

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

10(170)
2010

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ОПТИМИЗАЦИЯ

- Карпенко А. П., Мухлисуллина Д. Т., Овчинников В. А. Нейросетевая аппроксимация функции предпочтений лица, принимающего решения, в задаче многокритериальной оптимизации 2
- Гальченко В. Я., Якимов А. Н., Остапушенко Д. Л. Поиск глобального оптимума функций с использованием гибрида мультиагентной роевой оптимизации с эволюционным формированием состава популяции 9
- Мухачева Э. А., Валиахметова Ю. И., Хасанова Э. И., Телицкий С. В. Проектирование размещения ортогональных объектов на полигонах с препятствиями. 16

МОДЕЛИРОВАНИЕ

- Зелепухина В. А. Разработка интернет-ориентированных виртуальных лабораторий математического моделирования посредством разделения вычислительных и визуализационных задач 22
- Клосс Ю. Ю., Мартынов Д. В., Черемисин Ф. Г. Разработка методов компьютерного моделирования и анализа микронасоса Кнудсена 30

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Калинина Э. В., Ретинская И. В., Серова Д. А. Вероятностные и нечеткие модели оценки качества передачи информации в телекоммуникационных сетях 35
- Корячко В. П., Шибанов В. А., Ижванов Ю. Л., Шибанов А. П. Оценка времени передачи файла с учетом старения информации 40
- Сотов Л. С. Комбинаторная модель функционального формирователя разбиений бинарного множества. 46
- Давыдов А. И., Шахов В. Г., Ядрышников И. Б. Анализ протокола IPv6 в современных операционных системах. 52

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Грибова В. В., Черкезишвили Н. Н. Развитие онтологического подхода для автоматизации разработки пользовательских интерфейсов с динамическими данными. 54
- Зуев А. С., Петров Ю. И. Модернизация контекстных меню браузера, дерева папок и рабочего стола операционной системы Windows 59

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ

- Бушмелева К. И., Яценко Е. А. Специализированный пакет программ для калибровки измерительных приборов 64
- Курсков С. Ю., Ершова Н. Ю., Мурсагимов О. А. Организация удаленного доступа к микроконтроллерам MCS-96 на базе распределенной информационно-измерительной и управляющей системы поддержки научно-образовательного процесса 68

ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

- Гетманов В. Г., Орлов С. Е. Программный комплекс для спектрально-временного анализа нестационарных колебательных сигналов. 71

ИНФОРМАЦИЯ

- Тарнавский Г. А. Первый международный конгресс по "облачным" технологиям — Cloud Computing Congress CCC-2010. 77
- Contents 79
- Приложение. Барский А. Б. Оптимизационные задачи в основе пакета параллельных прикладных программ и системы информационного обслуживания Центра GRID-технологий

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.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
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.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ФОМИЧЕВ В. А.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.

Редакция:

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

Информация о журнале доступна по сети Internet по адресу <http://www.informika.ru/text/magaz/it/> или <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования.

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

УДК 519.6

А. П. Карпенко, д-р физ.-мат. наук, проф.,
e-mail: karpenko@nog.ru,

Д. Т. Мухлисуллина, студентка,
МГТУ им. Н. Э. Баумана,

В. А. Овчинников, канд. техн. наук,
ген. директор ООО "Ладуга", г. Одинцово,
e-mail: laduga@laduga.com

Нейросетевая аппроксимация функции предпочтений лица, принимающего решение, в задаче многокритериальной оптимизации

Рассматривается прямой адаптивный метод многокритериальной оптимизации на основе нейросетевой аппроксимации функции предпочтения лица, принимающего решение. Приводятся результаты исследования эффективности метода при решении двух- и трехкритериальных тестовых задач. Практическая апробация метода выполнена при решении двухкритериальной задачи оптимизации механической подсистемы двигателя внутреннего сгорания.

Ключевые слова: многокритериальная оптимизация, прямой адаптивный метод, аппроксимация, нейронные сети, двигатель внутреннего сгорания

Введение

При решении современных задач оптимизации, как правило, приходится учитывать различные противоречивые требования к качеству решения, поэтому выделяют класс задач многоцелевой, или многокритериальной, оптимизации (МКО-задач).

В задаче однокритериальной оптимизации (ОКО-задаче) полагается, что на некотором множестве альтернатив D_X задана скалярная критериальная функция $F(X)$, и выбор решения состоит в отыскании допустимой альтернативы, обеспечивающей наименьшее или наибольшее значение этой функции. Вообще говоря, критериальная функция $F(X)$ может быть как количественной, так и качественной. Типичными для технических задач являются количественные критериальные функции, которыми мы и ограничимся.

В МКО-задаче предполагается, что задана вектор-функция $\Phi(X) = (\phi_1(X), \phi_2(X), \dots)$, также оп-

ределенная на множестве альтернатив D_X . Лицу, принимающему решения (ЛПР), желательно найти в этом множестве альтернативу, которая, например, минимизировала бы все компоненты вектор-функции $\Phi(X)$, называемые частными критериями оптимальности.

При решении МКО-задачи ЛПР, принимая некоторую альтернативу $X \in D_X$ в качестве решения, всегда должно идти на компромисс, допуская ухудшение значений одних частных критериев для улучшения значений других критериев. При этом, как правило, нет объективных оснований для того, чтобы выбрать те или иные из указанных критериев и формализовать допустимые величины их уменьшения или увеличения. Вследствие этого МКО-задачи относятся к классу плохо формализуемых задач, решение которых принципиально требует привлечения предыдущего опыта и интуиции ЛПР. Указанное обстоятельство приводит также к тому, что решение современных практически значимых МКО-задач невозможно без использования программных систем многокритериальной оптимизации (МКО-систем).

Методы решения МКО-задачи чрезвычайно разнообразны. Существуют несколько способов классификации этих методов, например, классификация, основанная на содержании и форме использования дополнительной информации о предпочтениях ЛПР [1]. В соответствии с этой классификацией выделяются следующие классы методов решения МКО-задачи:

- методы зондирования;
- априорные методы;
- апостериорные методы;
- адаптивные методы.

Методы каждого из этих классов имеют свои достоинства, и ни один из них не свободен от недостатков, позволяющих признать его универсальным. Общей идеей всех методов решения МКО-задач является сужение множества допустимых значений вектора варьируемых параметров X вплоть до одной или немногих альтернатив.

Рассматриваемый в статье метод решения МКО-задачи относится к классу адаптивных методов, каждая итерация которых включает в себя фазу анализа, выполняемую ЛПР, и фазу расчетов, выполняемую МКО-системой. По характеру информации, получаемой МКО-системой от ЛПР

на фазе анализа, выделяются три класса адаптивных методов:

- методы, в которых ЛПР непосредственно назначает веса частных критериев оптимальности;
- методы, в которых ЛПР накладывает ограничения на значения частных критериев оптимальности;
- методы, в которых ЛПР выполняет только оценку (например, в терминах "лучше", "хуже", "одинаково") предлагаемых МКО-системой альтернатив [1]; методы этого класса называются *прямыми адаптивными методами*.

В основе прямых адаптивных методов решения МКО-задачи лежит предположение, что ЛПР может непосредственно сравнивать решения (в критериальном пространстве). Одной из наиболее известных процедур такой оценки решений является процедура Дайера—Джиофриона [2]. Другим известным методом данного класса является метод Зайонца—Валлениуса [3].

Прямой адаптивный метод решения МКО-задачи, который рассматривается в данной работе, основан на предположении существования функции предпочтений лица, принимающего решения $\psi(X)$, которая определена на множестве D_X и выполняет его отображение во множество действительных чисел R , т. е.

$$\psi : X \rightarrow R.$$

При этом задача многокритериальной оптимизации сводится к задаче выбора вектора $X^* \in D_X$ такого, что

$$\max_X \psi(X) = \psi(X^*), X \in D_X.$$

Предполагается, что при предъявлении ЛПР вектора параметров X , а также соответствующих значений всех критериев оптимальности $\phi_1(X)$, $\phi_2(X)$, ... это лицо может оценить соответствующее значение функции предпочтений $\psi(X)$.

В работе [4] предложен класс прямых адаптивных методов решения МКО-задачи, основанных на аппроксимации функции $\psi(X)$. В данной статье рассматривается один из методов этого класса — метод, базирующийся на аппроксимации функции предпочтения ЛПР с помощью нейронных сетей. Заметим, что методологической основой такой аппроксимации являются теоремы об универсальной аппроксимации [5].

Постановка задачи многокритериальной оптимизации

Пусть $X \in R^n$ — вектор параметров задачи (вектор варьируемых параметров), где R^n — n -мерное арифметическое пространство (пространство параметров). Множеством допустимых значений

вектора параметров X является ограниченное и замкнутое множество $D_X = \Pi \cap D$, где Π — "технологический" параллелепипед допустимых значений вектора параметров,

$$\Pi = \{X | x_i^- \leq x_i \leq x_i^+, i \in [1:n]\} \subset R^n;$$

множество D формируют ограничивающие функции $g_j(X)$, так что

$$D = \{X | g_j(X) \geq 0, j \in [1:k]\} \subset R^n.$$

Векторный критерий оптимальности $\Phi(X) = (\phi_1(X), \phi_2(X), \dots, \phi_m(X))$ со значениями в пространстве R^m определен в параллелепипеде Π . ЛПР стремится минимизировать на множестве D_X каждый из частных критериев оптимальности $\phi_1(X), \phi_2(X), \dots, \phi_m(X)$, что *условно* записывается в виде

$$\min_X \Phi(X) = \Phi(X^*), X \in D_X, \quad (1)$$

где X^* — искомое решение МКО-задачи.

Введем следующие обозначения: D_Φ — критериальное множество (множество достижимости) задачи, т. е. множество, в которое векторный критерий оптимальности $\Phi(X)$ отображает множество D_X ; D_Φ^* — фронт Парето задачи, $D_\Phi^* \subset D_\Phi$; D_X^* — множество Парето, т. е. множество векторов $X \in D_X$, которые порождают фронт Парето; $D_X^* \subset D_X$. Если $X \in D_X^*$, то будем говорить, что вектор X — эффективный по Парето вектор [1].

Положим, что частные критерии оптимальности $\phi_1(X), \phi_2(X), \dots, \phi_m(X)$ тем или иным образом нормализованы [4].

Рассмотрим решение задачи (1) методом скалярной свертки. Способ свертки не фиксируется — это может быть аддитивная свертка, мультипликативная свертка, свертка Джофриона и другие свертки [1]. Обозначим операцию свертки $\varphi(X, \Lambda)$, где $\Lambda \in D_\Lambda$ — вектор весовых множителей, $D_\Lambda = \{\lambda_i | \lambda_i \geq 0, \sum \lambda_i = 1, i \in [1:m]\}$ — множество допустимых значений этого вектора.

При каждом фиксированном векторе $\Lambda \in D_\Lambda$ метод скалярной свертки сводит решение задачи (1) к решению однокритериальной задачи глобальной условной оптимизации

$$\min_X \varphi(X, \Lambda) = \varphi(X^*, \Lambda), X \in D_X, \quad (2)$$

В силу ограниченности и замкнутости множества D_X решение этой задачи существует.

Хорошо известно [1], что в случае аддитивной свертки $\varphi(X, \Lambda)$ вектор X^* принадлежит множеству эффективных по Парето векторов. Для произвольной скалярной свертки вектор X^* , полученный в результате решения задачи (2), вообще говоря, не принадлежит множеству Парето. В связи с этим обратим внимание на то, что ЛПР не всегда выбирает решение, принадлежащее множеству Парето [6].

Если при каждом $\Lambda \in D_\Lambda$ решение задачи (2) единственно, то это решение ставит в соответствие каждому из допустимых векторов Λ единственный вектор X^* и соответствующие значения частных критериев оптимальности $\phi_1(X^*)$, $\phi_2(X^*)$, ..., $\phi_m(X^*)$. Данное обстоятельство позволяет полагать, что в этом случае функция предпочтений ЛПР определена не на множестве D_X , а на множестве D_Λ :

$$\psi: \Lambda \rightarrow R.$$

В результате МКО-задача сводится к задаче выбора вектора $\Lambda^* \in D_\Lambda$ такого, что

$$\max_{\Lambda} \psi(\Lambda) = \psi(\Lambda^*), \Lambda \in D_\Lambda. \quad (3)$$

Поскольку обычно $m \ll n$, переход от задачи (1) к задаче (3) важен с точки зрения уменьшения вычислительных затрат.

Если используется аддитивная свертка и множество достижимости D_Φ является выпуклым, то выражение (2) задает взаимно однозначное отображение множества D_Λ на множество D_Φ^* . При этом, как отмечалось, для любого $\Lambda \in D_\Lambda$ вектор X^* , являющийся решением задачи (2), принадлежит множеству Парето D_X^* . Если вместо аддитивной свертки используется свертка Дюффриона, то для получения того же результата не требуется выпуклость множества достижимости [1].

Величину ψ будем считать лингвистической переменной со значениями, представленными в таблице, где $\overset{\circ}{\psi}$ — ядро нечеткой переменной ψ [7].

Допустимые значения функции предпочтений ЛПР как лингвистической переменной

Функция предпочтений ψ	$\overset{\circ}{\psi}$
"Очень-очень плохо"	1
"Очень плохо"	2
"Плохо"	3
"Не совсем удовлетворительно"	4
"Удовлетворительно"	5
"Не совсем хорошо"	6
"Хорошо"	7
"Очень хорошо"	8
"Отлично"	9

В результате МКО-задача сводится к задаче отыскания вектора $\Lambda^* \in D_\Lambda$, обеспечивающего максимальное значение дискретной функции $\overset{\circ}{\psi}(\Lambda)$:

$$\max_{\Lambda} \overset{\circ}{\psi}(\Lambda) = \overset{\circ}{\psi}(\Lambda^*) = \overset{\circ}{\psi}^*, \Lambda \in D_\Lambda. \quad (4)$$

Метод решения задачи

Общая схема рассматриваемого метода является итерационной и состоит из следующих основных этапов.

Этап "разгона" метода (начальный этап). МКО-система некоторым образом (например, случайно) последовательно генерирует k векторов $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ и для каждого из этих векторов выполняет следующие действия:

- решает ОКО-задачу

$$\max_X \varphi(X, \Lambda_i) = \varphi(X_i^*, \Lambda_i), X \in D_X, i \in [1, k]; \quad (5)$$

- предъявляет ЛПР найденное решение X_i^* , а также соответствующие значения всех частных критериев оптимальности $\Phi_1(X_i^*)$, $\Phi_2(X_i^*)$, ..., $\Phi_m(X_i^*)$;
- ЛПР оценивает эти данные и вводит в МКО-систему соответствующее значение своей функции предпочтений $\overset{\circ}{\psi}(\Lambda_i)$.

Первый этап. На основе всех имеющихся в МКО-системе значений $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ вектора Λ и соответствующих оценок функции предпочтений $\psi(\Lambda_1), \psi(\Lambda_2), \dots, \psi(\Lambda_k)$ МКО-система выполняет следующие действия:

- строит функцию $\tilde{\psi}_1(\Lambda)$, аппроксимирующую функцию $\psi(\Lambda)$ в окрестности точек $\Lambda_1, \Lambda_2, \dots, \Lambda_k$;
- отыскивает максимум функции $\tilde{\psi}_1(\Lambda)$ — решает задачу

$$\max_{\Lambda} \tilde{\psi}_1(\Lambda) = \tilde{\psi}_1(\Lambda_1^*), \Lambda \in D_\Lambda;$$

- с найденным вектором Λ_1^* решает ОКО-задачу вида (5) — находит вектор параметров и соответствующие значения частных критериев оптимальности, а затем предъявляет их ЛПР. ЛПР оценивает указанные данные и вводит в систему соответствующее значение своей функции предпочтений $\psi(\Lambda_1^*)$.

Второй этап. На основе всех имеющихся в системе значений $\Lambda_1, \Lambda_2, \dots, \Lambda_k, \Lambda_1^*$ вектора Λ и соответствующих оценок функции предпочтений

$\psi(\Lambda_1), \psi(\Lambda_2), \dots, \psi(\Lambda_k), \psi(\Lambda_1^*)$ МКО-система выполняет аппроксимацию функции $\psi(\Lambda)$ в окрестности точек $\Lambda_1, \Lambda_2, \dots, \Lambda_k, \Lambda_1^*$ — строит функцию $\tilde{\psi}_2(\Lambda)$ и т. д. по схеме первого этапа до тех пор, пока ЛПП не примет решения о прекращении вычислений.

Исследование эффективности метода

Средствами известного программного комплекса MatLab 7.0 [8] разработана программа PREF-NNW, реализующая рассматриваемый метод решения МКО-задачи. Для построения и обучения нейронных сетей программа использует функции пакета Neural Network Toolbox, а для решения ОКО-задачи глобальной оптимизации — функцию пакета Optimization Toolbox, реализующую метод последовательного квадратичного программирования SQP (Sequential Quadratic Programming) в комбинации с методом мультистарта (из 100 случайных начальных точек).

Поскольку функция предпочтений ЛПП заранее неизвестна и может быть приближенно сформирована только в процессе диалога ЛПП с МКО-системой, ЛПП в процессе диалога может переоценить результаты предыдущих решений. Указанные оценки могут быть противоречивы, поэтому программа PREF-NNW предоставляет ЛПП механизм просмотра и корректировки предыдущих результатов, например, ЛПП может понизить или повысить введенные ранее оценки, "откатиться" на заданное число итераций и пр. Интерактивное взаимодействие программы PREF-NNW с ЛПП реализовано с помощью специально разработанного программного модуля, имеющего графический интерфейс.

С помощью программы PREF-NNW выполнено широкое исследование погрешности нейросетевой аппроксимации функции предпочтений ЛПП, а также исследование эффективности рассматриваемого метода в целом.

Исследование погрешности аппроксимации выполнено для двух двумерных двухкритериальных и одной двумерной трехкритериальной тестовых МКО-задач [9]:

- двухкритериальная задача 1, имеющая выпуклый фронт Парето: $\phi_1(X) = (x_1 - 5)^2 + x_2^2 + 10$;
 $\phi_2(X) = x_1^2 + (x_2 - 5)^2 + 20$;
 $D_X = \{X \mid 0 \leq x_i \leq 5, i = 1, 2\}$;

- двухкритериальная задача 2 (невыпуклый фронт Парето) [10];

$$\phi_1(X) = -x_1;$$

$$\phi_2(X) = -f(X) \left(1 - \sqrt{\frac{x_1}{f(X)}} - \frac{x_1}{f(X)} \sin(10\pi x_1)\right);$$

$$f(X) = 1 + 9x_2; D_X = \{X \mid 0 \leq x_i \leq 1, i = 1, 2\};$$

- трехкритериальная задача 3 (выпуклый фронт Парето):

$$\phi_1(X) = (x_1 - 5)^2 + x_2^2 + x_3^2 + 10;$$

$$\phi_2(X) = x_1^2 + (x_2 - 5)^2 + x_3^2 + 20;$$

$$\phi_3(X) = x_1^2 + x_2^2 + (x_3 - 5)^2 + 5;$$

$$D_X = \{X \mid 0 \leq x_i \leq 5, i = 1, 2, 3\}.$$

В качестве оценки погрешности аппроксимации использовали максимальную абсолютную ошибку $e = \max_i e = \max_i |\tilde{\psi}_i - \psi_i|$, где i — номер итерации. Исследовали многослойный персептрон (MLP) и нейронная сеть с радиальными базисными функциями (RBF) [8].

Исследование показало, что для МКО-задачи 1 MLP-сеть всего с двумя нейронами в скрытом слое обеспечивает удовлетворительную погрешность аппроксимации ($e < 0,53$); при числе нейронов в скрытом слое, превышающем 5, имеет место известный эффект переобучения сети. RBF-сеть для той же задачи дает несколько большую погрешность аппроксимации ($e \approx 0,6$); при этом каждая итерация требует увеличения числа нейронов в сети.

Аналогично, для МКО-задачи 2 MLP-сеть с пятью нейронами в скрытом слое обеспечивает приемлемую погрешность аппроксимации ($e < 0,17$). При этом RBF-сеть дает ошибку аппроксимации e , примерно в 4 раз превышающую минимальную ошибку MLP-сети.

Для МКО-задачи 3 удовлетворительную точность аппроксимации ($e < 0,32$) также обеспечивает MLP-сеть с пятью нейронами в скрытом слое. При этом RBF-сеть демонстрирует плохие экстраполяционные свойства.

Исследование эффективности метода выполнено для трех указанных выше МКО-задач при использовании MLP-сетей.

Для МКО-задачи 1 и MLP-сети с пятью нейронами в скрытом слое результаты исследования иллюстрируют рис. 1, 2, которые получены при $k = 3$ (три начальных случайных вектора $\Lambda = (\lambda_1, \lambda_2 = 1 - \lambda_1)$).

Светлые кружки на рис. 1 соответствуют "разгонным" точкам; темный кружок соответствует решению, предложенному программой PREF-

NNW на 4-й итерации. На рис. 2 светлые кружки показывают значения функции $\tilde{\psi}(\Lambda)$ на предыдущих четырех итерациях; темный кружок соответствует решению, предложенному программой PREF-NNW на 5-й итерации.

Исследование показывает, что для первой МКО-задачи нейросетевая аппроксимация функции предпочтений ЛПП позволяет быстро найти решение задачи с приемлемой точностью (всего пять итераций, включая $k = 3$ "разгонных" итерации).

Аналогичные результаты получены для МКО-задачи 2.

Для МКО-задачи 3 MLP-сеть с тремя нейронами в скрытом слое позволяет найти решение задачи за восемь итераций (включая $k = 7$ "разгонных" итераций со случайными векторами $\Lambda = (\lambda_1, \lambda_2, \lambda_3 = 1 - \lambda_1 - \lambda_2)$). Рис. 3 демонстрирует один из результатов исследования этой задачи. Темный кружок на рисунке соответствует решению, пред-

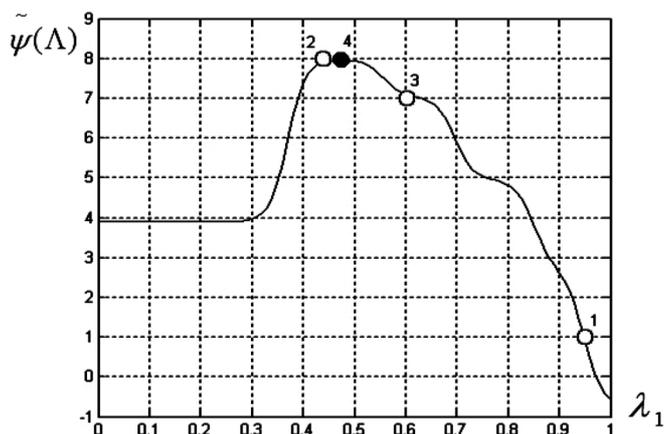


Рис. 1. Нейросетевая аппроксимация функции предпочтений МКО-задачи 1: MLP-сеть с пятью нейронами в скрытом слое; итерация 4

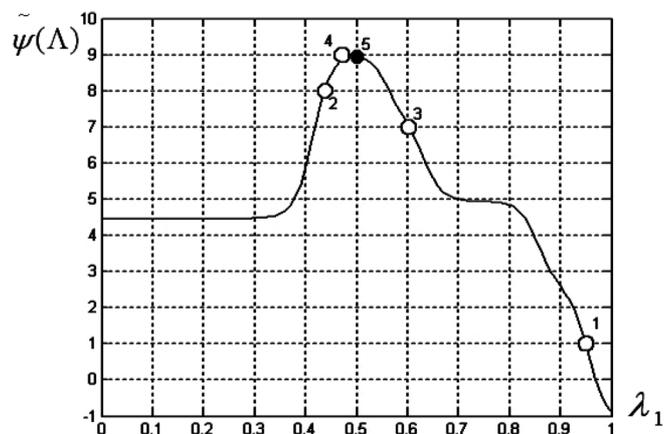


Рис. 2. Нейросетевая аппроксимация функции предпочтений МКО-задачи 1: MLP-сеть с пятью нейронами в скрытом слое; итерация 5

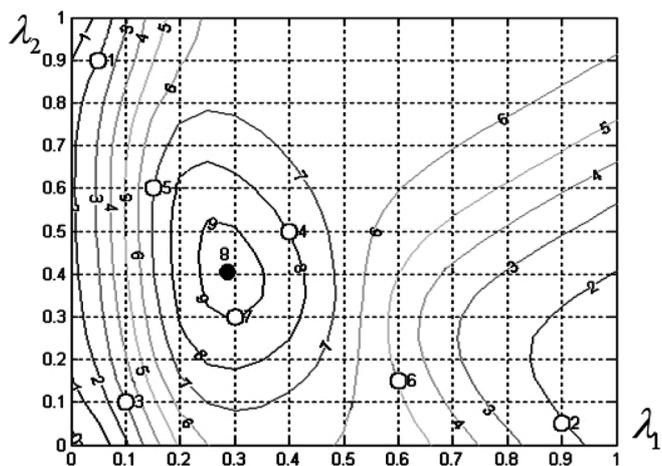


Рис. 3. Нейросетевая аппроксимация функции предпочтений МКО-задачи 3: MLP-сеть с тремя нейронами в скрытом слое; итерация 8

ложенному программой PREF-NNW на 8-й итерации.

Таким образом, и в случае трехкритериальной задачи нейросетевая аппроксимация функции предпочтений ЛПП позволяет быстро найти решение задачи (всего за восемь итераций, включая $k = 7$ "разгонных" итераций).

Практическая апробация метода

Рассмотренный адаптивный метод многокритериальной оптимизации интегрирован в программный комплекс анализа динамики систем различной физической природы — PRADIS [10]. Реализация метода выполнена на языке Python комплекса PRADIS с использованием библиотеки FANN, реализующей нейронные сети.

Практическая апробация метода выполнена для задачи двухкритериальной оптимизации двигателя внутреннего сгорания (ДВС). С этой целью средствами комплекса PRADIS разработана параметризованная трехмерная модель механической подсистемы ДВС [11]. Модель может быть использована для исследования работы одно-, двух- и четырехцилиндрового восьмиклапанного рядного автомобильного двигателя внутреннего сгорания с верхним расположением газораспределительного механизма.

PRADIS-модель ДВС. Общая структура модели представлена на рис. 4. Здесь и далее для представления структуры моделей используется входной графический язык препроцессора Qucs [10].

Маховик FLYWHEEL представлен библиотечной моделью, которая отображает инерционные свойства произвольного твердого тела при его пространственном движении. Модуль DRIVE моделирует ременную или цепную передачу, служащую для передачи вращения от коленчатого вала

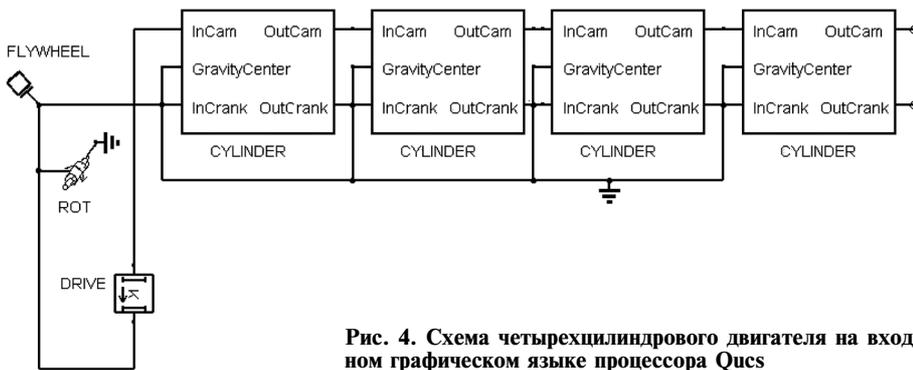


Рис. 4. Схема четырехцилиндрового двигателя на входном графическом языке процессора Qucs

к распределительному валу. Модель учитывает потери в передаче, которые определяются моментом трения в передаче. Подшипник ROT моделируется с помощью библиотечного цилиндрического шарнира. Модель учитывает упругие и вязкостные свойства подшипника [10].

Модель цилиндра CYLINDER является наиболее сложным элементом модели двигателя. Модель состоит из модуля кривошипно-шатунного механизма и модуля участка системы газораспределения. InCam и OutCam — входы и выходы модели распределительного вала; InCrank и OutCrank — входы и выходы модели коленчатого вала; GravityCenter — центр масс модели цилиндра.

Схема модуля кривошипно-шатунного механизма представлена на рис. 5. Модуль включает в себя: модель кривошипа CRANKSHAFT с противовесами COUNTERWEIGHT, которые образуют участок коленчатого вала; модель шатуна CONNECTING_ROD; модели поршня PISTON и системы индикаторного процесса INDICATE PROCESS_SYSTEM. Последняя модель по заданной зависимости давления газов в цилиндре двигателя от угла поворота кривошипа формирует силу, действующую на поршень. Указанная зависимость может быть задана как в аналитическом виде, так и в виде таблицы значений. PressureForce — модель силового воздействия на поршень.

Схема модуля участка системы газораспределения представлена на рис. 6. Модуль включает в себя модели участка распределительного вала CAM-SHAFT, кулачка CAM, клапана VALVE и пружины SPRING с тарелками TOP_PLATE и BOTTOM_PLATE.

Модели коленчатого и распределительного валов, а также

центром тяжести.

Модели поршней, тарелок, клапанов и маховика построены на основе библиотечной модели трехмерной инерционной массы. Противовесы моделируются с помощью той же модели трехмерной массы, но со смещенным центром тяжести.

Подшипники коленчатого и газораспределительного валов моделируются с помощью библиотечных моделей цилиндрических шарниров.

Модель контакта между поршнем и цилиндром представляет собой библиотечную модель цилиндрического шарнира, которая позволяет моделировать зазоры, нелинейную податливость, а также нелинейную вязкость в различных направлениях.

В модели системы газораспределения используется модель дискового кулачка с роликовым толкателем. Возможно также использование модели кулачка с тарельчатым толкателем. Профиль кулачка описывается кубическим сплайном, что обеспечивает гладкую зависимость силы, действующей на толкатель, от угла поворота кулачка. Моделируются податливость и инерция кулачка и ролика, трение скольжения и качения между кулачком и роликом.

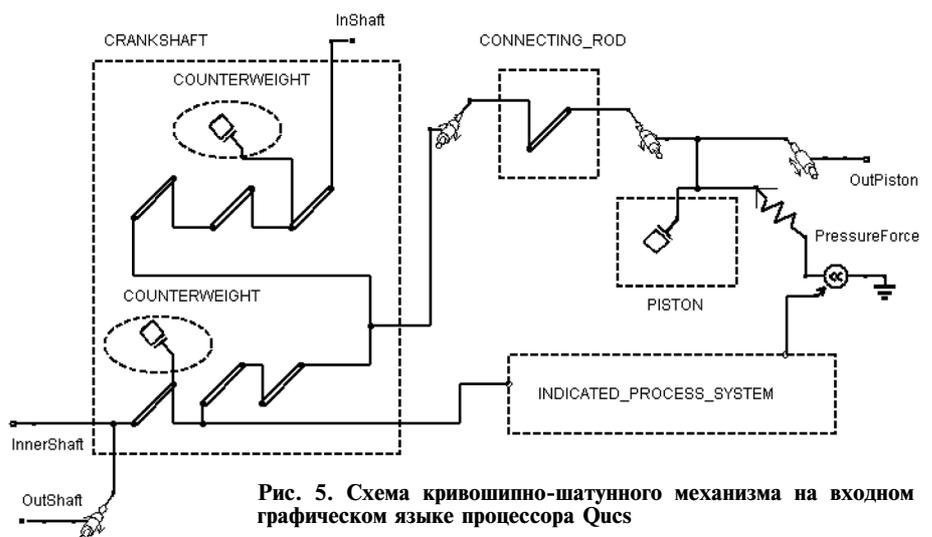


Рис. 5. Схема кривошипно-шатунного механизма на входном графическом языке процессора Qucs

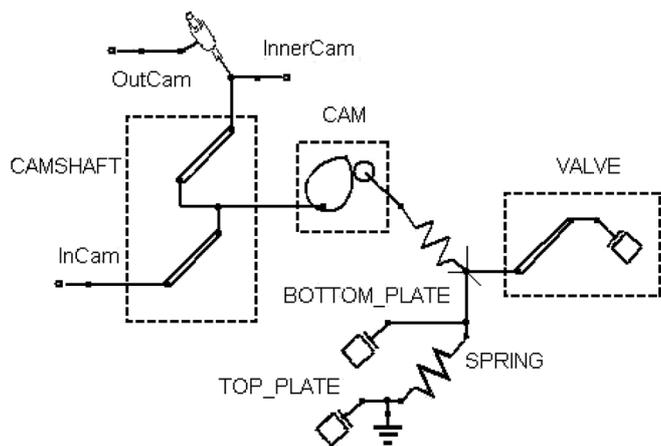


Рис. 6. Схема участка газораспределительного механизма на входном графическом языке процессора Qucs

Пружины моделируются с помощью библиотечной модели упругого линейного пространственного элемента.

Модели всех элементов ДВС являются параметризованными. В качестве параметров моделей используются геометрические характеристики двигателя (диаметр цилиндра, ход поршня, длины валов), инерционные характеристики деталей (массы и моменты инерции), физические свойства материалов и характеристики соединений (жесткость, вязкость).

МКО-задача. В качестве варьируемых параметров X задачи рассматривался ход поршня S и диаметр цилиндра D с ограничениями $0,06 \text{ м} \leq S \leq 0,15 \text{ м}$; $0,06 \text{ м} \leq D \leq 0,15 \text{ м}$, так что множество допустимых значений вектора варьируемых параметров имеет вид

$$D_X = \{X \mid 0,06 \leq S \leq 0,15; 0,06 \leq D \leq 0,15\}.$$

Критериями оптимальности являются максимальный крутящий момент $\phi_1(X)$ [Н·м], а также усилие реакции в шатунном подшипнике $\phi_2(X)$ [Н].

Для аппроксимации функции предпочтения ЛПР использовали MLP-сеть с пятью нейронами в скрытом слое. В качестве ЛПР выступал один из авторов работы — эксперт в области ДВС. Итерационный процесс поиска решения поставленной МКО-задачи иллюстрирует рис. 7.

Задача была решена всего за 11 итераций. Полученному решению соответствует точка 11 на рис. 7, для которой имеют место следующие значения варьируемых параметров и критериев оптимальности: $S^* = 0,118 \text{ м}$; $D^* = 0,123 \text{ м}$; $\phi_1^* = 143,92 \text{ Н·м}$; $\phi_2^* = 3049,20 \text{ Н}$. Заметим, что точки 1—11 на рис. 7 лежат на фронте Парето рассматриваемой

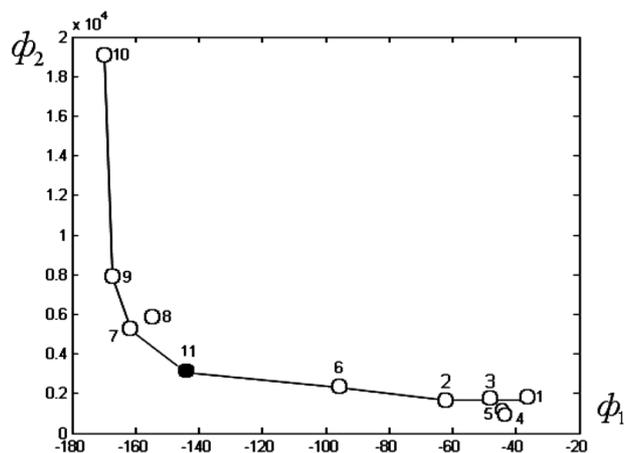


Рис. 7. Результаты решения задачи двухкритериальной оптимизации ДВС

МКО-задачи (с погрешностями, обусловленными погрешностями вычислений).

Опыт решения рассмотренной двухкритериальной задачи оптимизации ДВС показал удобство метода и возможность быстрого получения с его помощью решения МКО-задачи. Важно, что метод дополнительно позволяет ЛПР исследовать фронт и множество Парето решаемой МКО-задачи.

Выводы

Разработан адаптивный метод решения МКО-задачи, основанный на нейросетевой аппроксимации функции предпочтений ЛПР. Выполнено исследование эффективности метода, показавшее перспективность его развития. Метод реализован в виде MatLab программы PREF-NNW, а также в виде дополнительного модуля программного комплекса PRADIS. Практическая апробация метода и разработанного программного обеспечения выполнены при решении задачи многокритериальной оптимизации двигателя внутреннего сгорания.

В развитие работы предполагается сравнить эффективность нейросетевой аппроксимации функции предпочтений с эффективностью аппроксимации этой функции симплекс-планами, регрессионными планами второго порядка, а также нечеткими функциями. Кроме того, планируется сравнить эффективность рассматриваемого метода с эффективностью других адаптивных методов (например, метода, реализованного в известной МКО-системе NIMBUS [12]), провести практическую апробацию метода на МКО-задаче с тремя и более критериями.

Авторы благодарят В. Г. Федорука за плодотворное обсуждение работы, а также ценные советы по ее совершенствованию.

Список литературы

1. Лотов А. В. Введение в экономико-математическое моделирование / М.: Наука, 1984. 392 с.
2. Штойер Р. Многокритериальная оптимизация. Теория, вычисления и приложения. М.: Радио и связь, 1992. 504 с.
3. Дайер Дж. Многоцелевое программирование с использованием человеко-машинных процедур // Вопросы анализа и процедуры принятия решений. М.: Мир, 1976. С. 172–215.
4. Карпенко А. П., Федорук В. Г. Адаптивные методы решения задачи многокритериальной оптимизации, использующие аппроксимацию функции предпочтений лица, принимающего решения. [Электронный ресурс] // Электронное научно-техническое издание: наука и образование. 2008. № 8. URL: <http://technomag.edu.ru/doc/101804.html>.
5. Хайкин С. Нейронные сети. Полный курс. М.: Издат. Дом Вильямс, 2008. 1104 с.
6. Ларичев О. И. Теория и методы принятия решений. М.: Университетская книга, Логос, 2006. 392 с.
7. Леоненков А. Нечеткое моделирование в среде MatLab и fuzzyTECH. СПб.: БХВ-Петербург, 2003. 736 с.
8. Дьяконов В. П. MATLAB 7.*/R2006/2007. Самоучитель. М.: ДМК-Пресс, 2008. 768 с.
9. Мухлисуллина Д. Т. Исследование погрешности нейросетевой аппроксимации функции предпочтений лица, принимающего решения. [Электронный ресурс] // Электронное научно-техническое издание: наука и образование. 2009. № 12. URL: <http://technomag.edu.ru/>
10. Лобарева И. Ф., Черный С. Г., Чирков Д. В., Скороспелов В. А., Турук П. А. Многоцелевая оптимизация формы лопасти гидротурбины // Вычислительные технологии. 2006. Т. 11. № 5. С. 63–76.
11. PRADIS — анализ динамики систем различной физической природы [Электронный ресурс]. (<http://www.laduga.ru/pradis/pradis.shtml>).
12. Карпенко А. П., Мухлисуллина Д. Т., Овчинников В. А. Разработка математической модели двигателя внутреннего сгорания с использованием программного комплекса PRADIS. [Электронный ресурс] // Электронное научно-техническое издание: наука и образование. 2009. № 1. URL: <http://technomag.edu.ru/doc/120505.html>.
13. Карпенко А. П., Федорук В. Г. Обзор программных систем многокритериальной оптимизации. Отечественные системы // Информационные технологии. 2008. № 1. С. 15–22.

