sigma}}} \sqrt{\frac{U_0}{\sigma_{\Sigma}}} \right] dT, \quad (14)$$

можно рассматривать следующие функционалы для оптимизации:

$$H_1(T) = \frac{H(T)}{C_1};$$

$$H_2(T) = \frac{H(T)}{C_2}.$$

Как отмечено в [3], одной из процедур решения многокритериальных задач является метод Гермейера. При этом необходимым условием применения многокритериальных процедур оптимизации является наличие конкурирующих целей, когда одну из целей можно достигнуть лишь за счет другой.

Концептуально всякие разновидности информационных критериев, включая (14), а также фаззи-критерии оптимального выбора функций принадлежности дефаззификаторов (9) должны быть отнесены к конкурирующим целям в дуальных системах, где основной параметр системы может быть рассмотрен двояко: как детерминированная функция некоторого аргумента, а также как фаззи-величина ввиду воздействия на нее большого числа различных факторов. Задача оптимизации таких дуальных систем решается методом многоцелевой оптимизации [3].

Вначале рассмотрим решение поставленной задачи в общей постановке. Во избежание получения громоздких формул рассмотрим случай, когда обобщенный многокритериальный функционал цели оптимизируется согласно процедуре Гермейера, т. е. может быть отображен в виде линейной суммы двух частных критериев  $F_1$  и  $F_2$ :

$$F_0 = \alpha_1 F_1 + \alpha_2 F_2, \quad (15)$$

где  $\alpha_1, \alpha_2$  — положительные коэффициенты, устанавливаемые на основе экспертных оценок.

Для конкретной реализации решения задачи многокритериальной оптимизации применительно к сканерным системам, размещенным на носителях с нестационарной траекторией полета (рис. 3), воспользуемся результатами, полученными в [4].

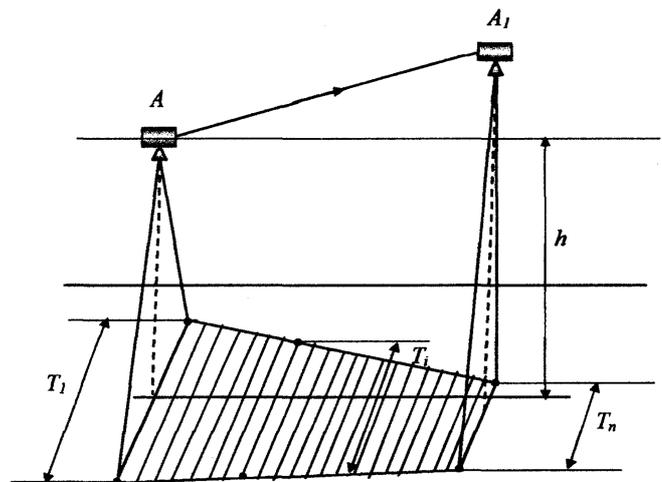


Рис. 3. Сканирующий фотометр  $A$ , установленный на носителе,двигающемся по нестационарной траектории  $AA_1$

В соответствии с [4], где была осуществлена информационная оптимизация работы сканерных систем дистанционного зондирования на борту носителя с нестационарной траекторией полета (рис. 3), критерий информационной оптимальности может быть выбран в следующем виде:

$$F_1 = \frac{\int_0^{T_m} k_1 T \log_2 H(T) dT}{\int_0^{T_m} H(T) dT}. \quad (16)$$

Как нетрудно видеть, критерий (16) подобен полученному критерию (9) с учетом ограничения (4).

Для решения задачи фаззи-оптимальности используем критерий оптимизации, подобный критерию, использованному в [4]:

$$F_2 = \frac{\int_0^{H_{\max}} H \ln[\mu_{\max}(H)] dH}{\int_0^{H_{\max}} \mu_{\max}(H) dH}. \quad (17)$$

Если искомой оптимальной функцией является  $f(T)$ , то с учетом ранее полученных выражений (11) и (13), а также (16) имеем:

$$F_{11} = \frac{H(T)}{C_2} = \frac{\int_0^{T_m} T \ln[A_1 \sqrt{f(T)}] dT}{\int_0^{T_m} f(T) dT}, \quad (18)$$

где  $A_1 = \varphi(T) \frac{U_0}{\sigma_\Sigma}$ .

Если искомой оптимальной функцией является  $\varphi(T)$ , то с учетом выражений (11) и (12), а также (16) получаем

$$F_{12} = \frac{H(T)}{C_1} = \frac{\int_0^{T_m} T \ln[A_2 \varphi(T)] dT}{\int_0^{T_m} \varphi(T) dT}, \quad (19)$$

где  $A_2 = \frac{U_0 \sqrt{f(T)}}{\sigma_\Sigma}$ .

Приняв в первом приближении линейный вид функций  $\mu_{\max}(H)$  и  $H(f)$ , имеем

$$\mu(H) = \mu'_H H[f(T)] = \mu'_H H_f f(T). \quad (20)$$

Аналогично (20) получаем

$$\mu(H) = \mu'_H H[\varphi(T)] = \mu'_H H_\varphi \varphi(T). \quad (21)$$

Функционал цели оптимального выбора функции принадлежности дефазификатора имеет вид [4]

$$F_2 = \frac{\int_0^{T_m} H \ln \mu_{\max}(H) dT}{\int_0^{T_m} \mu_{\max}(H) dT}. \quad (22)$$

С учетом (22), (20), (21) соответственно получаем

$$F_{21} = \frac{\int_0^{T_m} \ln \mu'_f f(T) dT}{\int_0^{T_m} \mu'_f f(T) dT}; \quad (23)$$

$$F_{22} = \frac{\int_0^{T_m} \ln \mu'_\varphi \varphi(T) dT}{\int_0^{T_m} \mu'_\varphi \varphi(T) dT}. \quad (24)$$

Составим полные функционалы, подлежащие оптимизации. С учетом (15), (18), (23) для случая поиска оптимальной функции  $f(T)$  имеем

$$F_{01} = \frac{\int_0^{T_m} \alpha_1 T \ln[A_1 \sqrt{f(T)}] dT}{\int_0^{T_m} f(T) dT} + \frac{\int_0^{T_m} \alpha_2 \ln[\mu'_f f(T)] f(T) dT}{\int_0^{T_m} \mu'_f f(T) dT}. \quad (25)$$

С учетом (15), (19) и (24) для случая поиска оптимальной функции  $\varphi(T)$  имеем

$$F_{02} = \frac{\int_0^{T_m} \alpha_1 T \ln[A_2 \varphi(T)] dT}{\int_0^{T_m} \varphi(T) dT} + \frac{\int_0^{T_m} \ln[\mu'_\varphi \varphi(T)] \varphi(T) dT}{\int_0^{T_m} \mu'_\varphi \varphi(T) dT}. \quad (26)$$

Дифференцируя (25) и (25) соответственно по  $f(T)$  и  $\varphi(T)$ , можно получить следующие уравнения:

$$\frac{\mu' \alpha_1 T}{f(T) \ln 2} + \alpha_2 \{1 + \ln[\mu' f(T)]\} = 0; \quad (27)$$

$$\frac{\mu' \alpha_1 T}{\varphi(T) \ln 2} + \alpha_2 \{1 + \ln[\mu' \varphi(T)]\} = 0. \quad (28)$$

Приняв  $Y(t) = \ln f(T)$  для уравнения (27) и  $Y(T) = \ln \varphi(T)$  для уравнения (28), уравнения (27) и (28) можно представить в виде следующего неоднородного дифференциального уравнения:

$$\frac{dY}{dT} + P_1(T)Y = P_2(T), \quad (29)$$

где

$$P_1(T) = \frac{\alpha_2 \ln 2}{\mu' \alpha_1 T}; \quad (30)$$

$$P_2(T) = \frac{\alpha_2 (1 + \ln \mu') \ln 2}{\alpha_1 \mu' T}. \quad (31)$$

Для дальнейшего упрощения выкладок примем  $1 + \ln \mu' = 0$ , т. е.  $\ln \mu' = -1$ ;  $\mu' = \frac{1}{e}$ .

В этом случае уравнение (29) превращается в линейное однородное уравнение

$$\frac{dY}{dT} + P_1(T)Y = 0. \quad (32)$$

Решение уравнение (32) имеет вид [5]:

$$Y = C e^{-P_1(T) dT}. \quad (33)$$

С учетом (33) для  $f(T)$  и  $\varphi(T)$  находим оптимальное выражение единого вида:

$$f(T) = \exp \left[ C \exp \left( -\frac{\alpha_2 \ln 2 \ln T}{\mu' \alpha_1} \right) \right]. \quad (34)$$

Как видно из выражения (34), в нестационарных системах рассматриваемого подкласса с увеличением нестационарного параметра  $T$  величина  $f(T)$  уменьшается.

Аналогичным образом также можно показать, что в нестационарных системах при оптимизации

по многоцелевой процедуре увеличение нестационарного параметра ( $T$ ) должно сопровождаться с уменьшением величины  $\varphi(T)$ . Следует отметить, что полученный результат качественно отличается от результата, приведенного в [4], где была осуществлена однокритериальная оптимизация по энергоинформационному критерию.

На основе полученных результатов конкретно для сканерной системы, размещенной на носителе, которая движется по нестационарной траектории, можно выработать следующие практические рекомендации по функционированию:

- многократные измерения в целях уменьшения влияния шумов целесообразно осуществлять при малом значении  $T$ , а не при большом;
- большие затухания в среде прохождения сигнала должны соответствовать малому значению  $T$ , и наоборот.

Аналогичные практические рекомендации могут быть выработаны и для других подобных нестационарных систем, что указывает на универсальность полученных результатов.

В заключение отметим, что правильный выбор соответствующей процедуры оптимизации существенно важно для нестационарных информационных систем, так как было показано, что переход от однокритериальной процедуры оптимизации к многокритериальной приводит к противоположному качественному результату в плане взаимосвязи основных функциональных параметров системы.

#### Список литературы

1. Романченко А. Ф. Информационно-измерительные системы нестационарного энергетического состояния. Уфа: Урал, 2000. 173 с.
2. Асадов Х. Г., Керимов М. Дж. Об оптимизации нестационарного режима работы систем дистанционного зондирования на примере теплового контроля энергоэффективности зданий и сооружений // Контроль, диагностика. 2006. № 11. С. 33–35.
3. Многоцелевая оптимизация. Принятие оптимальных многоцелевых решений. [http://web.petrstu.ru/~forest/courses/decision/chap7\\_a.htm](http://web.petrstu.ru/~forest/courses/decision/chap7_a.htm)
4. Асадов Х. Г., Керимов М. Дж. Транзитивно-фаззи-принцип оптимизации информационных систем. Применение к системам дистанционного зондирования // Авиакосмическое приборостроение. 2006. № 11. С. 26–29.
5. Эскольц Э. Дифференциальные уравнения в вариационное исчисление. М.: Наука, 1965.