УДК 004.021

В. Я. Гальченко, д-р техн. наук, проф.,
А. Н. Якимов, преподаватель,
Д. Л. Остапушенко, аспирант,
Луганский государственный
медицинский университет,
e-mail: alex_forever_82@mail.ru

Поиск глобального оптимума функций с использованием гибрида мультиагентной роевой оптимизации с эволюционным формированием состава популяции

Рассмотрен новый гибридный алгоритм оптимизации роем частиц с генетическим алгоритмом, позволяющий эффективно оптимизировать многоэкстремальные функции высокой размерности. Проведено его тестирование на целевых функциях, имеющих характерные особенности в своем поведении.

Ключевые слова: глобальный оптимум, оптимизация роем частиц, генетический алгоритм, гибридный, источник магнитного поля, синтез

Введение

Во многих современных инженерных и конструкторских задачах оптимизации целевые функции многоэкстремальны и имеют большую размерность. Стандартные детерминированные ме-

тоды поиска оптимума практически бессильны в этих случаях и могут использоваться лишь для решения ограниченного круга проблем. Поэтому существует необходимость разработки новых алгоритмов, позволяющих эффективно оптимизировать функции и находить глобальный оптимум.

К числу таких задач относятся, например, задачи проектирования источников магнитных полей с априори заданным распределением поля в рабочей зоне [1, 2]. Эти задачи при решении требуют многократного повторения прямой задачи анализа, что приводит к достаточно большим затратам времени счета и использованию значительных вычислительных ресурсов. Особенно важным является сокращение числа вычислений целевой функции и повышение скорости сходимости оптимизационного процесса при синтезе источников поля, содержащих в своей конструкции ферромагнитные элементы [3], где вычислительные затраты многократно возрастают.

Для решения подобных задач наиболее перспективными являются стохастические мультиагентные эволюционные методы, к которым относится оптимизация роем частиц (PSO — *Particle Swarm Optimization* [4]). Анализ литературных источников [5, 6] показал, что этот способ оптимизации в ряде случаев более эффективен, чем генетические алгоритмы, и позволяет найти приемлемое решение за меньшее число итераций. Также PSO имеет значительные преимущества перед классическими методами нелинейной оптимизации: не требуется, чтобы начальное приближение было близко к искомому; нет необходимости знать свойства оптимизируемой функции;

не требуется вычисление ее производных. При использовании оптимизации роem частиц легко задавать ограничения пространства поиска. В то же время в методе PSO моделируется лишь социальное поведение индивидуумов, но никак не учитывается их эволюционное развитие через естественный отбор. В реальных же стаях и роях слабейшие особи погибают или изменяются их социальные связи в коллективе, а наиболее приспособленные рожают новых представителей популяции. В последние годы проводился ряд попыток совместно использовать оптимизацию роem частиц и генетические алгоритмы (GA — *Genetic Algorithm*).

Авторы работы [7] использовали эти алгоритмы параллельно: популяцию разделяли на две части случайным образом, к одной из них применяли оптимизацию роem частиц, а к другой — генетический алгоритм. К недостаткам такого подхода следует отнести слабую взаимосвязь алгоритмов между собой, что не позволяет должным образом использовать принцип естественного отбора в оптимизации роem частиц.

В работе [8] родительский пул обособлен от PSO, что уменьшает долю частиц, приходящихся на этот алгоритм, и в целом понижает эффективность гибрида. Под пулом понимается часть популяции, являющаяся результатом применения соответствующей операции алгоритма. По этой же причине в пул PSO не попадают частицы с наилучшим значением оптимизируемой функции (фитнеса). Создание отдельного пула элитных частиц может приводить к вырождению популяции и преждевременной стагнации. Используемые авторами операторы мутации и кроссовера влияют лишь на координаты частиц. В существующих работах генетический алгоритм не оказывает влияния на скорости частиц и параметры PSO.

Цель данного исследования — разработка метода и алгоритма, которые обеспечивают более высокую эффективность оптимизации, чем PSO и GA, и наследуют лучшие элементы эволюционных стратегий, реализуемых в каждом из этих подходов в отдельности.

Описание гибрида PSO-GA

Предлагаемый гибрид включает в себя стандартный алгоритм PSO *gbest* [4], схема которого приведена ниже, и генетический алгоритм с вещественными генами в хромосоме [9], а также особый способ их совместного функционирования. Этот выбор обоснован тем, что по данным работы [10], подход *gbest* имеет наибольшую скорость сходимости по сравнению с другими известными разновидностями PSO. Приведем основные этапы алгоритма.

Шаг 1. Начальная инициализация частиц роя.

Шаг 2. Сортировка частиц по увеличению наилучшего значения фитнеса.

Шаг 3. Выбор из пула PSO элитных частиц с лучшим фитнесом.

Шаг 4. Создание наследников с помощью оператора кроссовера координат и скоростей случайно выбираемых элитных частиц.

Шаг 5. Создание из элитных представителей PSO мутировавших частиц.

Шаг 6. Применение итерации алгоритма оптимизации роem частиц для пула PSO.

Шаг 7. Расчет фитнеса для всех частиц роя.

Шаг 8. Если не выполнены условия остановки, то переход к шагу 2.

Шаг 9. Вывод результатов.

В предлагаемом гибриде рой частиц имеет структуру, показанную на рис. 1.

Обозначим через *pso* — число частиц в пуле алгоритма PSO, *elite* — число элитных (родительских) частиц роя, *offsprings* — число наследников, полученных из случайно выбираемых элитных частиц оператором кроссовера; *x-mutations* — число частиц, полученных применением оператора мутации координаты к случайно выбранным родительским частицам с последующим помещением мутантов в отдельный пул. Таким образом, структура популяции частиц однозначно определяется набором четырех параметров:

$$\text{structure} = \{pso, elite, offsprings, x\text{-mutations}\}$$

Каждая частица (индивидуум) популяции имеет две хромосомы:

$$\mathbf{X} = (x_0, x_1, \dots, x_{d-1});$$

$$\mathbf{V} = (v_0, v_1, \dots, v_{d-1}),$$

где x_i, v_i — соответствующие i -е компоненты координат и скоростей частиц; d — размерность пространства поиска Ψ .

Совокупность всех частиц популяции задается множеством точек

$$\mathbf{S} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s-1)}),$$

где s — число частиц в популяции; $s = pso + offsprings + x\text{-mutations}$.

Наилучшее найденное решение частица сохраняет в отдельной хромосоме X_p , различной для каждой частицы, а лучшее решение, найденное всем роem, заносится в глобальную хромосому X_g .

В начале работы алгоритма оптимизации проводится инициализация роя путем задания случайных положений каждой частицы в пределах d -мерного пространства поиска. Начальные скорости счита-

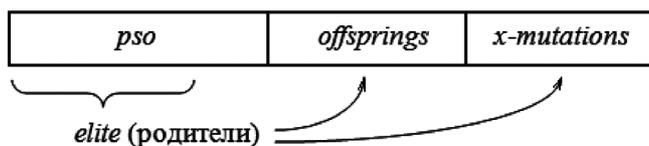


Рис. 1. Структура роя в алгоритме PSO-GA

ются равными нулю или, как вариант, генерируются случайным образом в интервале $[-x_{\max}; x_{\max}]$, где x_{\max} — наибольшее значение координаты.

При каждой итерации алгоритма PSO обновление скоростей выполняется по формуле

$$V^{(j)} = wV^{(j)} + c_1r_1(X_l^{(j)} - X^{(j)}) + c_2r_2(X_g^{(j)} - X^{(j)}),$$

где $X^{(j)}$ — вектор координат j -й частицы; $X_l^{(j)}$ — вектор наилучшего положения, найденного j -й частицей и характеризующегося наименьшим значением оптимизируемой функции; а $X_g^{(j)}$ — вектор наилучшего решения, найденного всем роем или соседями j -частицы; w — инерциальный коэффициент, определяющий память частицей своей прошлой скорости и влияющий на гладкость траектории ее движения; c_1 , c_2 — коэффициенты ускорения частиц к лучшему индивидуальному решению и к наилучшему решению, найденному роем; r_1 , r_2 — случайные числа, равномерно распределенные на интервале $[0; 1]$.

Такой подход в оптимизации роем частиц называется *gbest* с общим глобальным решением для роя. Если для каждой частицы глобальный оптимум определяется не всем роем, а только наилучшим значением фитнес-функции данной частицы и ее нескольких соседей, то для алгоритма PSO реализуется подход *lbest* [10]. Такая разновидность PSO имеет меньшие шансы попасть в локальный оптимум, но при ее использовании наблюдается более медленная сходимость, чем в *gbest*. Способ выбора соседей определяет топологию связей между частицами. В данной работе применялся метод случайного выбора топологии с небольшим числом соседей [11]. Для описания соседства использовалась матрица инцидентности L , имеющая размер $ps_o \times ps_o$, элементы которой $L_{i,j} = 1$, если i -я частица является соседом j -й частицы, и $L_{i,j} = 0$ — в остальных случаях. На каждой итерации алгоритма PSO матрица связей L обновляется таким образом, что у каждой частицы оказывается не более некоторого фиксированного числа случайно выбираемых с равной вероятностью соседей, оптимальное число которых по всей видимости определяется размерностью задачи и топографией целевой функции. В настоящей работе число соседей частицы не превышало трех. Возможным является также обновление матрицы инцидентности не на каждой итерации для всех частиц пула PSO, а только для вновь вошедших в этот пул частиц, состав которого в данном исследовании предлагается формировать с помощью генетиче-

ского алгоритма, тем самым сохраняя полезную топологическую информацию, полученную на предыдущих шагах алгоритма в целом.

Выбор весовых коэффициентов w , c_1 , c_2 в значительной мере влияет на сходимость алгоритма оптимизации роем частиц к оптимуму. Их значения приняты равными $w = 1/(2\ln(2)) \approx 0,72$; $c_1 = c_2 = 0,5 + \ln(2) \approx 1,19$, что обеспечивает устойчивую сходимость и баланс между локальным и глобальным поисками решения [12].

Перерасчет новых положений частиц проводится по формуле

$$X^{(j)} = X^{(j)} + V^{(j)}.$$

В каждой итерации выполняется проверка выхода скоростей и координат за допустимые границы. Если какая-то из компонент координат вышла за заданную границу, то она снова возвращается к ней и происходит остановка в данном направлении путем обнуления одноименной компоненты скорости.

При оптимизации естественный отбор реализуется с помощью обычной сортировки по наилучшему значению фитнес-функции, найденному каждой частицей. Таким образом, хромосомы с наилучшей приспособленностью попадают в пул PSO. Для генетического алгоритма в качестве родительских частиц отбираются случайным образом наиболее приспособленные индивидуумы из роя частиц PSO.

Пусть из элитных частиц выбраны точки p_1 и p_2 . Тогда хромосомы координат кроссоверов (рис. 2, а) можно найти по формулам

$$X' = \alpha \cdot X^{(p_1)} + (1 - \alpha)X^{(p_2)};$$

$$X'' = (1 - \alpha)X^{(p_1)} + \alpha X^{(p_2)},$$

где α — случайное равномерно сгенерированное число из интервала $[0; 1]$.

Кроме того, при применении оператора кроссовера от родителей предлагается наследовать не текущее положение в пространстве, а наилучшие найденные позиции. Это позволяет алгоритму GA в ряде случаев более эффективно использовать

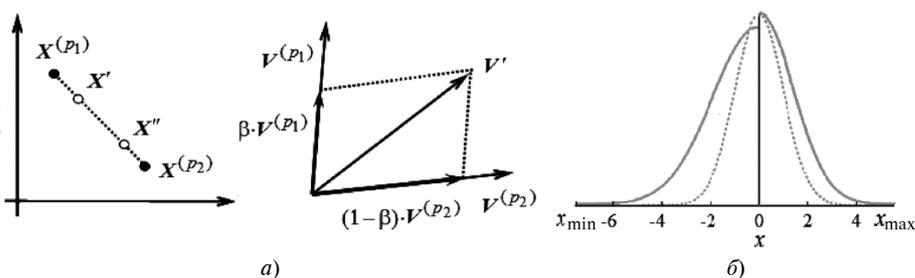


Рис. 2. Схемы генетических операторов роя:

а — кроссовер координат и скоростей; б — закон распределения для ассиметричных мутаций

память роя. Тогда схема образования хромосом координат у наследников примет вид

$$X' = \alpha X_I^{(p_1)} + (1 - \alpha) X_I^{(p_2)};$$

$$X'' = (1 - \alpha) X_I^{(p_1)} + \alpha X_I^{(p_2)}.$$

В отличие от большинства других гибридов при применении оператора кроссовера происходит обмен наследственной информацией не только о координатах, но и о лучших скоростях.

Для кроссоверов скоростей (рис. 2, а) используются аналогичные соотношения:

$$V' = \beta V^{(p_1)} + (1 - \beta) V^{(p_2)};$$

$$V'' = (1 - \beta) V^{(p_1)} + \beta V^{(p_2)},$$

где β — также случайно равномерно генерируется от нуля до единицы, а $V^{(p_1)}$ и $V^{(p_2)}$ — текущие скорости родителей.

Мутации в предложенном гибридном алгоритме строят на базе асимметричного нормального распределения следующим образом (рис. 2, б):

$$\Delta x = \begin{cases} Normal(0, 1)(x_{\max} - x)k/3, & \text{если } Normal(0, 1) \geq 0, \\ Normal(0, 1)(x - x_{\min})k/3, & \text{если } Normal(0, 1) < 0; \end{cases}$$

$$x' = x + \Delta x,$$

где $Normal(0, 1)$ — стандартизованное нормальное распределение; k — масштабирующий коэффициент, постепенно уменьшающийся с каждой итерацией алгоритма от единицы до нуля, например по линейному закону, что позволяет уменьшать степень мутаций с каждым новым поколением

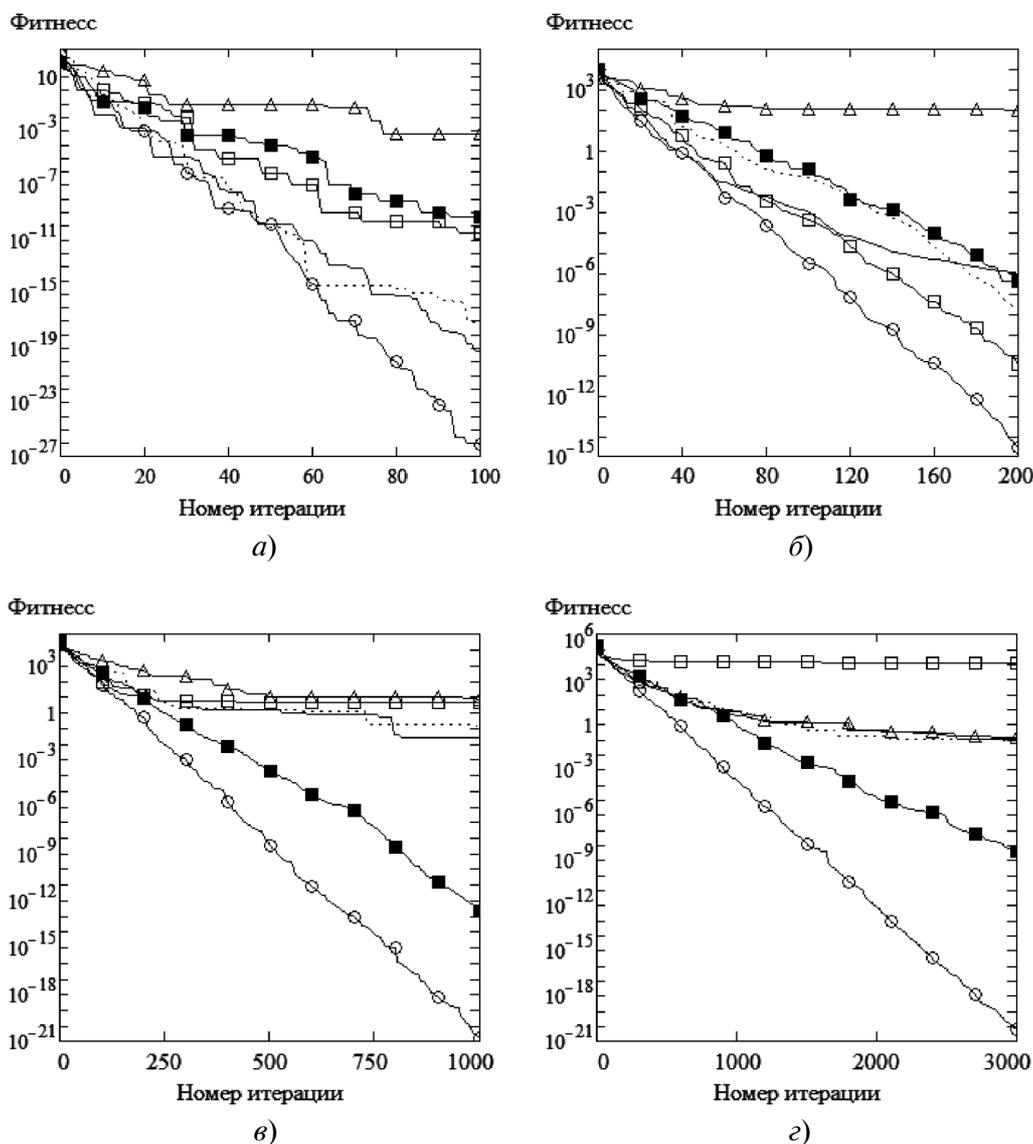


Рис. 4. Поиск оптимума квадратичной функции при размерностях: а — $d = 2$; б — $d = 10$; в — $d = 30$; з — $d = 100$

популяции. Величина Δx — смещение, на которое изменится на интервале $[x_{\min}; x_{\max}]$ мутируемая координата x .

Мутации позволяют разнообразить популяцию, способствуют глобальному поиску и уменьшают вероятность вырождения популяции.

Верификация гибридного алгоритма

Алгоритм применяли для задач с размерностью пространства поиска от 2 до 100. Поиск оптимума каждой из тестируемых функций проводился в d -мерном гиперкубе с ограничениями для каждой его стороны $-100 \leq x_i \leq 100$, где $i = \overline{0, d-1}$. Требовалось найти глобальные минимумы X^* этих функций и значения фитнес-функции $F(X^*)$ в них (см. таблицу).

Характеристики тестируемых функций

Вид функции	X^*	$F(X^*)$
Квадратичная функция $F_1 = \sum_{i=1}^d x_i^2$	(0; 0; ...; 0)	0
Функция Розенброка $F_2 = \sum_{i=1}^d 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2$	(1; 1; ...; 1)	0
Функция Растргина $F_3 = \sum_{i=1}^d 10 - 10\cos 2\pi x_i + x_i^2$	(0; 0; ...; 0)	0
Функция Гривонка $F_4 = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos \frac{x_i}{\sqrt{i}} + 1$	(0; 0; ...; 0)	0

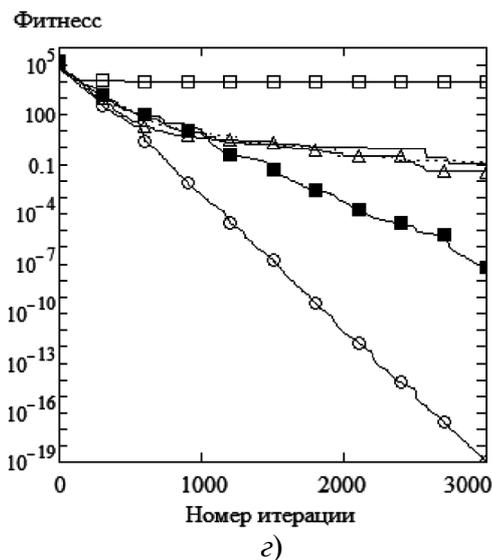
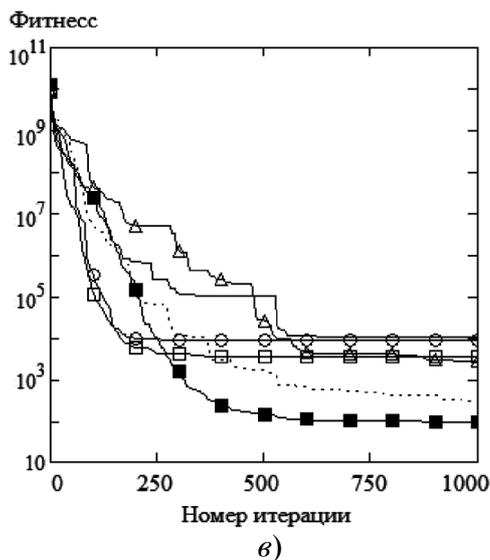
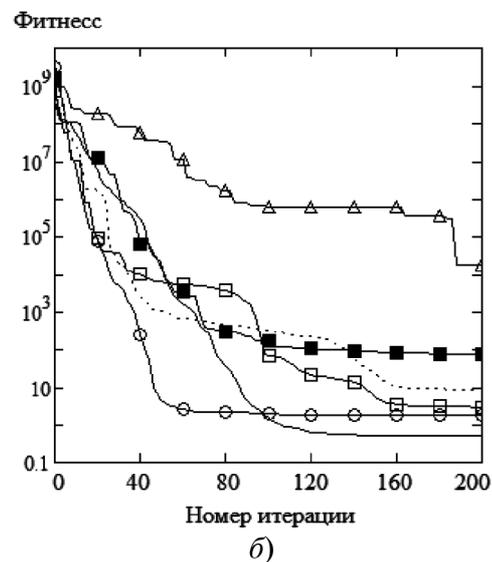
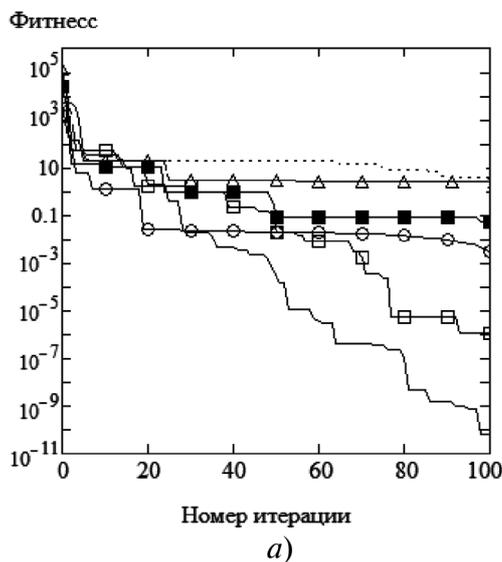


Рис. 5. Поиск оптимума функции Розенброка при размерностях: а — $d = 2$; б — $d = 10$; в — $d = 30$; г — $d = 100$

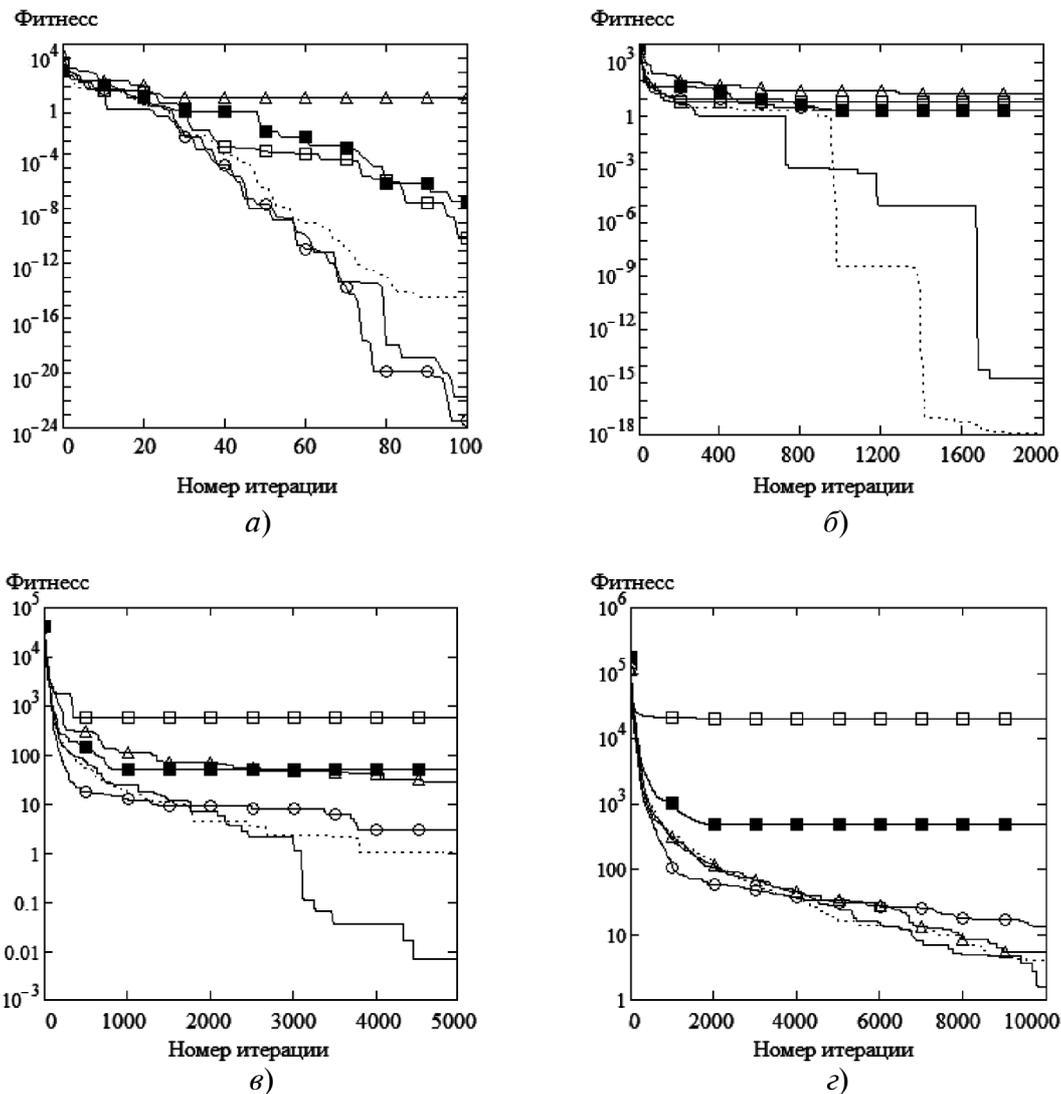


Рис. 6. Поиск оптимума функции Растргина при размерностях:
 $a - d = 2$; $b - d = 10$; $v - d = 30$; $z - d = 100$

Особенности в поведении исследуемых функций для случая двух аргументов иллюстрирует рис. 3 (см. третью сторону обложки): F_1 — при отдалении от экстремума функция быстро возрастает; F_2 — высокая степень овражности у оптимизируемой функции, при повышении размерности поиск оптимума значительно усложняется; F_3 — многоэкстремальная функция с множеством локальных экстремумов и одним глобальным оптимумом; F_4 — значение функции в ряде локальных оптимумов близко к значению в глобальном оптимуме, что приводит при оптимизации к частому попаданию в локальные решения.

Ниже приведены применяемые в алгоритме оптимизации структуры роя гибрида для разных размерностей пространства поиска:

$$\begin{aligned}
 d = 2: & \text{ pso} = 16, \text{ elite} = 5, \\
 & \text{ offsprings} = 4, \text{ x-mutations} = 0 \\
 d = 10: & \text{ pso} = 17, \text{ elite} = 5, \\
 & \text{ offsprings} = 2, \text{ x-mutations} = 1
 \end{aligned}$$

$$\begin{aligned}
 d = 30: & \text{ pso} = 14, \text{ elite} = 5, \\
 & \text{ offsprings} = 4, \text{ x-mutations} = 2 \\
 d = 100: & \text{ pso} = 21, \text{ elite} = 10, \\
 & \text{ offsprings} = 6, \text{ x-mutations} = 3
 \end{aligned}$$

При увеличении размерности задачи увеличение числа мутировавших частиц повышает эффективность глобального поиска оптимума.

Результаты численного моделирования приведены на рис. 4–7.

На приведенных рисунках показана сходимость к решению при использовании алгоритмов PSO, GA и PSO-GA в зависимости от числа итераций *iteration*. Сокращение PSO-GA* соответствует гибриднему алгоритму с оператором кроссовера, позволяющим наследовать от родителей наилучшие найденные позиции X_i . Этот алгоритм оказался наиболее эффективен для оптимизации функции F_1 .

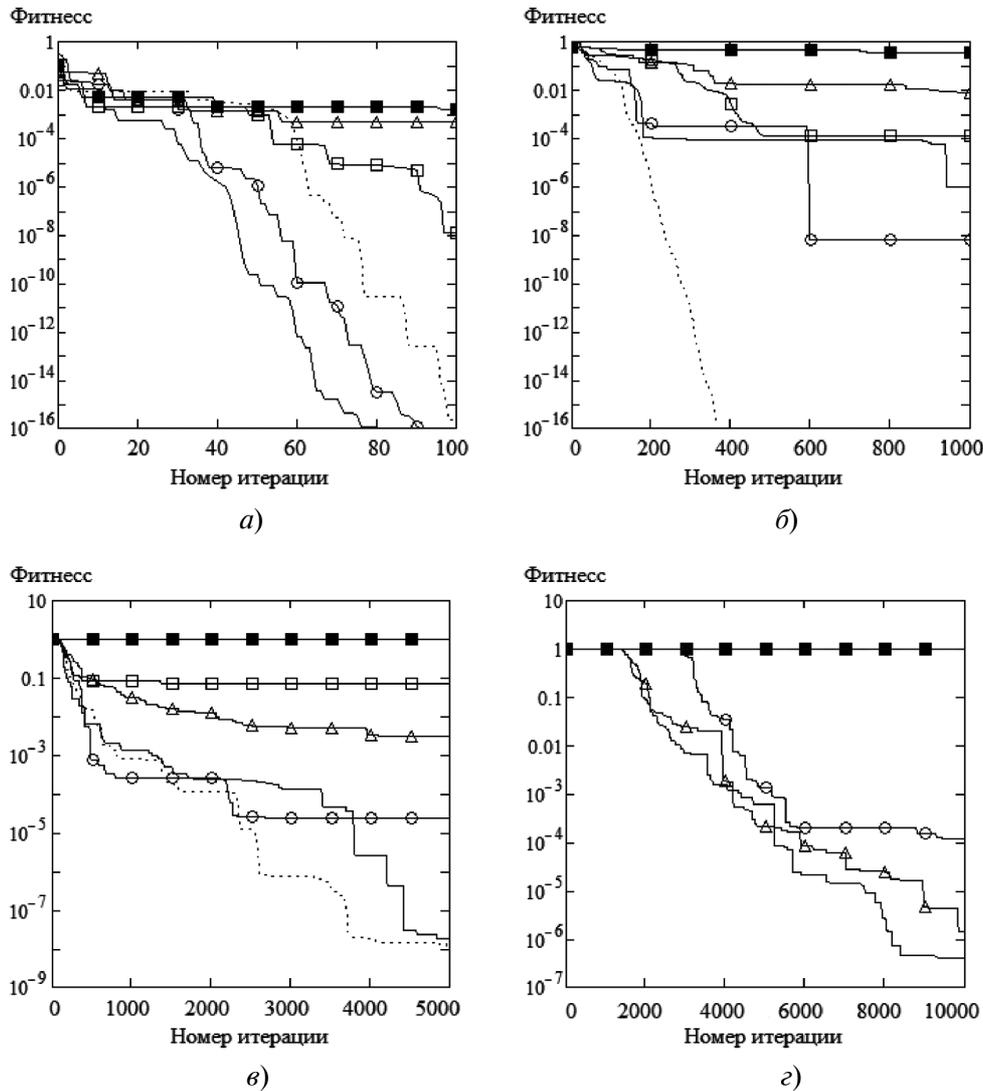


Рис. 7. Поиск оптимума функции Гривонка при размерностях:
 $a - d = 2$; $b - d = 10$; $v - d = 30$; $z - d = 100$

При оптимизации функции Розенброка для пространств поиска с небольшой размерностью наиболее эффективным оказался гибрид PSO-GA *gbest*. В случае $d = 100$ алгоритм PSO-GA* обеспечил наибольшую сходимость итерационного процесса. Для размерности 30, наиболее эффективный поиск провели методы оптимизации PSO и PSO-GA, использующие подход *lbest*.

Для оптимизации многоэкстремальных функций Растригина и Гривонка все разновидности гибридных методов оказались намного эффективнее, чем оптимизация роем частиц. Причем, как видно из графиков для размерности 100, с увеличением числа измерений возможность проводить глобальный поиск в гибриде PSO-GA обеспечивает в большей степени именно генетический алгоритм.

Выводы

1. Предложен новый гибрид алгоритмов PSO и GA, позволяющий эффективно оптимизировать

широкий класс целевых функций. При совместном использовании роевой оптимизации с генетическим алгоритмом удалось благодаря модифицированному оператору кроссовера улучшить локальный поиск роем оптимального решения, а оператор нормальной асимметричной мутации позволяет избежать попадания частиц роя в локальные экстремумы и улучшить способность к глобальному поиску.

2. Проведено тестирование работы гибридного алгоритма на различных типах функций в пространствах размерности от 2 до 100. Гибрид PSO-GA показал хорошие результаты, обеспечив в большинстве случаев более высокую скорость сходимости, чем стандартный PSO, что позволяет достичь требуемой точности решения за меньшее количество вычислений целевой функции.

3. Гибкость и управляемость метода, связанная с возможностью варьирования структуры роя частиц, высокая эффективность при поиске глобаль-

ного оптимума позволяют использовать предлагаемый алгоритм для решения широкого класса оптимизационных задач в науке и технике, в частности, в задачах синтеза источников магнитных полей с заданной конфигурацией поля.

Список литературы

1. Павлов А. К., Гальченко В. Я. Информационные модели для проектирования источников магнитных полей // Информационные технологии. 2002. № 7. С. 47–53.
2. Гальченко В. Я., Воробьев М. А. Использование генетических алгоритмов в структурном синтезе источников магнитных полей с заданными свойствами // Информационные технологии. 2003. № 7. С. 7–12.
3. Гальченко В. Я., Остапушенко Д. Л. Численный анализ пространственной конфигурации магнитных полей объектов сложной геометрической формы с учетом нелинейных характеристик веществ // Информационные технологии. 2008. № 8. С. 43–49.
4. Kennedy J., Eberhart R. Particle Swarm Optimization. // Proc. of the IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway. 1995. NJ. 5 (3). P. 1942–1948.
5. Eberhart R. C., Shi Y. Comparison between genetic algorithms and particle swarm optimization. // Proc. of the 7th Interna-

tional Conference on Evolutionary Programming. — Springer-Verlag, London. — 1998. — P. 611–616.

6. Hassan R., Cohanin B., de Weck O., Venter G. A comparison of particle swarm optimization and the genetic algorithm. // 46th IAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. — Austin, TX. 2005. April 18–21. P. 1–13.
7. Alfassion Grimaldi E., Gandelli A., Grimaccia F., Mussetta M., Zich R. E. A new hybrid technique for the optimization of large-domain electromagnetic problems. // Proc. of the XXVI-Ith General Assembly of the International Union of Radio Science, URSI GA 2005. — New Delhi, India. — 23–29 October 2005. URL: [www.ursi.org/Proceedings/ProcGA05/pdf/B05P.4\(01218\).pdf](http://www.ursi.org/Proceedings/ProcGA05/pdf/B05P.4(01218).pdf).
8. Li W. T., Shi X. W., Xu L. Improved GA and PSO culled hybrid algorithm for antenna array pattern synthesis. // Progress in Electromagnetics Research, PIER 80. 2008. P. 461–476.
9. Eshelman L. J., Schaffer J. D. Real-coded genetic algorithms and interval-schemata. // Foundation of Genetic Algorithms. 1993. Vol. 2. P. 187–202.
10. Нейтеративные, эволюционные и мультиагентные методы синтеза нечеткологических и нейросетевых моделей. / Под общ. ред. С. О. Субботина. — Запорожье: Изд-во ЗНТУ, 2009. 375 с.
11. Clerc M. Back to random topology. URL: http://clerc.maurice.free.fr/pso/random_topology.pdf.
12. Clerc M., Kennedy J. The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space // IEEE Transactions on Evolutionary Computation. 2002. Vol. 6. P. 58–73.

УДК 004.031

Э. А. Мухачева, д-р техн. наук, проф.,
e-mail: elitamuh@mail.ru,

Ю. И. Валиахметова, канд. техн. наук,
ст. преподаватель, e-mail: julikas@inbox.ru,

Э. И. Хасанова, аспирант, e-mail: hazel@mail.ru,
С. В. Телицкий, аспирант, e-mail: stasion@mail.ru,
Уфимский государственный
авиационный технический университет

Проектирование размещения ортогональных объектов на полигонах с препятствиями*

При размещении объектов на полигонах с препятствиями (строительных площадках, транспортных средствах и др.) возникают задачи "сквозного (гильотинного) размещения" и "декомпозиции свободной области полигона" на прямоугольные боксы. Для проектирования "сквозного размещения" применяются модификации "уровневого" подхода. Выбор лучшего размещения осуществляется в процессе эволюции с комбинированием мутаций. Задача "декомпозиции свободной области" сводится к проектированию покрытия полигонов с препятствиями. Для ее решения предложен метод с использованием матричного представления полигона.

Ключевые слова: гильотинное размещение, эволюционные метаэвристики, комбинирование эвристик, полигон, покрытие многоугольника

Введение

Объектом исследования является многосвязный ортогональный полигон (МОП). Под многосвязным полигоном понимается ортогональный многоугольник, внутри которого расположены односвязные многоугольники. МОП интерпретируется как полигон с уже размещенными в нем объектами многоугольной формы (здания, технические объекты и устройства, контейнеры). Требуется разместить новые прямоугольные объекты в свободной от препятствий области полигона.

Технические и организационные ограничения, диктуемые физическими условиями размещения предметов, такими как комфортность загрузки и выгрузки, обеспечение сквозных проходов и др., выдвигают задачу препроцессорной подготовки МОП к выполнению размещения. Для этого оказывается целесообразной декомпозиция свободной области полигона на прямоугольники. С такой проблемой встречаются при проектировании в строительной индустрии квартир и комнат, в логистике — складских помещений с учетом адресной системы склада, при загрузке товаров в транспортные средства, распределении сельскохозяйственных угодий и других ситуациях. Что касается функции цели, то она может быть различной в зависимости от конкретизации прикладной проблемы.