УДК 519.8+531+616.314-089.29-633

**Е. Н. Чумаченко**<sup>1</sup>, д-р техн. наук, проф.,  
**В. Р. Пашмурина**<sup>2</sup>, канд. мед. наук, доц.

<sup>1</sup> Московский государственный институт  
электроники и математики  
(технический университет),

<sup>2</sup> Московский государственный  
медико-стоматологический университет  
РОСЗДРАВА

## Прогнозирование взаимодействия структур костной ткани беззубой нижней челюсти с полными съемными протезами

*Обсуждается математическая модель биомеханической системы "съемный протез—имплантат—кость". Анализируется влияние геометрических параметров протеза и физического состояния костных тканей на напряженно-деформированное состояние системы. Устанавливается соответствие между числом имплантатов, на которые опирается протез, и прочностными характеристиками системы. Разработанная компьютерная система имитационного моделирования позволяет изучить биомеханические основы взаимодействия структур костной ткани нижней челюсти с полными съемными протезами различной конструкции и фиксирующими их дентальными внутрикостными имплантатами.*

**Ключевые слова:** биомеханика, математическое моделирование, зубные протезы, имплантаты, прочность, костные ткани челюсти.

### Введение

Реабилитация пациентов с полным отсутствием зубов является актуальной и до сих пор нерешенной проблемой [1]. Принято выделять три направления профилактики непереносимости зубных протезов: совершенствование используемых конструкционных материалов и клинико-лабораторных этапов изготовления протезов, профилактика прогрессирующей атрофии тканей протезного ложа, применение хирургических способов подготовки протезного ложа перед протезированием. Все эти направления требуют дальнейшей и более тщательной разработки.

Наибольшие трудности возникают при лечении пациентов с полным отсутствием зубов на нижней челюсти, особенно при неблагоприятных условиях полости рта и безуспешной психоэмоциональной адаптации к съемным протезам. Одним из эффективных методов

компенсации нарушенных функций у таких пациентов является дентальная имплантация [2—4]. Весомым фактором, определяющим противопоказания к установке имплантатов, является количественная и качественная перестройка костной ткани после удаления всех зубов на челюсти. Поэтому использование большого числа имплантатов для последующего протезирования несъемными конструкциями бывает затруднительно при наличии биомеханических проблем, связанных с атрофией челюсти, ее низкой минеральной плотностью, противопоказаниями к костной пластике. В связи с этим нередко имеются основания для установки небольшого числа имплантатов для улучшения фиксации съемных протезов. Такой подход имеет ряд преимуществ по сравнению с полными съемными пластиночными протезами. Он позволяет улучшить стабилизацию и равновесие протеза, перевести полный съемный протез из разряда неопирающегося в опирающийся, осуществить постановку зубов при нефизиологических вариантах соотношения челюстей, увеличить срок функционирования съемного протеза за счет укрепления его базиса фиксирующими матрицами аттачменов, замедлить резорбцию альвеолярного гребня; улучшить внешний вид большого за счет поддержки базисом мягких тканей лица [5, 6].

Обоснование выбора имплантатов и конструкции протеза зависит от состояния протезного ложа, других общих и местных факторов [4, 7, 8]. Однако приводимые в литературе данные о преимуществе и показаниях к тем или иным конструкциям фиксирующих систем, способах соединения супра- и мезоструктуры, выборе необходимого числа устанавливаемых имплантатов и оптимального расстояния между ними весьма противоречивы [9, 10]. До настоящего времени отсутствуют обоснованные рекомендации по планированию хирургического и протетического этапов лечения данной категории пациентов в зависимости от степени и вида атрофии, размеров и плотности кости нижней челюсти. Такой индивидуальный подход, обеспечивающий биомеханическую совместимость имплантатной системы и кости, повышение надежности фиксации и долговечного функционирования протезов не может быть осуществлен без выявления закономерностей напряженно-деформированного состояния тканей, окружающих дентальный имплантат и элементы съемного протеза, безопасного уровня нагрузки для ремоделирования кости.

Для решения задач биомеханического обоснования конструирования пластиночных протезов, перекрывающих внутрикостные имплантаты, используют разнообразные математические и физические модели с последующим тензометрическим, голографическим, поляризационно-оптическим и математическим анализом. Наиболее перспективным и доступным является математическое моделирование и конечно-элементный анализ [11—13].

### Материалы и постановка задачи

При создании математической модели следует учесть следующие связанные между собой параметры: размер, направление и неоднородность силового воздействия;

конструкцию и материал зубного протеза; форму и материал зубного имплантата; способ тканевой интеграции в области имплантата; качество и количество кости в области имплантации [14].

Биомеханическое исследование предполагает моделирование механических свойств костных тканей. Что касается дентина и кортикальной кости, то они изучены более подробно. Достоверность данных о плотности и достаточной независимости механических свойств дентина и кортикальной кости от индивидуальных особенностей человека подтверждена в рамках популяционных исследований. Эти данные неоднократно использовались при расчетах различных биомеханических конструкций минерализованных тканей [12, 15, 16]. Проблемы возникают при определении упругих механических свойств губчатой кости. Дело в том, что ее пористость носит индивидуальный характер и может существенно зависеть от состояния здоровья пациента. Следует отметить, если плотность кости выше, чем у здорового среднестатистического пациента из рассматриваемой группы, то это не приводит к ослаблению конструкции и не ограничивает допустимые для биологической системы нагрузки. Увеличение же пористости при соответствующем уменьшении плотности губчатой кости такие ограничения, несомненно, накладывает.

В работе [16] было установлено соответствие между упругими характеристиками губчатой кости и ее плотностью. За базовую плотность ( $\rho_0$ ) была принята плотность, соответствующая плотности губчатой кости у здорового среднестатистического пациента. Для этой плотности считались известными модуль Юнга  $E$  и коэффициент Пуассона  $\nu$ , а для установления общей связи  $E(\rho)$ ,  $\nu(\rho)$  воспользовались приемами механики композитных материалов [17, 18].

В результате было установлено, что при изменении относительной плотности  $\rho$  (действительная плотность  $\rho_i$ , отнесенная к начальной плотности  $\rho_0$ ) от 0,3 до 1,0, графики зависимости модуля Юнга  $E$ , предела упругости  $\sigma_s$  на сжатие и модуля сдвига  $G$  от плотности  $\rho$  имеют вид, показанный на рис. 1.

Механические свойства костных тканей, материалов имплантатов и протезов в рассматриваемой модели могут быть аппроксимированы следующим набором констант теории упругости (табл. 1).

Мы изучали характеристики биомеханической системы при условии равномерной атрофии беззубой нижней челюсти. В основу построения расчетных схем заложена классификация нижней челюсти по ее количественным и качественным характеристикам [6], соответствующие им единицы шкалы Хаунсфилда [19], физические свойства опорных тканей челюсти [11]. Все рассматриваемые модели нижней челюсти были классифицированы по признакам атрофии и плотности губчатой костной ткани. Таким образом, за основные были приняты девять состояний нижней челюсти — для трех различных значений плотности костной ткани  $\rho_1, \rho_2, \rho_3$  и для трех соответствующих степеней атрофии  $A, B, C$  [6]. Четвертая степень атрофии  $D$  нами не рассматривалась, так как является противопоказанием к имплантации.

В каждой из локально однородных односвязных подобластей (I) неоднородного многосвязного фрагмента костной ткани челюсти с имплантатом и протезом, должны выполняться:

- уравнения равновесия  $\sigma_{ij,j}^I = 0$ ; (1)

- экспериментально найденные соотношения  $\sigma_u^I = F^I(\epsilon_u)$ , (2)

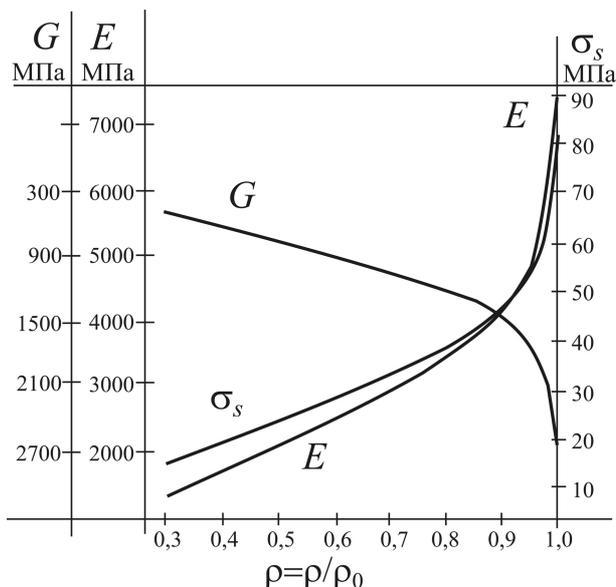


Рис. 1. Гипотетическая зависимость упругих модулей губчатой кости от ее относительной плотности

Таблица 1

Физические характеристики материалов модели

№ пп	Материал имплантата, протеза или костной структуры	$E_s$ , МПа	$\nu$	$\sigma_s^c$ (сжатие), МПа	$\sigma_s^p$ (растяжение), МПа
1.	Кортикальная кость	16 100	0,25	148	45
2.	Губчатая кость				
	$\rho(\rho_1) = 1,0$	7500	0,45	82	15
	$\rho(\rho_2) = 0,7$	2890	0,44	32,7	8
	$\rho(\rho_3) = 0,4$	1730	0,44	19	6
3.	Слизистая оболочка	7	0,45	10 000	10 000
4.	КХС	20 000	0,3	190	189
5.	Титановый сплав	150 000	0,34	840	600
6.	Пластмасса	3000	0,33	99	30
7.	Эластичная матрица	7	0,45	100 000	10 000

характеризующие физические инвариантные зависимости между интенсивностью напряжений и деформаций в костных тканях челюсти и искусственных включениях;

- соотношения, связывающие компоненты тензора напряжений и деформаций,

$$\sigma_{ij}^I = K^I \Theta^I \delta_{ij} + \frac{2\sigma_u^I}{3\epsilon_u^I} (\epsilon_{ij}^I - \epsilon^I \delta_{ij}); \quad (3)$$

- соотношения Коши

$$\epsilon_{ij} = \frac{u_{i,j} + u_{j,i}}{2}, \quad (4)$$

где средние деформации равны  $\epsilon = \frac{1}{3} \epsilon_{ij} \delta_{ij} = \frac{\Theta}{3}$ , а интен-

сивность деформаций  $\epsilon_u = \sqrt{2/3} \sqrt{(\epsilon_{ij} - \epsilon \delta_{ij})(\epsilon_{ij} - \epsilon \delta_{ij})}$ ,  $K$  — коэффициент объемного сжатия;  $\delta_{ij}$  — дельта Кронекера;  $\sigma_{ij}$  — компоненты тензора напряжений;  $\epsilon_{ij}$  — компоненты тензора деформаций;  $\sigma_u$  — интенсивность напряжений;  $u_i$  — компоненты вектора перемещений.