* Работа поддержана грантом РФФИ 10-07-91330-ННИО_a.

Аналогичная постановка декомпозиции свободной области полигона рассматривалась как задача его покрытия прямоугольниками с неопределенными размерами [1]. W. J. Masek показал, что задача является NP-полной.

В данной статье предложен оригинальный алгоритм декомпозиции с использованием матричного представления полигона. Незанятые в матрице позиции объединяются в прямоугольные области, которые не содержат препятствий. Далее решается задача сквозного размещения объектов в прямоугольных областях. Для этой цели используется уровневый алгоритм со стратегией "первый подходящий" (First Fit, FF) и приводятся подходы к реализации эволюционной метаэвристики с использованием технологии комбинирования эвристик [2, 3].

Декомпозиция свободной области МОП на прямоугольники

Постановка и комбинаторная модель задачи. Имеется многосвязный ортогональный полигон (рис. 1, а). Опишем вокруг него прямоугольник. Многоугольные области этого прямоугольника, не принадлежащие МОП, представим как виртуальные препятствия. Далее разделим многоугольные препятствия сквозными линиями так, чтобы они разбились на прямоугольники (рис. 1, б). Это можно сделать несколькими способами, но поскольку по условию задачи препятствия являются односвязными, то способом разбиения можно пренебречь. В результате интересующая нас задача состоит в следующем.

Дана прямоугольная область ширины W и длины L , а также набор прямоугольных препятствий заданных размеров $\omega_v, \lambda_v, v = \overline{1, \mu}$, где μ — число препятствий, ω_v, λ_v — ширина и длина v -го препятствия или его прямоугольной части. Введем прямоугольную систему координат: оси Ox и Oy совпадают соответственно с нижней и боковой сторонами области. Положение каждого прямоугольного препятствия B_v задается координатами (χ_v, η_v) его нижнего угла. Таким образом,

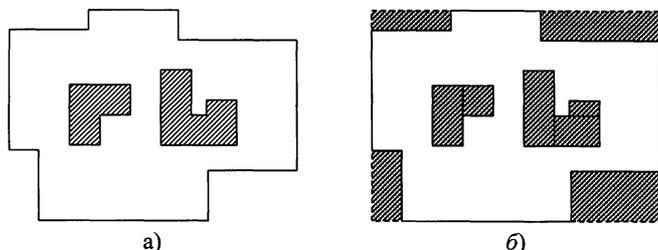


Рис. 1. Пример многосвязного ортогонального полигона: а — в исходном виде; б — в виде многосвязного прямоугольника с дополнительными препятствиями

МОП можно представить следующим набором исходных данных:

$$\langle W, L, (\chi_v, \eta_v), (\omega_v, \lambda_v), v = \overline{1, \mu} \rangle.$$

Требуется найти множество $\Pi = \{\Pi_1, \Pi_2, \dots, \Pi_m\}$ прямоугольников $\Pi_i = \langle x_i, y_i, w_i, l_i \rangle$ минимальной мощности, где (x_i, y_i) — координаты нижнего угла i -го прямоугольника, w_i и l_i — его ширина и длина, m — искомая мощность множества Π , удовлетворяющие следующим условиям:

1° **Ребра прямоугольников параллельны ребрам области:**

$$(\rho_x^i = l_i) \wedge (\rho_y^i = w_i), i = \overline{1, m},$$

где ρ_x^i, ρ_y^i — проекции прямоугольника Π_i на оси координат Ox и Oy .

2° **Взаимное неперекрывание прямоугольников:**

$$\forall i \neq j: i, j = \overline{1, m}, ((x_i \geq x_j + l_j) \vee (x_j \geq x_i + l_i)) \vee ((y_i \geq y_j + w_j) \vee (y_j \geq y_i + w_i)).$$

3° **Неперекрывание прямоугольников границами области:**

$$\forall i = \overline{1, m} (x_i \geq 0) \wedge (y_i \geq 0) \wedge (y_i + w_i \leq W) \wedge (x_i + l_i \leq L).$$

4° **Неперекрывание прямоугольников с препятствиями:**

$$\forall i, v: i = \overline{1, m}; v = \overline{1, \mu} \\ ((x_i \geq \chi_v + \lambda_v) \vee (\chi_v \geq x_i + l_i)) \vee ((y_i \geq \eta_v + \omega_v) \vee (\eta_v \geq y_i + w_i)).$$

5° **Прямоугольники должны покрывать всю свободную область МОП:**

$$\sum_{i=1}^m w_i l_i = WL - \sum_{v=1}^{\mu} \omega_v \lambda_v.$$

Задача разбиения свободной области на прямоугольные участки является частным случаем задачи покрытия прямоугольниками МОП.

Она получила название "Задача минимального покрытия" (Irreducible Covering Problem, ICP). Впервые W. J. Masek доказал, что задача является NP-полной [1]. Позднее J. C. Culberson и R. A. Reckhow свели ее к задаче о выполнимости и доказали, что она остается NP-полной даже в случае, если покрываемый полигон является простым (без препятствий) [4]. P. Verman и V. Dasgupta доказали, что если полигон имеет препятствия, задача является NP-трудной [5]. Задача разбиения прямоугольной области с препятствиями на боксы является более сложной, поскольку на прямоугольники дополнительно накладываеся условие неперекрывания. Следовательно, она

также является NP-трудной. Этим уже оправдана разработка эвристических алгоритмов.

Метод декомпозиции свободной области МОП на прямоугольные участки

Представление области с препятствиями в матричном виде. Через все грани препятствий проведем сквозные линии (рис. 2). В результате область с препятствиями оказывается покрытой сетью, каждая ячейка которой либо не содержит препятствий (пустая), либо является препятствием или частью препятствия (т. е. нет ячеек, в которых одновременно было бы и препятствие, и пустое место).

Сопоставим каждой ячейке (γ, σ) число

$$a_{\gamma\sigma} = \begin{cases} 0, & \text{если } (\gamma, \sigma) \text{ — свободная ячейка,} \\ 1, & \text{если } (\gamma, \sigma) \text{ — препятствие.} \end{cases}$$

Область с препятствиями будет записана в виде матрицы $A = [a_{\gamma\sigma}]$ с булевыми переменными.

Для выделения прямоугольных областей без препятствий из исходной области будем объединять пустые смежные ячейки. При этом под "смежной" понимается ячейка, имеющая значение индекса на 1 больше, чем у предыдущей.

Выбор исходной ячейки. В качестве исходной ячейки выбираем левую нижнюю ячейку исходной области. Если она является препятствием, движемся вверх по столбцу пока не найдем пустую ячейку, ее принимаем за исходную.

Объединение пустых ячеек. Можно выделить три направления объединения: вертикальное

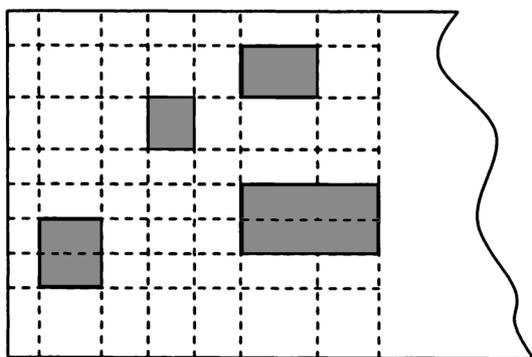


Рис. 2. Покрытие МОП матричной сетью

(рис. 3, а), горизонтальное (рис. 3, б), диагональное (рис. 3, в). При вертикальном способе в качестве ячеек-кандидатов на объединение рассматриваются только смежные ячейки из того же столбца, что и исходная ячейка, при горизонтальном — только смежные ячейки из той же строки, при диагональном способе объединение идет как по строкам, так и по столбцам.

В зависимости от параметров препятствий и их расположения эти способы могут давать различные результаты. Можно использовать разнообразные критерии выбора направления объединения, в частности, здесь также применима мультиметодная технология. Процесс объединения будет продолжаться до тех пор, пока не будет достигнута верхняя граница области с препятствиями.

"Уровневое" выделение пустых прямоугольных областей. Пусть в ходе предыдущих шагов было выделено несколько прямоугольных областей без препятствий и достигнута верхняя граница области с препятствиями (области 1 и 2 на рис. 4, а). Через правую границу области наибольшей длины проведем условный сквозной рез и продолжим выделение областей без препятствий из области справа от реза. Тогда, по аналогии с уровневой технологией, предложенной S. Martello и D. Vigo [6], исходная область с препятствиями окажется разделенной на прямоугольные уровни (рис. 4, а)

При выделении прямоугольников без препятствий в уровнях остаются "боковые пустоты", которые целесообразно использовать. Будем называть их вторичными МОП (рис. 4, б). Для них рекурсивно применяется метод матричной декомпозиции свободной области МОП на прямоугольники. Сформированные прямоугольники и вторичные МОП изображены на рис. 4 светло-серым и белым цветом соответственно, их границы отмечены жирными линиями. Работа алгоритма продолжается до тех пор, пока не останется пустых ячеек.

Сквозное размещение контейнеров в прямоугольной полосе на базе уровневой технологии

Постановка и комбинаторная модель задачи. Частным случаем задачи размещения прямо-

a_{31}	a_{32}	a_{33}	a_{34}
a_{21}	a_{22}	a_{23}	a_{24}
a_{11}	a_{12}	a_{13}	a_{14}

а)

a_{31}	a_{32}	a_{33}	a_{34}
a_{21}	a_{22}	a_{23}	a_{24}
a_{11}	a_{12}	a_{13}	a_{14}

б)

a_{31}	a_{32}	a_{33}	a_{34}
a_{21}	a_{22}	a_{23}	a_{24}
a_{11}	a_{12}	a_{13}	a_{14}

в)

Рис. 3. Способы объединения ячеек: а — вертикальное; б — горизонтальное; в — диагональное

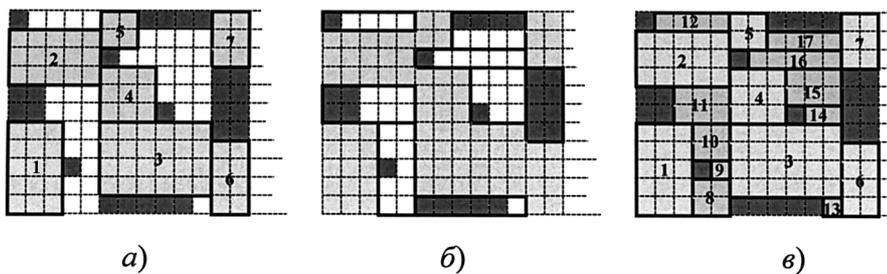


Рис. 4. Уровневое выделения областей без препятствий:

a — первичная декомпозиция на свободные прямоугольники; *б* — выделенные вторичные МОП; *в* — окончательная декомпозиция свободной области МОП на прямоуголь-

угольников в МОП является задача сквозного (гильотинного) размещения предметов в полосе.

В качестве основной будем рассматривать задачу гильотинного размещения предметов в полубесконечную полосу (2-Dimensional Guillotine Strip Placing, **2DGSP**). Имеется прямоугольная полоса заданной ширины W и неограниченной длины, а также набор прямоугольных деталей заданных размеров $w_i, l_i, i = \overline{1, n}$, где n — число прямоугольников, w_i, l_i — ширина и длина i -й детали. Таким образом, исходная информация задачи может быть представлена следующим набором данных:

$$\langle W; m; w = (w_1, w_2, \dots, w_n); l = (l_1, l_2, \dots, l_n) \rangle.$$

Введем прямоугольную систему координат: оси Ox и Oy совпадают соответственно с нижней и боковой сторонами полосы. Положение каждого прямоугольника P_i зададим координатами (x_i, y_i) его левого нижнего угла.

Решение рассматриваемой задачи может быть представлено в виде набора:

$$GP = \langle x = (x_1, \dots, x_i, \dots, x_n), \\ y = (y_1, \dots, y_i, \dots, y_n) \rangle,$$

где x_i, y_i — координаты нижнего угла прямоугольника, $i = \overline{1, n}$.

Набор $GP = (x, y)$ называется *допустимым гильотинным размещением*, если выполняются следующие условия:

1° **Ребра предметов параллельны ребрам полосы:**

$$(\rho_x^i = l_i) \wedge (\rho_y^i = w_i), i = \overline{1, n}, \text{ где } \rho_x^i, \rho_y^i \text{ — про-} \\ \text{екции предмета } i \text{ на оси координат } Ox \text{ и } Oy.$$

2° **Взаимное неперекрывание деталей:** $\forall i \neq j: i, \\ j = \overline{1, n},$

$$((x_i \geq x_j + l_j) \vee (x_j \geq x_i + l_i)) \vee \\ \vee ((y_i \geq y_j + w_j) \vee \\ \vee (y_j \geq y_i + w_i)).$$

3° **Неперекрывание деталей с гра-**

нями полосы: $\forall i = \overline{1, n}$

$$(x_i \geq 0) \wedge (y_i \geq 0) \wedge (y_i + w_i \leq W).$$

Набор $GP = (x, y)$ называется *гильотинным прямоугольным размещением*, если кроме условий 1°, 2°, 3° выполняется

4° **Условие гильотинности:**

для любого прямоугольника P с размерами $(w, l), w \neq w_i \vee l \neq l_i,$

$i = \overline{1, n}$, выполняется условие раз-

деления на два прямоугольника

$P'(w', l')$ и $P''(w'', l'')$ такие, что $((w' = w'' = w) \wedge (l' + l'' = l)) \vee ((w' + w'' = w) \wedge (l' = l'' = l))$ и $\forall i = \overline{1, n}$, если $P_i \cap P' \neq \emptyset$, то $P_i \cap P'' = \emptyset$ и если $P_i \cap P'' \neq \emptyset$, то $P_i \cap P' = \emptyset$.

Задача 2DGSP. При исходных данных $\langle W; m; w; l \rangle$ требуется минимизировать $L(x, y) = \max_{i = \overline{1, n}} (x_i + l_i)$

на множестве размещений $\{GP = (x, y)\}$, удовлетворяющих условиям 1°, 2°, 3°, 4°.

В общей задаче размещения объектов в МОП в качестве полосы рассматривается каждая из прямоугольных областей $P_i, i = \overline{1, m}$, с проверкой на ее ограниченность.

Рассматриваемая задача является NP-трудной, поэтому наряду с точными методами разрабатываются эвристические алгоритмы.

Уровневые алгоритмы используют подход, когда каждый новый элемент упаковывается с выравниванием по левому и нижнему краям. Через правую сторону прямоугольника максимальной длины, заполненного по ширине уровня, проводят сквозную вертикальную линию. Исходный прямоугольный объект оказывается разделенным на уровни прямоугольной формы, содержащие целиком входящие прямоугольники [6, 7]. Известны сходные с уровневыми послойные алгоритмы [8, 9]. Что касается стратегии выбора прямоугольников, то она может исходить из известных правил. При этом различаются конструктивные и эволюционные уровневые алгоритмы.

Конструктивные эвристики. Конструктивные эвристики можно использовать самостоятельно для решения задачи за один проход алгоритма или в качестве декодеров в рамках эволюционных или других метаэвристик. Обзор простых конструктивных эвристик подробно представлен в работе [10]. Здесь мы воспользуемся стратегией первый подходящий (FF).

Стратегия первый подходящий состоит в следующем:

Пусть все предметы упорядочены согласно списку p .

Шаг 1. Первый предмет помещаем в первый уровень и исключаем из списка π .

Шаг 2. На k -м шаге в текущий уровень помещаем первый из списка π предмет; если это невозможно, то размещаем в уровне первый подходящий предмет, иначе открываем новый уровень и переходим на шаг 1.

Цикл продолжается, пока $\pi \neq \emptyset$. При этом каждый новый предмет размещается с выравниванием по левому нижнему краю.

Одной из результативных модификаций конструктивной эвристики FF является первый подходящий по убыванию длины (First Fit Decreasing Length, FFDL). Согласно этому методу сначала все детали сортируются в порядке невозрастания длины l_i и в соответствии с полученным списком размещаются в полосу. Результат работы алгоритма всегда одинаков и зависит только от исходных данных и порядка их размещения в списке π .

Эволюционная гиперэвристика. Методы эволюционных вычислений используют конструктивные эвристики в процессе построения решений в качестве декодеров. На каждой итерации в соответствии с выбранной стратегией конструируется допустимое решение, которое сопоставляется с полученным на предыдущих итерациях рекордом. Процесс продолжается до выполнения критерия останова: затраченное время, число итераций либо достижение заданного значения целевой функции. Размещения, полученные при повторных запусках эволюционных алгоритмов, могут различаться при одних и тех же исходных данных.

Рассмотрим одноточечный эволюционный алгоритм (1 + 1)-EA с тремя способами мутации перехода от одной особи к следующей:

- (1 + 1)-EA(m) — наивный эволюционный алгоритм;
- (1 + 1)-EA(2) — эволюционный алгоритм поиска решения в окрестности;
- (1 + 1)-EA(SVC) — коррекция списка π , направленная на улучшение решения.

Первые два алгоритма подробно описаны в литературе, например в работе [10]. На каждой *простой* итерации (*интенсификации*) алгоритмов (1 + 1)-EA(m) и (1 + 1)-EA(2) текущий список π преобразуется путем перестановки всех или двух случайно выбранных элементов. Таким образом, формируется особь из окрестности начального решения. На *обобщенной* итерации (*диверсификации*) происходит возврат к исходному списку π .

Принципиально отличается от этих двух методов мутации алгоритм (1 + 1)-EA(SVC). Его опишем подробнее. Он использует для изменения списка π *метод последовательного уточнения оценок* (Sequential Value Correction, SVC). Под SVC понимается не конкретный метод, а направленная технология изменения списка π . Различные

алгоритмы на базе этой технологии представлены в работах [9, 12, 13 и др.]. Здесь используется простейшая версия SVC.

Метод последовательного уточнения оценок (SVC). Он предназначен для выполнения процедуры мутации на каждой итерации эволюционной метаэвристики (1 + 1)-EA.

На начальной итерации список π строится по невозрастанию длин предметов, и для него применяется алгоритм FF конструирования уровня упаковки (рис. 5).

На процедуре мутации вычисляются оценки предметов по формуле

$$y_i^0 = Ww_i / (W - h_j), i \in I_j, j = \overline{1, n},$$

где n — число уровней; I_j — множество предметов, размещенных в уровне j ; h_j — ширина остатка j -го

уровня. Величины y_i^0 имеют смысл попредметных норм занимаемой части полигона. В списке π предметы упорядочиваются по невозрастанию их оценок.

На момент выполнения мутации на k -й итерации имеем оценки y_i^{k-1} и информацию об уровне упаковки. В качестве новых оценок примем среднее арифметическое между прежними оценками и новыми попредметными нормами, т. е.

$$y_i^k = \frac{1}{2} \left(y_i^{k-1} + \frac{Ww_i}{W - h_j^{k-1}} \right), i \in I_j^k.$$

Далее строится новый список π .

В ходе численного эксперимента было проведено сравнение эволюционных эвристик (1 + 1)-EA(m), (1 + 1)-EA(2) и (1 + 1)-EA(SVC). Мутация SVC позволяет получать решения лучше, чем со случайными перестановками элементов в π .

В последние годы появились новые эвристические методы под названием *гиперэвристики*. Их применение позволяет расширить область локального поиска [11]. Простейшей реализацией гиперэвристики является генетический метод с комбинированием эвристик И. П. Норенкова [2] и его модификация под названием мультиметодный алгоритм [3]. Разные эвристики имеют различные

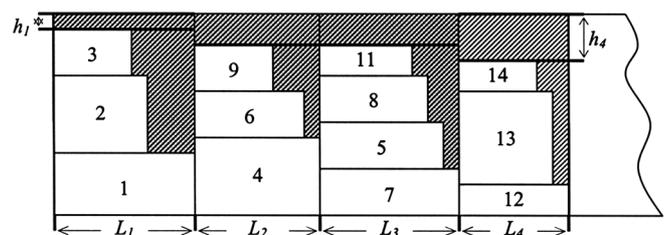


Рис. 5. Уровневая гильотинная упаковка

сильные и слабые стороны, гиперэвристики комбинируют их таким образом, чтобы каждая из них покрывала слабости других. Кроме того, гиперэвристики должны быть простыми в использовании. В формате гиперэвристик реализован алгоритм (1 + 1)-EA.

Заполнение боковых пустот. Назовем незаполненные области каждого уровня справа от размещенных прямоугольников *боковыми пустотами*.

Пусть имеется некоторое частичное размещение. Будем считать, что боковые пустоты текущего уровня k еще не заполнены (рис. 5).

Обозначим: π — исходный список элементов, π^+ — множество элементов исходного списка, размещенных в полосе, π^- — множество неразмещенных элементов исходного списка, т. е. $\pi = \pi^- \cup \pi^+$, $\pi^- \cap \pi^+ = \emptyset$.

Сначала необходимо выделить пустые прямоугольные области, которые можно гильотинно заполнить оставшимися прямоугольниками.

Будем различать горизонтальный и вертикальный способы выделения боковых пустот. Что касается их заполнения, то для выполнения этой процедуры воспользуемся полным перебором неразмещенных деталей. Для выбора одной из них определяем критерий размещения. Обе эти процедуры принадлежат мультиметодной технологии [2].

Горизонтальный способ выделения пустот заключается в том, что в качестве незанятой области рассматривается свободное пространство справа от каждого предмета в частичном размещении. При этом ширина пустой области равна ширине предмета, а длина — разности длины уровня и длины предмета (рис. 6, а).

Вертикальный способ выделения пустот предполагает, что ранее уже были выделены горизонтальные области, способ основан на их преобразовании. Горизонтальные пустоты рассматриваются в порядке невозрастания длин расположенных слева от них деталей. Для каждой пустой горизонтальной области ищутся смежные пустоты и исследуется целесообразность их объединения в одну вертикальную область (рис. 6, б).

Далее, следуя мультиметодной технологии, случайно определяем тип пустоты и размещаем в ней предмет, выбранный полным перебором из π^- (рис. 7).

В результате численного эксперимента [7] было показано, что введение процедуры заполнения

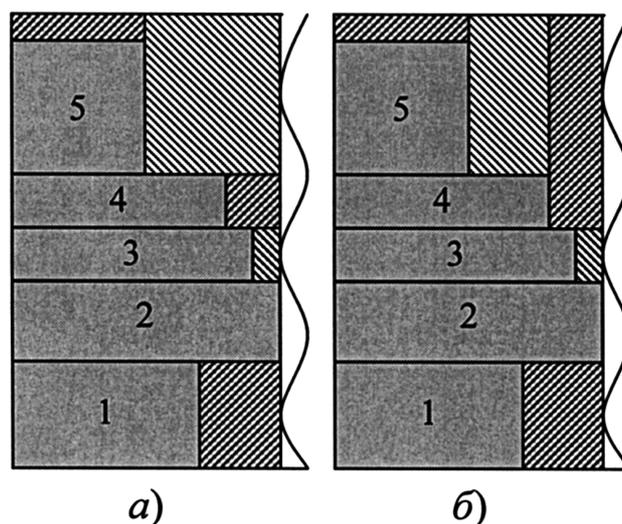


Рис. 6. Способы выделения боковых пустот: а — горизонтальный способ; б — вертикальный способ

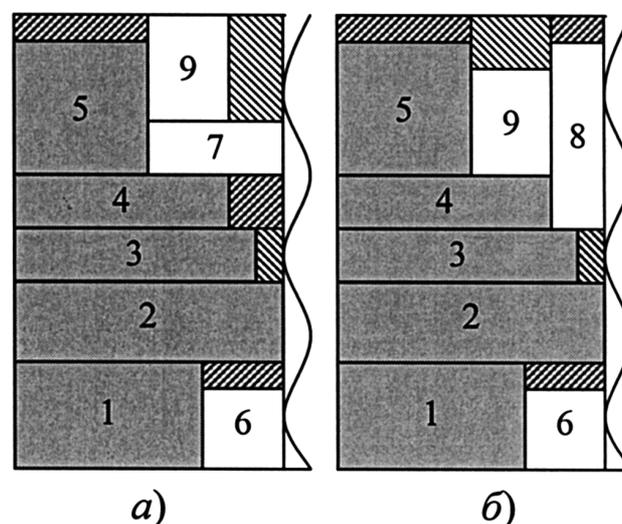


Рис. 7. Заполнение выделенных боковых пустот: а — заполнение горизонтальных пустот; б — заполнение вертикальных

боковых пустот позволяет существенно улучшить результаты уровневых алгоритмов.

Заключение

В статье приведена постановка задачи размещения крупногабаритных грузов в отсеках самолетов, на палубах судов, железнодорожных и автомобильных платформах и других транспортных средствах. Она описывается моделью гильотинного размещения предметов в МОП. Расчет карты размещения ведется только для одного транспортного средства. В статье рассмотрен алгоритм сквозного выделения прямоугольных боксов из многосвязного ортогонального полигона. Метод основан на представлении исходной области в виде матрицы и использует идеи мультиметодной и

уровневой технологий. Для расчета размещения грузов в выделенных боксах подходящим оказался уровневый подход. Однако при несомненном удобстве гильотинного размещения он оставляет в каждом уровне много свободного места. Предложена модификация уровневой технологии, сводящаяся к последовательному заполнению боковых пустот, образованных размещением предметов в данный уровень. На этом этапе предложено использовать процедуры комбинирования эвристики: случайный выбор направления формирования пустот и полный перебор предметов для его заполнения. Для решения задачи в целом используется комбинация алгоритмов мутации случайной перестановки и последовательного уточнения оценок. Ее реализация показала результаты, превосходящие решение задачи с одним из алгоритмов мутации. Данный комплекс позволяет решать различные задачи раскроя, ортогональной упаковки и покрытия. Приведенный алгоритм программно реализован и проведены тестовые расчеты.

Список литературы

1. **Masek W. J.** Some NP-Complete Set Covering Problems. Manuscript, MIT, 1979.
2. **Норенков И. П.** Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. 1999. № 1. С. 2—7.
3. **Валиахметова Ю. И., Филиппова А. С.** Мультиметодный генетический алгоритм для решения задач ортогональной упаковки // Информационные технологии. 2007. 12 (136). С. 50—57.
4. **Culberson J. C. and Reckhow R. A.** Covering Polygons is Hard // Journal of Algorithms. 1994. N 17. P. 2—44.
5. **Berman P. and Dasgupta B.** Approximating Rectilinear Polygon Cover Problems // Algorithmica. 1997. 17 (4). P. 331—356.
6. **Lodi A., Martello S., Vigo D.** Recent advances on two-dimensional bin packing problems // Discrete Applied Mathematics. 2002. N 123. P. 379—396.
7. **Мухачева Э. А., Хасанова Э. И.** Гильотинное размещение контейнеров в полосе: комбинирование эвристических технологий // Информационные технологии. 2009. № 11. С. 8—14.
8. **Adamovich A., Albano A.** Nesting two-dimensional shapes in rectangular Modules // Comput. Aided Design. 1976. N 8 (1). P. 27—33.
9. **Мухачева Э. А., Верхотуров М. А., Мартынов В. В.** Модели и методы расчета раскроя-упаковки геометрических объектов // Уфа: Изд. УГАТУ. 1998. 216 с.
10. **Филиппова А. С.** Проблемы декодирования прямоугольных упаковок: краткий обзор современных технологий // Информационные технологии. 2005. № 12. С. 13—20.
11. **Burke E. K., Mac Carty B. L., Petrovic S. and Gu R.** Knowledge Discovery in a Hyperheuristic for Course Timetabling using Case Based Reasoning. To appear in the Proceeding of the Fourth International Conference on the Practice and Theory of Automated Timetabling (PATAT'02) Ghent, Belgium. August. 2002.
12. **Мухачева Э. А., Мухачева А. С., Белов Г. Н.** Метод последовательного уточнения оценок: алгоритм и численный эксперимент для задачи одномерного раскроя // Информационные технологии. 2000. № 2. С. 11—17.
13. **Belov G., Scheithauer G., Mukhacheva E. A.** One-dimensional heuristics adapted for two-dimensional rectangular strip packing // Journal of the Operational Research Society. 2008. N 59. P. 823—832.

МОДЕЛИРОВАНИЕ

УДК 004.4

В. А. Зелепухина, канд., техн. наук, доц., Астраханский государственный университет,
e-mail: viktorina_82@mail.ru

Разработка интернет-ориентированных виртуальных лабораторий математического моделирования посредством разделения вычислительных и визуализационных задач

Описан способ разработки виртуальных лабораторий математического моделирования для Интернет, основанный на концепции разделения вычислительных и визуализационных процессов между серверной и клиентской сторонами. Предлагаемая методика базируется на использовании двух взаимодополняемых технологий: веб-сервисов специализированных математических пакетов для численных расчетов и Java-технологии для визуализации результатов моделирования.

Ключевые слова: виртуальные лаборатории, визуализация данных, математическое моделирование, компьютерное моделирование, веб-технологии, Java, MATLAB

Введение

В настоящее время одной из перспективных форм организации и проведения совместных научных исследований и реализации дистанционного обучения является создание информационно-вычислительных веб-ресурсов и виртуальных лабораторий (ВЛ) удаленного доступа, так как Интернет занял прочные позиции в современном научном сообществе и является важным средством коммуникации, обмена научной информацией и публикации научных результатов.

Несмотря на значительное число работ, посвященных разработке инновационного научно-исследовательского программного обеспечения [1, 2, 3], рынок интернет-технологий для создания виртуальных лабораторий находится еще в стадии становления. Разрабатываемые в настоящее время интернет-ориентированные виртуальные лаборатории реализуются с применением исключительно либо серверных (веб-сервисы специализированных математических пакетов), либо клиентских технологий (Java-апплеты), что не позволяет одновременно предоставить пользователю всю мощь вычислительных систем и интерактивную высококачественную визуализацию с гибкими возможностями настройки и взаимодействия с пользователем.

С одной стороны, обрабатывая большие массивы данных при численном моделировании, специализированные математические пакеты генерируют анимацию в виде серии растровых изображений, что становится бессмысленным вследствие существенного увеличения нагрузки на сеть и загрузки процессора конечного пользователя. Суммарный объем графического представления данных может во много раз превосходить объем исходной числовой информации. Включение дополнительных элементов в изображение в целях повышения его наглядности увеличивает размер конечного файла. С другой стороны, Java-технология, напротив, имеет мощные средства разработки интерактивной графики и анимации, но, одновременно, не позволяет проводить сложные математические расчеты вследствие "бедного" математического аппарата.

Описываемая в статье технология имеет практическую реализацию в виде специализированной системы визуализации данных и позволяет отображать данные в векторном формате, в том числе и анимацию, с возможностями интерактивного взаимодействия пользователя с изображением.

1. Тенденции разработки виртуальных лабораторий математического моделирования

В настоящее время прослеживаются две тенденции разработки интернет-ориентированных виртуальных лабораторий: с использованием соответствующих средств специализированных математических пакетов (веб-сервисов) и с применением клиентских технологий, в частности Java-технологии (апплеты).

1.1. Средства специализированных математических пакетов

Веб-сервисы специализированных пакетов математического моделирования (MATLAB Web Server¹, MapleNet, webMathematica, Mathcad Application Server и др.) позволяют развернуть приложение в сети с использованием стандартных веб-технологий, тем самым добавив численные и аналитические расчеты и визуализацию полученных результатов на веб-сайт. Для работы с пакетами не требуется установка программного комплекса на стороне конечного пользователя: клиент взаимодействует с программой посредством стандартного веб-обозревателя, команды которого обрабатываются запущенной на стороне сервера сессией математического пакета (MATLAB, Maple, Mathematica, Mathcad, соответственно) [3, 4]. Подобные веб-сервисы содержат не только средства обработки html-форм (чтение данных, возврат полученных данных в форму), но и могут генерировать статические графики в виде растровых изображений.

При использовании веб-сервисов математических пакетов для разработки виртуальных лабораторий главным достоинством, вероятно, является то, что конечный пользователь получает в свое распоряжение всю мощь встроенных численных методов [4, 5].

Основные недостатки при использовании веб-сервисов математических пакетов.

1. Возможности визуализации в основном определяются не возможностью самого математического пакета, а возможностями веб-интерфейса: степень интерактивности при изучении компьютерной модели зависит от дополнительных усилий, направленных на разработку и внедрение JavaScript-сценариев для растровых изображений, реагирующих на пользовательские запросы. При использовании таких сценариев команды от пользователя передаются серверу с установленным ма-

¹ MATLAB Web Server более не поддерживается компанией MathWorks, которая предлагает в качестве альтернативы продукты MATLAB Builder for Java и MATLAB Builder for .NET, позволяющие преобразовывать программу MATLAB в код на другом языке (в частности Java).

тематическим пакетом, и перегружается либо веб-страница полностью, либо ее часть (в случае использования технологии AJAX). Отсюда, как следствие, — существенная потеря времени. Ведь даже при стандартных манипуляциях с изображениями (например, операция масштабирования) потребуется повторный вызов серверного приложения для изменения параметров функции, отвечающей за генерирование графика по соответствующим координатам.

2. При оптимизации или сжатии растровых файлов в целях минимизации их размера происходит значительная потеря качества изображения.

3. Производительность работы с программой во многом определяется пропускной способностью сети, так как в процессе работы происходит обмен данными между клиентским компьютером и сервером.

Такие продукты как MATLAB Builder for Java позволяют компилировать код MATLAB в код на другом языке. Таким образом, можно организовать Java-библиотеки и для вычислений, и для визуализации. Просто, но дорого, как и все перечисленное выше программное обеспечение. Зачастую для реализации тех или иных виртуальных лабораторий, особенно для образовательных целей, не требуется всей мощи встроенного математического аппарата. Отсюда, как следствие, переплата за неиспользуемые модули и объекты.

Более подробно об опыте применения веб-средств математических пакетов для решения научных и прикладных задач можно узнать в работах [3—7].

1.2. Java-технология

Современным средством разработки виртуальных лабораторий для сети Интернет является Java-технология от Sun Microsystems, имеющая такие достоинства как платформенная независимость и сетевая направленность. Она позволяет создавать апплеты — приложения, встраиваемые в веб-документы и выполняемые за счет виртуальной машины Java (VJM, Virtual Java Machine), встроенной в веб-обозреватель. Java-технология все чаще и чаще используется как основа для разработки серьезных проектов в различных областях: в производстве, банковской сфере, коммерческой деятельности и научных исследованиях.

Главным достоинством Java-апплетов является тот факт, что все операции происходят на компьютере конечного пользователя, и после первоначальной загрузки апплета он функционирует в автономном режиме: работа в программе осуществляется непосредственно на том компьютере, где нужны результаты выполнения этой программы, что значительно сокращает трафик в сети.

Преимущества использования Java-апплетов по отношению к другим технологиям создания виртуальных лабораторий.

1. Java-апплеты работают на стороне клиента, следовательно, быстрее серверных приложений.

2. Апплеты могут взаимодействовать с другими апплетами, с программами на стороне сервера и с XML-документами.

3. Java имеет широкий набор элементов графического пользовательского интерфейса (GUI, Graphics User Interface), что обеспечивает более эффективное взаимодействие с пользователем.

4. Пользователи могут работать с программой, используя различные типы архитектуры ЭВМ и любые веб-обозреватели, оснащенные Java-машинной, причем информация поставляется в одном и том же виде и формате.

Основные недостатки использования Java для разработки виртуальных лабораторий.

1. Отсутствие стандартных классов, реализующих основные численные методы, порождает необходимость разработки соответствующих библиотек и потери времени при загрузке приложения за счет увеличения объема программы. Вероятно именно по этим причинам в Интернет представлены, в основном, простейшие модели физических процессов, охватывающих только школьную программу.

2. Работа по созданию проблемно-ориентированных библиотек на Java не всегда обоснована, поскольку имеется теоретическая возможность использования стандартных библиотек на Fortran (The NAG Fortran Library, IMSL Fortran Numerical Library), C (The NAG C Library, IMSL C Numerical Library) и т. д.

3. Быстродействие программы в существенной степени зависит от мощности компьютера клиента.

Сравнение функциональных возможностей веб-сервисов специализированных математических пакетов и Java-технологии при решении научных задач приведено в табл. 1.

2. Общая концепция предлагаемого метода

Виртуальным лабораториям удаленного доступа присущи следующие особенности:

- время одного сеанса работы пользователя превышает время, необходимое для перекачки по сети кода программ визуализации;
- объем данных, необходимых для построения изображения, меньше объема готового растрового изображения;
- в течение одного сеанса работы пользователь многократно обращается к визуализации данных в различных представлениях;

Таблица 1

Сравнение функциональных возможностей веб-сервисов специализированных математических пакетов и Java-технологии при решении научных задач

Веб-сервисы математических пакетов	Клиентские технологии (Java)
Обладают эффективной, надежной и проверенной библиотекой численных методов. Обеспечивают стандартные методы решения разнообразных систем (обыкновенные дифференциальные уравнения (ОДУ), дифференциальные уравнения в частных производных (ДУЧП), в том числе системы типа "реакция—диффузия", уравнения с запаздыванием и т. д.)	При решении сложных систем приходится либо разрабатывать библиотеки численных методов самостоятельно, либо прибегать к авторским средствам сторонних разработчиков
Результаты моделирования выдаются в виде растровых изображений	Мощные средства разработки интерактивных апплетов, встраиваемых в обозреватель, с возможностью анимации

- для построения одного изображения требуется комбинация из нескольких наборов данных, и число таких комбинации существенно превосходит число основных наборов.

Учитывая каждую из особенностей, предлагается использование стандартных математических пакетов (MATLAB, Maple, Mathematica, Mathcad, Scilab, JFreeMath и др.), имеющих возможность работы в сети Интернет, непосредственно для выполнения вычислений. Результаты расчета математической модели (ММ) логично выдавать исключительно в виде числовых массивов данных, а не растровых изображений, как это происходит обычно. После передачи данных клиенту происходит их визуализация в виде векторного изображения (анимации) с возможностями интерактивного взаимодействия с данными и визуализацией за счет использования Java-апплета (рис. 1).

Кроме того, что реализация визуализации в виде специализированной системы на стороне клиента лишает математическое программное обеспечение указанных выше недостатков, при таком решении сокращается сетевой трафик, уменьшается нагрузка на сервер и время реакции приложения на запросы пользователя.

По сравнению со стандартными подходами, предлагаемый способ **позволяет сократить объем передаваемой по сети информации** и проводить визуализацию полученных данных на клиентской стороне, не привязываясь к достаточно ограниченным возможностям визуализации стандартных математических пакетов и библиотек.

Использование специализированной системы визуализации на стороне клиента **позволяет сконцентрироваться на вычислительной части задачи**, ведь множество операций повторяются от реали-

зации к реализации и не связаны со спецификой задачи, а являются рутинными операциями графического вывода.