В каждой точке смежных границ соседних подобластей  $I$  и  $J$  принимаются специальные условия для функций перемещения

$$u_i^I(x_j) = u_i^J(x_j), \quad (5)$$

обеспечивающие жесткое соединение костных тканей между собой и непрерывность поля деформаций при приложении нагрузки к моделируемому фрагменту челюсти.

В силу того, что в костных тканях челюсти при приложении жевательной нагрузки недопустимо возникновение пластических деформаций, так как это может привести к нарушению целостности костных структур и необратимому изменению биомеханических параметров челюсти, механические свойства костных тканей (2), аппроксимированы упругопластической средой с линейным упрочнением (рис. 2), где утроенное значение модуля сдвига характеризует тангенс угла наклона линейного участка, а  $3G_1$  — тангенс угла наклона линейного упрочнения.

Модуль Юнга  $E$  связан с модулем сдвига  $G$ , коэффициентом объемного сжатия  $K$  и коэффициентом Пуассона  $\nu$  соотношениями

$$\frac{E}{G} = 2(1 + \nu), \quad \frac{E}{K} = 3(1 - 2\nu). \quad (6)$$

При расчете предельных состояний в костных тканях зубочелюстной системы, имплантатах и протезах в условиях функциональной и предельной системы внешних воздействий используются различные критерии, основанные на той или иной теории прочности.

Проблема рационального выбора критерия сводится к определению некоторой функции  $F(\sigma_1, \sigma_2, \sigma_3)$ , если значения главных напряжений соответствуют состоянию, предшествующему разрушению, то говорят об условии разрушения

$$F(\sigma_1, \sigma_2, \sigma_3) \leq k_{пр}, \quad (7)$$

$k_{пр}$  — критерий прочности. Эта величина обычно имеет определенную физическую интерпретацию: нормальное или касательное напряжение, интенсивность напряжений, максимальное удлинение, энергия формоизменения и т. д. Также необходимо учитывать, что костные ткани различным образом сопротивляются сжатию и растяжению, причем прочность на сжатие выше прочности на растяжение. Это влияние можно учесть, например, если принять, что значение критического касательного напряжения зависит от значения критического нормального напряжения, действующего в той же плоскости. На этом основывается теория разрушения Мора, которая не только позволяет охарактеризовать напряженное состояние при разрушении, но и предсказать ориентацию плоскости разрушения. Но само разрушение, при изучении биомеханической естественно-искусственной системы, не является главным. Нас в значительно большей степени интересует вероятный запас прочности исследуемой био-

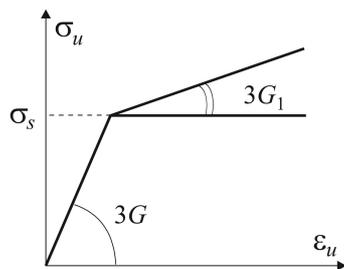


Рис. 2. Вариант модели физических соотношений в костных тканях зуба и челюсти

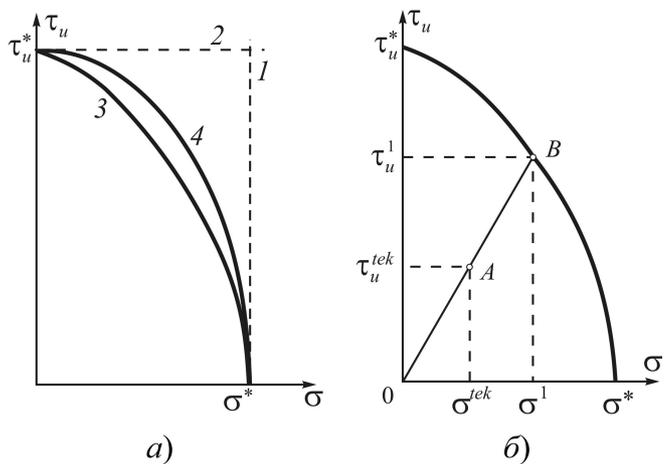


Рис. 3. Предельные кривые теории разрушения Шлейхера—Надаи (а) и расчетная схема для определения вероятности разрушения конструкции (б)

механической конструкции и связанные с этим вопросы выбора того или иного метода терапевтического или ортопедического лечения. В этом смысле более удобна теория разрушения Шлейхера—Надаи [12]. В известном смысле она аналогична теории Мора, но формулируется в терминах интенсивности касательных напряжений  $\tau_u$  и

средних напряжений  $\sigma = \frac{1}{3}(\sigma_{11} + \sigma_{22} + \sigma_{33})$ . В опасном состоянии интенсивность касательных напряжений является функцией средних напряжений, характерной для данного материала:

$$\tau_u = f(\sigma). \quad (8)$$

На плоскости  $\tau_u, \sigma$  уравнение (8) определяет некоторую кривую — границу разрушения (рис. 3, а). Штриховая прямая 1 соответствует условию наибольшей интенсивности касательных напряжений ("сдвиговое разрушение"); прямая 2 — условию наибольшего объемного растяжения ("хрупкое разрушение"). Общий случай характеризуется некоторыми кривыми 3 или 4. В зависимости от напряженно-деформированного состояния один и тот же материал может разрушаться хрупко и вязко. Теория Шлейхера—Надаи позволяет учесть двойственность характера разрушения и построить эффективные вероятностные оценки. Одной из приемлемых аппроксимаций кривой Шлейхера—Надаи является парабола, симметричная относительно оси  $O\tau_u$  и проходящая через критические точки  $(0, \tau_u^*)$  и  $(\sigma^*, 0)$ , где  $\tau_u^*$  — максимальное значение интенсивности сдвига, а  $\sigma^*$  — максимальное значение средних растягивающих напряжений. При значениях  $\sigma^* < 0$ , оценка запаса прочности осуществляется по критерию только сдвигового разрушения.

Так как значения  $\sigma_s^c$  и  $\sigma_s^p$  получены из опытов на одноосное растяжение-сжатие, то можно принять, что

$$\tau_u^* = \frac{1}{\sqrt{3}} \sigma_u^* = \frac{1}{\sqrt{3}} \sigma_s^c, \quad \text{а } \sigma^* = \frac{1}{3} \sigma_s^p.$$

Вероятность разрушения  $V_p$  и запас прочности  $V_{з.п}$  в окрестности произвольной точки рассматриваемой костной ткани с учетом индивидуальных особенностей всей конструкции и условий ее нагружения могут быть

Оценка запаса прочности зубочелюстной биомеханической конструкции

Относительная плотность губчатой кости						
	Приложение нагрузки					
	Центр	Край	Центр	Край	Центр	Край
$\rho = 1,0$	0,478	0,776	0,372	0,763	0,335	0,797
$\rho = 0,7$	0,467	0,742	0,369	0,745	0,334	0,796
$\rho = 0,4$	0,406	0,678	0,361	0,687	0,334	0,757

Относительная плотность губчатой кости						
	Приложение нагрузки					
	Центр	Край	Центр	Край	Центр	Край
$\rho = 1,0$	0,834	0,778	0,836	0,768	0,837	0,773
$\rho = 0,7$	0,790	0,776	0,828	0,767	0,744	0,772
$\rho = 0,4$	0,642	0,719	0,733	0,739	0,569	0,771

Относительная плотность губчатой кости						
	Приложение нагрузки					
	Центр	Край	Центр	Край	Центр	Край
$\rho = 1,0$	0,868	0,778	0,865	0,763	0,865	0,766
$\rho = 0,7$	0,821	0,776	0,845	0,762	0,861	0,765
$\rho = 0,4$	0,705	0,728	0,716	0,751	0,761	0,764

получены после расчета напряженно-деформированного состояния, в виде соответствующих отношений:

$$V_p = L^{tek}/L^*; V_{з.п} = 1 - V_p, \quad (9)$$

где  $(L^{tek})^2 = (\sigma^{tek})^2 + (\tau_u^{tek})^2$ , а  $L^*$  определяется как расстояние (по линии, проходящей через точку  $(\sigma^{tek}, \tau_u^{tek})$ ) от начала координат в системе осей  $(\sigma, \tau_u)$  до кривой  $\tau_u = f(\sigma)$ , имеющей следующий вид:

$$\tau_u = -\frac{\tau_u^*}{(\sigma^*)^2} \sigma^2 + \tau_u^* \text{ при } \sigma > 0. \quad (10)$$

Как уже упоминалось ранее, в случае, если  $\sigma \leq 0$ , полагаем  $L^* = \tau_u^*$ .

На рис. 3, б значение  $L^{tek}$  интерпретируется как отрезок  $OA$ ,  $L^*$  соответствует отрезку  $OB$ . Точка  $B$ , пересечение луча  $OA$  и кривой Шлейхера—Надаи, имеет координаты  $(\sigma^1, \tau_u^1)$ .

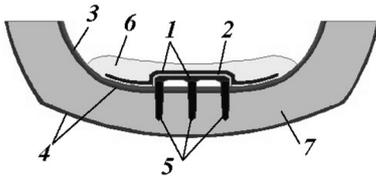
Решение поставленной задачи осуществлялось с помощью метода конечных элементов [12]. Построенная модель биомеханического взаимодействия костных тканей челюсти и искусственных включений реализована с помощью базового вычислительного комплекса SPLEN-K, разработанного

фирмой КОММЕК Лтд. ([www.kommek.ru](http://www.kommek.ru)), объединяющей ученых ряда ведущих технических вузов Москвы.

Разработанная модель позволяет проводить расчеты при переменных входных параметрах: плотности и атрофии костной ткани, длине имплантатов и расстоянию между ними, значению и направлению нагрузки. Действительно, прогнозирование биомеханических закономерностей в той или иной конструкции протеза зависит от значения прилагаемой нагрузки, т. е. характеристик зубов-антагонистов. Для соблюдения принципа идентичности условий исследования, а также учитывая частоту сочетания полного отсутствия зубов на верхней и нижней челюсти у пациентов пожилого возраста, мы определили среднюю вертикальную нагрузку как 10 МПа, сдвиговую — 2 МПа.

#### Обсуждение полученных результатов

Большинство авторов единодушны в том, что при полном отсутствии зубов для улучшения фиксации съемного протеза установку имплантатов следует проводить в межментальной области, длину имплантатов рекомендуют около 10 мм, если это не ограничивается индивидуальными размерами челюсти, а число их должно быть не менее двух.