3. Модели визуализации и представления данных результатов математического моделирования

Системам визуализации присущи свойства типичности не только пользовательского взаимодействия с данными (импорт данных, экспорт данных, пользовательский визуальный запрос и т. д.), но и построения на экране непосредственно изображения. Поэтому в разработанной системе для визуализации результатов моделирования используются правила и декларативное описание объектов, содержащиеся в модели взаимодействия математического пакета с системой визуализации и в модели представления результатов математического моделирования. Система визуализации унифицирована по отношению к различным типам математических моделей, так как визуализация объектов проводится динамически по соответствующей модели представления и правилам, описанным в ней.

Интересующие параметры математической модели пользователь указывает посредством веб-интерфейса, который автоматически генерируется по соответствующей модели представления: наличие четко определенных операций с данными, в данном случае, параметрами модели (отправить данные на сервер для расчета математической модели, принять данные на стороне клиента для визуализации), позволяет типизировать веб-интерфейс и запросы в целях автоматизации процесса генерации веб-интерфейса и соответствующих *html*-форм. Программа импортирует *XML*-представление модели данных, проводит анализ и генерирует веб-интерфейс с управляющими элементами.

Для обработки сгенерированной *html*-формы на сервер передается ассоциированный массив с

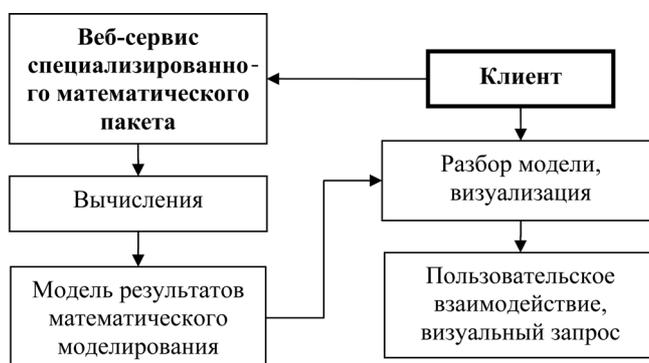


Рис. 1. Схема интеграции расчетов со стороны специализированного математического пакета с визуализацией на стороне клиента

именами элементов, соответствующих именам переменных формы и со значениями, соответствующими значениям элементов той же формы. Анализируя данный массив, программа автоматически перенаправляет расчетные параметры в соответствующий вычислительный модуль, путь к которому также указан в XML-представлении.

Основой модели представления результатов математического моделирования является совокупность различных визуализируемых объектов, для которых выполняется расчет координат в вычислительной системе, с присущими им атрибутами (цвет, форма, размер и т. д.). Интеграция всех описанных объектов с типом графического отображения позволит визуализировать исследуемый процесс. Объектом верхнего уровня является *математическая модель*, включающая в себя одну или несколько групп объектов. Массивы данных предназначены для хранения координат объектов. Для составления сцены объекты математической модели объединяются в *наборы данных*, которые визуализируются в разных окнах. Группа представляет собой множество поименованных объектов. Любой объект может входить в состав произвольного числа групп.

Изображение является иерархической структурой и состоит из следующих базовых элементов изображения: многоугольников, ломаных линий, секторов эллипсов, растровых изображений.

Каждый объект, входящий в состав математической модели, имеет первичный ключ, или уникальный идентификатор, который необходим для генерации графического представления и обработки пользовательских запросов. Значение идентификатора задается в декларативном описании объекта. Также заранее определяются координаты каждого объекта в определенный момент времени, параметры его маркера и соединительной линии с другими объектами. Если некоторый объект связан с другим объектом, то ссылка на объект представляется дополнительным ат-

рибутом в модели, который по своему смыслу является вторичным ключом. Непосредственно для совокупности объектов в составе исследуемого процесса определяются параметры визуализации и параметры вычислительной программы для последующего к нему обращения. Иерархию составляющих модели представления результатов математического моделирования можно посмотреть на рис. 2.

Модель визуализации распределена между тремя участниками процесса: удаленный пользова-



Рис. 2. Модель представления результатов математического моделирования

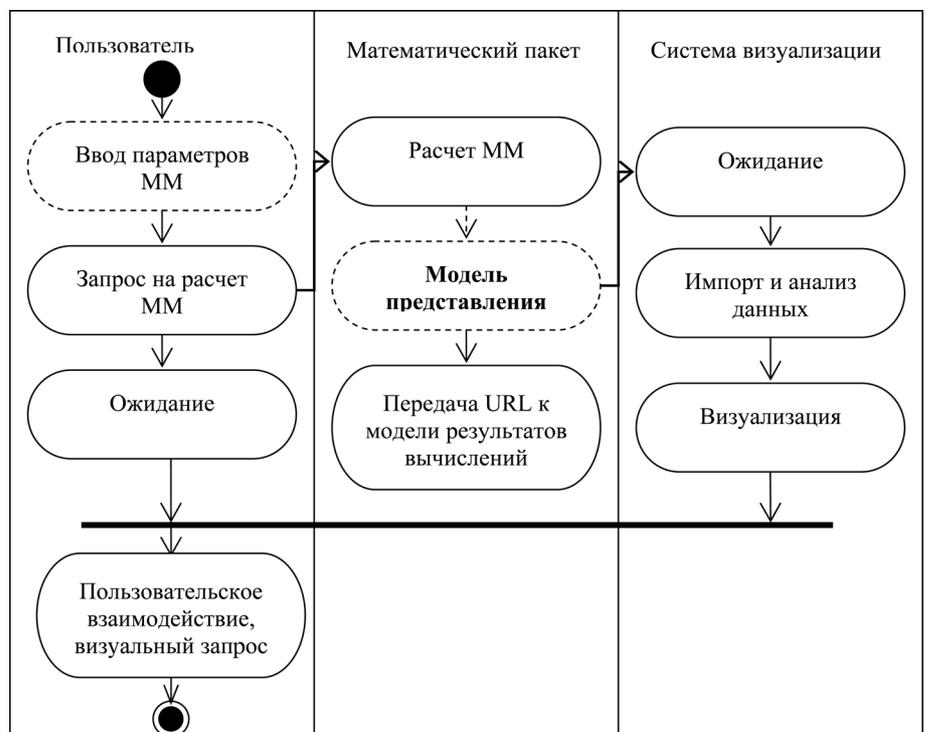


Рис. 3. Модель обмена и визуализации

тель; специализированный математический пакет; специализированная система визуализации данных (рис. 3). Источником данных может быть не только специализированная система математического моделирования, но и удаленная реально существующая установка, данные из СУБД (система управления базами данных) или же композиция данных этих источников.

Система визуализации и удаленный источник данных работают автономно, независимо друг от друга, находятся в режиме ожидания до поступления команды на визуализацию и расчет, соответственно. Вызов математического пакета осуществляет не удаленный пользователь непосредственно, а программа, автоматически составляющая запрос к системе математического моделирования.

Математический пакет, получив команду на выполнение вычислений, моделирует процесс по заданным пользователем параметрам, и в итоге своей работы на выходе не выдает ничего, кроме

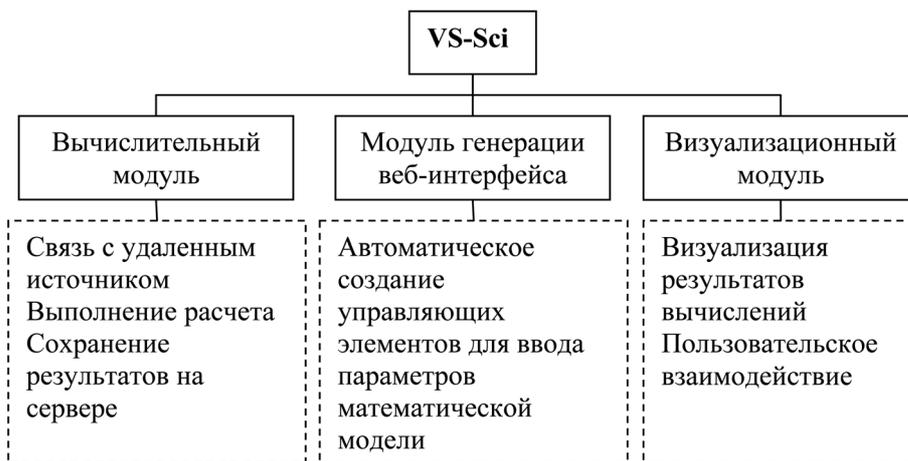


Рис. 5. Структурно-функциональная модель VS-Sci

пути к результатам математического моделирования, сохраненным на жестком диске сервера (рис. 3, 4). Система визуализации импортирует указанные вычислительной программой данные, анализирует XML-файлы, создает объекты и запускает процесс визуализации.

4. Практическая реализация подхода

В Астраханском государственном университете на основе предложенной технологии разработан программный комплекс VS-Sci, в котором представление характеристик исследуемых процессов основано на применении интерактивной векторной графики и анимации с возможностью пользовательского взаимодействия (рис. 5).

Основные задачи *вычислительного модуля*: проверка переданных пользователем параметров задачи, решение задачи, запись результатов вычислений на сервер в формате представления результатов математического моделирования, вывод URL-пути к XML-конфигурации результатов математического моделирования. После завершения процесса вычислений *визуализационный модуль* импортирует данные, разбирает их и строит визуализацию исследуемого процесса.

Веб-интерфейс, содержащий параметры математической модели и JavaScript-обработчики, автоматически генерируется по модели представления параметров математической модели (рис. 6).

Программный комплекс VS-Sci для визуализации данных, полученных с удаленных веб-серверов специализированных математических пакетов, был апробирован на виртуальной лаборатории нелинейной динамики и некоторых математических моделях живых систем² (рис. 7–9).

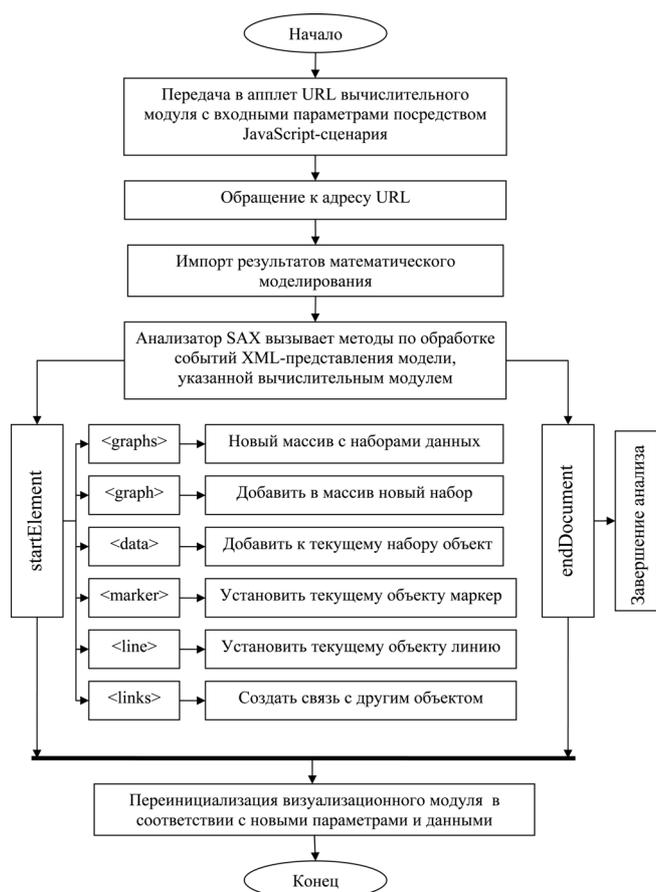


Рис. 4. Блок-схема алгоритма работы визуализационной системы

² Примеры реализации компьютерных моделей с использованием программного комплекса VS-Sci находятся на сайте <http://mathmod.aspu.ru>.

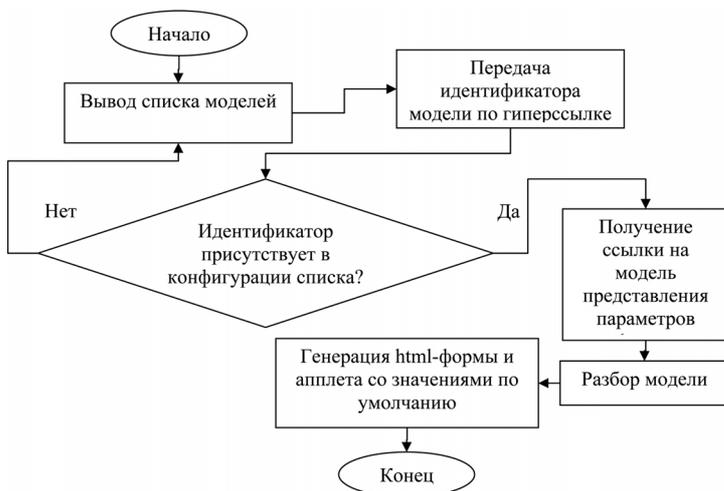


Рис. 6. Блок-схема алгоритма генерации веб-интерфейса

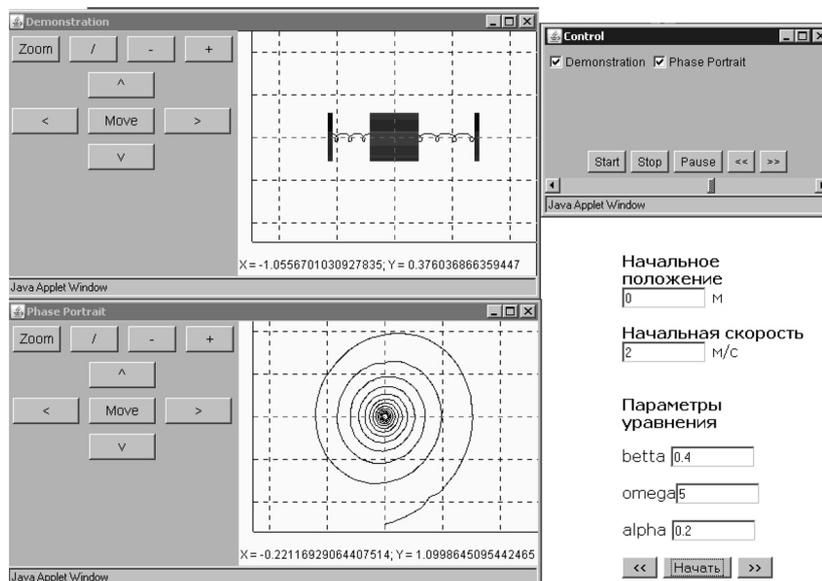


Рис. 7. Интерфейс модели "Механическая система" с параметрами $\alpha = 0,2$, $\beta = 0,4$, $\omega = 5$, $x_0 = 0$, $v_0 = 2$

тельно генерирует анимацию по исходным конфигурации и числовым данным и предоставляет встроенные средства для взаимодействия пользователя с изображением.

Заключение

Описанный в статье способ и разработанная система визуализации основаны на использовании двух взаимодополняемых технологий и повышают эффективность виртуальных лабораторий, в частности:

- существенно расширяют графические возможности, а также увеличивают быстродействие системы, поскольку единожды полученные данные могут быть использованы на клиентской стороне многократно для получения различных графических представлений;
- сокращают объем пересылаемой информации и тем самым увеличивают быстродействие процесса и снижают нагрузку сети, так как вместо пересылки изображений, что типично для веб-сервисов математического программного обеспечения, передаются только файлы с числовыми данными;
- позволяют создавать модели, расчет которых выполняется

Расчеты моделей выполнены с использованием средств пакета MATLAB. Применение разработанной системы существенно эффективнее стандартных методов (табл. 2), так как уменьшаются общее время загрузки страницы с результатами математического моделирования и время отклика на запрос пользователя. Кроме того, система визуализации самостоя-

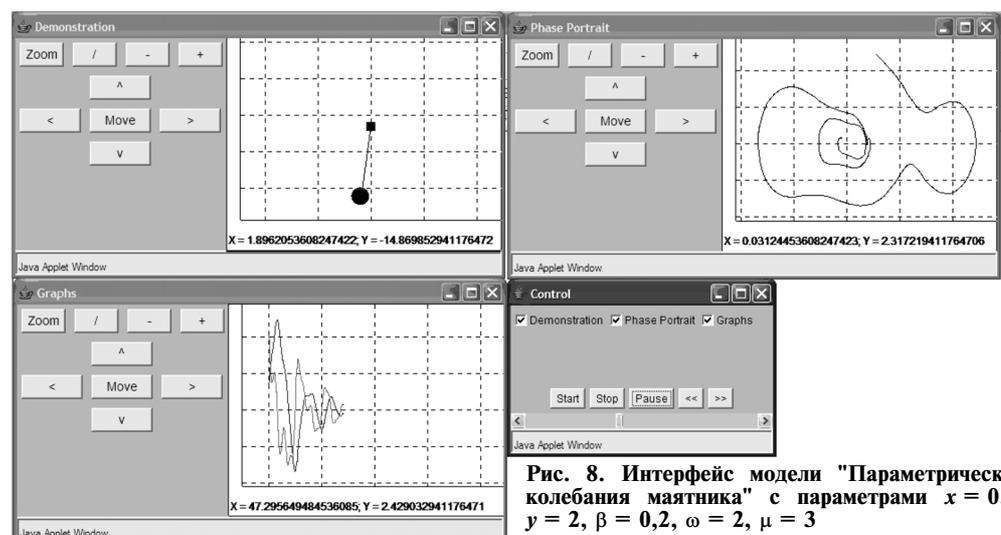


Рис. 8. Интерфейс модели "Параметрические колебания маятника" с параметрами $x = 0,5$, $y = 2$, $\beta = 0,2$, $\omega = 2$, $\mu = 3$

Сравнительная характеристика реализации виртуальных лабораторий на основе MATLAB Web Server и реализации на основе MATLAB Web Server+VS-Sci

Характеристика приложения	MATLAB Web Server	MATLAB Web Server+VS-Sci
Возможность повторного использования результатов вычислений	—	+
Возможность масштабирования без повторного обращения к серверу	—	+
Возможность фоновой подкачки данных	—	+
Возможность пользовательского взаимодействия с изображением	—	+
Перезагрузка веб-страницы при обращении к серверу для проведения расчетов	+	—
Время загрузки приложения, с		
Модель механической системы	4	2
Параметрические колебания математического маятника	3	3
Колебания двух связанных осцилляторов	14	3
Объем пересылаемой информации, Кбайт		
Модель механической системы	41	18
Параметрические колебания математического маятника	43	26
Колебания двух связанных осцилляторов	10 035	23

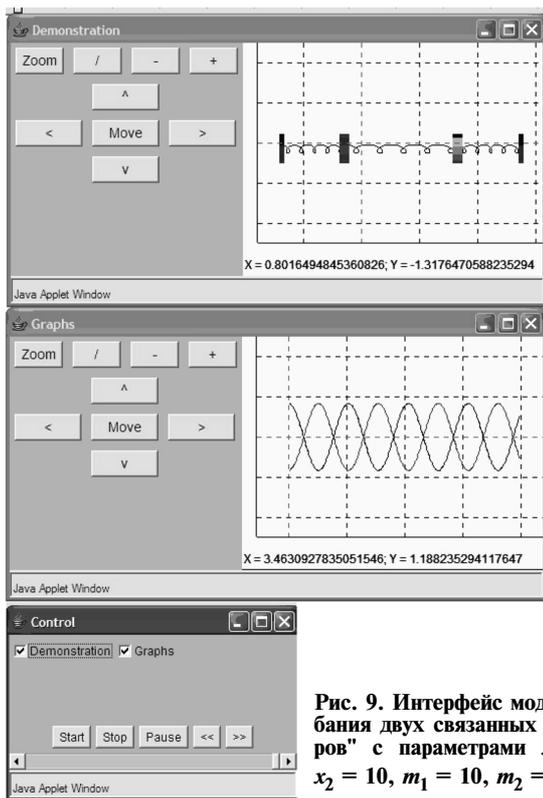


Рис. 9. Интерфейс модели "Колебания двух связанных осцилляторов" с параметрами $x_1 = -10$, $x_2 = 10$, $m_1 = 10$, $m_2 = 10$

на различных вычислительных кластерах, вследствие независимости визуализации от источника данных.

Разработка технологии и программного обеспечения на ее основе получила финансовую поддержку Российского фонда фундаментальных исследований (проект РФФИ № 07-07-00128-а) и Фонда содействия развитию малых форм предприятий в научно-технической сфере (проект У. М. Н. И. К. "Интернет-ориентированная кроссплатформенная система визуализации данных для сопряжения со специализированными пакетами научного программного обеспечения и базами данных").

Список литературы

1. **Konzes R. T., Myers J. D., Wolf W. A.** Collaboratories: doing science on the Internet // IEEE Computer. 1996. № 29.

2. **Vdovitsyn V. T., Tarasov V. V.** Multi-Agent System for International Support of Collaborative Researchers Work in a Computer Network // Proc. Of FDPW'99. Developments in Distributed Systems and Data Communications. Petrozavodsk, 1999. P. 139–145.

3. **Пономарева И. С., Зелепухина В. А., Тарасевич Ю. Ю., Манжосова Е. Н., Панченко Т. В.** Применение веб-технологий в физическом практикуме // Физическое образование в вузах. 2006. Т. 12, № 1. С. 103–114.

4. **Пономарева И. С., Зелепухина В. А., Тарасевич Ю. Ю.** Некоторые аспекты создания web-приложения на базе MATLAB Web Server // Информационные технологии. № 9. 2006. С. 68–72.

5. **Пономарева И. С., Зелепухина В. А., Тарасевич Ю. Ю.** Разработка приложений для MATLAB Web Server // Компьютерные инструменты в образовании. 2005. № 4. С. 48–56.

6. **Одинцов В. И., Конрадов А. А., Кукса Ю. И.** Применение MATLAB Web Server в геофизике для интерактивной адаптивной обработки данных, распределенных в сети Интернет // Труды II Всероссийской научной конференции "Проектирование научных и инженерных приложений в среде MATLAB". — М.: ИЛУ РАН, 2004. С. 1821–1835.

7. **Котельников И. А., Черкасский И. А.** MATLAB Web Server: Вычисления в Интернете // Exponenta Pro. Математика в приложениях. 2004. № 1 (4). С. 4–11.

Ю. Ю. Клосс^{1, 2}, д-р физ.-мат. наук, нач. лаб.,
 Д. В. Мартынов^{1, 2}, студент,
 e-mail: mart1989@list.ru
 Ф. Г. Черемисин^{1, 3}, д-р физ.-мат. наук,
 гл. научн. сотр.,

¹ Московский физико-технический институт,
 г. Долгопрудный, М. О.

² Российский научный центр
 "Курчатовский институт", Москва

³ Вычислительный центр имени
 А. А. Дородницына РАН, Москва

Разработка методов компьютерного моделирования и анализа микронасоса Кнудсена

Проведено компьютерное моделирование движения газа в насосе Кнудсена на основе численного решения уравнения Больцмана. Оператор переноса аппроксимировался методом конечных объемов с использованием тетраэдрических сеток. Интеграл столкновений вычислялся проекционным методом. Проведено параметрическое исследование работы насоса при различных числах Кнудсена.

Ключевые слова: проекционный метод, неструктурированные сетки, метод конечных объемов, уравнение Больцмана, насос Кнудсена

Введение

Кинетическое уравнение Больцмана позволяет изучать движение разреженного газа и определять его макропараметры. Однако теоретически решить кинетическое уравнение ввиду его сложности возможно лишь для узкого ряда задач. В большинстве случаев приходится применять численные методы.

В данной работе на основе численного моделирования кинетического уравнения Больцмана изучается один из возможных вариантов трехмерного насоса Кнудсена. Это устройство представляет собой последовательно соединенные длинные цилиндрические трубки, радиусы которых различны. Вдоль широкой трубки температура линейно возрастает от температуры T_1 до T_2 , а вдоль узкой линейно убывает от T_2 до T_1 .

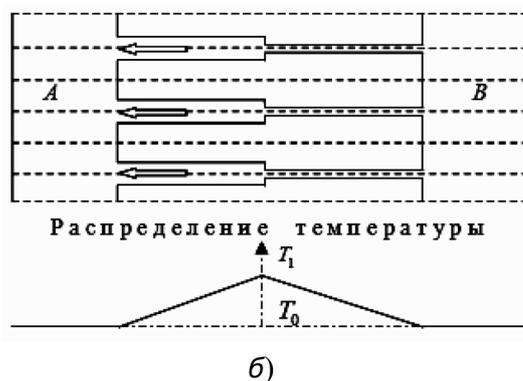
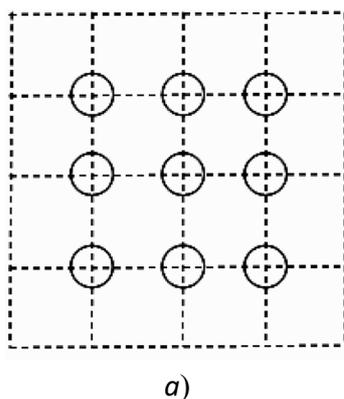


Рис. 1. Однокаскадный насос Кнудсена:
 а — вид в поперечном соединении; б — вид сбоку

Насосы Кнудсена не содержат движущихся механических частей и могут применяться в микро- и наномасштабах для работы таких устройств как масс-спектрометр [1], оптический спектрометр [2], газовый анализатор [3], требующих высокого вакуума.

Впервые предложенные Кнудсенем в 1910 году [4] эти насосы получили большой практический интерес в настоящее время. Современные эксперименты [5, 6] по исследованию насоса Кнудсена показывают его применимость в нано- и микро-масштабах и высокую экономичность по расходу энергии. Возникает необходимость оптимизации работы и конструкции насоса для разных режимов.

1. Постановка краевой задачи

На рис. 1 показан однокаскадный насос Кнудсена, состоящий из цилиндрических трубок разного диаметра. А — накачиваемый резервуар, В — откачиваемый резервуар. Штрихом показаны линии симметрии насоса. Стрелки указывают направление потока газа в насосе.

Движение газа во всех симметричных областях подобно, поэтому для уменьшения объема вычислений целесообразно моделировать движение газа в одной из таких областей. Поступая таким образом, мы решим задачу для всей системы.

На рис. 2 приведена одна из симметричных областей. Радиус узкой трубки равен r , а широкой трубки — $R = 2r$. Длина каждой из трубок $L = 12r$, геометрические размеры параллелепипедных резервуаров вдоль трех координатных осей равны $0,5L \times 0,5L \times L$. В начальный момент функция распределения по скоростям газа внутри системы имеет максвелловский вид при температуре T_0 и числовой плотности молекул n_0 :

$$f(\xi_x, \xi_y, \xi_z) = n_0 \left(\frac{m}{2\pi k T_0} \right)^3 \exp \left(-\frac{m(\xi_x^2 + \xi_y^2 + \xi_z^2)}{2k T_0} \right),$$

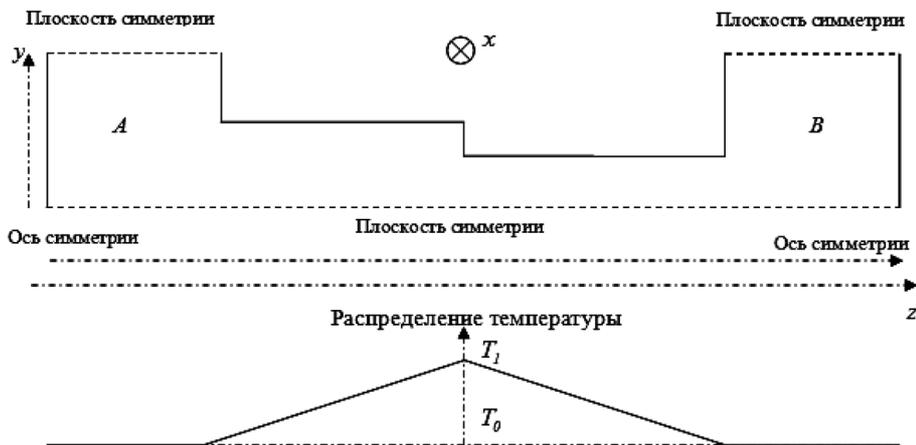


Рис. 2. Симметричная область. Распределение температуры вдоль стенок

где ξ_x, ξ_y, ξ_z — проекции скорости молекулы на координатные оси; m — масса молекулы газа; k — постоянная Больцмана.

Температуры стенок левого и правого резервуаров также равны T_0 . Вдоль трубок температура изменяется линейно, как показано на рис. 2. Максимальное значение температуры T_1 достигается в месте контакта узкой и широкой трубок. Отражение молекул от стенок резервуаров и трубок является диффузным с максвелловским распределением по скоростям. Температура отраженных молекул равна температуре стенки. От плоскостей симметрии молекулы отражаются зеркально.

2. Построение пространственной сетки

Время, затрачиваемое на численное решение уравнения Больцмана, линейно зависит от числа ячеек в координатном пространстве, поэтому сетка должна отвечать следующим требованиям:

1) сгущаться в тех областях, в которых движение газа представляет наибольший интерес, и быть менее плотной там, где важно только распределение макропараметров;

2) сгущаться у искривленных поверхностей расчетной области для того, чтобы лучше повторять геометрию системы.

Этим требованиям удовлетворяют неравномерные тетраэдрические сетки. Такая сетка представлена на рис. 3. Ребра тетраэдров в резервуарах больше, чем в трубках, где изучаются потоки газа. Тетраэдры становятся еще меньше вблизи стенок цилиндрических каналов, чтобы точнее повторить их геометрию.

Сетка была построена с помощью программы GMSH [7]. Использовался алгоритм построения тетраэдров Делоне (Delaunay). Качество тетраэдров было

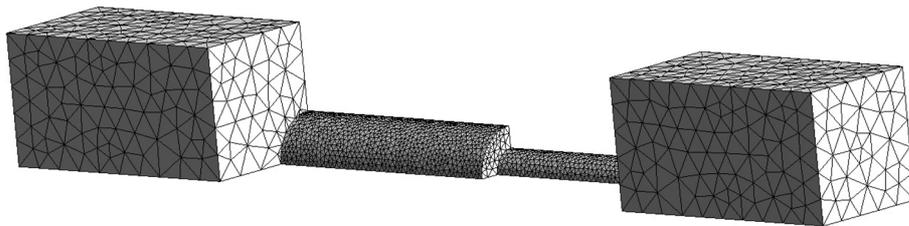


Рис. 3. Неравномерная тетраэдрическая сетка

улучшено встроенными оптимизаторами GMSH и Netgen.

3. Переход к безразмерным переменным и функциям

Перейдем к безразмерным переменным и безразмерной функции распределения

$$\xi^* = \frac{\xi}{v_0}; t^* = \frac{t}{\tau_0}; b^* = \frac{b}{\sigma_{eff}};$$

$$\mathbf{x}^* = \frac{\mathbf{x}}{\lambda}; T^* = \frac{T}{T_0}; n^* = \frac{n}{n_0};$$

$$f^* = \frac{f}{n_0 v_0^{-3}};$$

$$v_0 = \sqrt{kT_0/m}; \lambda = 1/(\sqrt{2} \pi n_0 \sigma_{eff}^2), \tau_0 = \lambda/v_0,$$

где b — прицельное расстояние сталкивающихся молекул; ξ — скорость молекулы; v_0 — нормировочная скорость; t — время; τ_0 — нормировочное время; k — постоянная Больцмана; m — масса молекул газа; \mathbf{x} — координаты молекулы; λ — длина свободного пробега молекулы газа; σ_{eff} — эффективный диаметр молекул; T — температура газа; T_0 — нормировочная температура газа; n — числовая плотность газа; n_0 — нормировочная числовая плотность газа; f — функция распределения по скоростям.

При решении задачи использовалась модель молекул — твердых сфер, в которой σ_{eff} является диаметром молекулы [8], а также потенциал Ленарда—Джонса, где σ_{eff} определяется следующим

образом: $\sigma_{LJ, eff} = \sigma \sqrt{\Omega^{(2,2)}}$, $\Omega^{(2,2)}$ табулированы в работах [9, 10] и мало отличаются от 1.

Опустим знак * и вернемся к прежним обозначениям. Получим уравнение Больцмана в безразмерных переменных:

$$\frac{\partial f}{\partial t} + \xi \frac{\partial f}{\partial \mathbf{x}} = \frac{1}{\sqrt{2}\pi} I, \quad (1)$$

где $I = \int (f' f'_* - ff_*) g b db d\varphi d\xi_*$.

Здесь обозначено $f \equiv f(\xi)$; $f_* \equiv f(\xi_*)$; $f' \equiv f(\xi')$; $f'_* \equiv f(\xi'_*)$; $g = \xi_* - \xi$; φ — параметр столкновения; ξ и ξ_* — скорости двух молекул до столкновения; ξ' и ξ'_* — скорости молекул после столкновения.

4. Конечно-разностная схема

Для численного решения уравнения (1) необходимо перейти от непрерывной функции распределения к дискретной. Для этого введем скоростную сетку, построение которой осуществляется следующим образом. Вначале определяется куб в пространстве скоростей и в нем строится равномерная сетка узлов. Выбор равномерной сетки узлов обусловлен методом вычисления интеграла столкновений. Ребро куба равно $4,8\sqrt{T_1}$. Затем в куб вписывается сфера, и исключаются все внешние узлы. Число узлов сетки сокращается примерно вдвое. Далее уравнение (1) решается в каждом из N_0 равноотстоящих скоростных узлов ξ_γ в каждой ячейке пространственной тетраэдрической сетки.

В скоростном пространстве функция распределения и интеграл столкновений могут быть представлены на базисе δ -функций в виде

$$f(\xi, \mathbf{x}, t) = \frac{V}{N_0} \sum_{\gamma=1}^{N_0} f_\gamma(\mathbf{x}, t) \delta(\xi - \xi_\gamma),$$

$$I(\xi, \mathbf{x}, t) = \frac{V}{N_0} \sum_{\gamma=1}^{N_0} I_\gamma(\mathbf{x}, t) \delta(\xi - \xi_\gamma).$$

Для определения дискретных значений $I_\gamma(\mathbf{x}, t)$ интеграла столкновений используется проекционный метод [11–13], сохраняющий массу, импульс и энергию, а также дающий точное значение интеграла, равное нулю, для максвелловской функции распределения. Последнее свойство метода выполняется для сколь угодно грубых кубатурных сеток, применяемых для вычисления интеграла столкновений.

После вычисления дискретных значений интеграла столкновений задача сводится к решению конечно-разностным методом системы уравнений

$$\frac{\partial f_\gamma}{\partial t} + \xi_\gamma \frac{\partial f_\gamma}{\partial \mathbf{x}} = I_\gamma(\mathbf{x}, t). \quad (2)$$

Система (2) из N_0 уравнений решается симметричным методом расщепления с шагом по времени $\tau = t^{j+1} - t^j$;

$$\frac{f_\gamma^{1,j} - f_\gamma^j}{\tau/2} = \frac{1}{V} (\Phi_{1,\gamma}^1 + \Phi_{2,\gamma}^1 + \Phi_{3,\gamma}^1 + \Phi_{4,\gamma}^1); \quad (3)$$

$$\frac{f_\gamma^{2,j} - f_\gamma^{1,j}}{\tau} = \frac{1}{\sqrt{2\pi}} I_\gamma; \quad (4)$$

$$\frac{f_\gamma^{j+1} - f_\gamma^{2,j}}{\tau/2} = \frac{1}{V} (\Phi_{1,\gamma}^2 + \Phi_{2,\gamma}^2 + \Phi_{3,\gamma}^2 + \Phi_{4,\gamma}^2), \quad (5)$$

$f_\gamma^{1,j}$ и $f_\gamma^{2,j}$ — промежуточные значения функции распределения; Φ_k^1 , Φ_k^2 — потоки функции распределения через k -ю грань тетраэдра в уравнениях (3) и (5) соответственно; V — объем тетраэдра.

Уравнение переноса аппроксимируем явной консервативной схемой первого порядка точности. На каждом временном шаге оно решается два раза для смягчения условия Куранта.

Определим Φ_k для всех тетраэдров. Через \mathbf{n}_k обозначим нормаль к k -й грани, направленной внутрь тетраэдра. Возможно три случая:

1. Рассматриваемый тетраэдр под номером i граничит с другим тетраэдром под номером m . Тогда справедлива формула

$$\Phi_k = \begin{cases} S_k f(t, i, \xi)(\xi, \mathbf{n}_k), & (\xi, \mathbf{n}_k) < 0, \\ S_k f(t, m, \xi)(\xi, \mathbf{n}_k), & (\xi, \mathbf{n}_k) > 0, \end{cases} \quad (6)$$

где S_k — площадь k -й грани тетраэдра.

Верхняя формула отвечает вытеканию газа из тетраэдра, в то время как нижняя формула означает приток газа в i -й тетраэдр из соседнего k -го тетраэдра.

2. Грань тетраэдра граничит со стенкой, тогда имеет место диффузное отражение:

$$\Phi_k = \begin{cases} S_k f(t, i, \xi)(\xi, \mathbf{n}_k), & (\xi, \mathbf{n}_k) < 0 \\ - \frac{\sum_{(\xi, \mathbf{n}_j) < 0} (\xi, \mathbf{n}_k) f(t, i, \xi)}{\sum_{(\xi, \mathbf{n}_j) > 0} (\xi, \mathbf{n}_k) \exp\left(-\frac{\xi^2}{2T_{\text{ст}}}\right)} \exp\left(-\frac{\xi^2}{2T}\right), & (\xi, \mathbf{n}_k) > 0, \end{cases} \quad (7)$$

где $T_{\text{ст}}$ — безразмерная температура стенки.

3. Грань тетраэдра лежит на плоскости симметрии, тогда имеет место зеркальное отражение:

$$\Phi_k = \begin{cases} S_k f(t, i, \xi)(\xi, \mathbf{n}_k), & (\xi, \mathbf{n}_k) < 0, \\ S_k f(t, i, \xi - 2\mathbf{n}_k)(\xi, \mathbf{n}_k), & (\xi, \mathbf{n}_k) > 0. \end{cases}$$

Уравнение (4) решается как интегральное с контролем положительной определенности получаемого решения, изложенным в работах [11–13].

5. Результаты расчетов

Приведем описание процесса накачки для аргона в случае $r = 1,25$, $T_1/T_0 = 1,5$. Число Кнудсена с учетом симметрии $Kn = \frac{\lambda}{2r} = 0,4$. Число узлов в пространственной сетке равно 20 000, а в скоростной — 3 000. Степень накачки будем характеризовать с помощью отношения

давлений $a = \frac{p_2}{p_1}$, где p_1 — давление

газа в откачиваемом резервуаре; p_2 — давление газа в накачиваемом резервуаре. На рис. 4 представлена зависимость отношения давлений от времени. Здесь рассмотрено два потенциала взаимодействия молекул: потенциал упругих сфер и потенциал Леннарда—Джонса. Как видно из графика, результаты для этих потенциалов при одинаковом эффективном сечении отличаются незначительно. Вблизи нуля накачка растет быстро до значения 1,014. Это связано с тем, что в широкой трубке в начальный момент находится больше газа, чем в узкой, и из нее в начале процесса вытекает больше газа, чем из узкой.