**Рис. 4. Расчетная схема биомеханической системы "челюсть—протез":**

1 — сплав КХС; 2 — эластичная прокладка; 3 — слизистая; 4 — кортикальная кость; 5 — титановые имплантаты; 6 — пластмассовый протез; 7 — губчатая кость

Рассмотрим сравнительную характеристику запаса прочности у биомеханической системы нижней челюсти и полностью съемного протеза с балочным полулабильным соединением для трех уровней атрофии и различного числа опорных имплантатов. Распределенную нагрузку будем прикладывать к протезу в одном случае в центре, имитируя прямую нагрузку на имплантаты, в другом — на край протеза, имитируя максимальную вывихивающую нагрузку. Расчеты, выполненные в соответствии с описанной методикой, сведены в табл. 2, где в соответствии с заданными нагрузками, атрофией и плотностью губчатой кости, приведены вероятностные значения запасов прочности рассматриваемой биомеханической конструкции. Расчетная схема и примеры рассчитанных полей вероятности разрушения для центральной и краевой нагрузки приведены на рис. 4 и 5 (см. вторую сторону обложки). Из приведенных результатов следует, что при эксплуатации протезов данного типа, их эффективность и запас прочности будут тем выше, чем больше число опорных имплантатов. Однако эта зависимость имеет сильно нелинейный характер. При переходе от двух опорных имплантатов к трем запас прочности в некоторых случаях (для центральной нагрузки) возрастает почти в 2 раза. А при переходе от трех опор к четырем преимуществ остаются минимальными. Таким образом, можно заключить, что опору на четыре и большее число имплантатов следует рассматривать только в случае, если при значительном уровне атрофии плотность губчатой кости пациента существенно понижена.

На рис. 5 (см. вторую сторону обложки) приведены рассчитанные поля вероятностей разрушения биомеханической конструкции с относительной плотностью губчатой кости 0,7 при центральной распределенной нагрузке. Анализ этих полей и обратных к ним полей запаса прочности, а также всего комплекса полученных результатов, показывает, что в большинстве случаев целесообразно рекомендовать трехопорную систему протезирования. Четырех- и более опорная система целесообразна для пациентов с пониженной плотностью губчатой кости и существенной атрофии нижней челюсти. Двухопорная система может применяться у пациентов с высокой плотностью губчатой кости и при отсутствии атрофии. Конкретные рекомендации целесообразно уточнять с помощью компьютерного имитационного моделирования по данным компьютерной томографии пациента.

### Заключение

Дальнейшие исследования с использованием разработанной имитационной модели позволят оптимизировать конструкции полных съемных протезов с опорой на им-

плантаты, обеспечить равномерное распределение нагрузки между элементами биомеханической системы, разработать алгоритм действия врача при планировании хирургического этапа лечения пациентов, у которых полностью отсутствуют зубы, с применением съемных зубных протезов, фиксируемых с помощью имплантатов.

### Список литературы

1. Каливрадджян Э. С., Голубев Н. А., Лещева Е. А., Рыжова Л. П. Применение модифицированной конструкции съемного протеза с целью повышения эффективности лечения больных с полным отсутствием зубов // Актуальные проблемы стоматологии. Сб. трудов под ред. проф. Лебедеко И. Ю. — М.: Изд. ММСУ. 2002. С. 104—107.
2. Иванов С. Ю., Бизяев А. Ф., Ломакин М. В. и др. Стоматологическая имплантология: Стоматология XXI века. — М.: Изд. ММСУ, 2000. 63 с.
3. Олесова В. Н., Мушеев И. Ю., Поздеев А. И., Рогатнев В. П., Осипов А. В. Титан — оптимальный конструкционный материал для протезов на денальных имплантатах // Российский вестник денальной имплантологии. 2003. № 1. С. 24—27.
4. Олесова В. Н., Набоков А. Ю., Дмитренко Л. Н., Гарифудинов Д. М., Мушеев И. У., Дахно Л. А., Золотарев А. С. Компьютеризированное планирование денальной имплантации // Российский вестник денальной имплантологии. 2004. № 2 (6). С. 54—57.
5. Перевезенцев А. П. Замещение дефектов зубных рядов с использованием малого количества имплантатов // Современная ортопедическая стоматология, 2004. № 2. С. 50—53.
6. Misch C. E. Contemporary Implant Dentistry. 2<sup>nd</sup> ed. Mosby, Inc., 1999. 684 p.
7. Олесова В. Н., Осипов А. В. Изучение процессов напряженно-деформированного состояния в системе протез—имплантат—кость при ортопедическом лечении беззубой нижней челюсти. Ч. 2. Несъемное протезирование // Проблемы нейростоматологии. 1998. № 4. С. 8—11.
8. Трезубов В. Н., Арутюнов С. Д. Стоматология. М.: Медицинская книга, 2003. 576 с.
9. Киришин М. А., Лебедеко И. Ю., Ревякин А. В. Напряженно-деформированное состояние в системе "Полный съемный пластиночный протез — нижняя челюсть" со сферической и балочной системами фиксации на четырех внутрикостных винтовых имплантатах // Современная ортопедическая стоматология. 2005. № 4. С. 92—94.
10. Renouard F., Rangert V. Факторы риска в денальной имплантологии. М.: Медицинская книга, 2004. 182 с.
11. Чумаченко Е. Н., Воложин А. И., Портной В. К., Маркин В. А. Гипотетическая модель биомеханического взаимодействия зубов и опорных тканей челюсти при различных значениях жевательной нагрузки // Стоматология. 1999. № 5. Т. 78. С. 4—8.
12. Чумаченко Е. Н., Арутюнов С. Д., Лебедеко И. Ю. Математическое моделирование напряженно-деформированного состояния зубных протезов. М.: Молодая гвардия, 2003. 270 с.
13. Чумаченко Е. Н., Шашмурина В. Р. Математическая модель съемного зубного протеза // Сб. трудов 2-й МНПК. Исследование, разработка и применение высоких технологий в промышленности. Санкт-Петербург. 2006. Том 4. С. 255—256.
14. Weiss C. M., Weis A. Principles and Practice of Implant Dentistry. Mosby, Inc., 2001. 447 p.
15. Сухарев М. Ф., Бобров А. В. Изучение биомеханического взаимодействия имплантатов и кости методом математического моделирования // Клиническая имплантология и стоматология. 1997. № 2. С. 34—37.
16. Чумаченко Е. Н., Воложин А. И., Маркин В. А. Биомеханическая модель и методика расчета напряженно-деформированного состояния пародонтального комплекса нижней челюсти // Научные технологии. 2001. № 1. — С. 49—60.
17. Композиционные материалы / Под ред. Л. Браутмана и Р. Крока: Пер. с англ. В 8 томах. М.: Наука, 1978.
18. Композиционные материалы: справочник / Под ред. В. В. Васильева, Ю. М. Тарнопольского. М.: Машиностроение, 1990. 512 с.
19. Иванов С. Ю., Васильев А. Ю., Буланова Т. В., Гончаров И. Ю., Бучнев Д. Ю. Методика предимплантационного обследования пациентов // Российский вестник денальной имплантологии. 2003. № 2. С. 42—43.

**М. А. Короткова\***, канд. техн. наук,

**В. А. Рекубратский\*\***, аспирант,

**А. Е. Коротков\***, студент,

**О. А. Шульга\*\***, канд. хим. наук,  
ведущий научный сотр.,

**К. Г. Скрябин\*\***, д-р биолог. наук,  
зав. лаб./директор

\* Московский инженерно-физический институт,

\*\* Центр "Биоинженерия" РАН

## Динамическая модель генетического контроля развития растения на примере *Arabidopsis thaliana*

*Описаны система моделирования и созданная в этой системе модель развития растения. В модели учитываются мутации генов и их влияние на определение типа органов цветка, развитие стебля и листьев, а также влияние окружающей среды на процесс развития. Представлены основные подходы и модели, реализованные в системе. Приведены результаты работы программы, иллюстрирующие состоятельность разработанной модели.*

**Ключевые слова:** визуализация, ген, генная сеть, математическое моделирование, модель, морфогенез, мутация, растение, сети Петри.

### Введение

В предлагаемой работе описывается динамическая адаптивная программная модель развития растения на примере цветка *Arabidopsis thaliana* (L.) Heynh. Выбор объекта моделирования обусловлен тем, что значительное число генетических и молекулярно-биологических исследований, данные которых можно использовать для построения математических моделей, проводится именно на этом растении. Разработанная модель отражает влияние на динамику развития и плодородность/фертильность растения внешних условий и наличия мутаций в генах, определяющих морфогенез. Разработка таких моделей позволяет сделать шаг вперед в понимании механизмов работы генетического аппарата. Вместе с тем, проведение модельных экспериментов может позволить в ряде случаев избежать ненужных натуральных экспериментов, требующих значительных материальных ресурсов и временных затрат.

В настоящее время существует ряд работ, посвященных моделированию развития *Arabidopsis thaliana*, однако ранее не было предложено модели, позволяющей продемонстрировать как нормальное развитие растения, так и его развитие при наличии одной или нескольких мутаций.

В развитии растения выделяется вегетативный этап и этап разметки органов (этап формирования

цветка). Наиболее интересным для моделирования является этап определения местоположения (разметки) органов цветка. Если в растении дикого типа (в отсутствие мутаций) расположение органов фиксировано, то при различных мутациях (в генах, определяющих структуру цветка) число и вид органов может изменяться. Определить вид цветка при всех возможных комбинациях мутаций в настоящее время не представляется возможным.

Существует ряд теоретических моделей разметки органов. Все разработанные модели в той или иной степени учитывают изменение активности генов в процессе развития растения. Активности различных генов не являются независимыми величинами. Для отражения взаимозависимостей этих активностей в теоретической биологии введено понятие генной сети. *Генная сеть* — группа генов, взаимозависимое функционирование которых обеспечивает выполнение определенных жизненно важных функций организмов. Нас интересуют генные сети, работающие в период инициации цветения и формирования цветка.

Выделим три основные теоретические модели разметки органов цветка. Все они предлагают разделение генов на несколько классов, а меристемы, в которой происходит инициация развития органов цветка, на несколько зон — концентрических колец. Зоны нумеруются в порядке их образования. Тип органов в каждой зоне определяется тем, гены каких классов активны в этой зоне. Все модели учитывают только два возможных состояния гена: активное и неактивное.

Первая — базовая модель — это **ABC-модель**, описанная во многих работах (например, [1, 2]). Она предусматривает разделение генов, определяющих развитие цветка, на три класса, обозначаемых заглавными латинскими буквами: А, В и С. Согласно этой модели, в зоне с преимущественной активностью генов группы А формируются чашелистики. Преимущественная активность генов группы С соответствует формированию плодolistиков. Для формирования лепестков требуется активность генов классов А и В, тычинок — генов классов С и В. При этом в растениях дикого типа преимущественная активность генов класса А наблюдается в первой и второй зонах, генов класса В — второй и третьей зонах, генов группы С — третьей и четвертой зонах. Таким образом, в первой зоне растения дикого типа формируются чашелистики. Сочетание активности генов класса В с генами класса А, характерное для второй зоны растения дикого типа, приводит к формированию лепестков во второй зоне. В третьей и четвертой зонах соответственно формируются тычинки и плодolistики. При мутациях в генах этих классов наблюдается замена типов органов. Однако данная модель неспособна описать некоторые изменения в мутантных растениях, особенно в части изменения числа органов. Кроме того, эта модель учитывает работу не всех генов, влияние которых на формирование цветка известно. В развитие этой модели были предложены ABCDE-модель [3] и LABC-модель, описанные в работах [4, 5], свободные от некоторых ее недостатков.

В **ABCDE-модели** учитывается большее число генов, активность которых требуется для образования органов цветка. Дополнительные гены отнесены к классам D и E. Кроме генов тех классов, активность которых учитывает ABC-модель, для образования лепестков, тычи-

нок и плодолистиков требуется еще и активность генов класса E, а для развития семязачатков — еще и активность генов класса D. Таким образом, в зоне, где активны гены только классов A и E, образуются чашелистики, сочетание активностей классов A, B и E приводит к образованию лепестков, при сочетании активностей генов классов B, C и E образуются тычинки, при сочетании активности генов классов C и E образуются пестики, а если имеется еще и активность генов класса D в этой зоне, растение фертильно.

**LABC-модель** предлагает новое разбиение генов на классы, новый класс L определяется для некоторых генов, ранее принадлежавших классу A. В этой модели все органы цветка рассматриваются как производные двух базовых типов органов: листа и плодолистика. Формирование базовых типов органов обеспечивается функционированием генов класса L (лист) и C (плодолистик). Активность генов класса L соответствует образованию вегетативных листьев, одновременная активность классов L и A — чашелистиков, классов L, A и B — лепестков, классов C и B — тычинок, активность только генов класса C — плодолистиков.

Кроме моделей, отражающих работу генов в процессе морфогенеза, разработан ряд компьютерных моделей, отражающих динамику развития растения. Наиболее зрелищной из них является модель развития растения *Arabidopsis thaliana* (L.) Heynh [6], созданная в 2005 г. Эта модель воспроизводит развитие растения дико типа во времени. Модель является чисто иллюстративной, не связана с моделированием работы генов и отражает только внешние свойства растения. В другой динамической модели [7] заложено определение местоположения органов растения в зависимости от зоны влияния органа на базе ABC-модели. Недостатком представленной реализации модели является невозможность изменения заложенной в нее модели взаимодействия генов, а также отсутствие удобного интерфейса для настройки работы модели при наличии мутаций: мутации могут быть заданы только изменением областей влияния органов, что не позволяет непосредственно задавать конкретные мутации.

Поскольку исследования работы генов в настоящее время интенсивно развиваются, меняются и представления о взаимодействиях генов. Невозможно рассчитывать, что в ближайшее время появится окончательная модель их взаимодействия. Поэтому актуально создание адаптивной модели развития, позволяющей как задавать генную сеть, так и отражать возможные мутации и изменения внешних условий развития растения. Такая модель позволит не только осуществлять проверку работы уже известных генных сетей и при необходимости вносить в них изменения, но и определять критические точки для экспериментальных исследований, которые позволят уточнить существующие и вновь разрабатываемые модели взаимодействий генов и динамику их функционирования.

Разработанная модель отражает динамику развития растения и дает возможность вносить изменения в генную сеть. Кроме того, модель учитывает внутренние и внешние условия развития растения. Внешние условия — это влажность, освещенность, количество питательных веществ; внутренние условия — наличие или отсутствие мутаций в геноме растения.

## 1. Основы для моделирования

Основой для моделирования послужили теоретическая модель развития растения, способ представления активностей генов, способ представления генной сети и методы демонстрации динамики развития.

В качестве базовой теоретической модели формирования органов была сформирована LABCDE-модель, сочетающая в себе свойства как LABC [4, 5], так и ABCDE-модели [3]. Эта модель предусматривает шесть классов генов: L (AP2, LUG); A (AP1); B (AP3, PI); C (AG); D (FBP11, ANT, BEL1, SUP, SHP1 (= AGL1), SHP2 (= AGL5), AGL8, AGL11, AGL13, AGL15); E (SEP1, SEP2, SEP3). Активность генов класса L приводит к образованию вегетативных листьев, одновременная активность L и A — чашелистиков, классов L, A, B и E — лепестков, C и E — плодолистиков, C, B и E — тычинок, C, D и E — семязачатков в пестике. Модель отражает влияние внешней среды на активацию генов в процессе развития.

Использованная в модели генная сеть растения, работа которой приводит к образованию цветочной меристемы, представлена на рис. 1 (см. третью сторону обложки).

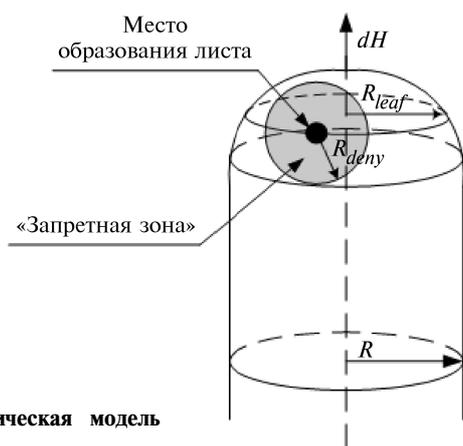
Для представления генных сетей используют различные сетевые графические модели, в которых вершины представляют гены, а дуги между вершинами — взаимосвязи между генами. Поскольку взаимосвязи бывают различных типов, базовыми из которых являются активация и подавление (активности групп или отдельных генов, а также процессов), эти особенности отражаются на схеме.

На схемах приняты следующие условные обозначения. Гены обозначаются краткими буквенными названиями: "AP1", "SEP3", "FT" и т. д. Гены, составляющие группу, выделяются рамкой. Гены, входящие в группу, активируются или деактивируются одновременно в результате одних и тех же воздействий. Процессы взаимодействия генов обозначаются стрелкой от гена A к гену B, если ген A активирует ген B, и линией с коротким перпендикулярным отрезком у гена B, если ген A деактивирует ген B. В том случае, когда во взаимодействиях участвуют группы генов, а не отдельный ген, вместо отдельного гена на схеме изображается прямоугольник, в котором перечислены гены группы. Условия активации генов сети и результаты работы генных сетей, отличные от активации генов или групп генов, обозначаются словами, например, условия: "Свет красный", "Яровизация", "Питательные вещества" и т. д., результаты: "Чашелистики", "Удлинение побега" и т. д.

Активация гена LFY на этой схеме соответствует окончанию вегетативного развития растения и образованию меристемы соцветия.

На стадии цветковой меристемы рассматриваем работу генных сетей в уже образовавшихся четырех зонах. Процесс образования этих зон в данной модели не рассматривается. Структуры генных сетей в этих зонах одинаковы, но начальные условия их работы различны. Вид генных сетей меристемы соцветия, определяющей идентичность цветковых органов, представлен на рис. 2 (см. третью сторону обложки).

Поскольку развитие растения определяется активностью генов, важнейшей задачей является моделирование работы генной сети. Основной характеристикой генов определенных классов является наличие или отсутствие их активности, поэтому генная сеть моделируется



**Рис. 3. Математическая модель развития побега**

с помощью элементарных сетей Петри, в которых каждому гену сопоставляется позиция, фишка в позиции соответствует активному гену, пустая позиция соответствует неактивному гену. Изменение состояния сети Петри соответствует изменению состояния генной сети. Мутантному гену, т. е. гену, который не может проявить активность, сопоставляется позиция, которая никогда не может содержать фишку.

Влияние внешней среды в модели отражается на состоянии генной сети, которая, в свою очередь, определяет внешний вид цветка и всего растения.

Для описания развития побега используется математическая модель, предложенная в работе [8] (рис. 3).

Стебель моделируется цилиндром радиуса  $R$ , на конце которого расположена полусфера того же радиуса — зона роста. Рост стебля происходит со скоростью  $dH$ , что соответствует увеличению высоты цилиндра на величину  $dH$  за единицу времени  $dt$  (шаг модельного времени). В зоне роста на окружности радиуса  $R_{leaf}$  расположено место, на котором возникает лист (на рисунке это место отмечено жирной черной точкой). Далее в некоторой "запретной зоне" радиуса  $R_{deny}$  лист образоваться не может. Место самого первого листа на стебле выбирается случайно на указанной окружности. Каждый последующий лист может образоваться через время  $dt$  на минимальном допустимом расстоянии от предыдущего. Таким образом, при функционировании модели во времени все точки образования листьев будут лежать на спирали, что соответствует реальному природному процессу.

В модели выделено три группы генов, мутации в которых по-разному влияют на развитие побега. В зависимости от группы, в которую входит мутантный ген, реализуются следующие эффекты:

- скорость  $dH$  выходит из-под контроля и может значительно варьироваться от шага к шагу;
- "запретная зона" перестает иметь строгие границы, т. е.  $R_{deny}$  также варьируется от шага к шагу;
- процесс развития листа идет не до конца, останавливаясь на одной из промежуточных фаз.

## 2. Используемая система моделирования

В разработанной комплексной системе моделирования развития растения приняты упрощения, приведенные ниже:

- гены разделяются на два вида: нормальный ген и мутантный, при этом все мутации рассматриваются

как сильные, т. е. приводящие к полной потере функциональности;

- в процессе работы генной сети для гена рассматриваются два состояния: активный и неактивный ген. Это позволило для моделирования функционирования генной сети использовать элементарные сети Петри;
- все влияния внешних условий на развитие растения отображаются только в состоянии его генных сетей;
- для демонстрации изменения фенотипа в зависимости от влияния внешней среды и состояния генной сети основными являются замена типов органов в случае мутаций и степень развития органов цветка. Поэтому изображение строится для одного цветка, а не для соцветия;
- не рассматривается процесс образования зон в меристеме, зоны считаем заданными;
- любая мутация генов класса D ведет к полной стерильности растения.

В системе для представления вегетативного развития и процесса развития цветка в системе реализовано взаимодействие разнотипных моделей: дискретной модели функционирования генных сетей растения и меристемы и непрерывных моделей для задания внешних условий. Общая программная модель развития растения включает в себя несколько подсистем, составляющих единую систему моделирования. В данную систему входят:

- подсистема реализации работы генной сети, реализованной в виде сети Петри;
- программа задания мутаций и внесения соответствующих изменений в сеть Петри, реализующую генную сеть;
- программа учета влияния окружающей среды на состояние генной сети;
- модель вегетативного развития растения (роста стебля и листьев);
- модель определения типа органов в цветке и развития цветка;
- программа определения фертильности растения;
- программа визуализации результатов развития.

Возможности изменения генных сетей растения в данной модели реализуются путем изменения сетей Петри для всего растения и меристемы соцветия. Программа задания мутаций генных сетей задает параметры реализованной генной сети и возможности работы генов. Динамика параметров внешней среды задает начальное состояние генных сетей и момент начала их работы.

**Реализация генной сети.** Любой процесс в сети имеет инициатора, которым всегда выступает ген или группа генов, и объект, на который это взаимодействие направлено. Объектом может выступать:

- другой ген или другая группа генов;
  - другой процесс взаимодействия.
- По характеру взаимодействия выделяют:
- активацию, когда активность одних генов приводит к появлению активности других генов либо к активации стороннего процесса (взаимодействия);
  - деактивацию, когда активность генов приводит к потере активности других генов либо к блокировке стороннего процесса.

Таким образом, в генной сети существует всего четыре различных типа взаимодействия.