Полученная кривая при $t > 200$ с большой точностью аппроксимируется уравнением вида $f(t) = A - b \cdot \exp\left(-\frac{t}{\tau_{rel}}\right)$ при $A = 1,039$, $b = 0,0243$, $\tau_{rel} = 5867$. Эти параметры несут определенный физический смысл: $A = \lim_{t \rightarrow \infty} a(t)$ показывает степе-

нь накачки $\frac{p_2}{p_1}$ в установившемся режиме, τ_{rel} — время релаксации процесса. В предельном случае, когда объем трубок бесконечно мал по сравнению

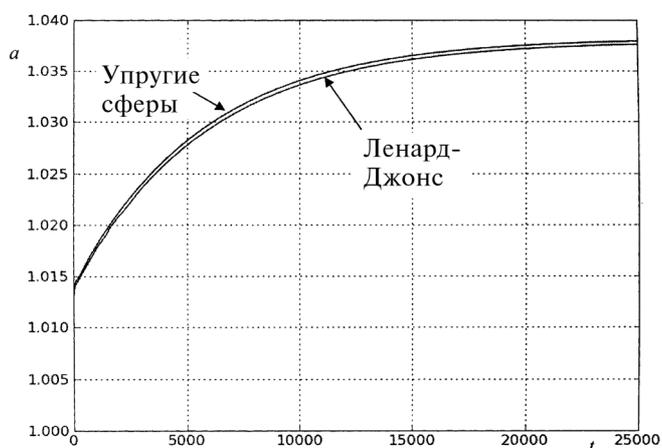


Рис. 4. Зависимость накачки от времени

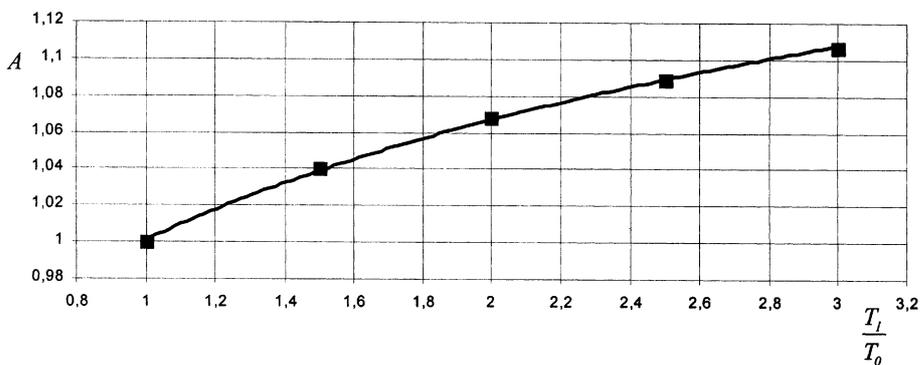


Рис. 6. График зависимости накачки от отношения температур

с объемами резервуаров, $b \rightarrow a$. Аппроксимация проводилась с помощью программы *gnuplot* [14]. Точность аппроксимации равна 10^{-5} .

На рис. 5 (см. третью сторону обложки) показано распределение температуры, плотности и давления в установившемся режиме.

Зависимость отношения давления от времени $a(t)$ рассматривали и при других отношениях температур T_1/T_0 . Форма кривой накачки газа имеет тот же вид, что и при $T_1/T_0 = 1,5$, однако отношения давлений в левом и правом резервуарах оказываются различными в стационарном режиме для разных T_1/T_0 . На рис. 6 изображена точками зависимость предельной накачки A от отношения температур T_1/T_0 . Полученная зависимость со средней квадратичной ошибкой 0,02 описывается следующей формулой: $A = (T_1/T_0)^{0,0924}$.

Исследования накачки газа были также проведены для других чисел Кнудсена. График зависимости накачки газа от числа Кнудсена представлен на рис. 7. Интересно то, что при малых числах Кнудсена значение величины A возрастает, а при больших числах убывает, при этом максимум приходится на $Kn = 0,5$. Это означает, что самый эффективный режим работы ступенчатого насоса Кнудсена наступает при $\lambda = r$.

Если объемы резервуаров неизменны, то τ_{rel} убывает с возрастанием числа Кнудсена, причем этот закон со средней квадратичной ошибкой 0,05 аппроксимируется функцией $\tau_{rel}(Kn) = 2517 \cdot Kn^{-0,85}$. На рис. 8 точками изображена зависимость $\tau_{rel}(Kn)$, а линией — аппроксимирующая функция.

Остановимся теперь на детальном изучении потоков в системе. Представим их структуру на характерных этапах развития процесса. Можно выделить три этапа течения.

1. В начале процесса температура газа в трубках повышается, давление там возрастает, и газ начинает вытекать из этой области в резервуары.

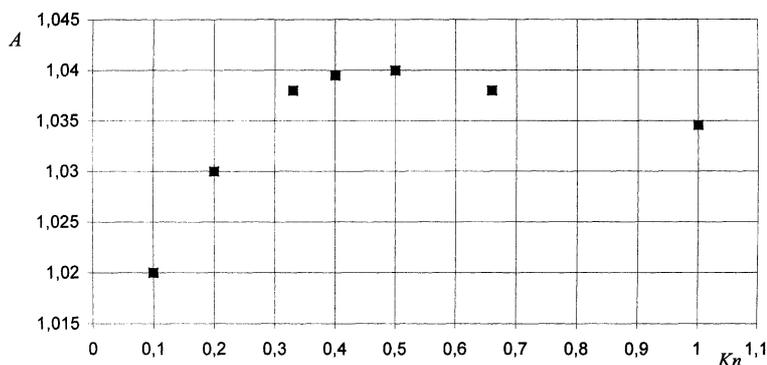


Рис. 7. Зависимость накачки от числа Кнудсена

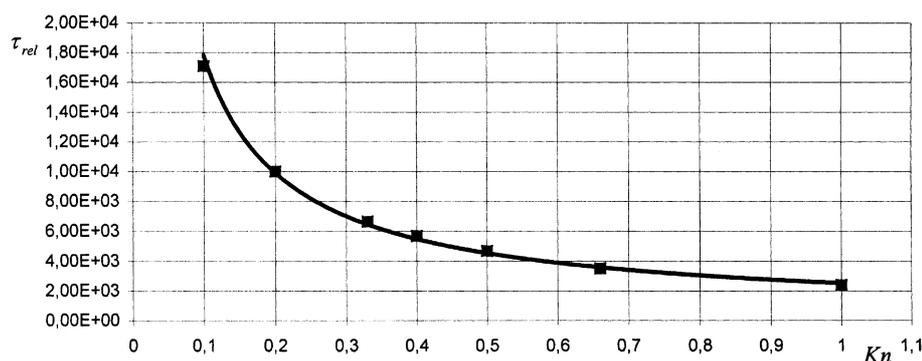


Рис. 8. Зависимость времени релаксации от числа Кнудсена

2. По истечении времени $t \approx 200$ установится режим перекачки газа из правого резервуара в левый. При этом в резервуарах образуются вихри вследствие взаимодействия газа со стенками.

3. Стационарный режим наступает тогда, когда эффект теплового скольжения уравнивается разностью давлений в левом и правом резервуарах. Этот этап интересен тем, что в трубках на расстоянии длины свободного пробега от цилиндрических стенок газ скользит вдоль градиента температуры, а в центре течет в противоположном направлении, образуя вихревые структуры.

На рис. 9, 10 и 11 (см. четвертую сторону обложки) изображены потоки газа в продольном сечении системы на 1-м, 2-м и 3-м этапах соответственно при $Kn = 0,2$.

Заключение

В данной работе изучен однокаскадный насос Кнудсена на основе решения кинетического уравнения Больцмана. Интеграл столкновений вычислялся консервативным проекционным методом. Оператор переноса кинетического уравнения аппроксимировался методом конечных объемов и использования тетраэдрических пространственных сеток. Получена зависимость отношения давлений в процессе накачки от времени. Проведено параметрическое исследова-

ние накачки в зависимости от отношения температур на концах трубки и числа Кнудсена. Максимальная степень накачки газа при отношении температур $T_1/T_0 = 1,5$ была получена при числе Кнудсена $Kn = 0,5$. По мере увеличения отношения температур значение накачки растет. Рассмотрена зависимость времени релаксации процесса накачки при одинаковых резервуарах в зависимости от числа Кнудсена. Детально рассмотрена структура потока газа в трубках на каждом из трех выделенных этапов. Представлены графики потоков в продольном срезе. Получены простые аналитические зависимости текущей накачки от времени, предельной накачки от отношения температур на концах трубки, времени релаксации процесса накачки от числа Кнудсена.

Расчеты выполнены на суперкомпьютере МФТИ-60 в рамках проекта "Применение суперкомпьютерных вычислительных систем и технологий для анализа и моделирования газокинетических процессов в микро- и наноструктурах" (№ 2.1.2/2837, руководитель Н. Н. Пономарев-Степной) в рамках ведомственной целевой программы Министерства образования и науки РФ "Развитие научного потенциала высшей школы (2009–2010 годы)".

Список литературы

1. Nathanson H. C., Liberman I., Freidhoff C. Novel functionality using micro-gaseous devices // Proc. of IEEE Int. Conf. on MEMS. 1995. P. 72–76.
2. Blomberg M., Rusanen O., Keranen K., Lehto A. A silicon microsystem-miniaturised infrared spectrometer // Proc. of IEEE Int. Conf. on Solid-State Sensors and Actuators. 1997. V. 2. P. 1257–1258.
3. Terry S. C., Jerman J. H., Angell J. B. A gas chromatographic air analyzer fabricated on a silicon wafer // IEEE Trans. on Electron Devices. 1979. V. 26, I. 12. P. 1880–1886.
4. Knudsen M. Eine Revision der Gleichgewichtsbedingung der Gase. Thermische Molekularströmung // Ann. der Phys. 1910. Bd. 31, N 9. S. 205–229.
5. Han Y., Alexeenko A., Young M., Muntz E. P. Experimental and Computational Studies of Temperature Gradient Driven Molecular Transport in Gas Flows Through Nano/Micro-Scale Channels // Nanoscale and Microscale Thermophysical Engineering. 2007. Vol. 11, Is. 1 & 2. P. 151–175.
6. Sone Y., Sato K. Demonstration of a one-way flow of a rarefied gas induced through a pipe without average pressure and temperature gradients // Physics of Fluids. 2000. V. 12. N 7. P. 1864–1868.
7. <http://www.geuz.org/gmsh/>
8. Коган М. Н. Динамика разреженного газа. М.: Наука, 1967.
9. Чепмен С., Каулинг Т. Математическая теория неоднородных газов. М.: Иностранная литература, 1960.

10. Лифшиц Е. М., Питаевский Л. П. Теоретическая физика / Под ред. Л. Д. Ландау, Е. М. Лифшица. В 10 т. Том 10. М.: Физматлит, 2002.

11. Черемисин Ф. Г. Консервативный метод вычисления интеграла столкновений Больцмана // Доклады РАН. 1997. Т. 357. № 1. С. 53–56.

12. Черемисин Ф. Г. Решение кинетического уравнения Больцмана для высокоскоростных течений // ЖВМ и МФ. 2006. Т. 2. № 46. С. 329–343.

13. Tcheremissine F. G. Solution of the Boltzmann Kinetic Equation for Low Speed Flows // Transport Theory and Statistical Physics. 2008. Vol. 37, N 5. P. 564–575.

14. www.gnuplot.info

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

УДК 004.738.5:004.415.53

Э. В. Калинина, канд. техн. наук, ст. науч. сотр.,

И. В. Ретинская, д-р техн. наук, проф.,

Д. А. Серова, аспирант,

Российский государственный университет
нефти и газа им. И. М. Губкина

e-mail: kto@migmail.ru

Введение

Неоднородность телекоммуникационных и компьютерных сетей, сетевых информационных ресурсов и пользователей Интернет определяет необходимость разработки адекватных математических методов для решения прикладных задач, связанных с мониторингом IP-каналов связи.

Существующие подходы к мониторингу и анализу телекоммуникационных сетей, как правило, опираются на возможность непосредственного доступа ко всем инфраструктурным элементам телекоммуникационной сети, что зачастую трудно выполнимо. Здесь может быть полезен анализ сетевого трафика локальных сетей на основе открытой потоковой информации.

Для оценки качества работы сети можно использовать следующие характеристики открытой потоковой информации:

- уровень загрузки канала (*utilization level*);
- время доставки пакета (*round trip time*), которое оценивается по данным команды *Ping*;
- число потерянных пакетов (*packet loss rate*), также оцениваемое по данным команды *Ping*.

Время доставки и число потерянных пакетов, получаемые с помощью команды *Ping*, также можно использовать для анализа работоспособности канала связи и оценки качества соединения между двумя удаленными точками.

1. Процедура мониторинга IP-сети и анализируемые данные

Для построения модели были использованы статистические данные научно-образовательной сети. В ее структуру входили три вида технологий (ATM-OC/3, Fast Ethernet, Gigabit Ethernet) передачи любого вида трафика — компьютерного, телефонного или видео — пакетами фиксированной длины.

Каждые 5 мин посылался пакет из 10 сигналов по 56 байт на 41 узел-приемник, с каждого узла

Вероятностные и нечеткие модели оценки качества передачи информации в телекоммуникационных сетях

Рассматривается тестирование каналов IP-сетей командой Ping, которое на основе открытой потоковой информации позволяет независимо от технологии передачи данных и типа канала связи оценить как его работоспособность, так и качество передачи информации различного вида.

Оценивание проводится на основе анализа результатов непрерывного и продолжительного тестирования канала через равные промежутки времени. Изучаемые характеристики — число потерянных пакетов в серии и задержка сигнала. Предложены вероятностные математические модели для временных рядов, образуемых данными непрерывного мониторинга командой Ping, адекватные методы анализа, процедуры и специализированный программный комплекс для вычисления ряда параметров, формируются статистические критерии, характеризующие работоспособное состояние канала связи. Оценка качества трафика проводится с помощью операторов усреднения и агрегирования нечетких экспертных суждений системного администратора, определяющего меру допустимых искажений информации.

Предложенные методы демонстрируются на примере анализа каналов связи существующей научно-образовательной сети.

Ключевые слова: телекоммуникации, моделирование, качество связи

регистрировалось время возвращения или задержки и число потерянных сигналов из 10 посланных. Данные собирались в течение месяца.

В ходе тестирования каналов связи регистрировалось число отсутствующих откликов, а также выдавались значения минимального, максимального и среднего времени отклика сервера.

Для узлов сети собрано в среднем по 7,5 тысяч указанных выше значений.

Потеря всех 10 пакетов в сигнале означает неработоспособное состояние канала связи в данный момент времени. При этом данные по времени возврата отсутствуют, и оно может оцениваться как бесконечно большое. Такие случаи при тестировании квалифицируются как состояние отказа канала связи.

Потеря пакетов и время возврата сигналов в серии зависят от изменяющегося рабочего состояния узла приемника, связанного со случайной вариацией его загрузки и состоянием аппаратуры, поэтому регистрируемые числовые значения следует рассматривать как случайные величины. Математическое моделирование результатов тестирования через равные промежутки времени сводится к изучению и анализу временных рядов: ряда случайного числа потерянных пакетов и ряда случайных значений времени задержки пакетов.

Статистические характеристики рядов оцениваются по выборочным данным, собранным в режиме нормальной эксплуатации канала связи [6]. Анализом установлено, что ряды потерянных пакетов и времен задержек между собой не коррелированы, и изучение их статистических свойств можно проводить независимым образом.

2. Статистический анализ числа потерянных пакетов

В процессе получения эхо-отклика, в соответствии с описанным выше способом получения статистической информации, реализуется случайное событие с одиннадцатью возможными исходами с вероятностями потерь от 0 до 10 пакетов:

$$p_0, p_1, \dots, p_{10}.$$

Вероятности p_0, p_1, \dots, p_9 соответствуют рабочему состоянию канала связи, однако качество передаваемой информации при потере пакетов ухудшится, так как потеря пакетов приводит к искажениям и ошибкам в получаемой приемником информации.

Тестированием, в первую очередь, идентифицируется работоспособное состояние канала, поэтому можно ограничиться анализом двух исходов — рабочим состоянием с вероятностью $q = \sum_{i=0}^9 p_i$ и отка-

зом с вероятностью $p = p_{10}$, $p + q = 1$, а результаты тестирования описываются схемой испытаний Бернулли.

Вероятностная математическая модель для временной последовательности результатов тестирования с общим числом испытаний n и числом отказов x описывается биномиальным распределением [8].

Эффективной и состоятельной оценкой вероятности отказа p является эмпирическая частота отказа — отношение числа зарегистрированных потерь всех 10 пакетов k к общему числу тестирования N :

$$h_{10} = k/N, \quad (1)$$

а вероятность q работоспособного состояния канала в момент послышки сигнала оценивается через частоту события, состоящего в потере всех 10 пакетов, $q \approx 1 - h_{10}$.

В качестве критерия, оценивающего уровень безотказности канала связи, предлагается рассматривать вероятность работоспособного состояния в течение определенного временного промежутка — в течение суток.

Если n — число запросов, посылаемых за выбранный промежуток времени (в суточном временном отрезке $n = 288$ при тестировании каждые 5 мин), а вероятность работоспособного состояния в одном акте тестирования равна $q = 1 - p$, то вероятность сохранять работоспособное состояние $Q(n)$ в n актах тестирования задается выражением

$$Q(n) = q^n = (1 - p)^n, \quad (2)$$

вероятность хотя бы одного отказа $P(n)$ в n актах тестирования — равенством

$$P(n) = 1 - Q(n) = 1 - (1 - p)^n. \quad (3)$$

При $pn \ll 1$ вероятности безотказной работы и хотя бы одного отказа вычисляются как

$$Q(n) \approx 1 - pn \text{ и } P(n) \approx pn. \quad (4)$$

Для сохранения работоспособного состояния в течение суток с вероятностью $Q(n) > 0,99$ необходимо $p < 3,4 \cdot 10^{-5}$, а для $Q(n) > 0,95$ — $p < 1,7 \cdot 10^{-4}$.

Под уровнем работоспособности канала связи здесь будет пониматься вероятность его безотказной работы (2) в течение суток.

В табл. 1 для тестируемой сети представлены оценки уровня работоспособности (2) каналов связи в течение суток.

По уровню работоспособности каналы разделены по группам и только 9 каналов сети имеют вероятности отказа в течение суток менее 0,05. Для 23 каналов уровень работоспособности менее

Таблица 1

Уровни работоспособности каналов исследуемой сети

Вероятность рабочего состояния в течение суток	Более 0,95	От 0,75 до 0,95	От 0,50 до 0,75	Менее 0,5
Число каналов	9	8	1	23

0,5, что можно трактовать как отсутствие связи в течение большей части суток. Длительная потеря связи свидетельствует об обрыве канала связи или об отключении питания на узле сети, возможно связанном с режимом работы персонала.

Анализ результатов тестирования показал, что уровень работоспособности канала связи не зависит от технологии передачи информации: одни и те же значения уровня встречаются при использовании технологий ATM-OC/3, Fast Ethernet и Gigabit Ethernet. Можно заключить, что уровень работоспособности является независимой оценкой состояния канала.

3. Статистический анализ ряда временных задержек

Для временной задержки от момента отправки эхо-запроса до момента получения эхо-отклика регистрируются четыре характеристики: минимальное и максимальное время возврата, среднее значение задержки и ее вариация.

В табл. 2 представлены данные тестирования нескольких IP-каналов в условиях нормального режима работы и аномально больших задержек. При аномальных состояниях канала связи выборочные оценки в серии из 10 пакетов для среднего значения, максимальной задержки и среднего квадратичного отклонения существенно смещены относительно значений нормального режима. Этот факт приводит к необходимости применения устойчивых (робастных) к аномальным задержкам статистических оценок для выборочных характеристик.

Большие задержки свидетельствуют либо о чрезмерной загруженности канала в момент тестирования, либо о сбое аппаратуры, либо о несанкционированной атаке на сеть пользователя и приводят к искажениям и ошибкам в получаемой информации в некоторых типах трафика.

Пусть на протяжении достаточно большого промежутка времени отсутствуют аномальные данные, а ряд задержек представляет собой стационарную последовательность (сохраняет неизменными вероятностные свойства), и пусть время возврата является случайной величиной с функцией распределения $P(x)$ и плотностью $p(x)$. Совокупность времен возврата 10 пакетов образует выборку объемом 10 одинаково распределенных

случайных величин или выборку из генеральной совокупности с функцией распределения $P(x)$.

Ограничимся рассмотрением случаев возврата всех 10 пакетов серии, так как анализ данных тестирования показывает, что частота случаев с потерями пакетов мала, и этими исходами можно пренебречь.

В условиях стационарности ряда параметры генеральной совокупности (математическое ожидание и квадратичное отклонение) можно оценить по данным тестирования с числом серий L . Пусть $\bar{x}^{(l)}$ и $s^{(l)}$ соответственно выборочные значения среднего и квадратичного отклонений в l -й серии ($l = 1 \dots L$), тогда оценки генерального среднего ξ и квадратичного отклонения σ получаются усреднением по всем сериям.

В математической статистике достаточно хорошо изучены вероятностные свойства различных статистик, образованных членами конечных выборок.

Предлагается использовать критерии, построенные на порядковых статистиках: критерий наибольшего отклонения максимальной задержки в серии от генерального среднего $d = |\max - \xi|$ и критерий размаха сериальных наблюдений $\omega = (\max - \min)$ [5, 8].

Распределение вероятности $P(d)$ определяется функцией $P(x)$ и задается равенством:

$$P(d) = (P\{|x - \xi| < d\})^n, \quad (5)$$

где $n = 10$; $P(x)$ — функция распределения; $p(x)$ — его плотность для случайного времени возврата одного пакета в серии.

Расчеты показывают, что при $n = 10$ и доверительном уровне $P = 0,995$ выполняется соотношение

$$P(d < 3,5\sigma) = 0,995. \quad (6)$$

С уровнем доверия 0,995 или уровнем значимости 0,005 можно утверждать, что разница между максимальной задержкой и генеральным средним

Таблица 2

Данные тестирования по задержкам

IP-адрес канала	Нормальный режим связи	Аномальная задержка
	min/avg/ /max/stddev, мс	min/avg/ /max/stddev, мс
194.190.244.161	15,944/16,850/ /17,637/0,676	15,677/22,378/ /59,800/12,693
www.cisco.com	198,454/198,851/ /199,139/0,203	198,857/201,302/ /203,920/1,780
www.achbank.com	0,868/1,175/ /1,559/0,197	0,889/2,412/ /5,830/1,592
192.168.5.1	0,449/0,574/ /0,720/0,082	0,410/31,525/ /184,496/63,379

не превышает $3,5\sigma$. Это правило предлагается использовать для идентификации аномальных задержек в процессе мониторинга тестированием командой Ping.

Идентификацию аномальных задержек можно проводить по критерию, построенному на размахе $\varpi = (\max - \min)$.

В случае нормально распределенных наблюдений с параметрами (ξ, σ^2) и их числе в серии $n = 10$ для размаха ϖ с уровнем доверия 0,9999 выполняется соотношение [5]:

$$P(\varpi_n < 6,90\sigma) = 0,9999, \quad (7)$$

при этом для статистики $W_n = \frac{\varpi_n}{\sigma}$ при $n = 10$ математическое ожидание $M\{W_n\} = 3,077$, дисперсия $D\{W_n\} = 0,635$, среднее квадратичное отклонение $\sqrt{D\{W_n\}} = 0,797$.

Применение критериев (6) или (7) подразумевает знание величины σ . Однако регистрация чрезмерно большой задержки (выброса) в l -й серии ($l = 1 \dots L$) приводит к резкому смещению относительно стационарных значений как максимального (max) и среднего времени возврата (avg), так и вариации задержки ($stddev$). При большом числе аномальных серий будут смещены и выборочные средние этих параметров. Поэтому для идентификации выбросов нужно использовать устойчивые к помехам оценки средних [7].

Если аномальные наблюдения встречаются не более чем в 30—35 % случаев в L актах тестирования, устойчивыми состоятельными и несмещенными оценками параметров ξ и σ являются медианы вариационных рядов сериальных оценок avg и $stddev$ для всего времени тестирования.

Наличие аномальной задержки не свидетельствует о неработоспособности канала, но сигнализирует об ухудшении качества связи.

4. Нечеткие методы оценки качества связи

Требования к качеству работы канала определяются, в первую очередь, характером передаваемой информации и необходимостью передачи данных в режиме реального времени.

Особенности сетевого трафика предлагается учитывать путем формирования критериев оценки качества на основе экспертных суждений администратора сети о влиянии потерь пакетов и больших задержек на снижение качества связи.

Лингвистические суждения носят нечеткий, расплывчатый характер и требуют усилий для своей аналитической формализации. Для исследуемой сети системным администратором были сформулированы следующие суждения относительно данных в акте тестирования:

1. Для качественной связи недопустимыми являются потери 6 и более пакетов из 10, качество теряется примерно пропорционально числу потерь. Отсутствие потерь соответствует наивысшему уровню качества.

2. Недопустимыми являются случаи, когда разница между максимальной и минимальной задержкой превышает 50 мс, качество монотонно падает с ростом задержки относительно режима стационарной работы и может считаться недопустимым, когда она превышает среднюю задержку примерно в 100 раз.

Экспертные суждения можно промоделировать нечеткими множествами на исходах акта тестирования. При этом будем полагать, что наивысший уровень качества оценивается как 1, а недопустимый уровень — как 0.

5. Оценка качества работы сети на основе нечеткого анализа ряда потерь пакетов

Формализуем суждения о потере пакетов функцией принадлежности нечеткого множества, задающего меру соответствия результата тестирования качественной связи.

Пусть $m^{(i)}$ — число потерянных пакетов в i -м акте тестирования, $m^{(i)} \in M^{(i)}$, $M^{(i)} = \{0, 1, \dots, 10\}$ — множество возможных потерь. Введем нечеткое множество $A^{(i)} = \{(m^{(i)}, \mu_p(m^{(i)}))\}$, с функцией принадлежности $\mu_p(m^{(i)})$, характеризующей степень соответствия потери $m^{(i)}$ пакетов качественной связи:

$$A^{(i)} = \{0 \mid 1; 1 \mid 0,8; 2 \mid 0,6; 3 \mid 0,4; 4 \mid 0,2; 5 \mid 0,1; 6, \dots, 10 \mid 0\}, i = 1 \dots L. \quad (8)$$

Нечеткое множество $A^{(i)}$ формализует мнение системного администратора о качестве связи в i -м акте тестирования: при отсутствии потерь ($\mu_p = 1$), при потерях 6 и более пакетов ($\mu_p = 0$) μ_p монотонно убывает с ростом числа потерь.

Обозначим $\mathbf{m} = (m^{(1)}, \dots, m^{(L)})$ — вектор потерь в серии длиной L , введем нечеткое множество $P_L = \{(\mathbf{m}, \overline{\mu}_p(\mathbf{m}))\}$, заданное на декартовом произведении подпространств $M = M^{(1)} \times M^{(2)} \times \dots \times M^{(L)}$, ($M^{(i)} = \{0, 1, \dots, 10\}$ — множество возможных потерь пакетов в i -м тестировании, $i = 1 \dots L$), с функцией принадлежности $\overline{\mu}_p(\mathbf{m})$ точки $\mathbf{m} \in M$ размерности L множеству реализаций высокого уровня качества по потерянным пакетам.

Функция принадлежности $\overline{\mu}_p(\mathbf{m})$ является композицией функций принадлежности отдельных нечетких множеств (8) и вводится хорошо известным усредняющим оператором, имеющим

целый ряд полезных свойств [3], — арифметическим средним:

$$\begin{aligned} \bar{\mu}_p(\mathbf{m}) &= \bar{\mu}_p(\mu_p(m^{(1)}), \dots, \mu_p(m^{(L)})) = \\ &= \frac{1}{L} \sum_{i=1}^L \mu_p(m^{(i)}). \end{aligned} \quad (9)$$

Обобщенное среднее $\bar{\mu}_p = 1$, когда для любого i $\mu_p(m^{(i)}) = 1$. Величина $\bar{\mu}_p$ является выборочной оценкой для математического ожидания уровня качества тестируемого канала связи, она задает усредненную по множеству тестирований степень или уровень качества передачи информации.

Выбор в данном случае агрегирующих операторов *min* или геометрического среднего приводил бы к слишком жесткому оцениванию уровня качества. В этом случае нулевой уровень присваивался бы всему ряду, если при тестировании произошел хотя бы один случай потери шести и более пакетов.

6. Оценка качества работы сети на основе нечеткого анализа ряда задержек пакетов

Суждения о недопустимом качестве связи, высказанные системным администратором, сформулированы в терминах размаха наблюдений $\varpi^{(i)} = (\max^{(i)} - \min^{(i)})$ в i -м акте тестирования. Поэтому для формализации нечетких высказываний используется критерий (7) $P(\varpi_{10} < 6,90\sigma) = 0,9999$ в предположении, что значение σ получено на основе описанных выше процедур статистического оценивания, и из вероятностных свойств нормированного размаха для нормального режима $W_n = \frac{\varpi_n}{\sigma}$ при $n = 10$ математическое ожидание $M\{W_{10}\} = 3,077$.

Следующие неравенства задают условия нормального режима и качественной передачи информации в i -м акте тестирования, сформулированные системным администратором: $\varpi^{(i)} \leq 50$ мс и $(\varpi^{(i)}/\sigma) \leq 6,90$. С ростом значения $(\varpi^{(i)}/\sigma)$ качество падает и становится недопустимым, когда отношение $(\varpi^{(i)}/\sigma) / M\{W_{10}\} > 100$, что эквивалентно $(\varpi^{(i)}/\sigma) > 307,7$.

Построим нечеткие множества B и C для моделирования принадлежности к высокому уровню качества результата тестирования по задержке.

Пусть $\varpi^{(i)} = (\max^{(i)} - \min^{(i)})$ — размах для 10 значений задержек в i -м акте тестирования, где $\varpi^{(i)} \in \mathcal{W}^{(i)}$, $\mathcal{W}^{(i)} = \{\varpi^{(i)} : \varpi^{(i)} > 0\}$ — множество возможных размахов.

Введем нечеткое множество $B^{(i)} = \{(\varpi^{(i)}, \mu_1(\varpi^{(i)}))\}$, $\varpi^{(i)} > 0$, с функцией принадлежности $\mu_1(\varpi^{(i)})$, характеризующей степень соответствия $\varpi^{(i)}$ режиму нормальной работы с высоким уровнем качества передачи информации:

$$\mu_1(\varpi^{(i)}) = \begin{cases} 1, & \text{при } \varpi^{(i)} \leq 50 \text{ мс,} \\ 0, & \text{при } \varpi^{(i)} > 50 \text{ мс.} \end{cases} \quad (10)$$

и нечеткое множество $C^{(i)} = \{(\varpi^{(i)}, \mu_2(\varpi^{(i)}))\}$, $\varpi^{(i)} > 0$, с функцией принадлежности $\mu_2(\varpi^{(i)})$, характеризующей степень соответствия $\varpi^{(i)}$ режиму нормальной работы с высоким уровнем качества передачи информации:

$$\mu_2(\varpi^{(i)}) = \begin{cases} 1 & \text{при } (\varpi^{(i)}/\sigma) \leq 6,90, \\ 1 - 0,00325(\varpi^{(i)}/\sigma) & \text{при } 6,90 < (\varpi^{(i)}/\sigma) \leq 307,7, \\ 0 & \text{при } (\varpi^{(i)}/\sigma) > 307,7. \end{cases} \quad (11)$$

Нечеткое множество $D^{(i)} = (\varpi^{(i)}, \mu_3(\varpi^{(i)}))$, $\varpi^{(i)} > 0$, являющееся пересечением нечетких множеств $B^{(i)}$ и $C^{(i)}$ с функцией принадлежности $\mu_3(\varpi^{(i)})$, обеспечивает выполнение приведенных выше условий для режима связи с высоким качеством передачи информации:

$$\begin{aligned} \mu_3(\varpi^{(i)}) &= \mu_1(\varpi^{(i)}) \wedge \mu_2(\varpi^{(i)}) = \min(\mu_1(\varpi^{(i)}), \\ &\mu_2(\varpi^{(i)})), \quad \varpi^{(i)} \in \mathcal{W}^{(i)}, \quad \varpi^{(i)} > 0. \end{aligned} \quad (12)$$

Обозначим $\varpi = (\varpi^{(1)}, \dots, \varpi^{(L)})$ — вектор размахов в L тестированиях, введем нечеткое множество $T_L = \{(\varpi, \bar{\mu}_3(\varpi))\}$, заданное на декартовом произведении подпространств $W = W^{(1)} \times W^{(2)} \times \dots \times W^{(L)}$, $(W^{(i)} = \{\varpi^{(i)} : \varpi^{(i)} > 0\})$ — множество возможных размахов в i -м тестировании, $i = 1 \dots L$ с функцией принадлежности $\bar{\mu}_3(\varpi)$ точки $\varpi \in W$ множеству реализаций по задержкам высокого уровня качества.

Функция принадлежности $\bar{\mu}_3(\varpi)$ агрегирует функции принадлежности отдельных нечетких множеств оператором усреднения — арифметическим средним.

Обобщенное среднее $\bar{\mu}_3 = 1$, когда для любого i $\mu_3(\varpi^{(i)}) = 1$, оно задает осредненный по множеству тестирований уровень качества передачи информации, определяемый задержками.

Таким образом построенные функции принадлежности и операции над ними позволяют получить количественную оценку качества работы канала.

7. Вычислительные аспекты мониторинга трафика

Основываясь на разработанных статистических и нечетких моделях, был разработан комплекс программ для оценки качества телекоммуникационной сети "Ping Loss". В состав программного комплекса входят две самостоятельные программы, написанные на объектно-ориентированном языке программирования Delphi 7 с помощью компилятора Borland Delphi версии 7.0. Программы являются инструментом администратора сети и позволяют выявить "узкие" моменты в работе каналов.

Заключение

Гибкость оценок уровня качества по данным команды Ping предоставляет широкие возможности для проведения аудита работы сети и построения объективных оценок качества ее работы.

Разработанные математические модели определения состояния отказа, уровней безотказности и качества передачи информации применимы для оперативной диагностики всех типов каналов: кабельная связь, высокоскоростные каналы и кана-

лы, использующие телефонные сети, беспроводная передача информации.

Список источников

1. Ретинская И. В., Калинин Э. В., Серова Д. А. Оценка качества связи при мониторинге IP-сетей с помощью команды PING // Труды XVI конференции представителей научно-образовательных сетей RELARN. 2009.
2. Kalinina E., Serova D. Stochastic and fuzzy methods of communication quality estimation for IP networks monitoring. Conference Proceedings, 15th Zittau East-West fuzzy colloquium, Zittau 13–15.09.2008. P. 134–141.
3. Wagenknecht M., Kalinina E. Some Remarks on Crisp and Fuzzy Number Averaging and Aggregating. Conference Proceedings, 13th Zittau East-West Fuzzy Colloquium, Zittau, 13–15.09.2006. P. 190–201.
4. Kalinina E., Besrukavnyi D., Domrachev V. Fuzzy Methods in Tasks of Monitoring of the Network Traffic (at Administration of a Network) // Conference Proceedings, 12th Zittau East-West Fuzzy Colloquium, Zittau, 21–23.09.2005. P. 35–44.
5. Большов Л. Н., Смирнов Н. В. Таблицы математической статистики. М.: Наука, 1983.
6. Anderson T. W. The Statistical Analysis of Time Series. John Wiley & Sons, 1971.
7. Huber P. J. Robust Statistics A. review // Ann. Math. Statist. 1972. V. 43, N 4. P. 1041–1067.
8. Hald A. Statistical Theory with Engineering Applications. 1952.

УДК 004.324

В. П. Корячко¹, д-р техн. наук, зав. каф.,

В. А. Шибанов¹, канд. техн. наук, доц.,
e-mail: vova_shibanov@mail.ru,

Ю. Л. Ижванов², канд. техн. наук,
1-й зам. директора по НР,

А. П. Шибанов¹, д-р техн. наук, проф.

¹ Рязанский государственный
радиотехнический университет

² Государственный научно-исследовательский
институт информационных технологий
и телекоммуникаций

Оценка времени передачи файла с учетом старения информации

Рассматривается GERT-модель передачи файла по каналу связи с оконным управлением при условии потере ценности информационных кадров со временем. Число кадров в файле задается случайной величиной с дискретным распределением вероятностей. Приводятся соотношения для оценки вероятности старения файла в зависимости от времени.

Ключевые слова: GERT-сеть, старение кадров, старение файлов, оконное управление, сети с многопротокольной коммутацией по меткам MPLS

Введение

В компьютерных сетях ценность передаваемой информации может уменьшаться тем значительнее, чем больше времени занимает процесс передачи сообщения из конца в конец сети. Особенно отчетливо это проявляется тогда, когда предварительно выполняется сборка файла из нескольких информационных кадров, и только потом выполняется сама передача. Например, в сетях с многопротокольной коммутацией по меткам (*Multi Protocol Label Switching — MPLS*) для задания предельно допустимого значения времени жизни метки MPLS отведено поле TTL (*Time To Live*) длиной 8 бит. В случае непредвиденной задержки в тракте передачи или возникновения петли маршрутизации, например в случае сбоя оборудования, дальнейшее передвижение таких пакетов может проводиться по специальным алгоритмам. Во многих случаях после истечения некоторого интервала времени принимается решение об уничтожении "устаревшего" пакета. Если такие меры не предпринимать, то нахождение в сети пакетов с просроченными метками времени негативно сказывается на ее работе. Устаревшие кадры занимают лишнюю память в маршрутизаторах, их передвижение по сети уменьшает ресурсы телекоммуникационной системы, приводит к увели-

чению очередей, а в конечном итоге ухудшает показатели качества сети. Оценка степени негативного влияния устаревших пакетов позволяет сделать вывод о целесообразности применения процедур исключения кадров, "засоряющих" телекоммуникационные тракты. Поэтому вероятностно-временные характеристики канала связи должны уточняться с учетом эффекта старения информации.

В данной статье рассматривается GERT-сеть процесса передачи кадров по каналу связи в режиме оконного управления по алгоритму с возвращением на N кадров GBN (*GBN — Go Back N*) [1]. В этом алгоритме кадры передаются непрерывно без ожидания подтверждения приема определенного числа кадров. При получении отрицательной квитанции неподтвержденный и все последующие кадры передаются повторно. Положительная квитанция служит подтверждением правильной передачи не только данного кадра, но и всех предшествующих ему.