Для моделирования работы генной сети использованы детерминированные одноцветные сети Петри с дискретным временем срабатывания [9]. Реализация приве-



ждом шаге проводится оценка значений всех параметров окружающей среды и при попадании значения какого-либо из параметров в определенный диапазон активируется соответствующий ген в общей генной сети. После этого сеть Петри, представляющая генную сеть, начинает работу и функционирует до некоторого конечного состояния. По результатам работы сети определяется момент наступления цветения, а также будет ли растение на данном шаге развиваться. Развитие здесь подразумевает удлинение стебля, рост и появление новых листьев, рост органов цветка, если они уже появились.

**Алгоритм вычисления местоположения нового листа на стебле.** Моделирование развития побега происходит согласно принципам, изложенным в работе [8] (см. рис. 3). Для вычисления местообразования следующего листа по известным положениям предыдущих листьев и их запретным зонам разработан следующий алгоритм.

На очередном шаге моделирования имеем набор листьев, для каждого из которых заданы координаты их местообразования  $\{O_i\}$  и радиусы  $\{R_{ie}\}$  "запретных зон", в которых новый орган появиться не может. Необходимо определить местоположение  $P$  нового листа, который не должен попасть ни в одну из имеющихся запретных зон. Используем цилиндрическую систему координат с осью  $Oz$ , в которой каждая точка  $M$  задается тройкой координат  $M_z$  (координата по оси  $Oz$ ),  $M_R$  (расстояние точки от оси  $Oz$ ) и  $M_\varphi$  (полярный угол).

Координата  $P_R$  задается параметром  $P_{leaf}$ :  $P_R = R_{leaf}$ .

Координата  $P_z$  однозначно определяется по  $P_R$ :

$$P_z = \sqrt{R^2 - (P_R - H)^2}.$$

Для определения координаты  $P_\varphi = \varphi$  вычисляются значения промежуточных параметров

$$q_i = \frac{O_{R_i}^2 + P_R^2 + P_z^2 + O_{z_i}^2 - 2P_z O_{z_i} - R_i^2}{2P_R O_{R_i}}$$

и рассматриваются следующие случаи.

1. Если хотя бы одно  $q_i < -1$ , то система не имеет решения и искомая позиция не может быть найдена при выбранной величине  $P_z$ , т. е. на данном шаге лист не образуется.

2. Если все  $-1 \leq q_i < 1$ , то вводится обозначение  $\arccos q_i = \alpha_i$  и рассматриваются варианты:

а) если при некотором  $i$   $\alpha_i + \varphi_i > 2\pi$ , то вычитаем  $2\pi$  из  $(\alpha_i + \varphi_i)$  и из  $(2\pi - \alpha_i + \varphi_i)$ ;

б) находим  $\max_i \{\alpha_i + \varphi_i\} = \beta$  и  $\min_i \{2\pi - \alpha_i + \varphi_i\} = \beta_2$ ;

в) если  $\beta_1 > \beta_2$ , то искомая позиция не может быть найдена при выбранной величине  $P_z$ , иначе  $\varphi = \beta_1$ .

3. Если некоторые (но не все) коэффициенты  $q_i \geq 1$ , то соответствующие ограничения исключаются из рассмотрения. Далее нахождение  $\varphi$  происходит аналогично случаю 2.

4. Если все  $q_i \geq 1$ , то координата  $\varphi$  может быть любой. В качестве значения  $\varphi$  выбирается случайный угол от 0 до  $2\pi$ .

Алгоритм либо определяет, что на данном шаге новый лист появиться не может, либо определяет координаты местоположения нового листа.

**Работа системы в целом.** Моделирование может проводиться как в автоматическом, так и пошаговом режиме. Результаты моделирования представляются динамически в графическом виде: строится трехмерный макет

полученного растения, который можно вращать для получения более подробного представления о строении растения, и двухмерная схема цветка как наиболее важной части растения. Также выдается информация о фертильности растения. Для облегчения процесса проведения исследований предусмотрена возможность протоколирования результатов работы модели. На рис. 6 (см. четвертую сторону обложки) представлено главное окно разработанной системы с результатами работы модели: трехмерным макетом растения без мутаций (слева) и схемой цветка (в центре).

### 3. Результаты

Для оценки прогностических возможностей разработанной модели было проведено исследование работы модели, в ходе которого модель определяла тип органов цветка для известных фенотипов растения *Arabidopsis thaliana* с мутациями по одному, двум и трем генам. Модель прошла испытание успешно, корректно определила типы органов во всех случаях. На рис. 7 (см. четвертую сторону обложки) приведен внешний вид некоторых фенотипов, полученный в системе моделирования.

Проведенные эксперименты позволяют сделать вывод, что разработанная система моделирования позволяет правильно определять тип органов в цветках с различными мутациями генов. Она устраняет некоторые недостатки ABCDE- и LABC-моделей определения типа органов цветка, учитывает влияние внешних факторов на развитие растения и осуществляет динамическое визуальное представление прогноза в виде трехмерного макета развивающегося растения и схемы цветка растения.

Система была включена в программный комплекс "Системная биология" в рамках комплексного проекта ИТ-КП.5/001, разработанного по заказу Министерства образования и науки РФ.

#### Список литературы

1. Pelaz S., Pitta G., Baumann E., Wisman E., Yanofsky M. F. B and C floral organ identity functions require SEPALLATA MADS-box genes // Nature. 2001. № 405. P. 200–203.
2. Bowman J. L., Drews G. N., Meyerowitz E. M. Expression of the Arabidopsis floral homeotic gene AGAMOUS is restricted to specific cell types late in flower development // Plant Cell. 1991. August (№ 8). P. 749–758.
3. Angenent G. C., Colombo L. Molecular control of ovule development // Trends Plant Sci. 1996. № 1. P. 228–232.
4. Honma T., Goto K. Complexes of MADS-box proteins are sufficient to convert leaves into floral organs // Nature. 2001. № 409. P. 525–529.
5. Шестаков С. В., Пенин А. А., Логачева М. Д., Ежова Т. А. Новая модифицированная схема генетического контроля развития цветка // Технологии живых систем. 2005. Т. 2. № 1. С. 37–46.
6. Mundermann L., Erasmus Y., Lane B., Coen E. and Prusinkiewicz P. Quantitative Modeling of Arabidopsis Development // Plant Physiology. 2005. October. Vol. 139. P. 960–968.
7. Алексеев Д. В., Ежова Т. А., Козлов В. Н., Кудрявцев В. Б., Носов М. В., Пенин А. А., Скрябин К. Г., Чуб В. В., Шульга О. А., Шестаков С. В. Закономерности разметки органов цветка *Arabidopsis thaliana* // Математическое моделирование. ДАН. 2005. Т. 401. № 4. С. 133–135.
8. Алексеев Д. В., Ежова Т. А., Козлов В. Н., Кудрявцев В. Б., Носов М. В., Пенин А. А., Скрябин К. Г., Чуб В. В., Шестаков С. В., Шульга О. А. Определение типа и положения органов цветка: динамическая модель развития // Изв. РАН. Сер. биологическая. 2006. № 6. Ноябрь–декабрь. С. 645–659.
9. Семенов Ю. А. Сети Петри: [http://www.podgoretsky.com/ftp/Docs/Internet/Semenov/10/petri.htm]

УДК 004.4

**Ю. И. Шемакин**, д-р техн. наук, проф.,  
Российская академия государственной службы  
при Президенте РФ

## Системантика и информационные нанотехнологии

*Излагается системно-семантическая основа содержания предложенной автором новой науки "системантика". Раскрывается ее роль в информационной нанотехнологии.*

**Ключевые слова:** система, семантика, системантика, информационная нанотехнология.

Системантика — новая наука, изучающая естественные и искусственные системы [1]. Название науки и заглавие книги несет большую смысловую нагрузку. К этому заглавию автор пришел не сразу. Оно сформировалось им в результате обобщения жизненного научного и практического опыта слиянием слов *система* и *семантика*. Понятие "система" тесно связано с мирозданием и с картиной Мира. Материи свойственна системная организация. Вне системы материя не существует. Материя и система — понятия синонимичные. Познать сущность Мира и его составляющих вне систем утопично. Понятие "семантика" отражает связь формы и содержания и обладает большой смысловой силой.

Попытки создания картины мира целостной системой предпринимались многими исследователями, начиная с античных времен. В религии термин "система" не нашел применения. В науке впервые наметок на определение понятия "система" сделал еще Аристотель формулировкой "целое несводимо к сумме частей, его образующих". В "целом" определение верно, но "частностей" оно не исключает. Так в Словаре русского языка С. И. Ожегова дается около десятка определений слову "система", отражающих различные частности.

В сети Интернет в статьях с ключевым словом "система" (С) основным определением выступает "принципиальная несводимость свойств системы к сумме свойств составляющих ее элементов и невыводимость из последних свойств целого; зависимость каждого элемента, свойства и отношения (С) от места и функций внутри целого".

Из новейших работ необходимо отметить книгу декана факультета системологии Университета штата Нью-Йорк (США) проф. Дж. Клира "Системология: Автоматизация решения системных задач" [2]. В названии предложенной им науки и учебной дисциплины несущими выразительными средствами выступают слова "система" и "логика" с их формальным толкованием. К этому же формату относятся книги В. В. Дружинина и Д. С. Конторова "Системотехника" [3] и Д. М. Жилина "Теория систем: Опыт построения курса" [4].

Недостатками приведенных в этих книгах определенных систем являются их антропоцентричность и отсутствие инвариантности, что неизбежно ведет к неограниченному выделению различных типов систем и соответствующих специализированных теорий, раскрывающих содержание понятий "система" каждого типа и вида. Это вызывает необходимость введения ряда классификаций систем с использованием разных оснований. Например, на атрибутом уровне таких классификаций исторически сложилось разделение понятий "материальные" и "идеальные", предлагаются адекватные понятия — "материальные" и "абстрактные". Следовательно, предусматривается возможность бесконечной детализации, небезобидной для практики.

Все это отразилось и на развитии самой науки, где в настоящее время системный анализ превалирует над синтезом. Наука же начинается там, где все многообразие сводится к единообразию. В средние века это стремление наиболее четко проявилось в работах нидерландского философа Б. Спинозы (XVII век). Опираясь на механико-математическую методологию, он стремился к созданию целостной картины природы.

"Общую теорию систем" выдвинул впервые в 1968 г. австрийский биолог-теоретик, проживающий с 1949 г. в США и Канаде, Людвиг фон Бергаланфи. В задачи этой теории входило: разработка математического аппарата описания разных типов систем, установление изоморфизма законов в различных областях знаний.

В обеих этих работах доминантой выступает формальный аппарат, содержательная истинность которого остается открытой.

С позиций самоорганизации с принципиальными трудностями столкнулась и синергетика, гимном которой является также форма, направленная на выяснение законов построения организации, возникновения упорядоченности в приложениях к прогнозам.

Наиболее продвинутые и содержательные взгляды на систему принадлежат Норберту Винеру как изобретателю слова "кибернетика" (в переводе с греческого "kebernetes" — "лоцман" или "рулевой"). Однако интерпретацию Н. Винером перевода следует подправить как объединение значений его переводных соответствий "лоцман" и "рулевой". Лоцман привлекается временно для проводки судов в труднопроходимых местах.

В общем случае кораблевождения и вождения летательных аппаратов на большие расстояния кинематические траектории, ведущие к цели, вычисляет штатный специалист — штурман, работающий на постоянной основе. Рулевой (в авиации — пилот) выполняет функции ее реализации воздействием на рулевые механизмы, которые вызывают динамический процесс. Подобные функциональные структуры присущи любым самоорганизующимся системам Вселенной.

Автором в течение многих лет велись исследования и практические работы, синтез которых привел к построению общей модели Вселенной с позиций единства ориентированных на знание информационных и вещественно-энергетических процессов в системах неживой, живой и социальной природы. *На этих основах впервые система определена как совокупность элементов, объеди-*

ненных самоорганизацией, единством цели и функциональной целостностью.

При этом понятию самоорганизация дается единое, обобщенное с управлением, следующее определение.

*Самоорганизация выражается в свойстве системы прогнозировать на основе открытости изменения своей структуры и функций при выборе цели с адаптацией к окружающей среде и выполнять управление на основе системной закрытости определенными целью реактивными действиями путем включения функциональных механизмов.* Из этой формулы следует, что самоорганизация включает управление и необходимые для этого в системах структурные и функциональные элементы.

Важнейшим методологическим принципом было принято изучение семантической сущности материи как выражения системной связи формы с содержанием. Функциональные процессы, свойственные природе, выделены дедуктивным анализом системной роли знаний и информации. Комплексное моделирование средствами вычислительной техники, расширение трактовки результатов моделирования на принципах системного единства, объединения материальных и идеальных, рациональных и иррациональных начал, а не их противопоставление и системное взаимоисключение, составляют основное содержание выдвигаемых в книге идей и положений, на основе которых проводится описание функциональных процессов в естественных и искусственных системах.

Простейшую систему, исходя из ее определения, могут составить два элемента. Один отдельно взятый элемент или два не взаимодействующих элемента не могут составить систему. Систему могут составить два взаимодействующих элемента с разными ролями. Образованный ими контур представляет элементарную бинарную структуру. Один ее элемент выполняет функции СУБЪЕКТА, а другой — ОБЪЕКТА, находящегося в оппозиции при единстве цели взаимодействия. Субъект воздействует на объект по ПРЯМОЙ связи, объект воздействует на субъект по ОБРАТНОЙ связи. Прямая и обратная связь составляют контур, обеспечивающий СОЗНАТЕЛЬНОЕ свойство всей природы, отражающей ее способность к самоорганизации и мутации. Окружающая среда выступает третьим участником взаимодействия, проявляющимся через системную цель. Взаимодействие этих элементов в системе определяется потребностями ее существования и развития. Из таких простейших систем по иерархическому принципу формируются системы более сложной структуры. Но эта сложность преодолевается, в свою очередь, такой же бинарной субъектно-объектной соподчиненностью иерархических уровней.

ПРОСТРАНСТВО и ВРЕМЯ — параметры, определяющие границы и период существования систем. Время необратимо и представляет собой кинематическую координату — "стрелу времени", определяющую идеальное положение системы в эволюционном процессе, пространство представляет собой динамическую координату реального положения системы относительно стрелы времени.

Функциональную целостность системы любой природы обеспечивают четыре атрибутивных элемента:

1) "вещество", как обладающее массой покоя, — статическая компонента — носитель знания;

2) "Энергия", как физическое поле, — динамическая компонента — носитель информации;

3) "знание", как стратегическая информация, необходимая для определения идеального состояния системы путем выбора цели и формирования кинематической траектории движения к ней;

4) "информация", как оперативные знания, необходимые системе в динамическом процессе, который переводит ее из фактического реального состояния в желаемое идеальное.

Взаимодействие элементов внутри системы и системы со средой связано с переносом вещества, энергии, знаний и информации. Обмен между системой и средой требует определенного контакта между ними и базируется на принципе открытости, а обмен между элементами внутри системы осуществляется на принципе закрытости.

Самоорганизация включает статические, кинематические и динамические процессы в их единстве. Статические компоненты обеспечивают накопление и хранение знаний в системной памяти. Кинематические процессы, основанные на знаниях, определяют необходимость присутствия сознания как внутрисистемного механизма. Динамические свойства, основанные на информации, определяют подсознательные (автоматические, регулирующие) действия. Сознание и подсознание, составляющие ИНТЕЛЛЕКТ, базируются на обратных связях. Через обратные связи осуществляется просмотр посредством периодической подачи на вход хранящихся в памяти знаний и поступающей из окружающей среды информации. Для осознания события время цикла должно быть меньше характерных временных изменений в окружающей среде — необходимо успеть за событиями, обладать способностью восприятия их непрерывного развития и реакции на них [5].

Различные виды материи (систем) различаются только своей структурой. Структура полностью и однозначно определяет свойства систем любой природы, поэтому структуру вещества как носителя свойства можно отождествить с языком, средствами которого кодируются хранящиеся знания, а структурные параметры энергетических процессов — с языком кодирования передающейся информации. Это утверждение относится и к такому виду материи (системы), какой в организме выступает мозг.

Более развитые системы наследуют полезные признаки прототипов и приобретают новые свойства, адекватные изменениям среды, что обеспечивает восходящее развитие и свидетельствует о творческом начале, присутствующем в природе.

Концептуальной основой науки "Системантика" является теоретически доказанная и подтвержденная практикой семантическая системность самоорганизации Вселенной единством формы и содержания, открытости и закрытости, прогнозирования и управления с адаптацией к окружающей среде, реализующих целевые реактивные структуры и функции, включением знаний ориентированных, информационных и вещественно-энергетических механизмов.

Термин "технология" традиционно употребляют применительно к искусственным процессам производства и понимают ее как совокупность (последовательность) приемов, направленных на создание продукта. Это определение относится и к "созданию" информации. Первой серьезной информационной технологией было книгопечатание, которое к настоящему времени привело к перегрузке общества информационными "шумами". Преодоление кризиса становится возможным с появлением "новых информационных технологий", основанных на использовании более систематических ЭВМ и методов.

*Системантика* — метанаука о семантической системности эволюционных переходов естественных и искусственных систем по критерию ассоциативности взаимодей-

ствия на основе единства формы и содержания. Она адекватно отражает системную семантику и ассоциативный принцип мышления единством смыслов, присутствующих в сознании (то есть идеальных компонентов), и их носителей, элементов в строении мозга (то есть реальных компонентов). Невозможно "извлечь" смысл, не затрагивая его носители — языковые и нейронные. Это следует иметь в виду науке, называемой "инженерия знаний". Нельзя отрицать и влияние искусственной среды на эволюцию естественной среды и особенно влияния развития вычислительной техники на эволюцию человеческого мозга. Рост объема памяти и быстродействия человеко-машинных систем при ассоциативной способности мышления и взаимодействия структур мозга создают кумулятивный эффект в развитии умственных способностей человека. По высказыванию Хеннинга Кёлера, сославшегося на работу Мюнхенского Института рациональной психологии, ученые которого пришли к выводу, что человеческий мозг в ускоренном темпе меняется так сильно, что через 50 лет мы будем иметь других людей, которые на основе других мозговых структур будут иначе думать, иначе ощущать и иначе действовать [6].

Вместе с тем, такое развитие событий, в свою очередь, приведет к коренному изменению облика искусственной среды. Убедительным основанием для этого являются успехи развития нанонауки, наноматериалов и нанотехнологий [7], позволяющих формировать на субмолекулярном уровне сложные структуры любых наносистем, с различными свойствами и функциями. Прежде всего, это касается

процесса познания, формирования наноструктур мозга и моделирования наноинтеллекта, обеспечивающих безграничное поле приложений, в том числе и самой науки систематики. Ее раздел, которому предстоит изучать семантические основы самоорганизации и системного синтеза на наноуровне, справедливо в этих условиях назвать "Наносистемантика".

Таким с позиций систематики можно представить коэволюционное развитие человеческой цивилизации с взаимными циклическими эволюционными переходами на основе критерия ассоциативности естественных и искусственных систем в земных условиях и межпланетном пространстве.

#### Список литературы

1. **Шемакин Ю. И.** Системантика. М.: Изд-во РАГС, 2006.
2. **Клир Дж.** Системология: Автоматизация решения системных задач. М.: Радио и связь, 1990.
3. **Дружинин В. В., Конторов Д. С.** Системотехника. М.: Радио и связь, 1985.
4. **Жилин Д. М.** Теория систем: Опыт построения курса. М.: Изд. УРСС, 2003.
5. **Сергий В. М.** Сознание как система внутреннего видения // Журнал высшей нервной деятельности. 1994. Т. 44. Вып. 4—5. С. 627—639.
6. **Пелехова Ю.** Синяя аура // Совершенно секретно. Апрель 2007. № 4 (215).
7. **Ковальчук М. В.** Курчатовский институт от "экса" до "нано" // Атом-пресса. 2007. № 10—11. С. 4.

## ИНФОРМАЦИЯ

### Пленарное заседание ИСО/МЭК СТК1/ПК36 в Корее

16—21 марта 2008 г. на острове Джеджу (Республика Корея) состоялось Пленарное заседание ИСО/МЭК СТК1/ПК36 "Информационные технологии в обучении, образовании и тренинге" (<http://jtc1sc36.org>).

В работе пленарного заседания приняли участие национальные делегации: Австралии, Великобритании, Германии, Канады, Китая, Люксембурга, Новой Зеландии, Норвегии, Республики Корея, Российской Федерации, США, Финляндии, Франции и Японии.

Национальная делегация Российской Федерации была представлена в следующем составе: Позднеев Б. М. (председатель), Борисов П. П., Косульников Ю. А., Куракин Д. В., Марков К. И.

Пленарному заседанию предшествовало проведение Открытого форума (Сеул), организованного Службой образования и исследования Республики Корея. Основными темами обсуждения были стандарты и технологии электронного обучения, электронные учебники, новые модели интерактивного электронного обучения.

В резолюциях пленарного заседания и рабочих групп отражен ряд важных решений относительно взаимодействия ПК 36 с другими подкомитетами СТК1, техническими комитетами ИСО и международными организациями, а также аспекты синхронизации разработки стандартов в рамках рабочих групп и редакторов, систематизации используемой терминологической базы, регламентов проведения пленарных заседаний подкомитета и рабочих групп и др.

Российская национальная делегация приняла активное участие в следующих рабочих группах (РГ): РГ1 "Словарь", РГ4

"Управление и доставка контента", РГ5 "Обеспечение качества и описание структуры".

На заседаниях РГ1 и РГ5 большое внимание было уделено обсуждению направлений деятельности и обеспечению сотрудничества с другими рабочими группами, подкомитетами и техническими комитетами ИСО, международными организациями. Были зафиксированы новые официальные взаимосвязи с техническими комитетами, подкомитетами ИСО/МЭК (ISO TC 176, JTC1/SC) и с международными организациями (Cartago Alliance, CEN/TC 353).