Постановка задачи

Процесс передачи файла рассмотрим при следующих предположениях:

- передаваемый файл чувствителен к задержкам и обладает высшим приоритетом в очереди коммутатора и маршрутизатора;
- передается файл, содержащий кадры экспоненциально распределенной длины;
- файл может содержать от одного до n кадров. Число кадров в файле случайное и задается дискретным распределением с вероятностями q_1, \dots, q_n ;
- суммарное время, включающее время распространения сигнала от передатчика к приемнику, время передачи кадра от передатчика к приемнику, время распространения сигнала от приемника к передатчику и время передачи отрицательной или положительной квитанции через канал связи, распределено по экспоненциальному закону с параметром λ ;
- время передачи кадра от передатчика к приемнику распределено по экспоненциальному закону с параметром μ ;
- вероятность успешной передачи кадра через канал связи за одну попытку передачи равна p ;
- время старения кадра задано постоянной величиной.

Нахождение условных распределений времени передачи файла с учетом старения

Пусть $g(t)$ и $G(t)$ — соответственно плотность вероятности и функция распределения времени передачи файла через сеть. Время передачи файла t может быть больше или меньше допустимого

времени старения α . Вероятности своевременной доставки файла и старения соответственно равны

$$P(t < \alpha) = G(\alpha);$$

$$P(t > \alpha) = 1 - P(t < \alpha) = 1 - G(\alpha). \quad (1)$$

Условные плотности вероятностей определяются выражениями

$$\text{для } t < \alpha: f_{t < \alpha}(t) = \begin{cases} g(t)/G(\alpha), & t < \alpha; \\ 0, & t \geq \alpha; \end{cases} \quad (2)$$

$$\text{для } t > \alpha: f_{t > \alpha}(t) = \begin{cases} 0, & t \leq \alpha, \\ g(t)/[1 - G(\alpha)], & t > \alpha. \end{cases}$$

Найдем условные функции распределения.

При $t < \alpha$:

$$F_{t < \alpha}(t) = \int_0^t f_{t < \alpha}(t) dt = \int_0^t \frac{g(t)}{G(\alpha)} dt = \frac{G(t)}{G(\alpha)}.$$

При $t > \alpha$:

$$F_{t < \alpha} = 0.$$

$$\text{Следовательно, } F_{t < \alpha}(t) = \begin{cases} G(t)/G(\alpha), & t < \alpha, \\ 0, & t \geq \alpha. \end{cases} \quad (3)$$

При $t \leq \alpha$ $F_{t > \alpha}(t) = 0$.

При $t > \alpha$

$$F_{t > \alpha}(t) = \int_0^t f_{t > \alpha}(t) dt = F_{t > \alpha}(\alpha) + \int_{\alpha}^t \frac{g(t)}{1 - G(\alpha)} dt = \frac{G(t) - G(\alpha)}{1 - G(\alpha)},$$

$$\text{откуда } F_{t > \alpha}(t) = \begin{cases} 0, & t \leq \alpha; \\ [G(t) - G(\alpha)]/[1 - G(\alpha)], & t > \alpha. \end{cases}$$

Для оценки быстродействия канала связи используем аппарат GERT-сетей (*GERT: Graphical Evaluation and Review Technique* — метод графической оценки и анализа систем). Поведение моделируемой системы рассматривается как процесс последовательного выполнения операций, каждая из которых характеризуется независимой случайной величиной. Выбор каждой следующей операции осуществляется в общем случае на основе вероятностного механизма. Выполняемые операции связываются с дугами GERT-сети, а ее узлы соответствуют состояниям системы. Случайное время выполнения операции характеризуется W -функцией, которая представляет собой произведение вероятности выполнения операции на производящую функцию моментов времени выполнения операции. Для замкнутой GERT-сети справедливо топологическое уравнение Мейсона [2], которое выражает эквивалентную W -функцию выходной случайной величины GERT-сети через эквивалентные W -функции всех петель первого по-

рядка (их аналогом являются простые контуры в ориентированном графе). По эквивалентной W -функции выходной величины GERT-сети можно найти соответствующую плотность распределения вероятностей [3].

Рассмотрим процесс передачи файла с повторной пересылкой всех кадров, начиная с места ошибки, на следующем примере. Пусть передается файл, состоящий из пяти кадров, и второй кадр передается ошибочно (рис. 1).

Время передачи файла $t_{\text{пер}}$ равно сумме следующих слагаемых:

а) времени распространения сигнала от передатчика к приемнику при передаче первого кадра $t_{\text{расп}}$;

б) длительностей передачи кадров $t_{\text{кад}}$ (учитывается время только успешных передач);

в) длительностей межкадровых интервалов $t_{\text{м.и}}$ для всех кадров, кроме последнего;

г) $t_{\text{ош}}$ — суммарного времени, затрачиваемого на передачу кадра с ошибкой, на распространение сигнала от приемника к передатчику, на передачу отрицательной квитанции и на распространение сигнала от передатчика к приемнику.

Составляющие а) и в) являются постоянными величинами, а составляющие б) и г) — случайными величинами. Постоянная составляющая времени передачи файла равна $t_{\text{пер}}^{\text{пост}} = t_{\text{расп}} + (k-1)t_{\text{м.и}}$, где k — число кадров в файле. В дальнейшем будем рассматривать только случайную составляющую времени передачи, имея в виду, что время старения файла уменьшено на величину $t_{\text{пер}}^{\text{пост}}$.

Для определения случайной составляющей времени передачи файла рассмотрим GERT-сеть,

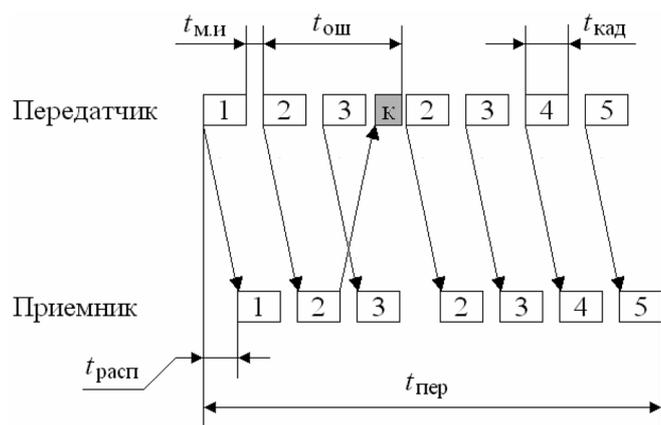


Рис. 1. Временная диаграмма процесса передачи файла

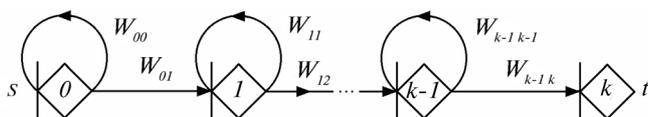


Рис. 2. GERT-сеть передачи файла по каналу связи

отображающую передачу через канал связи файла, состоящего из k кадров (рис. 2).

Дуги $(0,1), (1,2), \dots, (k-1, k)$ соответствуют успешной передаче 1-го, 2-го, ..., k -го кадров и отображают время, затраченное на передачу кадра от передатчика к приемнику. Время передачи положительной квитанции не учитывается, поскольку ее передача осуществляется параллельно с приемом следующих кадров. Дуги $(0,0), (1,1), \dots, (k-1, k-1)$ характеризуют неудачные попытки передачи соответственно 1-го, 2-го, ..., k -го кадров и связаны со временем, затраченным на двойное распространение сигнала, передачу кадра и отрицательной квитанции. Дуги, отражающие успешную передачу файла, характеризуются экспоненциальным временем передачи с параметром μ , а дуги, отражающие неудачные попытки передачи файла, — экспоненциальным временем передачи с параметром λ .

Дуги GERT-сети описываются следующими W -функциями:

$$W_{00}(s) = W_{11}(s) = \dots = W_{k-1, k-1}(s) = (1-p)\lambda/(\lambda-s);$$

$$W_{01}(s) = W_{12}(s) = \dots = W_{k-1, k}(s) = p\mu/(\mu-s).$$

Согласно [2] эквивалентная W -функция выходной величины данной GERT-сети

$$W_E(s) = \left(\frac{W_{01}(s)}{1-W_{00}(s)} \right) \cdot \dots \cdot \left(\frac{W_{k-1, k}(s)}{1-W_{k-1, k-1}(s)} \right) = \left(\frac{p\mu(\lambda-s)}{(p\lambda-s)(\mu-s)} \right)^k.$$

Эквивалентная производящая функция моментов

$$M_E(s) = W_E(s)/W_E(0) = \left(\frac{p\mu(\lambda-s)}{(p\lambda-s)(\mu-s)} \right)^k. \quad (4)$$

От производящей функции моментов $M_E(s)$ в формуле (4) заменой переменной s на $i\zeta$, где ζ — действительная переменная, перейдем к характеристической функции выходной величины GERT-сети $X_E(\zeta)$:

$$X_E(\zeta) = \left(\frac{p\zeta(\lambda-i\zeta)}{(p\lambda-i\zeta)(\mu-i\zeta)} \right)^k. \quad (5)$$

Из выражения (5) найдем плотность вероятности времени прохождения GERT-сети по формуле обращения характеристической функции [4]:

$$g_{k,E}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\zeta t} \left(\frac{p\zeta(\lambda-i\zeta)}{(p\lambda-i\zeta)(\mu-i\zeta)} \right)^k d\zeta.$$

Для вычисления последнего интеграла воспользуемся методами теории аналитических функций комплексного переменного. Проведем

замену переменной $z = -i\zeta$. Тогда последний интеграл сведется к следующему:

$$g_{k,E}(t) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} e^{zt} \left(\frac{p\mu(\lambda+z)}{(p\lambda+z)(\mu+z)} \right)^k dz. \quad (6)$$

Все полюсы функции $F(z) = \left(\frac{p\mu(\lambda+z)}{(p\lambda+z)(\mu+z)} \right)^k$

лежат левее мнимой оси. В левой полуплоскости функция $F(z)$ является аналитической, за исключением конечного числа полюсов, и стремится к нулю при $|z| \rightarrow \infty$ равномерно относительно $\arg z$, следовательно, для данной функции выполняются условия леммы Жордана [5]. Поэтому последний интеграл представляет собой интеграл Бромвича—Вагнера

$$g(t) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} e^{zt} F(z) dz = \sum_{k=1}^n \operatorname{Res} [e^{zt} F(z)].$$

Подынтегральная функция в формуле (6) имеет два полюса порядка k : $z = -p\lambda$ и $z = -\mu$. Вычет подынтегральной функции относительно точки z_k , являющейся полюсом k -го порядка, определяется как

$$\operatorname{Res}_{z=z_k} F(z) = \frac{1}{(k-1)!} \lim_{z \rightarrow z_k} \frac{d^{k-1} [(z-z_k)^k F(z)]}{dz^{k-1}}.$$

Введем следующие обозначения:

$$F_1(t, k) = \lim_{z \rightarrow -\mu} \frac{d^{k-1}}{dz^{k-1}} \left(\frac{(\lambda+z)^k}{(p\lambda+z)^k} e^{zt} \right);$$

$$F_2(t, k) = \lim_{z \rightarrow -p\lambda} \frac{d^{k-1}}{dz^{k-1}} \left(\frac{(\lambda+z)^k}{(\mu+z)^k} e^{zt} \right). \quad (7)$$

Выражения для функций (7) несложно получить в аналитическом виде лишь для малых значений k , в частности:

$$F_1(t, 1) = \frac{(\lambda-\mu)}{(p\lambda-\mu)} e^{-\mu t}, \quad F_2(t, 1) = \frac{\lambda(1-p)}{(\mu-p\lambda)} e^{-p\lambda t};$$

$$F_1(t, 2) = \frac{(\lambda-\mu)^2}{(p\lambda-\mu)^2} e^{-\mu t} \left(t - \frac{2}{p\lambda-\mu} + \frac{2}{\lambda-\mu} \right);$$

$$F_2(t, 2) = \frac{\lambda^2(1-p)^2}{(p\lambda-\mu)^2} e^{-p\lambda t} \left(t - \frac{2}{\mu-p\lambda} + \frac{2}{\lambda(1-p)} \right);$$

$$F_1(t, 3) = \frac{\lambda-\mu}{(p\lambda-\mu)^3} e^{-\mu t} (t^2(\lambda-\mu)^2 +$$

$$+ 6t \frac{\lambda(p-1)(\lambda-\mu)}{(p\lambda-\mu)} + 6 - 18 \frac{\lambda-\mu}{p\lambda-\mu} + 12 \frac{(\lambda-\mu)^2}{(p\lambda-\mu)^2});$$

$$F_2(t, 3) = \frac{\lambda(1-p)}{(\mu-p\lambda)^3} e^{-p\lambda t} \left(t^2 \lambda^2 (1-p)^2 + \right.$$

$$+ 6t \frac{\lambda(1-p)(\mu-\lambda)}{(\mu-p\lambda)} + 6 - 18 \frac{\lambda(1-p)}{\mu-p\lambda} +$$

$$\left. + 12 \frac{\lambda^2(1-p)^2}{(p\lambda-\mu)^2} \right).$$

По индукции для произвольного значения k получаем следующие выражения для функций (7):

$$F_1(t, k) = e^{-\mu t} \sum_{j=0}^{k-1} t^j b_{k,j}; \quad F_2(t, k) = e^{-p\lambda t} \sum_{j=0}^{k-1} t^j c_{k,j},$$

где $b_{k,j}$, $c_{k,j}$ — постоянные коэффициенты, значения которых определяются значениями параметров модели p , λ , μ .

С учетом введенных обозначений плотность вероятности времени прохождения GERT-сети определяется выражением

$$g_{k,E}(t) = \frac{p^k \mu^k}{(k-1)!} [e^{-\mu t} \sum_{j=0}^{k-1} t^j b_{k,j} + e^{-p\lambda t} \sum_{j=0}^{k-1} t^j c_{k,j}].$$

В общем случае файл может состоять из переменного числа кадров. Пусть заданы q_k — вероятности того, что файл состоит из k кадров. Тогда по формуле полной вероятности [6] плотность вероятности времени передачи файла

$$g(t) = \sum_{k=1}^n q_k \frac{p^k \mu^k}{(k-1)!} [e^{-\mu t} \sum_{j=0}^{k-1} t^j b_{k,j} +$$

$$+ e^{-p\lambda t} \sum_{j=0}^{k-1} t^j c_{k,j}].$$

Функцию распределения находим, интегрируя плотность от 0 до t :

$$G(t) = \int_0^t \sum_{k=1}^n q_k \frac{p^k \mu^k}{(k-1)!} [e^{-\mu t} \sum_{j=0}^{k-1} t^j b_{k,j} +$$

$$+ e^{-p\lambda t} \sum_{j=0}^{k-1} t^j c_{k,j}] dt;$$

$$G(t) = \sum_{k=1}^n q_k \frac{p^k \mu^k}{(k-1)!} \left[\sum_{j=0}^{k-1} b_{k,j} \int_0^t e^{-\mu t} t^j dt + \right.$$

$$\left. + \sum_{j=0}^{k-1} c_{k,j} \int_0^t e^{-p\lambda t} t^j dt \right].$$

После интегрирования получаем итоговую формулу для нахождения функции распределения времени передачи файла:

$$G(t) = \sum_{k=1}^n q_k \frac{p^k \mu^k}{(k-1)!} \left[\sum_{j=0}^{k-1} b_{k,j} \left(\frac{j!}{\mu^{j+1}} - e^{-\mu t} \sum_{i=0}^j \frac{j! \cdot t^i}{i! \cdot \mu^{j+1-i}} \right) + \sum_{j=0}^{k-1} c_{k,j} \left(\frac{j!}{(p\lambda)^{j+1}} - e^{-p\lambda t} \sum_{i=0}^j \frac{j! \cdot t^i}{i! \cdot (p\lambda)^{j+1-i}} \right) \right]$$

Отсюда по формуле (1) находим вероятность старения файла при фиксированном значении времени старения α :

$$P_c(\alpha) = 1 - \sum_{k=1}^n q_k \frac{p^k \mu^k}{(k-1)!} \left[\sum_{j=0}^{k-1} b_{k,j} \left(\frac{j!}{\mu^{j+1}} - e^{-\mu \alpha} \sum_{i=0}^j \frac{j! \cdot \alpha^i}{i! \cdot \mu^{j+1-i}} \right) + \sum_{j=0}^{k-1} c_{k,j} \left(\frac{j!}{(p\lambda)^{j+1}} - e^{-p\lambda \alpha} \sum_{i=0}^j \frac{j! \cdot \alpha^i}{i! \cdot (p\lambda)^{j+1-i}} \right) \right] \quad (8)$$

По формулам (2) и (3) находим соответствующие выражения для условных плотности вероятностей и функции распределения времени передачи файла для неустаревших файлов:

$$f_c(t) = \frac{\sum_{k=1}^n q_k \frac{p^k \mu^k}{(k-1)!} \left[e^{-\mu t} \sum_{j=0}^{k-1} t^j b_{k,j} + e^{-p\lambda t} \sum_{j=0}^{k-1} t^j c_{k,j} \right]}{\int_0^\alpha \sum_{k=1}^n q_k \frac{p^k \mu^k}{(k-1)!} \left[e^{-\mu t} \sum_{j=0}^{k-1} t^j b_{k,j} + e^{-p\lambda t} \sum_{j=0}^{k-1} t^j c_{k,j} \right] dt}$$

$$F_c(t) = \frac{\int_0^t \sum_{k=1}^n q_k \frac{p^k \mu^k}{(k-1)!} \left[e^{-\mu t} \sum_{j=0}^{k-1} t^j b_{k,j} + e^{-p\lambda t} \sum_{j=0}^{k-1} t^j c_{k,j} \right] dt}{\int_0^\alpha \sum_{k=1}^n q_k \frac{p^k \mu^k}{(k-1)!} \left[e^{-\mu t} \sum_{j=0}^{k-1} t^j b_{k,j} + e^{-p\lambda t} \sum_{j=0}^{k-1} t^j c_{k,j} \right] dt}$$

В качестве примера на рис. 3 приведены графики зависимостей вероятности старения файла от значения параметра времени старения, построен-

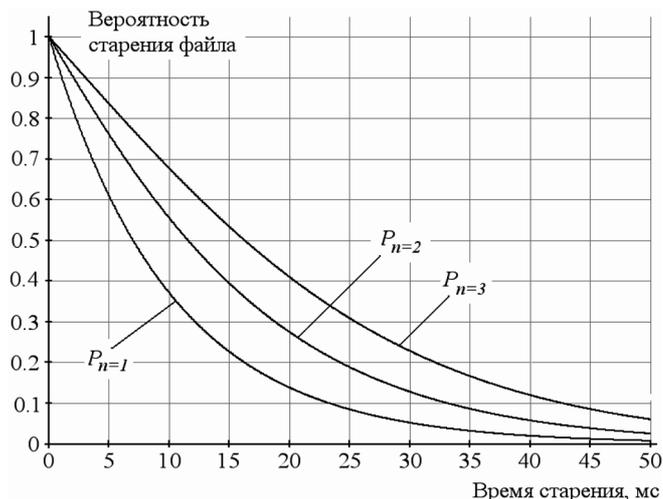


Рис. 3. Вероятности старения файла в зависимости от значения параметра времени старения

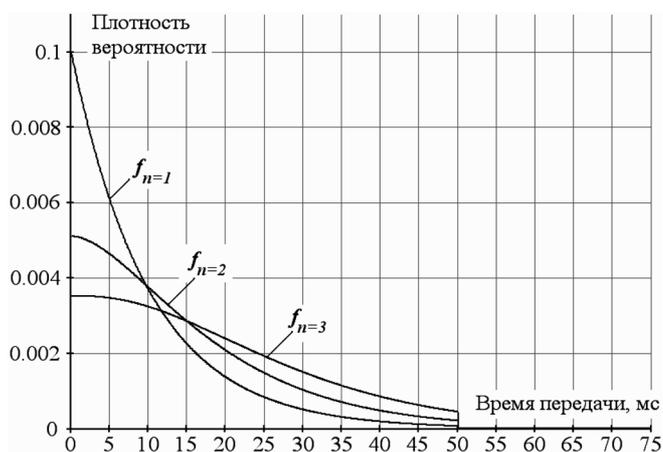


Рис. 4. Условные плотности распределения времени передачи файла

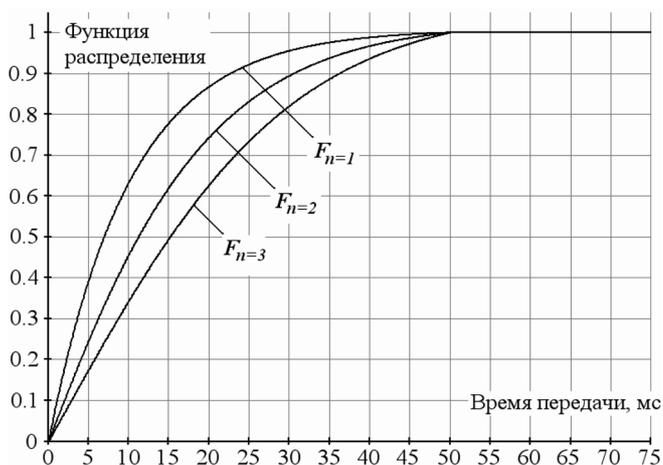


Рис. 5. Условные функции распределения времени передачи файла

ные по формуле (8), для $n = \overline{1, 3}$, $q_k = 1/n$, $p = 0,995$, $\lambda = 0,05 \text{ мс}^{-1}$, $\mu = 0,1 \text{ мс}^{-1}$.

Графики условных плотностей вероятности и функций распределения для различных значений времени старения и $n = 1, 2, 3$; $q_k = 1/n$; $p = 0,995$; $\lambda = 0,05 \text{ мс}^{-1}$; $\mu = 0,1 \text{ мс}^{-1}$, $\alpha = 50 \text{ мс}$ при дискретном, равномерно распределенном числе кадров в файле приведены соответственно на рис. 4 и 5.

Выводы

Разработанная модель отражает процесс передачи файла достаточно большой длины по предварительно установленному соединению. В общем случае файл разбивается на кадры различной длины. Это характерно для передачи в реальном времени зашифрованной речи (со сжатием информации), информации видеокамер со сжатием отдельных фрагментов изображения, наборов команд (файла команд) для исполнительного оборудования при управлении промышленными объектами, передаваемых с определенной периодичностью и т. д.

Рассмотренная модель может быть применена в качестве отдельных компонентов в более сложных моделях. Можно использовать GERT-сети как базовые модели для расчета характеристик агрегированного канала, состоящего из параллельно соединенных звеньев. Вводится общая начальная вершина, выполняющая выходную функцию направления кадров без задержки на входы GERT-сетей, отражающих работу отдельных параллельных каналов. Вычисляются распределения F_1, F_2, \dots, F_n выходных случайных величин t_1, t_2, \dots, t_n GERT-сетей с учетом значения параметра старения. Распределение G времени прохождения кадров через агрегированный канал находится как распределение максимальной из n случайных величин $G = \max(t_1, t_2, \dots, t_n) = F_1 F_2 \dots F_n$.

Время выполнения отдельных операций процесса передачи данных может характеризоваться непрерывными распределениями произвольного вида. В этом случае выходные распределения GERT-сетей находятся с использованием численных методов [3]. Если в базовой GERT-сети выделяются частичные графы, выходные распределения которых находятся с использованием разных методов расчета, например, на основе численных методов и с использованием теории

аналитических функций, то мы имеем дело с неоднородными GERT-сетями [7].

Функциональные возможности анализа и синтеза систем передачи информации существенно возрастают, если применяются комбинированные модели, создаваемые с использованием GERT-сетей и средств имитации. При этом условные функции распределения с учетом фактора старения информации, полученные на основе расчета GERT-сетей, используются для получения выборочных случайных задержек в системе имитации. Например, с позиции теории очередей канал передачи данных может рассматриваться как обслуживающий прибор, а указанные выборки определяют время передачи заявки в каждом имитационном прогоне. Тогда моделирование работы сетевой транспортной среды с учетом заполнения и освобождения входных и выходных буферов канала может быть выполнено с использованием средств компьютерной имитации системы массового обслуживания. Описание такой системы моделирования приведено, в частности, в работе [8]. Комбинированные модели такого рода можно использовать для решения задач нахождения наилучших проектных вариантов при создании или модернизации современных широкополосных сетей на базе технологий MPLS, 10 Gigabit Ethernet, высокоскоростных магистралей синхронной цифровой иерархии SDH (*Synchronous Digital Hierarchy*) и уплотненного волнового мультиплексирования DWDM (*Dense Wave Division Multiplexing*).

Работа поддержана Российским фондом фундаментальных исследований, грант 07-07-0146 а.

Список литературы

1. **Олифер В. Г., Олифер Н. А.** Компьютерные сети. Принципы, технологии, протоколы. СПб.: Питер, 2000. 672 с.
2. **Pritsker A. A.** GERT-Graphical evaluation and review technique. Memorandum RM-4973-NASA, 1966.
3. **Шибанов А. П.** Нахождение плотности распределения времени исполнения GERT-сети на основе эквивалентных упрощающих преобразований // Автоматика и телемеханика, 2003. № 2. С. 117–126.
4. **Феллер В.** Введение в теорию вероятностей и ее приложения: Пер. с англ. В 2-х томах. Т. 2. М.: Мир, 1984. 738 с.
5. **Привалов И. И.** Введение в теорию функций комплексного переменного. М.: Наука, 1984. 432 с.
6. **Вентцель Е. С., Овчаров Л. А.** Теория вероятностей и ее инженерные приложения. М.: Наука, 1988. 480 с.
7. **Шибанов А. П.** Нахождение закона распределения выходной величины GERT-сети большой размерности // Информационные технологии, 2002. № 1. С. 42–45.
8. **Корячко В. П., Шибанов А. П., Шибанов В. А., Чернышев А. С.** Имитационная система моделирования телекоммуникационных сетей // Телекоммуникации. 2001. № 10. С. 28–32.

Комбинаторная модель функционального формирователя разбиений бинарного множества

Представлена модель формирователя разбиений бинарного множества. Предложенное решение имеет эффективную аппаратную реализацию и высокое быстродействие.

Ключевые слова: разбиение множества, многоуровневая коммутационная сеть, баньян переключатель

Введение

В ряде прикладных задач, например, в криптографии или в системах телекоммуникаций при организации нескольких линий одновременного соединения с одним "клиентом" представляют интерес управляемые внешним кодом устройства, осуществляющие произвольные разбиения исходного множества входных данных. При этом n выходов формирователя разбиений разделены на m кластеров, и каждый из его n входов может быть соединен с выходом любого выходного кластера. Рассматриваемые в данной статье формирователи разбиений являются управляемыми внешним кодом, поэтому согласно терминологии, введенной в [1], их можно отнести к классу функциональных формирователей разбиений.

Задача осуществления произвольного разбиения входного множества данных может быть решена с использованием коммутационных сетей с топологией Бенеша [2, 3]. Эти сети состоят из соединенных между собой одинаковых переключателей, расположенных в матричном порядке с $n/2$ линиями и $2k - 1$ уровнями, где $k = \log_2 n$. Каждый переключатель имеет два входа, два выхода данных и битовый управляющий вход. Причем в зависимости от значения бита управляющего кода переключатель осуществляет соединение либо первого входа с первым выходом, а второго входа со вторым выходом, либо первого входа со вторым выходом, а второго входа с первым выходом. Используя сеть с топологией Бенеша, можно реализовать любую перестановку входных данных или осуществить одновременное соединение каждого из n входов сети с любым из n выходов.

Предлагаемое в статье решение дает возможность уменьшить число уровней переключателей матрицы формирователя разбиений, упрощая, та-

ким образом, его конструкцию и увеличивая быстродействие.

Целью данной работы является разработка модели функционального формирователя упорядоченных разбиений исходного n -элементного множества S , где $n = 2^k$, на подмножества одинаковой мощности, равной 2^u , где $u \in \{0, 1, 2, \dots, k - 1\}$.

Отличительными особенностями предложенного решения являются эффективная аппаратная реализация формирователя разбиений, большое число возможных топологий соединений переключателей матрицы формирователя, меньшее время задержки преобразования.

Топологию коммутационной матрицы формирователя можно представить в виде орграфа

$G(P, \bar{A})$, где P — множество вершин, а \bar{A} — множество ребер графа. Множество вершин P состоит из множества n входных вершин, множества n выходных вершин и множества промежуточных вершин. Промежуточные вершины образуют матрицу с $n/2$ линиями и $2\log_2 n - u - 1$ уровнями и интерпретируются как переключатели коммутационной матрицы формирователя. Множество ребер графа описывает соединения между переключателями. Коды управления переключателями являются битовыми атрибутами промежуточных вершин орграфа. В зависимости от их значений формируются разрешенные пути прохождения вершин графа. Каждая промежуточная вершина является смежной двум вершинам предыдущего уровня и двум вершинам следующего уровня. Каждая вершина входного множества является смежной двум промежуточным вершинам первого уровня. Каждая вершина выходного множества является смежной двум промежуточным вершинам последнего уровня.

Таким образом, для построения модели необходимо:

- определить последовательность преобразований, реализуемых на каждом уровне вершин графа;
- определить возможные соединения между вершинами графа;
- разработать алгоритм определения атрибутов промежуточных вершин орграфа, формирующей динамическую топологию соединений коммутационной матрицы формирователя.

При этом удобно считать, что множество входных данных разбивается на подмножества, и на любом уровне орграфа осуществляется разбиение каждого подмножества входных данных на два других подмножества.

Пусть S — множество входных данных.

Разбиением множества S называется совокупность непустых, попарно непересекающихся под-

множеств множества S , объединение которых совпадает с S . Число упорядоченных разбиений n -элементного множества на m частей фиксированного размера $|S_1|, |S_2|, \dots, |S_m|$ определяется мульти-

$$\text{номиальным коэффициентом } \binom{n}{|S_1|, |S_2|, \dots, |S_m|} = \frac{n!}{|S_1|! \cdot |S_2|! \cdot \dots \cdot |S_m|!}.$$

Покажем, что осуществляя рекурсивно разбиения множества S на два подмножества за $2\log_2 n - u - 1$ шагов можно реализовать такое разбиение S на множества S_m мощностью $|S_m| = 2^u$, что любой из элементов $\tilde{s}_i \in S$ попадет в любое из множеств S_m . При разбиении S используются два различных алгоритма. В соответствии с этим разбиения разделяются на две последовательности: входную и выходную. Входная последовательность разбиений содержит $\log_2 n - 1$ шагов, выходная — $\log_2 n - u$ шагов.

Зададим на множестве S биективное отношение $I: S \leftrightarrow S_{00}$, где $S_{00} = \{0, 1, 2, \dots, n - 1\}$. Элементы S_{00} необходимы для настройки переключателей матрицы. Каждое число $s_i \in S_{00}$ можно представить в виде двоичного разложения

$$s_i = b_k^i 2^{k-1} + b_{k-1}^i 2^{k-2} + \dots + b_1^i 2^0, \quad (1)$$

где $b_j^i \in \{0, 1\}$.

Требуемое разбиение $S = \bigcup_{m=0}^{2^k-u} S_m$ зададим выражением

$$S_m = \{\tilde{s}_i | \tilde{s}_i = I(s_i) \& (b_k^i 2^{k-u-1} + b_{k-1}^i 2^{k-u-2} + \dots + b_{u+1}^i 2^0) = m\}. \quad (2)$$

Таким образом, задача построения произвольного разбиения входных данных сводится к разбиению элементов множества S_{00} на блоки, элементы которых имеют $k - u$ одинаковых старших разрядов. Мы будем использовать элементы множества S_{00} в качестве входных данных и, осуществляя их сортировку по выходным кластерам, настраивать последовательно каждый уровень переключателей матрицы.

Пусть

$$(\nu_1^i, \nu_2^i, \dots, \nu_k^i) = \eta(b_1^i, b_2^i, \dots, b_k^i) \quad (3)$$

— перестановка двоичных разрядов разложения (1). Положим также, что перестановка $\eta(b_1^i, b_2^i,$

$\dots, b_k^i)$ из выражения (2) относится к старшим $k - u$ разрядам разложения (1), т. е. $(\nu_1^i, \nu_2^i, \dots, \nu_u^i) = (b_1^i, b_2^i, \dots, b_u^i)$.

Входная последовательность разбиений

Построим разбиение $S_{00} = S_{11} \cup S_{21}$ такое, что

$$(|S_{11}| = |S_{21}|) \& (\forall s_i \in S_{11})(\exists s_j \in S_{21}) \times ((\nu_2^i, \dots, \nu_k^i) = (\nu_2^j, \dots, \nu_k^j)).$$

Таким образом,

$$(\forall s_i, s_j \in S_{11})(i \neq j \& (\nu_3^i, \nu_4^i, \dots, \nu_k^i) = (\nu_3^j, \nu_4^j, \dots, \nu_k^j)) \rightarrow (\nu_2^i \neq \nu_2^j);$$

$$(\forall s_i, s_j \in S_{21})(i \neq j \& (\nu_3^i, \nu_4^i, \dots, \nu_k^i) = (\nu_3^j, \nu_4^j, \dots, \nu_k^j)) \rightarrow (\nu_2^i \neq \nu_2^j).$$

На втором шаге построим рекурсивно следующее разбиение множества S_{00} на равные подмно-

$$\text{жества } S_{00} = \bigcup_{i=1}^4 S_{i2};$$

$$|S_{12}| = |S_{22}| = |S_{32}| = |S_{42}|,$$

причем $\{S_{12}, S_{22}\}$ — разбиение S_{11} , а $\{S_{32}, S_{42}\}$ — разбиение S_{21} и

$$(\forall s_i \in S_{12})(\exists s_j \in S_{22})((\nu_3^i, \nu_4^i, \dots, \nu_k^i) = (\nu_3^j, \nu_4^j, \dots, \nu_k^j));$$

$$(\forall s_i \in S_{32})(\exists s_j \in S_{42})((\nu_3^i, \nu_4^i, \dots, \nu_k^i) = (\nu_3^j, \nu_4^j, \dots, \nu_k^j)). \quad (4)$$

Аналогично на третьем шаге построим разбиение каждого из множеств $S_{12}, S_{22}, S_{32}, S_{42}$ и т. д. Рекурсивные разбиения выполняем $k - 1$ раз.

Для каждого множества S_{pq} на шаге разбиения q справедливо

$$(\forall s_i \in S_{pq})(\exists s_j \in S_{pq})((\nu_{q+2}^i, \nu_{q+3}^i, \dots, \nu_k^i) = (\nu_{q+2}^j, \nu_{q+3}^j, \dots, \nu_k^j));$$

$$(\forall s_i, s_j \in S_{pq})(i \neq j \& (\nu_{q+2}^i, \nu_{q+3}^i, \dots, \nu_k^i) = (\nu_{q+2}^j, \nu_{q+3}^j, \dots, \nu_k^j)) \rightarrow (\nu_{q+1}^i \neq \nu_{q+1}^j); \quad (5)$$

$$|S_{pq}| = 2^{k-q}.$$

Множества $S_{p, k-1}$ содержат по два элемента $|S_{p, k-1}| = 2$. Согласно (5) для пары элементов $s_i, s_j \in S_{p, k-1}$ справедливо

$$(v_k^i = 1) \& (v_k^j = 0) \vee (v_k^i = 0) \& (v_k^j = 1). \quad (6)$$

Выходная последовательность разбиений

Выходная последовательность разбиений множества S_{00} предназначена для реализации распределения элементов s_i по выходным кластерам.

На первом шаге построим разбиение $S_{00} = D_{11} \cup D_{21}$, причем

$$\begin{aligned} (\forall s_i \in D_{11})(v_k^i = 1); \\ (\forall s_i \in D_{21})(v_k^i = 0). \end{aligned} \quad (7)$$

Согласно (6), (7), множества D_{11}, D_{21} содержат по одному элементу каждого из двухэлементных множеств $S_{p, k-1}$.

На втором шаге построим разбиение $S_{00} = \bigcup_{i=1}^4 D_{i2}$, причем

$$\begin{aligned} D_{11} = D_{12} \cup D_{22}, \quad D_{21} = D_{32} \cup D_{42} \text{ и} \\ (\forall s_i \in D_{12})(v_{k-1}^i = 1); \\ (\forall s_i \in D_{22})(v_{k-1}^i = 0); \\ (\forall s_i \in D_{32})(v_{k-1}^i = 1); \\ (\forall s_i \in D_{42})(v_{k-1}^i = 0). \end{aligned} \quad (8)$$

Множества $D_{12}, D_{22}, D_{32}, D_{42}$ содержат по одному элементу каждого из множеств $S_{p, k-2}$.

На шаге r построим разбиение $S_{00} = \bigcup_{i=1}^{2^r} D_{ir}$

$$\begin{aligned} (\forall s_i \in D_{2p-1, r})(v_{k-r+1}^i = 1); \\ (\forall s_i \in D_{2p, r})(v_{k-r+1}^i = 0), \end{aligned} \quad (9)$$

где $p = (1, 2, \dots, 2^{r-1})$.

Указанные выше разбиения (7), (8), (9) возможны, так как число элементов $s_i \in S_{00}$, имеющих $v_k^i = 1$, равно числу элементов, у которых $v_k^i = 0$. Для любого множества D_{wr} число элемен-

тов $s_i \in D_{wr}$, имеющих $v_{k-r+1}^i = 1$, равно числу элементов, у которых $v_{k-r+1}^i = 0$.

Разбиения повторяются $k-u$ раз, причем

$$\begin{aligned} (\forall s_i, s_j \in D_{m, k-u})(v_{u+1}^i, v_{u+2}^i, \dots, v_k^i) = \\ = (v_{u+1}^j, v_{u+2}^j, \dots, v_k^j). \end{aligned} \quad (10)$$

Доказательство утверждения (10) проведем от противного. Пусть $(\exists s_i, s_j \in D_{m, k-u})(\exists l \in \{u+1, u+2, \dots, k\})(v_l^i = 1)(v_l^j = 0)$, тогда согласно (9) $s_i \in D_{2p-1, l}, s_j \in D_{2p, l}$ но элементы s_i, s_j не могут принадлежать разным множествам, так как или $D_{m, k-u} \subset D_{2p-1, l}$ или $D_{m, k-u} \subset D_{2p, l}$. Полученное противоречие доказывает утверждение (10).

Таким образом, множества $D_{m, k-u}$ образуют выходные кластеры. Причем $k-u$ старших двоичных разрядов чисел s_i каждого кластера совпадают. Каждый кластер содержит 2^u чисел. Реализация приведенных выше входной и выходной последовательностей преобразований дает возможность осуществить любое разбиение элементов множества S . Управление этим разбиением осуществляется с использованием элементов множества S_{00} .

Реализация входной последовательности разбиений с использованием переключателей 2×2

Реализация рассмотренных выше последовательностей разбиений имеет эффективное аппаратное решение. Входная последовательность разбиений осуществляется с использованием $(k-1)$ -го уровня переключателей 2×2 . Каждый из переключателей имеет два входа, два выхода данных и бинарный вход управляющего кода $c \in \{0, 1\}$. Данные на входах переставляются местами (при значении управляющего кода $c = 1$) либо не переставляются (при $c = 0$), т. е.

$$\begin{aligned} y_1 = \bar{c}x_1 + cx_2; \\ y_2 = cx_1 + \bar{c}x_2, \end{aligned}$$

где x_1, x_2 — входные данные, а y_1, y_2 — выходные (черта над символом управляющего кода c обозначает операцию отрицания). Такая функциональная схема реализуется с помощью простейшей комбинационной схемы с управляемыми ключами, которые в зависимости от значения управляющего сигнала либо замыкают, либо размыкают цепь.

Определим компоненты бинарного вектора $C = (c_1, c_2, \dots, c_w)$ так, чтобы реализовать требуе-

мое разбиение множества S_{pq} на два подмножества $S_{p, q+1}$ и $S_{p+1, q+1}$:

$$\begin{aligned} S_{p, q+1} &= \{c_i s_{2i-1} + \bar{c}_i s_{2i} | i = (1, 2, \dots, w)\}; \\ S_{p+1, q+1} &= \{\bar{c}_i s_{2i-1} + c_i s_{2i} | i = (1, 2, \dots, w)\}; \\ S_{p, q+1} \cup S_{p+1, q+1} &= S_{pq}, S_{p, q+1} \cap S_{p+1, q+1} = \emptyset, \\ |S_{p, q+1}| &= |S_{p+1, q+1}|, \end{aligned} \quad (11)$$

где $w = \frac{|S_{pq}|}{2}$, $s_i \in S_{pq}$.

Утверждение.

$$\begin{aligned} (\forall S_{pq})(\exists C = (c_1, c_2, \dots, c_w))(\forall s_i \in S_{p, q+1})(\exists s_j \in \\ \in S_{p+1, q+1})((v_{q+2}^i, \dots, v_k^i) = (v_{q+2}^j, \dots, v_k^j)). \end{aligned} \quad (12)$$

Доказательство.

Для доказательства приведен алгоритм определения компонент вектора C . Операцию вычисления значения элемента $c_m \in \{0, 1\}$ будем рассматривать как настройку соответствующего переключателя матрицы формирователя. Выберем произвольный элемент s_i . Согласно (5)

$$\begin{aligned} (\forall s_i \in S_{p, q})(\exists s_j \in S_{p, q})((v_{q+2}^i, \dots, v_k^i) = \\ = (v_{q+2}^j, \dots, v_k^j)). \end{aligned}$$

Согласно (11)

$$\begin{aligned} y_{1m} &= c_m s_{i+1} + \bar{c}_m s_i; \\ y_{2m} &= c_m s_i + \bar{c}_m s_{i+1}. \end{aligned} \quad (13)$$

Для определенности будем считать, что $(\forall m)(y_{1m} \in S_{p, q+1})(y_{2m} \in S_{p+1, q+1})$. Пусть $c_m = 0$, тогда $y_{1m} = s_i$, $y_{2m} = s_{i+1}$ и $s_i \in S_{p, q+1}$, $s_{i+1} \in S_{p+1, q+1}$.

Алгоритм определения компонент вектора C :

1. Если в выражении (13) $i + 1 = j$, тогда для пары (s_i, s_j) утверждение (12) выполняется автоматически, так как $s_j \in S_{p+1, q+1}$.

2. Если $(i + 1 \neq j)$, то $(\exists c_l)(l \neq m)(y_{1l} = s_{i+1})(y_{2l} = s_j)$. Таким образом, $s_j \in S_{p+1, q+1}$, $s_{i+1} \in S_{p, q+1}$. Элемент s_{j+1} , как и элемент s_i , принадлежит множеству $S_{p, q+1}$.

3. Если $(v_{q+2}^{j+1}, \dots, v_k^{j+1}) \neq (v_{q+2}^{i+1}, \dots, v_k^{i+1})$, то повторяются приведенные выше рассуждения по пп. 1, 2 для элемента с номером $i = j + 1$.

4. Если $(v_{q+1}^{j+1}, \dots, v_k^{j+1}) = (v_{q+1}^{i+1}, \dots, v_k^{i+1})$ и определены не все компоненты бинарного вектора C , тогда $(\exists c_m \in Q)(y_{1m} = c_m s_{i+1} +$

$+ \bar{c}_m s_i)(y_{2m} = c_m s_i + \bar{c}_m s_{i+1})$, где Q — множество неопределенных элементов вектора C . Рассуждения для следующего произвольно выбранного элемента s_i повторяются.

5. Если $(v_{q+1}^{j+1}, \dots, v_k^{j+1}) = (v_{q+1}^{i+1}, \dots, v_k^{i+1})$, и все компоненты бинарного вектора C определены ($Q = \emptyset$), доказательство завершено.

Из анализа приведенного выше алгоритма следует, что для любого множества S_{pq} по крайней мере один элемент $c_i \equiv 0$. Соответствующий переключатель матрицы формирователя является неуправляемым.

Для ускорения работы определение компонент вектора C может быть выполнено с использованием параллельного алгоритма [4].

Таким образом, в соответствии с (11) множество переключателей (вершин орграфа) на каждом уровне q входной части матрицы формирователя можно разбить на подмножества так, что на входы переключателей одного подмножества P_{pq} поступают данные из множества S_{pq} . Выходы каждого переключателя этого подмножества соединены с переключателями двух подмножеств следующего уровня $P_{p, q+1}$ и $P_{p+1, q+1}$.

Выше был приведен один из возможных алгоритмов определения кодов управления переключателями входной части матрицы формирователя для реализации заданного разбиения.

Реализация выходной последовательности разбиений с использованием переключателей 2×2

Доказательство возможности соединения и настройки переключателей для реализации выходной последовательности разбиений проведем по индукции.

1. Любой переключатель первого уровня можно настроить, если его входными данными являются элементы из множества $S_{p, k-1}$. Согласно (6) для пары элементов $s_i, s_j \in S_{p, k-1}$ справедливо

$v_k^i = 1 \wedge v_k^j = 0 \vee v_k^i = 0 \wedge v_k^j = 1$. Если такие пары являются входными данными переключателя, разбиение (7) реализуется на первом уровне, содержащем $\frac{n}{2}$ переключателей.

2. Пусть возможно настроить переключатели до уровня q . На входы каждого переключателя уровня $q + 1$ поступают числа из множества $S_{p, k-q}$. Предположим противное, на уровне $q + 1$ существует переключатель, который невозможно настроить, т. е. $(\exists s_i, s_j \in S_{p, k-q})(i \neq j \ \& \ v_{k-q+1}^i = v_{k-q+1}^j)$.

Согласно (10) $(\forall s_i, s_j \in D_{mq})((v_{k-q+1}^i, v_{k-q+2}^i, \dots, v_k^i) = (v_{k-q+1}^j, v_{k-q+2}^j, \dots, v_k^j))$. Но согласно (5) $(\forall s_i, s_j \in S_{p, k-q})(i \neq j \& (v_{k-q+2}^i, v_{k-q+3}^i, \dots, v_k^i) = (v_{k-q+2}^j, v_{k-q+3}^j, \dots, v_k^j)) \rightarrow (v_{k-q+1}^i \neq v_{k-q+1}^j)$.

Полученное противоречие доказывает утверждение.

На рис. 1 приведена диаграмма орграфа для фрагмента формирователя разбиений, иллюстрирующая соединения переключателей выходной части. Один из выходов промежуточных переключателей не показан. На входы переключателя уровня G_3 поступают элементы $s_i, s_j \in S_{11}$. Согласно (5) $(\forall s_i, s_j \in S_{11})(i \neq j \& (v_3^i, v_4^i, \dots, v_k^i) = (v_3^j, v_4^j, \dots, v_k^j)) \rightarrow (v_2^i \neq v_2^j)$, поэтому $(v_2^i = 1 \& v_2^j = 0) \vee (v_2^i = 0 \& v_2^j = 1)$.

Таким образом, на входы каждого переключателя первого уровня G_1 поступают данные только из соответствующего двухэлементного множества $S_{p, k-1}$. Переключатели на первом уровне образуют одно множество \tilde{P}_{11} . На входы переключателей, расположенных на уровне G_q ($q > 1$), поступают данные только из соответствующего множества $S_{p, k-q}$. Причем множество переключателей

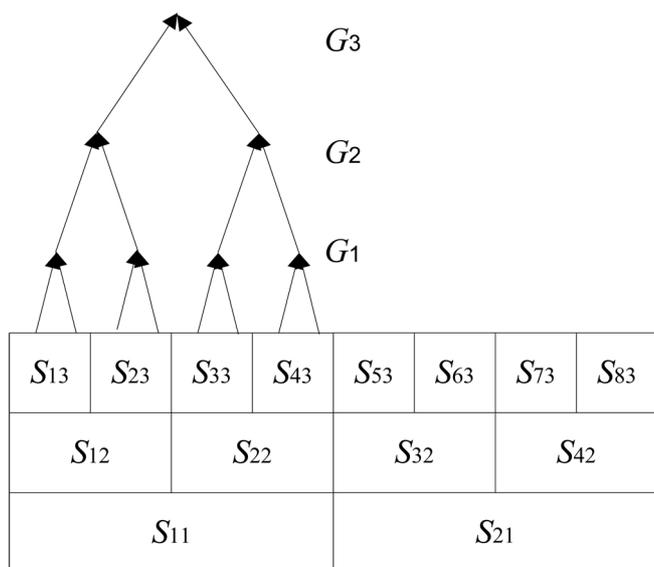


Рис. 1. Диаграмма орграфа для фрагмента коммутационной матрицы формирователя разбиений для случая $n = 16, u = 1$

на этом уровне разбито на 2^{q-1} подмножеств \tilde{P}_{mq} , где $m = \overline{1, 2^{q-1}}$. На входы каждого переключателя из некоторого подмножества \tilde{P}_{mq} поступают данные из соответствующего множества D_{mq} . Эти правила определяют возможные соединения между переключателями соседних уровней, а также входной и выходной частей матрицы формирователя разбиений.

Алгоритм настройки переключателей определяется следующим правилом. Согласно (9) входные данные переключателей из подмножества \tilde{P}_{pq} ($p = \overline{1, 2^{q-2}}$) разбиваются на выходах на два множества $D_{2p-1, q}$ и $D_{2p, q}$, причем $|D_{2p-1, q}| = |D_{2p, q}|$, и у любого элемента s_i из множества $D_{2p-1, q}$ бит $v_{k-q+1}^i = 1$, а у любого элемента s_i из множества $D_{2p, q}$ бит $v_{k-q+1}^i = 0$.

Пример. Формирователь разбиений для случая $n = 16, u = 1$

На рис. 2 приведена диаграмма орграфа одного из вариантов коммутационной матрицы формирователя разбиений для случая $n = 16, u = 1$.

Входная часть состоит из управляемых и неуправляемых переключателей T_{ij} , где $i = \overline{1, n}$, $j = \overline{1, k-1}$. Выходная часть состоит из аналогичных управляемых переключателей V_{sg} где $s = \overline{1, n}$, $g = \overline{1, k-u}$, $u < k$. Неуправляемые переключатели не имеют входа управляющего кода и осуществляют фиксированное соединение первого входа с первым выходом, второго входа со вторым выходом. На первом уровне один переключатель T_{31} неуправляемый. На втором уровне неуправляемых переключателей два — T_{42}, T_{52} , на третьем — четыре — T_{13}, T_{33}, T_{53} и T_{73} .

Переключатели составляют входную часть матрицы с $n/2$ линиями и $k-1$ уровнями F_1, \dots, F_{k-1} и выходную часть матрицы с $n/2$ — линиями и $k-u$ уровнями G_1, \dots, G_{k-u} . Каждый переключатель T_{im} уровня $m = \overline{1, k-2}$ входной части матрицы своими выходами соединен с входами данных переключателей $T_{h, m+1}, T_{p, m+1}$ уровня $m+1$. Каждый переключатель V_{im} уровня $m = \overline{1, k-u-1}$ выходной части матрицы своими выходами соединен с входами данных переключателей $V_{h, m+1}, V_{p, m+1}$

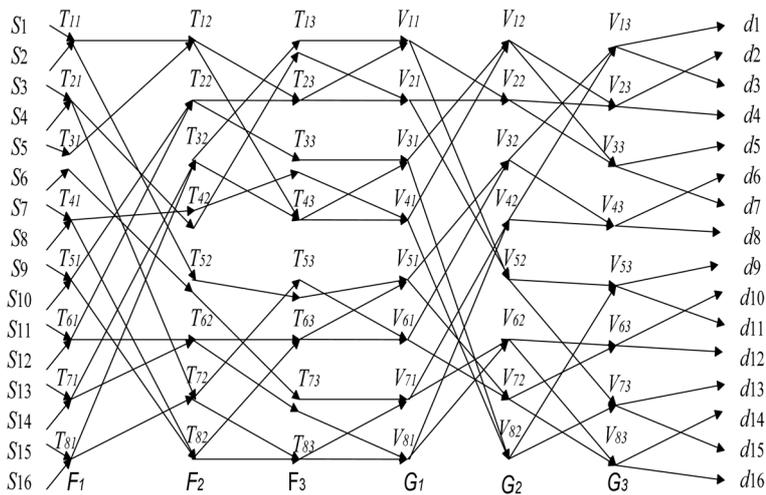


Рис. 2. Диаграмма орграфа одного из вариантов коммутационной матрицы формирователя разбиений для случая $n = 16$, $u = 1$

уровня $m + 1$, причем на соединения накладываются следующие ограничения:

$$2^{k-m-1} \text{int} \left(\frac{i-1}{2^{k-m-1}} \right) + 1 \leq h \leq 2^{k-m-1} \text{int} \left(\frac{i-1}{2^{k-m-1}} + 1 \right);$$

$$2^{k-m-1} \text{int} \left(\frac{(i + 2^{k-m-1} - 1) \bmod n}{2^{k-m-1}} \right) + 1 \leq p \leq 2^{k-m-1} \left(\text{int} \left(\frac{(i + 2^{k-m-1} - 1) \bmod n}{2^{k-m-1}} \right) + 1 \right),$$

где int — функция выделения целой части; $(i + 2^{k-m-1}) \bmod n$ — операция вычисления остатка от частного $\frac{i + 2^{k-m-1} - 1}{n}$.

Переключатели входной части матрицы соединены так, что на уровне F_1 входное множество S исходных данных разбивается на два подмножества S_{11} и S_{21} , причем $|S_{11}| = |S_{21}|$. На следующем уровне F_2 каждое из множеств S_{11} и S_{21} разбивается на два подмножества S_{12} , S_{22} и S_{32} , S_{42} , причем $|S_{12}| = |S_{22}| = |S_{32}| = |S_{42}|$. Для обеспечения работы алгоритма настройки матрицы, приведенного выше, каждый переключатель уровня $q < k$ выходной части матрицы может быть соединен через промежуточные переключатели с переключателями *только одного из множеств* $S_{i, k-q}$ уровня $k - q$ входной части матрицы. На уровне G_1 входные данные выходной части матрицы разбиваются на два подмножества

D_{11} и D_{21} , причем $|D_{11}| = |D_{21}|$, и любой элемент множества D_{11} больше каждого из элементов множества D_{21} . На следующем уровне G_2 каждое из множеств D_{11} и D_{21} разбивается на два подмножества D_{12} , D_{22} и D_{32} , D_{42} . Причем $|D_{12}| = |D_{22}| = |D_{32}| = |D_{42}|$, и элементы множества D_{12} больше элементов множества D_{22} , элементы множества D_{22} больше элементов множества D_{32} , элементы множества D_{32} больше элементов множества D_{42} .

Заключение

В статье предложена модель и алгоритм работы устройства, осуществляющего заданное разбиение входных данных на блоки равной мощности 2^u . Операция выполняется параллельно и может быть осуществлена за один такт внешнего генератора тактовых импульсов. Размер строки входных данных $n = 2^k$.

Число переключателей матрицы составляет $n(\log_2(n) - u/2 - 1) + 1$ и растет практически линейно с ростом n , что делает технически возможной реализацию функционального формирователя разбиений при больших значениях n . Если использовать для реализации кросс-кластерной перестановки коммутационную матрицу "butterfly" [3] или "omega-flip" [5], потребуется на $n(u + 1)/2 - 1$ переключателей больше. Соответственно для управления матрицей потребуется код на $n(u + 1)/2 - 1$ бит длиннее.

Время задержки преобразования $\tau(2 \log_2 n - u - 1)$, где τ — задержка на одном переключателе. Если использовать для реализации разбиения коммутационную матрицу "butterfly" [4] или "omega-flip" [5], время задержки преобразования будет на τu больше.

Число различных разбиений исходного множества, осуществляемых данным устройством, составляет

$$\frac{n!}{((n/2^u)!)^u}.$$

Если входы и выходы коммутационной матрицы, а также входы и выходы всех переключателей поменять местами, то с использованием матрицы можно формировать все перестановки мультимножества, содержащего 2^{k-u} наборов из 2^u одинаковых элементов.

В предложенной модели коммутационной матрицы соединения между ее уровнями заданы не жестко, что дает возможность их изменения при оптимизации топологии матрицы.

Список литературы

1. Курейчик В. М., Глушань В. М., Щербаков Л. И. Комбинаторные аппаратные модели и алгоритмы в САПР // Радио и связь, 1990. 216 с.
2. Benes V. Mathematical Theory of Connecting Networks and Telephone Traffic. New York Academic, 1965.
3. Патент US № 6,922,472, МПК 380/37 Method and system for performing permutations using permutation instructions based on butterfly networks/ Lee; Ruby B. (Princeton, NJ), Yang; Xiao (Princeton, NJ), Vachharajani; Manish (Hamilton, NJ). July 26, 2005 Appl. No.: 09/850, 237.

4. Lee C. Yi., Oruc A. Y. A Fast Parallel Algorithm for Routing Unicast Assignments in Benes Networks // IEEE Transactions On Parallel And Distributed Systems. March 1995. V. 6. No. 3. P. 329—334.
5. Патент US № 6,952,478, МПК 380/37 Method and system for performing permutations using permutation instructions based on modified omega and flip stages/ Lee; Ruby B. (Princeton, NJ), Yang; Xiao (Princeton, NJ). — October 4, 2005 Appl. No.: 09/850, 238.
6. Патент US № 5175539, МПК H04J 3/00 Interconnecting network/Richter; Harald. — December 29, 1992, Appl. No.: 07/617, 914.

УДК 004.057.4

А. И. Давыдов, аспирант,
В. Г. Шахов, канд. техн. наук, проф.,
И. Б. Ядрышников, аспирант,
Омский государственный университет
путей сообщения
e-mail: pr12yad@yandex.ru

Анализ протокола IPv6 в современных операционных системах

Основным недостатком протокола IPv4 и основным мотивом перехода на IPv6 оказалось ограниченное число IP-адресов, катастрофическая нехватка которых ощущается уже в наши дни. DHCP-серверы (выдающие динамические IP-адреса) и системы трансляции сетевых адресов (NAT) до некоторой степени смягчают проблему. Тем не менее, 32-битное поле, отведенное разработчиками протокола под IP-адрес в IPv4, обеспечивает 4 294 967 296 уникальных адресов, часть из которых зарезервирована под служебные нужды.

Ключевые слова: протокол, префикс, маска, безопасность, сессия, сервер, клиент

Протокол IPv4 в настоящее время столкнулся с рядом проблем, таких как масштабируемость сети, непригодность протокола к передаче мультисервисной информации с поддержкой различных классов обслуживания, включая обеспечение информационной безопасности. Указанные проблемы обусловили развитие классической версии протокола IPv4 в направлении разработки версии IPv6.

При этом к проблемам расширяемости протокола IPv4 следует отнести:

- недостаточность объема 32-битного адресного пространства;
- сложность агрегирования маршрутов и, как следствие, разрастание таблиц маршрутизации;
- сложность массового изменения IP-адресов;
- относительная сложность обработки заголовков пакетов в IPv4.

Для решения обозначенных проблем было разработано множество версий протокола IP для различных вычислительных платформ и операционных систем. До некоторого момента существовало несколько альтернативных вариантов протокола нового поколения: IPv7 (разработчик — Ullman), TUBA (Gallon), ENCAPS (Hinden), SIP (Deering) и PIP (Fracis). В результате развития линии ENCAPS (с промежуточной версией IPAE), SIP и PIP слились в 1993 г. в предложение SIPP, которое в июле 1994 г. было принято в качестве основы для создания протокола IP нового поколения, получившего название IPv6, где "6" обозначает номер версии протокола. Альтернативным названием протокола является IPng (IP next generation), хотя иногда под IPng понимают все варианты модернизации IP, включая также не вошедшие в проект IPv6, но продолжающие развиваться.

Документом, фиксирующим появление IPv6, является спецификация RFC 1752 "The Recommendation for the IP Next Generation Protocol". Базовый набор протоколов IPv6 был принят IETF в сентябре 1995 г. и получил статус Proposed Standard.

В IPv6 все изменения планировались таким образом, чтобы минимизировать изменения на других уровнях стека протоколов TCP/IP. В результате размер IP-адреса увеличен до 128 бит. Даже с учетом неэффективности использования адресного пространства, являющейся оборотной стороной эффективной маршрутизации и автоматического конфигурирования, этого достаточно, чтобы обеспечить объединение миллиарда сетей, как того требовали документы IETF. Обеспечена возможность простого и гибкого автоматического конфигурирования адресов для сетей произвольного масштаба и сложности. IPv6 остался масштабируемым протоколом, причем поля расширений (дополнительные заголовки) могут добавляться без снижения эффективности маршрутизации.

В большинстве узкоспециализированных серверных платформ, таких как корневые контролеры доменов сети Internet, стек IPv6 уже давно реализован и используется. Однако это

достаточно малая часть всей мировой компьютерной сетевой инфраструктуры, большую же часть составляют частично изолированные локальные вычислительные сети (ЛВС) с одним сетевым интерфейсом.

Рассмотрим возможность использования существующих реализаций протокола IPv6 в офисной сети.

Большинство офисных ЛВС представляют собой несколько коммутаторов, один или несколько серверов с операционной системой (ОС) семейства Windows Server и клиентскими станциями на базе Windows или Linux.

Из существующих ОС с официально заявленной поддержкой IPv6 наиболее распространены Windows Vista Business, Windows 2008 Server и UBUNTU 8.10.

Для определения возможности функционирования офисной сети на основе протокола IPv6 авторами был проведен эксперимент:

- на операционной системе Windows 2008 Server настроены инфраструктура Active Directory, DNS- и DHCP-серверы;
- на клиентских ПК установлены ОС UBUNTU 8.10 Desktop edition и Windows Vista Business.

Целью эксперимента была организация взаимодействия в сети с помощью протокола IPv6.

Для анализа сетевого взаимодействия между клиентской и серверной операционными системами был использован сетевой анализатор Wireshark версии 1.0.5.

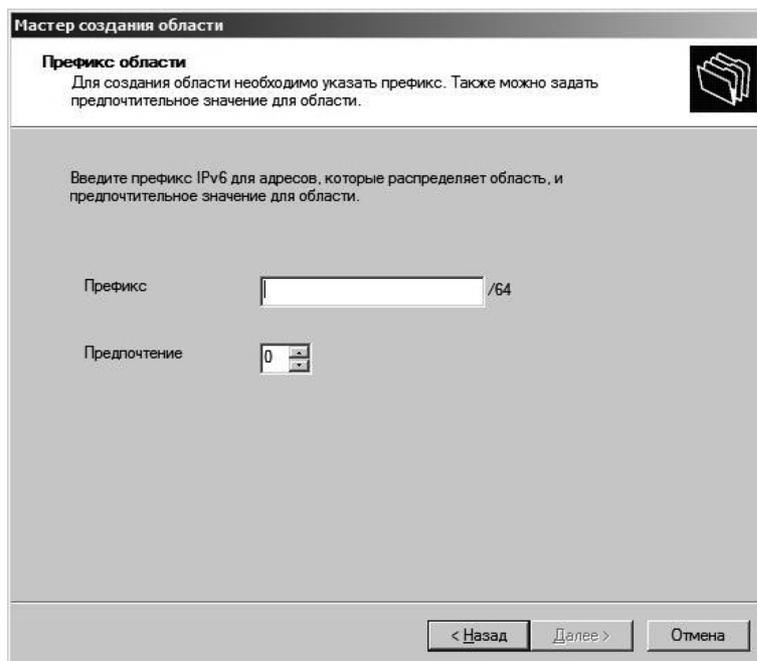
На сервере были заданы статические IPv4- и IPv6-адреса. При конфигурировании DHCP-сервера были использованы настройки по умолчанию.

При создании пула адресов для DHCP-сервера обращает на себя внимание отсутствие возможности задавать префикс маски сети, который равен 64, что представлено на рисунке.

Можно выделить $\frac{64}{16} = 4$ октета под адрес сети и 4 октета под адрес хоста. Этот факт необходимо учитывать, так как существует вероятность, что клиент и сервер могут оказаться в разных IP-сетях.

При конфигурировании сети в ОС UBUNTU 8.10 было исключено использование "двойного стека", для поддержки DHCP установлен пакет Wide-dhcpv6-client, в Windows Vista Business отключен IPv4.

Вначале проанализируем способы согласования IPv6-адресов в рамках созданного клиент-серверного взаимодействия.



Префикс области

Для ОС UBUNTU основные этапы согласования адресов состоят из:

- первый пакет: UBUNTU сообщает имеющимся активным сетевым устройствам о появлении новой мультикастовой группы;
- второй пакет: UBUNTU на основе MAC-адреса генерирует IPv6-адрес и пытается определить, есть ли такой адрес в сети; для этого посылается мультикастовый запрос всем узлам с адресом хоста "2dlc", если в течение одной секунды ответа не последовало, тогда заявленный адрес присваивается интерфейсу;
- третий пакет: запрашивает адреса маршрутизаторов;
- четвертый пакет: по протоколу DHCPv6 запрашивает IPv6-адреса всех имеющихся DNS-серверов;
- пятый пакет: DNS-сервер запрашивает MAC-адрес компьютера клиента;
- шестой пакет: UBUNTU сообщает свой MAC-адрес;
- седьмой пакет: DNS-сервер передает свой IPv6-адрес.

Отсутствие операции по сопоставлению IP- и MAC-адресов связано с тем, что из IPv6 протокол ARP был полностью исключен. Ему на смену пришел ICMPv6 Neighbor Solicitation (работает на запрос) и Neighbor Advertisement (работает на ответ).

Для ОС Windows Vista Business согласование IPv6-адресов осуществляется аналогичным образом, за исключением того, что в этой ОС отсутствует поддержка DHCPv6, поэтому IPv6-адреса

всех имеющихся DNS-серверов задаются. Каждый из пакетов сессии отправляется два раза.

Алгоритм работы протокола IPv6 при копировании файлов с сервера и на сервер, добавлении клиентских компьютеров в домен не отличается от аналогичной операции в IPv4.

После проведенных экспериментов можно сделать ряд выводов:

- способ генерации IPv6-адреса на основе MAC не универсален и плохо адаптируем. В ЛВС возможна только одна логическая сеть вида x:x:x:x:x:x/64, в которой согласуется только последний октет. В частности, для работоспособности созданной для проведения экспериментов IPv6-сети необходимо, чтобы IPv6-адрес сервера имел вид: FE80::X:X:X:X, где FE80 — префикс локальной связи (по функциональности аналогичен сети 192.168.0.0 в IPv4);
- в сети, использующей протокол IPv6, возможна простая DoS-атака. Суть заключается в том, что по стандарту RFC 2462 узел не может использовать адрес, если он уже используется в сети. Таким образом, компьютер злоумышленника может просто отвечать на все запросы подключающегося к сети компьютера о том, что все возможные адреса сети в данный момент заняты;
- существует потенциальная опасность протокола ICMPv6. Одна из задач, решаемых этим протоколом, — это обмен маршрутной информа-

цией с серверами и другими компьютерами сети. Поскольку аутентификация сообщений NDP обычно не используется, злоумышленник имеет возможность подделывать сообщения для выполнения ряда атак. Например:

а) внедрение ложных шлюзов для клиентов (аналог ложного DHCP-сервера) и маршрутизаторов путем отправки ответов на запросы клиентского компьютера;

б) реализация DoS атак путем снижения MTU в сети (минимум 1260), внедрения ложного маршрута, изменения маски сети, снижения ограничения на число ретрансляций пакета.

Таким образом, использование IPv6 в офисных ЛВС не целесообразно. Большое число ошибок в стандартах и их программных реализациях создает необходимость проведения дополнительных работ по обеспечению безопасности сети.

Конечно, отказ от использования IPv6 — слишком радикальная мера защиты, но на данный момент она единственная. Все остальные решения ненадежны.

Список литературы

1. Моримото Р., Ноэл М., Драуби О. Microsoft Windows Server 2008. Полное руководство. М.: Вильямс, 2008. 1392 с.
2. Мэлоун Д., Мэрфи Н. IPv6: администрирование сетей. М.: КУДИЦ-ПРЕСС, 2006. 320 с.
3. Уолтон Ш. Создание сетевых приложений в среде Linux: Руководство разработчика. М.: Вильямс, 2004. 464 с.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

УДК 004.5, 004.8

В. В. Грибова, д-р техн. наук, зав. лаб., e-mail: gribova@iacp.dvo.ru,

Н. Н. Черкезишвили, аспирант, Институт автоматизации и процессов управления ДВО РАН, г. Владивосток

Развитие онтологического подхода для автоматизации разработки пользовательских интерфейсов с динамическими данными¹

Предлагается концепция автоматизации разработки пользовательских интерфейсов с динамическими данными, которая является дальнейшим развитием онтологического подхода. Интерфейсы с динамическими данными допускают формирование наборов входных/выходных данных динамически во время работы приложения, тогда как для интерфейсов со статическими данными требуется точное определение наборов входных/выходных данных во время их разработки. Интерфейсы с динамическими данными дают более высокую степень гибкости по сравнению с интерфейсами со статическими данными, что представляет принципиально новые возможности их использования и расширяет круг применения таких интерфейсов. Однако современные средства автоматизации разработки интерфейсов поддерживают проектирование и автоматическую генерацию только интерфейсов со статическими данными. Предлагаемая в работе концепция направлена на устранение их недостатков.

Ключевые слова: модель интерфейса, модель задач, автоматическая генерация интерфейса, онтологии

¹ Работа выполнена при финансовой поддержке ДВО РАН в рамках Программы фундаментальных исследований ОЭММПУ РАН, проект "Модели мультиагентных систем для управления распределенной обработкой информации".

Введение

Существующие интерфейсы можно разделить на интерфейсы со статическими данными и интерфейсы с динамическими данными. Интерфейсы со статическими данными характеризуются фиксированными наборами входных/выходных данных, для которых структура каждого набора, а также сценарий диалога известны на этапе проектирования пользовательского интерфейса. Для интерфейсов с динамическими данными наборы входных/выходных данных, структуру каждого набора, а также сценарий диалога невозможно определить на этапе его проектирования. Примерами интерфейсов с динамическими данными являются различные редакторы, программы, в которых наборы входных/выходных данных генерируются логикой приложения, а также гибко конфигурируемые приложения, в которых пользователь может определять собственные структуры данных.

Для автоматизации разработки пользовательских интерфейсов в настоящее время используют построители WIMP-интерфейсов, моделиориентированные средства и средства, основанные на онтологическом подходе [1].

Однако все перечисленные средства ориентированы на автоматизацию разработки пользовательских интерфейсов со статическими данными; автоматизация разработки интерфейсов с динамическими данными не поддерживается. Такие интерфейсы либо полностью реализованы на языках программирования (C++, Java, Pascal и др.), либо некоторые компоненты интерфейса выполнены с использованием специализированных средств автоматизации разработки, остальные — на языках программирования. В результате проектирование, реализация и особенно сопровождение пользовательских интерфейсов оказываются чрезвычайно трудоемкими.

Цель данной работы — представление концепции автоматизации разработки интерфейсов с динамическими данными.

1. Онтологический подход для автоматической генерации пользовательских интерфейсов

Предлагаемая в работе концепция автоматизации разработки интерфейсов с динамическими данными является развитием онтологического подхода [2, 3]. Инструментарий, основанный на данном подходе, широко используется для автоматической генерации пользовательских интерфейсов; с его помощью разработаны пользовательские интерфейсы для приложений в различных предметных областях.

Основными положениями автоматизации проектирования, реализации и сопровождения поль-

зовательского интерфейса на основе онтологического подхода являются:

- *анализ профессиональной деятельности, связанной с разработкой пользовательского интерфейса*: выделение групп специалистов, осуществляющих разработку и сопровождение пользовательского интерфейса, а также систем понятий, которые они используют в своей работе;
- *построение онтологий пользовательского интерфейса*, необходимых для того, чтобы в их терминах разработчики интерфейса могли определять и модифицировать структуру конкретной модели пользовательского интерфейса;
- *разработка модели пользовательского интерфейса в терминах онтологий*, которая является конкретизацией онтологии пользовательского интерфейса;
- *построение алгоритма автоматического преобразования модели интерфейса в программный код*, который управляется онтологиями пользовательского интерфейса, при этом характеристики конкретной модели являются входными данными для этого алгоритма.

Таким образом, в онтологическом подходе модель интерфейса является основной информационной составляющей, на основе которой автоматически генерируется код интерфейса. Модель интерфейса состоит из нескольких компонентов, описывающих всю необходимую информацию о пользовательском интерфейсе, необходимую для его генерации (рис. 1).

Компоненты модели интерфейса создают разработчики интерфейса (эксперты предметной области, дизайнеры, программисты) с помощью редакторов, управляемых соответствующими онтологиями (модель задач формируется с помощью редактора, управляемого онтологией задач, модель системы понятий диалога формируется с помощью редактора, управляемого онтологией системы понятий диалога и т. д.). В результате разработчики интерфейса освобождаются от изучения специализированных языков спецификаций и программирования и в терминах понятных им систем понятий формируют соответствующие компоненты модели интерфейса. После формирования модели интерфейса генератор кода пользовательского интерфейса автоматически генерирует код интерфейса, в который интегрируется код логики приложения, затем полученный код компилируется в приложение, с которым работает пользователь.

2. Основные положения концепции

Концепция разработки интерфейсов с динамическими данными расширяет модель интерфейса онтологического подхода новыми компонентами

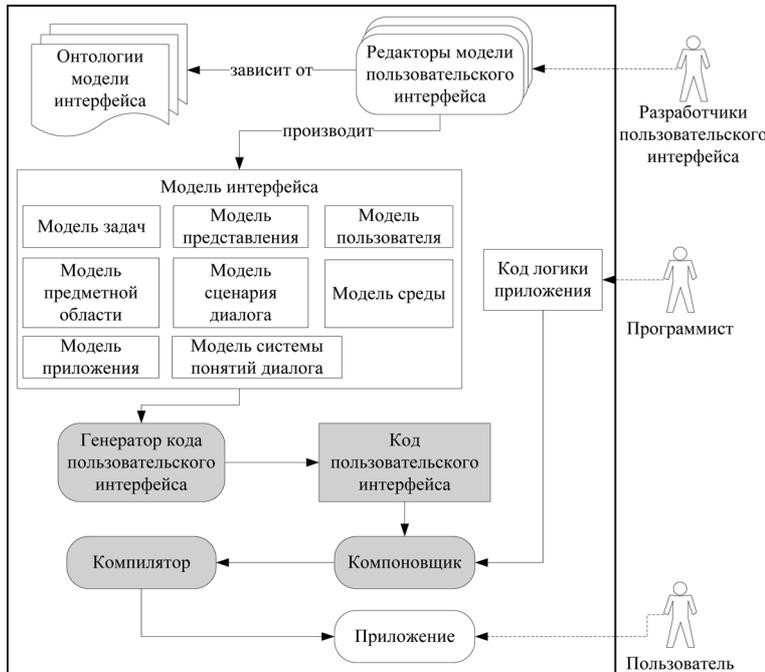


Рис. 1. Концептуальная схема онтологического подхода

и, соответственно, онтологиями, в терминах которых новые компоненты будут формироваться и модифицироваться [4]. Для описания модели интерфейса с динамическими данными были выделены статические характеристики модели интерфейса, которые не изменяются в процессе работы приложения, и динамические характеристики модели интерфейса, определяющие структуру возможных изменений, но конкретные их значения определяются в процессе работы приложения.

Анализ динамических характеристик интерфейса привел к следующим изменениям:

I. Онтология представления пользовательского интерфейса заменена онтологиями абстрактного и конкретного представления. Онтология представления онтологического подхода предназначена для формирования модели представления в терминах WIMP-интерфейсов (кнопки, поля ввода, списки, чекбоксы и т. д.) [5, 6]. Однако в силу специфики приложений с динамическими данными, для таких приложений невозможно определить их конкретное представление во время проектирования.

Онтология абстрактного представления позволяет описывать представление интерфейса в терминах абстрактного пользовательского интерфейса [7], независимого от среды исполнения приложения и типов входных/выходных данных интерфейса. Элементы абстрактного интерфейса позволяют определять структуру представления информации, не уточняя их конкретное представление.

Онтология конкретного представления предназначена для формирования конечного представления пользовательского интерфейса, описанного в терминах WIMP-интерфейсов. Модель конкретного представления автоматически формируется во время работы приложения по модели абстрактного представления, соответствиям между: онтологиями пользователя и конкретного представления, онтологиями абстрактного и конкретного представлений, а также по конкретным характеристикам входных/выходных данных, например, число элементов ввода влияет на выбор элемента представления (рис. 2).

II. Онтология задач пользователя (статическая) и сценарий диалога заменены онтологией динамических задач. Онтология задач пользователя (статическая) онтологического подхода предназначена для описания иерархии задач пользователя и связей между задачами, определяющих порядок и условия их выполнения: объединение (обе задачи в связке могут выполняться параллельно); выбор (в один момент времени может выполняться только одна задача связки); допуск (устанавливает строгую последовательность выполнения задач); деактивация (выполнение одной задачи делает другую задачу недоступной для выполнения).

Онтология сценария диалога напрямую связана с онтологией представления и предназначена для описания множества возможных состояний диалога и переходов из состояния в состояние. Состояния диалога и условия переходов описываются в терминах интерфейсных элементов модели представления.

В динамических приложениях на этапе проектирования невозможно задать интерфейсные элементы и, соответственно, сценарий диалога. Введение в онтологию задач динамических характеристик и правил выполнения задач позволяет задавать сценарии выполнения задач, заместив тем самым модель сценария диалога.

К динамическим характеристикам относятся переменные задачи и атрибуты задач.

1. Переменные задачи определяют динамический контекст задачи, формируемый в процессе ее выполнения. Они могут хранить промежуточ-



Рис. 2. Пример соотношений между онтологиями

ные данные выполнения алгоритмов и влиять на сценарии выполнения задач. Выделены следующие типы переменных: логические, целочисленные, строковые, действительные и перечислимые. Перечислимые переменные определяются разработчиками модели задач и могут содержать множество допустимых значений для каждого типа перечисления, например, перечисление "дни недели" содержит множество значений {понедельник, вторник, среда, ...}. Перечислимые переменные могут принимать одно из множества значений, относящихся к этому типу.