Руководитель Российской национальной делегации Позднеев Б. М. выступил на заседании РГ5 с докладами "Продвижение международных стандартов в Российской Федерации и государствах-членах Содружества Независимых Государств" и "Предложение по коррекции сферы деятельности РГ5".

На заседании РГ5 был утвержден обновленный список редакторов разрабатываемых стандартов, в том числе по стандарту "ISO/IEC 19796 Управление качеством, гарантии и метрики — Часть 5. Руководство по применению 19796-1" (в качестве со-редактора был утвержден Позднеев Б. М.).

На заседаниях РГ4 обсуждались вопросы использования профиля спецификаций SCORM в качестве документов ISO (технических отчетов, международных стандартов). В дискуссии активно принимали участие представители национальных делегаций Канады, США, Германии, а также представители ряда организаций ADL, IMS, AICC.

#### Источники:

ИСО/МЭК СТК1/ПК36 (<http://www.jtc1sc36.org>),  
TK461 (<http://www.tk461.stankin.ru>).

# CONTENTS

<b>Alguliev R. M., Aliguliyev R. M., Huseynova A. A. Increase of Efficiency of Corporate Network with Application CDN Technology.</b> . . . . .	2
The method of increase of efficiency of corporate networks which is based on CDN-technology is offered. Its efficiency is estimated by minimization of delivery cost of web-contents to end users. For minimization of delivery cost the mathematical model determination of optimum points of arrangement of servers and optimum replication of web-contents is offered. The model is described by discrete programming problem. The Method Particle Swarm Optimization is applied to solving the discrete programming problem.	
<b>Samoilin E. A. Synthesis of Optimum Algorithms Estimation of Pulse Handicaps at Restoration of Signals in Digital Radio Systems</b> . . . . .	9
Synthesis of optimum by criterion Neiman—Pirson algorithms estimation of pulse handicaps arising in reception devices of digital radio systems owing to many conditions of broadcasting and transformation of the digital information are consider. The algorithms allow to receive an optimum estimation of handicaps used in two stages procedures of restoration of digital information processes.	
<b>Keywords:</b> pulse handicaps, optimum estimation, restoration of signals, digital radiosystems.	
<b>Domnin A. O. The Analytical Model of a Client-Server Application.</b> . . . . .	13
In the article is suggested the mathematical model of task processing in a network application that is implemented as a closed queuing network system.	
<b>Keywords:</b> analytical model, queuing theory, theory of queuing, client-server model, response time.	
<b>Mukhacheva E. A., Bukharbaeva L. Y., Filippov D. V., Karipov U. A. Optimization Problems of Transport Logistics: Operating Bin Stowage for Cargo Transportation.</b> . . . . .	17
This paper contains schemes where one of the modules is the compact cargo stowage inside the transportation vehicles providing the ease of loading by warehouse conveyor and unloading by customer. For this purpose the optimization problem of packing with additional limitations is solved.	
<b>Keywords:</b> transportation-and-bin packing model, traffic routing, orthogonal packing, multi-method technology, metaheuristics.	
<b>Galimyanova N. N. Experimental Research Parallel Implementations Combinatorial Algorithms for Knapsack Problem</b> . . . . .	22
Parallel algorithms for one-dimensional zero-one knapsack problem are considered in this paper. Parallel branch and bound and dynamic programming algorithms are presented. Combined branch and bound with dynamic programming algorithm is proposed. Results of computational experiments are presented. Comparison with other works is made.	
<b>Keywords:</b> knapsack problem, parallel algorithm, parallel branch and bound algorithm, parallel dynamic programming algorithm, combined algorithm, experimental result.	
<b>Subbotin S. A. Method of Synthesis of Classifying Neuro-Fuzzy Networks with Using of Feature Term Importance.</b> . . . . .	31
The non-iterative method of classifying neuro-fuzzy network synthesis is offered. It is based on splitting of feature value ranges into equal length intervals and on belonging evaluation of a recognizing example to classes on belongings to overlapped feature value intervals with taking into account of their importance.	
<b>Zhukov I. S. Application of a Neural Network for Identification of Signals of Space Vehicles Transmitted on Command of Radio Lines</b> . . . . .	33
In article is considered the question of application of a neural network for identification of the information circulating in command radio lines of a space vehicle and subject to distortion because of handicaps, as in the equipment, so and at passage of a radio signal in atmosphere.	
<b>Tarasov G. V., Kharitonov D. I. Construction of Parallel Program Visualization System Based on Petri Nets</b> . . . . .	35
This paper presents architecture of parallel programs visualization system based on Petri nets. Modeling principles of sequential process structure in terms of Petri nets are examined. New method of parallel processes interaction visualization is proposed, which based on additional model application for describing communication environment behaviour.	
<b>Keywords:</b> Petri nets theory, parallel programming, program debugging.	
<b>Ryzhkov E. A., Karpov A. N. Approaches to 64-bit Applications Verification and Testing</b> . . . . .	41
The coming of 64-bit processors to the PC market causes a problem which the developers have to solve: the old 32-bit applications should be ported to the new platform.	
After such code migration an application may behave incorrectly. The article concerns questions related to software verification and testing. It also highlights difficulties a developer of 64-bit applications may face and the ways of solving them.	
<b>Keywords:</b> static analysis, source code analysis, software verification, software testing.	
<b>Bakaev V. V. Contractor Interactive Technical Information Support Service</b> . . . . .	46
A reliable operational system of communication between organizations possessing the product during its life cycle provides reduction in occupancy expenses. One of the solutions of the problem is to create Technical Support System for Customers. State-of-the-arte information technologies enable you to arrange the process on a twenty-four hour basis, 7 days a week, 365 days a year. This article presents a short description of the system ensuring solution of organizational and secure problems during informational technical provision of customers of aeronautical engineering.	
<b>Keywords:</b> catalogue, front server, knowledge data base, user logon, request engager.	
<b>Alexeev A. A., Kulikov A. I., Starushkin V. G. Visual Geological Media Designer "Seismic Model Manager"</b> . . . . .	48
There are many problems with the correct and non-contradictory geological cross-section model making. These problems are very actual today in geophysics. Also there are a lot of different algorithms and geophysical formats. Therefore geophysicist-specialists need to have a special product to build geophysical models and to run algorithms.	

Our product affords an opportunity to create, modify and save in different formats geophysical models. Seismic Model Editor as a part of Seismic Model Manager is a special vector editor and it allows geophysicist to build any model of geological crosssection. Editor supports model validation according to the rules described in science-based model. An important point is that any model modifying also changes corresponding geophysical data of the current model. So, visual editing of this model changes model parameters too.

Seismic Model Manager Client forms a part of Seismic Model Manager and allows user to upload a created model to a server and to run a task using available algorithm. Then Client should get a task results from a server and send it to client's computer. This solution provides a remote work for geophysicists

**Keywords:** geological media, data visualization, visual editor, isogeometric smoothing.

**Malykhina G. F., Merkusheva A. V. Using the Neural Network and Wavelet Transformation Jointly for Nonstationary Signal Analysis** . . . . . 51

We present some applications of joint using the neural network and wavelet-transformation (WT) in the problems of non-stationary signal analysis. The first problem relates to speech signal (SP) monitoring system where WT implementation is used for SP compression, probability distribution identification of signal WT and wavelet-filtration with adapted threshold. Neural network algorithm is used for recognizing the signal frame type (speech/pause) and controlling two phases: thresholds determination (in the pause frames) and wavelet-filtration of speech signal on the basis of these thresholds (in the speech frames). By such a method, the adaptation to noise level variations is achieved. The expansion of a signal is obtained on the basis of wavelet packet, frequency sub-bands of which approximate the critical bands of perceptual model. New relations are obtained for threshold calculation on the basis of exponential-power distribution, which generalize Cramer's expression for maximum value distribution for Gauss's variable. That makes it possible to filter (in the wavelet domain) the noises of most general type. In the second problem, we suppose the algorithm for estimating free-gas levels in commodity-oil pipelines. The method is based on non-contact radiation densitometer-device. Neural network implements identification of informational significance of WT signal frames: time intervals with gas in the oil flow and intervals having no gas content.

**Kirsanov A. P. A Combinatorial Method for Describing and Checking Processes** . . . . . 57

**Asadov H. G., Nabiev N. A. Multicriteria Optimization of Fuzzy and Non-Fuzzy Data Measurement Systems of Non-Stationary Energetic Condition** . . . . . 61

The questions on optimization of data measurement system of non-stationary energetic condition using multicriteria procedure are considered in the article. The multicriteria optimization of given subclass of data measurement systems is carried out using the combined target functional consisting of linear combination of information and fuzzy functionals. The practical recommendations on practical realization of similar systems are given.

**Chumachenko E. N., Shashmurina V. R. Forecasting of Interplay of Patterns of a Bone Tissue of an Edentulous Mandible with Full Removable Prosthesis** . . . . . 66

The mathematical model of a bio-mechanical system "removable prosthesis—implant—bone" is considered. Influencing geometrical arguments of a prosthesis and physical condition of bone tissues on stress — strain state of a system is analysed. The conformity between number of implants and strength properties of a system is mounted. The designed computer system of a simulation modeling allows to learn biomechanical fundamentals of interplay of patterns of a bone tissue of a mandible with full removable prostheses of a different construction and arresting them implants.

**Keywords:** biomechanics, mathematical simulation, tooth prostheses, implants, hardness, bone of a jaw.

**Korotkova M. A., Rekratskiy V. A., Korotkov A. E., Shulga O. A., Skryabin K. G. Dynamic Model of Genetic Control of Plant Development (on the Example of Arabidopsis thaliana)** . . . . . 71

Computer simulation system of plant development is described. Gene mutations and their influence upon definition of the flower organ type, development of stem and leaves, and also environment influence on the process of development are taken into account. Methods and models implemented in the system are submitted. Simulation results showing the system justifiability are demonstrated in the article.

**Keywords:** Arabidopsis thaliana, visualization, gene, gene network, mathematical simulation, model, morphogenesis, mutation, plant, Petri nets.

**Shemakin Yu. I. Systemantics and Information Nanotechnology** . . . . . 76

The author of the article studies the system and semantic principals of content of a new science "Systemantics" proposed by him and develops its importance in information nanotechnology.

---

---

**Адрес редакции:**

107076, Москва, Стромьинский пер., 4/1

Телефон редакции журнала (495) 269-5510

E-mail: it@novtex.ru

Дизайнер Т.Н. Погорелова. Художник В.Н. Погорелов.  
Технический редактор О. А. Ефремова. Корректор Т. В. Арбузова

Сдано в набор 07.05.2008. Подписано в печать 23.06.2008. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.

Усл. печ. л. 9,8. Уч.-изд. л. 11,15. Заказ 684. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"  
142110, Московская обл., г. Подольск, ул. Кирова, 15