2. Для каждой задачи определены два типа атрибутов: "состояние" и "статус задачи". Атрибут "состояние" может принимать одно из следующих значений: задача выполнена; задача не выполнена; задача прервана. Атрибут "статус" может принимать одно из следующих значений: задача доступна; задача недоступна; задача в процессе выполнения.

Правила выполнения задач представляют собой пару вида: < условие, действие >. Условие является логическим выражением вида "если (условие 1) И | ИЛИ ... (условие N)". Действие представляет собой набор инструкций, которые выполняются только в том случае, если условие истинно. В условиях правил выполнения задач указываются значения атрибутов задач и переменных, в зависимости от которых выполняются действия: изменение значений атрибутов задач и переменных, вызовы системных функций интерфейса [8] либо функций логики приложения.

Онтология модель задач (статическая) также расширена рядом дополнительных возможностей:

а) добавлен новый тип задач — "задача приложения"; задачи приложения в отличие от задач пользователя не связаны с пользовательским интерфейсом, а связаны с логикой приложения и предназначены для автоматизации важных для интерфейса функций управления наборами входных/выходных данных задач (сохранение, редактирование, просмотр, удаление данных и др.);

б) к атрибутам задач каждого типа добавлен атрибут "входные/выходные данные", необходимый для установки связи между элементами модели системы понятий диалога и логикой приложения.

III. Онтология пользователя расширена онтологией ролей пользователей. Онтология пользователя онтологического подхода позволяет описывать характеристики пользователя, такие как уровень опытности, пол, возраст, предпочтение ввода, профессия пользователя и другие. Эти характеристики составляют портрет пользователя, необходимый для генерации пользовательского интерфейса.

Средства онтологического подхода позволяют разрабатывать только однопользовательские при-

ложения, однако они не предназначены для генерации приложений, у которых разным типам пользователей доступна различная функциональность. В связи с этим онтология пользователя была расширена онтологией ролей пользователей, позволяющей задавать для каждого пользователя/группы пользователей допустимую функциональность.

Таким образом, полученная в результате преобразований модель интерфейса (каждый компонент модели интерфейса является результатом конкретизации соответствующей онтологии) состоит из следующих компонентов: модели связи с приложением; динамической модели задач, включающей модели задач пользователя и приложения, динамические характеристики задач; модели системы понятий диалога, описывающей предметную область приложения; модели среды, описывающей характеристики среды исполнения приложения; модели пользователя, состоящей из модели профиля пользователя и модели ролей; модели представления, состоящей из модели абстрактного представления и модели конкретного представления (рис. 3).

IV. Генератор кода заменен интерпретатором модели интерфейса. При использовании генератора кода интерфейса, прежде чем пользователь



Рис. 3. Модель пользовательского интерфейса с динамическими данными

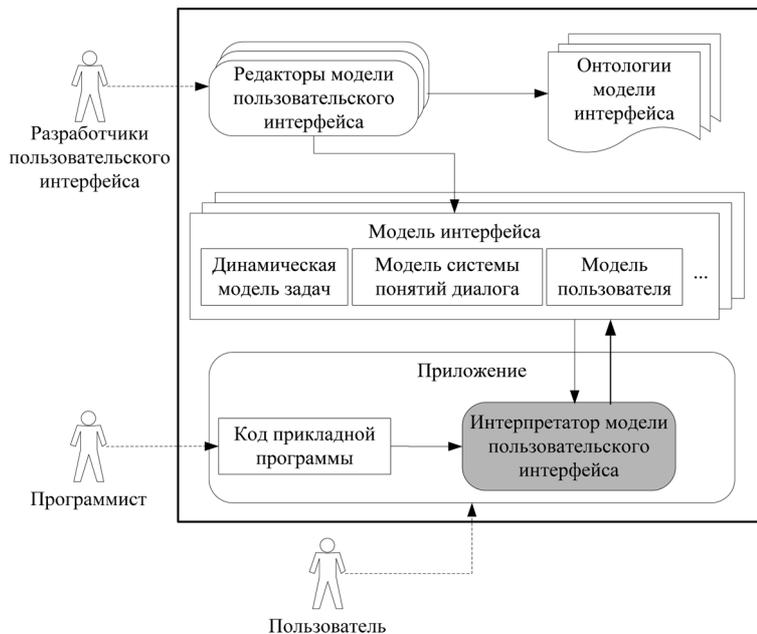


Рис. 4. Схема автоматической генерации интерфейсов с динамическими данными

получает исполнимое приложение, необходимо выполнить генерацию кода приложения, компоновку и компиляцию кода. Только после выполнения этих действий пользователь может начать работать с приложением. Таким образом, при каждом изменении модели интерфейса необходимо повторять эти действия заново, что неизбежно влечет за собой перезапуск приложения. Но в интерфейсах с динамическими данными изменения в модели интерфейса могут происходить очень часто в процессе работы пользователя с приложением. Каждое такое изменение будет требовать от пользователя повторного выполнения описанной выше последовательности действий, необходимых, чтобы изменения в модели интерфейса вступили в силу.

Интерпретатор в отличие от генератора кода не генерирует код интерфейса (рис. 4), а интерпретирует модель интерфейса сразу в исполняемое приложение. Это позволяет мгновенно отражать

изменения в пользовательском интерфейсе при изменении модели интерфейса, не требуя перезапуска приложения.

Заключение

Приложения с динамическими данными в настоящее время являются очень распространенными. Однако современные средства автоматической генерации пользовательских интерфейсов не позволяют автоматизировать их разработку. Предлагаемая в работе концепция является расширением онтологического подхода к разработке и автоматической генерации пользовательских интерфейсов и предназначена для автоматизации разработки интерфейсов с динамическими данными. В настоящее время начата разработка инструментария, основанного на предложенной концепции.

Список литературы

1. Грибова В. В., Клешев А. С. Методы и средства разработки пользовательского интерфейса: современное состояние // Программные продукты и системы. 2001. № 1. С. 2—6.
2. Грибова В. В., Клешев А. С. Использование методов искусственного интеллекта для проектирования пользовательского интерфейса // Информационные технологии. 2005. № 8. С. 58—62.
3. Грибова В. В., Клешев А. С. Управление проектированием и реализацией пользовательского интерфейса на основе онтологий // Проблемы управления. 2006. № 2. С. 58—62.
4. Грибова В. В., Черкезишвили Н. Н. Концепция автоматизации проектирования и реализации пользовательских интерфейсов с динамическими данными // Материалы рабочего семинара "Научное программное обеспечение" в рамках 7-й международной конференции памяти академика А. П. Ершова "Перспективы систем информатики". — Новосибирск: Сибирское научное издательство, 2009. С. 97—103.
5. Гультяев А. К., Машин В. А. Проектирование и дизайн пользовательского интерфейса. СПб.: КОРОНА принт. 2000. 352 с.
6. Мандел Т. Дизайн интерфейсов: пер. с англ. — М.: ДМК Пресс, 2005. 416 с.
7. Trættemberg H. Model-based User Interface Design // Information Systems Group Department of Computer and Information Sciences. 2002. 211 с.
8. Грибова В. В., Тарасов А. В. Гибкие инструментальные средства для разработки пользовательского интерфейса // Известия вузов. Приборостроение. 2007. Т. 50, № 3. С. 35—38.

А. С. Зуев, канд. техн. наук, доц.,
Ю. И. Петров, студент,
Московский государственный университет
приборостроения и информатики
e-mail: zuev_andrey@mail.ru

Модернизация контекстных меню браузера, дерева папок и рабочего стола операционной системы Windows*

Представлено описание модернизированного контекстного меню браузера, дерева папок и рабочего стола операционной системы Windows, приведены результаты сравнения с аналогичными элементами интерфейса, обоснованы эргономичность и эффективность применения данной разработки.

Ключевые слова: контекстное меню, дерево папок, браузер, рабочий стол, человеко-компьютерное взаимодействие, графический интерфейс, эргономика программного обеспечения, проектирование графических интерфейсов, оптимизация графических интерфейсов, окно операционной системы

Введение

Одним из важнейших этапов разработки современного программного обеспечения (ПО) является проектирование качественного графического пользовательского интерфейса. Понятие качества в данном контексте является многоаспектным и, например, в соответствии со стандартом ISO 9126:2001 "Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению" может трактоваться как удобство использования ПО, определяемое следующими параметрами:

- соответствие стандартам удобства использования (*usability compliance*);
- удобство обучения (*learnability*);
- привлекательность (*attractiveness*);
- удобство работы (*operability*);
- понятность (*understandability*).

Программный продукт, не только реализующий требующиеся функциональные возможности, но и обеспечивающий удобство их использования оператором, обладает высокой конкурентоспособностью на рынке программного обеспечения. Вместе с тем, помимо задачи проектирования ка-

* Работа выполнена при финансовой поддержке грантов Президента РФ молодым российским ученым кандидатам наук (грант № МК-948.2008.9).

чественного интерфейса может рассматриваться задача совершенствования (модернизации) реализованных в нем технических решений [1].

В статье описана модель Containers, представляющая результаты модернизации контекстных меню браузера, дерева папок и рабочего стола операционной системы (ОС) Windows. В модели реализованы дополнительные функциональные возможности, сокращающие сложность навигации пользователя по файловой системе и доступные к внедрению в современные ОС.

Описание функциональных возможностей модели

Интерфейс модели Containers представлен на рис. 1 (см. вторую сторону обложки). Ключевым элементом, реализующим основные функциональные возможности, является модернизированное контекстное меню, вызываемое нажатием правой клавиши мыши на каталоге, выбранном в браузере, дереве папок или на рабочем столе ОС.

Заголовок окна, панель навигации, адресная панель, панель папок, файловый браузер и строка состояния являются стандартными элементами интерфейса окон ОС Windows. Модернизированное контекстное меню является ключевым элементом интерфейса, новшеством, преимуществом и отличительной особенностью модели Containers. Помимо типовых пунктов данное меню содержит три "контейнера": *Папки*, *Файлы* и *Ярлыки*. Данные пункты отображаются в меню, если в выбранном каталоге присутствуют соответствующие вложенные объекты. При вызове контекстного меню для пустой папки его вид соответствует типовому контекстному меню операционной системы. В модели предусмотрен режим эмуляции рабочего стола, позволяющий оценить удобства использования модернизированного контекстного меню при работе с каталогами на рабочем столе ОС. Включение данного режима выполняется нажатием кнопки "*Режим рабочего стола*" на *Панели навигации*.

Проиллюстрируем возможности, предоставляемые "контейнерами":

- отображение списков соответствующих объектов, вложенных в выбранный каталог, с помощью контекстных иерархических меню (рис. 2 и 3);
- быстрый переход по иерархии вложенных папок, а также между вложенными папками, файлами и ярлыками с минимальными затратами времени и минимальным числом манипуляций с мышью и клавиатурой;
- открытие файлов и запуск программ с помощью ярлыков, отображенных в контекстных меню соответствующих "контейнеров".

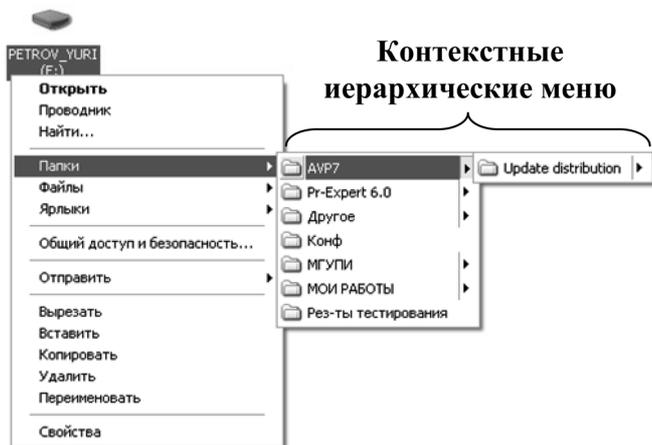


Рис. 2. Пример контекстного меню папок

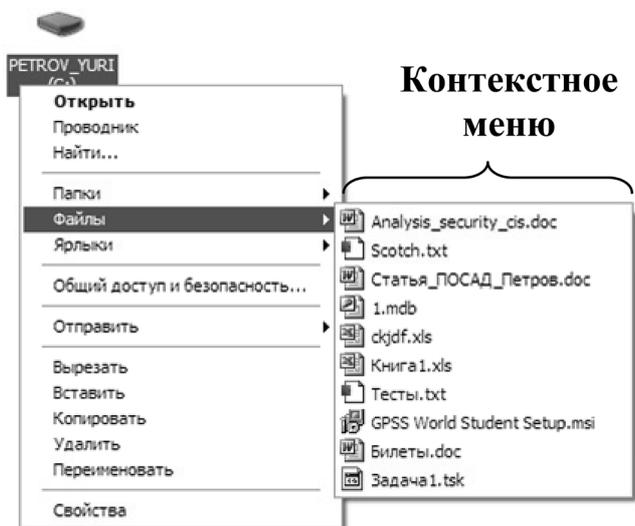


Рис. 3. Пример контекстного меню файлов

На рис. 4 представлен пример модернизированного контекстного меню, соответствующего каталогу, выбранному в дереве папок.

В результате использования модернизированного контекстного меню доступ к содержанию каталогов может быть осуществлен без непосредственного их открытия в окне ОС, что повышает скорость навигации по вложенным объектам — папкам, файлам и ярлыкам.

Работа с модернизированным контекстным меню может заключаться в просмотре списков объектов, вложенных в выбранный каталог, в навигации по вложенным папкам, а также в открытии файлов и запуске программ с помощью ярлыков. Принципы работы с контекстными меню каталогов, выбранных в браузере, дереве папок и на рабочем столе ОС (в режиме его эмуляции) аналогичны. Функциональность модели ориентирована на повышение эргономичности работы с клавиатурой и мышью, в том числе в случае комбинирования данных средств манипулирования.

- Для работы с помощью клавиатуры требуется:
- установить фокус на файловый браузер или дерево папок нажатиями клавиши *Tab*;
 - нажатиями клавиш-стрелок *Вниз*, *Вверх*, *Вправо*, *Влево* выбрать каталог и нажать на клавиатуре кнопку вызова контекстного меню;
 - в открывшемся меню с помощью клавиш-стрелок *Вниз* и *Вверх* выбрать требующийся "контейнер" из доступных;
 - с помощью клавиш-стрелок *Вниз*, *Вверх*, *Вправо*, *Влево* выполнить переходы между объектами, отображаемыми в последующих контекстных меню, и выбрать требующийся объект нажатием клавиши *Enter*.

Для работы с помощью мыши требуется:

- выбрать каталог в браузере, в дереве папок или на рабочем столе ОС (в режиме его эмуляции) нажатием правой клавиши мыши;
- в открывшемся меню наведением курсора выбрать требующийся "контейнер" из доступных и аналогично выполнить переходы между объектами, отображаемыми в последующих контекстных меню;
- выбрать требующийся объект нажатием левой клавиши мыши.

Режим эмуляции рабочего стола предоставляет возможность оценить удобства использования модернизированного контекстного меню при работе с каталогами, расположенными на рабочем столе ОС. В данном режиме интерфейс модели Containers принимает вид, представленный на рис. 5 (см. вторую сторону обложки), при этом фоновый рисунок соответствует рисунку, установленному на рабочем столе ОС, а в браузере отображается содержание текущего открытого каталога.

Настройки программы

Настройки программы представляют собой опции дополнительного управления ее функцио-

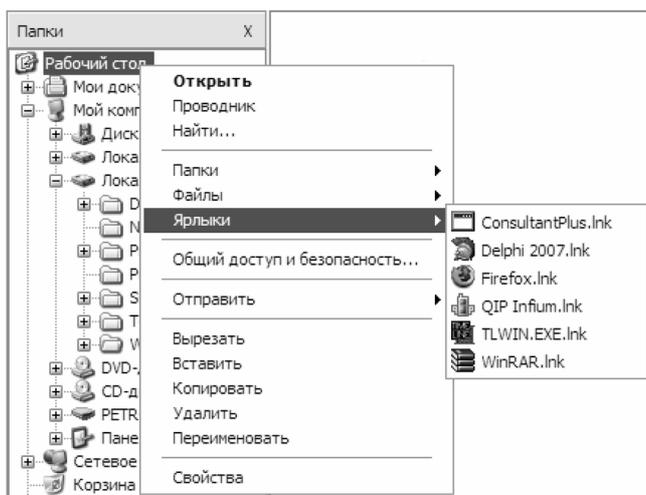


Рис. 4. Пример контекстного меню ярлыков для каталога в дереве папок

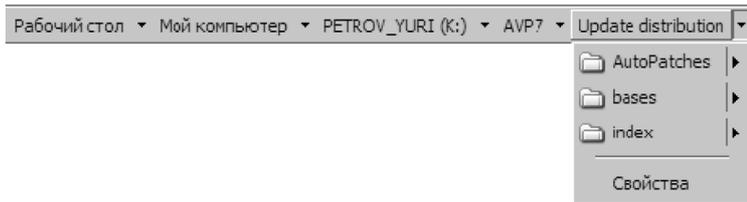


Рис. 6. Адресная панель UltraExplorer

нальными возможностями, повышающие удобство эксплуатации и позволяющие пользователю реализовать свои предпочтения.

Вызвать окно настроек модели можно одним из следующих способов:

- выбором пункта главного меню *Инструменты* → → *Настройки...*;
- нажатием правой клавиши мыши на метке *Адрес* адресной панели;
- нажатием горячей клавиши *F10*.

Настройки модели Containers подразделяются на общие, связанные с деревом папок и контекстное меню.

Общие настройки:

- *Число позиций истории браузера* (от 3 до 30) определяет число возможных переходов к предыдущим местоположениям в файловой системе компьютера в текущем сеансе работы программы.
- *Сделать доступным режим рабочего стола* — обеспечивает возможность включения режима эмуляции рабочего стола и отображение соответствующей кнопки на панели навигации.
- *Шрифт* надписей на элементах интерфейса в окне приложения Containers может быть оформлен различными стилями и кеглем.

Настройки дерева папок:

- *Разрешить изменение размеров* — обеспечивает возможность масштабирования области отображения дерева папок.
- *Выделять выбранную в "контейнере" папку* — обеспечивает автоматическое выделение в дереве папок каталога, выбранного в браузере с помощью контекстных иерархических меню. Данная опция функционирует, если в настройках контекстного меню задан параметр "*Открывать в ... Containers*".

Настройки контекстного меню позволяют задать действие, выполняемое автоматически при выборе объектов в контекстных меню "контейнеров":

- *Открывать в...* — определяет приложение, в браузере которого будет открыта папка, содержащая выбранный объект: Containers или окно Windows.
- *Выделять выбранный объект* — обеспечивает автоматическое выделение выбранного объекта в браузере указанного выше приложения.
- *Запускать файлы и ярлыки* — обеспечивает автоматическое открытие файлов и запуск про-

грамм с помощью ярлыков, выбранных в контекстных меню соответствующих "контейнеров".

- *Отображать скрытые и системные объекты* — обеспечивает отображение в контекстных меню "контейнеров" *Папки*, *Файлы* и *Ярлыки* соответствующих скрытых и системных объектов.
- *Упорядочить пункты контекстных меню "контейнеров"* — обеспечивает варианты автоматического упорядочивания имен объектов в списках контекстных меню "контейнеров". По умолчанию объекты отображаются в соответствии с алфавитным порядком имен, возможны также следующие варианты их упорядочивания по возрастанию или убыванию:
 - *Папки* — по размеру или дате создания;
 - *Файлы* — по типу, размеру, датам создания, изменения или открытия;
 - *Ярлыки* — по датам создания или запуска соответствующего приложения.

Сравнение с аналогами

Сравним модернизированное контекстное меню модели Containers с аналогичными элементами интерфейса приложения UltraExplorer и панели инструментов ОС Windows. Как показывает представленный далее материал, модель Containers обладает большей функциональностью, эффективностью и эргономичностью навигации по файловой системе компьютера.

UltraExplorer. Это бесплатный многофункциональный файловый менеджер, последняя версия которого датируется 25.05.2009 г. Ключевой особенностью данного приложения является оригинальная адресная панель, содержащая контекстные иерархические меню, каждое следующее из которых отображает список папок, вложенных в папку, выбранную в предыдущем меню. Переход к папке в файловой системе компьютера организован посредством последовательного выбора имен папок, в которые она вложена. Пример рассматриваемого элемента интерфейса представлен на рис. 6.

Представленное техническое решение позволяет эффективно перемещаться по структуре вложенных папок, однако его применение не распространено на случаи работы с отдельными каталогами, отображаемыми в браузере, в дереве папок или на рабочем столе ОС. Также рассматриваемый элемент интерфейса не позволяет открывать файлы и запускать программы с помощью ярлыков.

Панели инструментов ОС Windows. В операционной системе Windows XP предусмотрена возможность доступа к объектам посредством контекстных списков панели инструментов "*Рабочий стол*", размещаемой на *панели задач*. По умолчанию рассматриваемый элемент интерфейса не дос-

тупен, его добавление может быть выполнено с помощью контекстного меню панели задач (рис. 7).

Пример панели "Рабочий стол" представлен на рис. 8.

Панель "Рабочий стол" обладает следующими недостатками:

- доступ к объектам реализован только с рабочего стола ОС, что увеличивает число требуемых переходов по вложенным папкам;
- в контекстных списках отображены имена вложенных папок, файлов и ярлыков, что затрудняет визуальный поиск требуемых объектов;
- контекстные списки доступны только при наличии панели "Рабочий стол" на панели задач Windows;
- реализованные функциональные возможности не доступны при работе с каталогами, отображаемыми в браузере, в дереве папок или на рабочем столе ОС.

Эффективность работы с модернизированным контекстным меню

В качестве параметра, характеризующего эффективность работы с графическим интерфейсом и реализованными в нем техническими решениями, могут рассматриваться суммарные затраты времени, требующиеся пользователю для выполнения конкретного набора задач.

В настоящее время навигация по файловым системам ОС основана на работе с окнами, отображающими содержание открываемых папок. Это обуславливает наличие ряда недостатков, исключенных при использовании модернизированного контекстного меню модели Containers (см. таблицу).

Сравнение принципов работы с окнами ОС и контекстным меню модели Containers

Работа с окнами ОС	Работа с меню модели Containers
Просмотр содержания выбранного каталога организован в окне ОС	Просмотр содержания выбранного каталога организован в контекстных списках
Необходим визуальный поиск требуемого объекта среди всех, представленных в окне	Необходим визуальный поиск требуемого объекта среди одноименных объектов в списке
Поиск вложенной папки связан с просмотром каталогов (с изменением содержания окна ОС)	Поиск вложенной папки связан с просмотром контекстных списков
Доступ к папке организован в результате последовательного отображения в окне ОС содержания каталогов, в которые она вложена	Доступ к папке организован в результате переходов по контекстным спискам с именами каталогов, в которые она вложена
Просмотр содержания каталога выполняется наведением курсора, нажатием клавиши Enter или двойным нажатием клавиши мыши	Просмотр содержания каталога выполняется наведением курсора на пункты контекстного меню ("контейнеры")
Содержание каталогов, в которые вложена папка, не доступно пользователю	Содержание каталогов, в которые вложена папка, отображено в контекстных списках
Поиск папки в дереве папок связан с просмотром его ветвей, что осложняет работу пользователя и рассеивает его внимание	Поиск папки в дереве папок связан с просмотром контекстных списков, что не рассеивает внимание пользователя
Каждый переход по структуре вложенных папок связан с обновлением информации, отображаемой в окне ОС и дереве папок	Каждый переход по структуре вложенных папок связан с обновлением содержания одного из отображенных контекстных списков
Область концентрации внимания пользователя — окно операционной системы	Область концентрации внимания пользователя — контекстные списки
Точки концентрации внимания пользователя расположены в браузере и на панели навигации окна ОС	Точки концентрации внимания пользователя расположены в контекстно связанных списках

Представленные в таблице результаты сравнения позволяют утверждать, что модернизированное контекстное меню обеспечивает возможность быстрого просмотра содержания каталогов, эффективную навигацию по структуре вложенных папок, меньшую сложность требуемых от пользователя манипуляций с мышью и клавиатурой, а также оперативность визуального поиска объектов, вложенных в выбранный каталог. При этом внимание пользователя сконцентрировано в небольшой области экрана, где представлены контекстно связанные списки, отображающие содержание просматриваемых папок, а точки концентрации внимания находятся на незначительном удалении друг от друга.

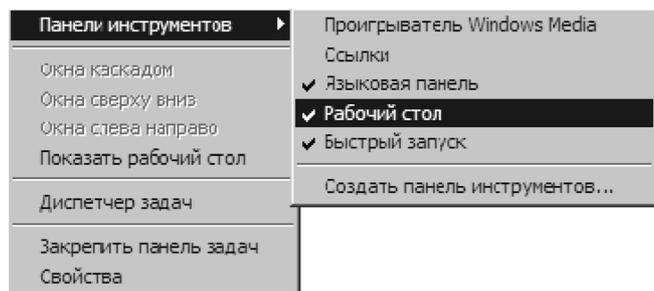


Рис. 7. Добавление панелей инструментов на панель задач ОС Windows XP

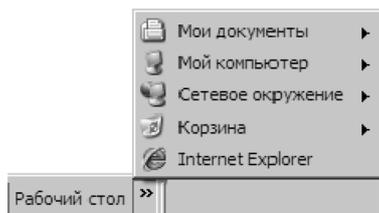


Рис. 8. Панель инструментов "Рабочий стол"

В результате разделения вложенных в каталог объектов по контекстным спискам папок, файлов и ярлыков сокращаются энтропия информации и затраты времени на визуальный поиск требуемого объекта. Сложность работы с контекстными списками ниже, чем с окном ОС, что обусловлено меньшими расстояниями требующихся от пользователя перемещений курсора и внимания.

Количественные оценки результатов сравнения эффективности работы с окном ОС и контекстным меню модели Containers зависят от конкретных условий — числа и состава объектов, вложенных в каждую из открываемых пользователем папок, — и поэтому могут существенно варьироваться. Вместе с тем, получить данные оценки можно с помощью представленных ниже формул, принципы использования которых рассмотрены в [2–4].

Время t (миллисекунды) визуального равновероятного выбора одного из N объектов позволяет сравнить эффективность использования окна ОС и контекстных меню при каждом переходе по структуре вложенных папок:

$$t = a + b \log_2(N + 1),$$

где a и b — коэффициенты, зависящие от навыков пользователя (для приближенных вычислений $a = 50$, $b = 150$).

Время t_v (миллисекунды) перемещения курсора к объекту позволяет оценить сложность выполняемых пользователем действий:

$$t_v = a + b \log_2\left(\frac{D}{S} + 1\right),$$

где a и b — коэффициенты, зависящие от навыков пользователя и технических характеристик средства манипулирования; D — длина прямой линии, соединяющей начальную позицию курсора и ближайшую точку объекта; S — размер объекта вдоль линии перемещения курсора.

В общем случае процесс работы пользователя с графическим интерфейсом может быть описан с помощью нотаций методов GOMS [4] (Goals, Operators, Methods, Selection Rules — Цели, Операторы, Методы и Правила выбора). Выполняемые пользователем задачи декомпозируются на последовательности ментальных и моторных действий (например, прочесть надпись, вспомнить, перевести взгляд и внимание, нажать клавишу мыши и т. д.), для которых опытным путем установлены значения затрат времени. Сравнение суммарных затрат времени на выполнение одинакового набора задач с помощью различных вариантов графического интерфейса позволяет сделать вывод о том, какой из них более эффективен

и эргономичен. Вместе с тем, вне зависимости от конкретных условий сравнения работа с контекстными списками требует меньшего числа ментальных и моторных действий, чем использование окна ОС. Это гарантирует большую эффективность и эргономичность представленной в данной статье модернизации контекстных меню операционной системы.

Заключение

Реализованное в модели Containers модернизированное контекстное меню браузера, дерева папок и рабочего стола операционной системы Windows является конкурентоспособной оригинальной разработкой, соответствующей передовому уровню организации человеко-компьютерного взаимодействия. Реализованные технические решения позволяют повысить эффективность навигации пользователя по файловой системе компьютера — благодаря наличию "контейнеров" и использованию контекстных иерархических меню сокращаются затраты времени на доступ к требуемому объекту, а также число и сложность манипуляций, выполняемых с мышью и клавиатурой.

Представленная модель является прототипом, позволяющим оценить преимущества реализованных в ней технических решений, а также обосновать целесообразность их внедрения не только в операционные системы, но и в прикладное программное обеспечение.

В работах [2, 3] были представлены описания моделей, являющихся результатами модернизации дерева папок и строки адреса ОС Windows. Объединение данных разработок с рассмотренной в настоящей статье моделью позволяет сформировать оригинальное окно операционной системы, превосходящее существующие аналоги не только по функциональным возможностям, но также по эргономичности и удобству работы пользователя.

Список литературы

1. Зуев А. С. Математическое и программное обеспечение средств проектирования и совершенствования интерактивных графических человеко-машинных интерфейсов: Дис. ... канд. технич. наук. М.: МГУПИ, 2006. 125 с.
2. Зуев А. С., Петров Ю. И. Описание модификации дерева папок проводника Windows Explorer // Информационные технологии. 2010. № 1. С. 45–53.
3. Зуев А. С., Петров Ю. И. Описание модификации строки адреса проводника Windows Explorer // Информационные технологии. 2009. № 3. С. 11–19.
4. Раскин Д. Интерфейс: новые направления в проектировании компьютерных систем: Пер. с англ. СПб.: Символ-Плюс, 2005. 272 с.

УДК 53.089.6:004.031.2

К. И. Бушмелева, канд. физ.-мат. наук, доц.,
e-mail: bki@ivt.surgu.ru,

Е. А. Яценко, ассистент, e-mail: kea@ivt.surgu.ru
Сургутский государственный университет

Специализированный пакет программ для калибровки измерительных приборов

Приведено описание специализированного программного продукта для калибровки измерительных приборов: дана оценка его возможностей, сфера применения, информация о структуре пакета программ, пользовательский интерфейс.

Ключевые слова: калибровка, измерительные приборы, оценка адекватности, линейная зависимость, пакет программ, калибровочные модели

Введение

Калибровка измерительных систем является важным инструментом обеспечения достоверности и заданной точности измерительной информации, а также неотъемлемой составляющей государственной системы единства измерений.

Из практики исследований известно, что во многих случаях результаты измерений, проведенных разными аттестационными лабораториями, значительно отличаются друг от друга. Возможная причина сложившейся ситуации состоит в отсутствии унифицированной методики расчета калибровочной функции для измерительных приборов. Традиционно при калибровке измерительных приборов аттестационные лаборатории используют программные продукты, поставляемые с приборами, при этом алгоритм их работы неизвестен, а оценка адекватности полученных градуировочных зависимостей не проводится.

Предмет проводимых исследований — совместное применение математического и физического моделирования градуировочных функций и современных достижений в области информационных технологий при решении метрологических задач.

В качестве объекта исследования рассматриваются метрологические характеристики градуировочных моделей фотометрических приборов, применяемых для количественного химического анализа показателей качества питьевой воды.

Пакет программ для калибровки измерительных приборов

В рамках научно-исследовательской работы разработан программный продукт (ПП), предназначенный для

калибровки измерительных систем по набору стандартных образцов (СО) [1, 2].

Данный программный продукт может быть применен при расчете градуировочных функций для любых измерительных приборов с линейной зависимостью между измеряемой величиной и выходным сигналом прибора, или если эта зависимость может быть линеаризована. Также необходимым требованием является нормальный закон распределения результатов измерений.

Главной целью создания пакета программ являлось исключение постоянной систематической погрешности измерений, связанной с градуировкой измерительных приборов.

Разработанный ПП позволяет рассчитывать параметры следующих калибровочных моделей стандартных методик ISO 11095 [3] и РМГ 54-2002 [4]:

модель основного метода ISO с предположением о постоянстве среднего квадратичного отклонения;

модель основного метода ISO с предположением о пропорциональности среднего квадратичного отклонения;

метод заключения в скобки ISO;

точечный метод ISO;

метод усреднения оценок РМГ 54-2002;

метод наименьших квадратов РМГ 54-2002.

Главным показателем адекватности градуировочных характеристик в стандартных методиках является критерий Фишера. Каждая из представленных моделей предусматривает расчет критерия Фишера (F) и сравнение его с табличным значением ($F_{\text{табл}}$). Если полученное значение не превосходит заданное, то нет причин отказываться от полученной модели, в противном случае калибровочную функцию использовать нельзя. В работе предполагается, что из моделей, инвариантных по другим показателям адекватности, оптимальной считается та, у которой расчетное значение критерия Фишера наименьшее.

Пакет программ позволяет проводить анализ адекватности полученных моделей согласно критерию Фишера, что, прежде всего, дает возможность устранить постоянную систематическую погрешность, связанную с неверным предположением о виде зависимости между выходным сигналом прибора и значением измеряемой величины.

Для анализа моделей ISO в программе предоставлена визуализация всех сопутствующих графиков и таблиц, среди них таблица дисперсионного анализа, график остатков.

В пакете программ реализован новый алгоритм поиска оптимальной калибровочной модели с учетом интервального значения аттестованных значений СО. Данный алгоритм применяется для моделей стандартных методик, которые при расчетах калибровочной функции используют точечные значения СО, что, в свою очередь, противоречит действительности. На практике, аттестованное значение стандартного образца — это доверительный интервал, все точки внутри которого равновероятны. В случае, когда для калибровки используются аттестованные смеси с большими погрешностями значе-

ний величин, предположение о безошибочности принятых значений уже не может считаться верным, что не позволяет применять методику с использованием точечного значения стандартного образца.

Блок-схема алгоритма поиска оптимальной калибровочной модели с учетом интервального значения представлена на рис. 1.

В соответствии с выбранным пользователем числом точек H каждый интервал аттестованного значения образцов делится на $H - 1$ равных интервала, крайние точки интервалов рассчитываются и заносятся в массив P :

$$P = \begin{Bmatrix} \{a_{11}, a_{12}, \dots, a_{1H}\} \\ \{a_{21}, a_{22}, \dots, a_{2H}\} \\ \dots \\ \{a_{N1}, a_{N2}, \dots, a_{NH}\} \end{Bmatrix}, \quad (1)$$

где a_{ij} — это j -я точка внутри доверительного интервала i -го образца.

Целью алгоритма является перебор всех возможных комбинаций $\{(a_{1j_1}, a_{2j_2}, \dots, a_{Nj_N}), j_1, j_2, \dots, j_N = 1 \dots H\}$ и выбор той, что позволяет определить оптимальную калибровочную модель, имеющую наименьшее расчетное значение критерия Фишера $F_{\text{расч}}$.

Для повышения быстродействия алгоритма перебор осуществляется манипуляциями индексов массива P , а не самих точек доверительных интервалов, при этом индексы точек текущей комбинации хранятся в массиве V :

$$V = \{v_1, v_2, \dots, v_N\},$$

где v_i — индекс точки внутри интервала аттестованного значения i -го образца.

Для сохранения оптимальной комбинации точек и соответствующего ей расчетного значения критерия Фишера в процессе выполнения алгоритма используются глобальные переменные, обозначенные на блок-схеме соответственно: G_V и F .

Предложенный алгоритм позволяет рассчитывать оптимальные модели внутри доверительных интервалов аттестованных значений, т. е. модели с наименьшим значением критерия Фишера.

При расчете калибровочной функции возможно предположение о том, что нулевому значению измеряемой величины соответствует нулевое значение выходного сигнала прибора. В связи с этим возможны варианты использования точки начала координат. В результате был разработан и реализован алгоритм, позволяющий преобразовывать базовые калибровочные модели в соответствии с приведенными ниже вариантами:

- использование точки с координатами (0,0) как элемента входного массива измерений;
- обязательное построение калибровочной функции из точки начала координат;
- не учитывать точку (0, 0) особым образом.

Алгоритм преобразования калибровочных моделей в соответствии с предположениями о точке начала координат делает процесс калибровки более гибким, обеспечивая дополнительные возможности при поиске адекватных моделей, что способствует исключению система-

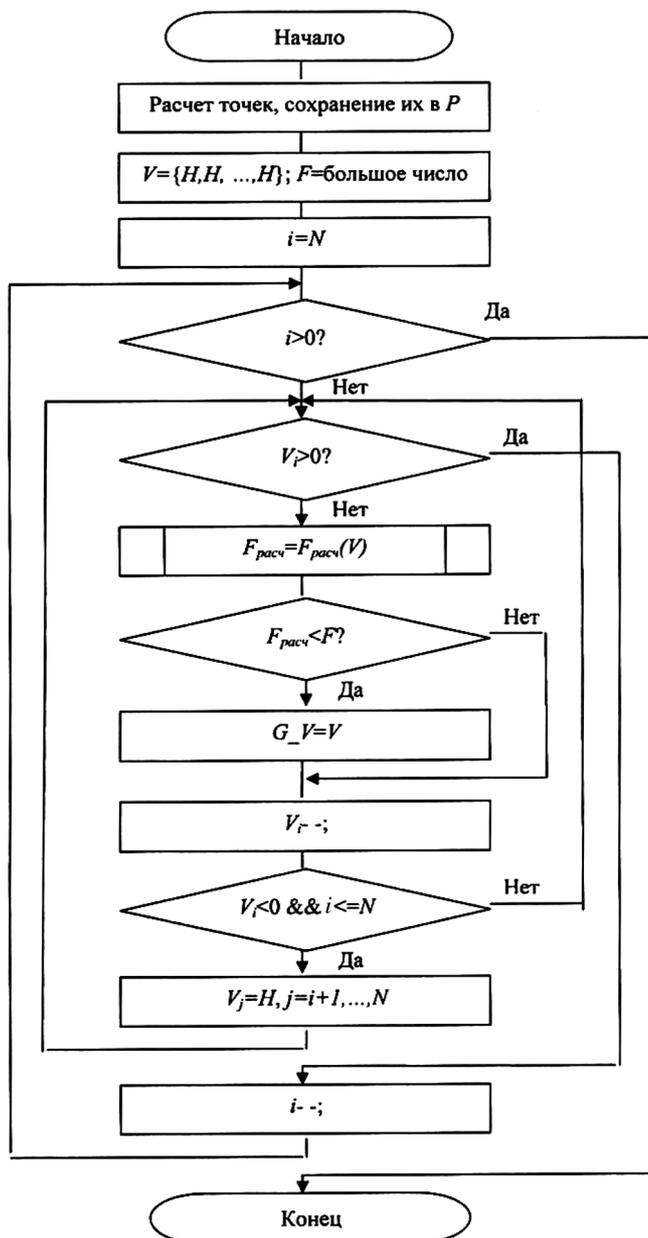


Рис. 1. Блок-схема поиска оптимальной калибровочной модели

тической погрешности измерений, связанной с неверно выполненной градуировкой приборов.

Использование пакета программ позволяет обеспечить:

- сокращение времени, необходимого для вычислений при калибровке измерительных комплексов;
- снижение требований к квалификации персонала, занимающегося калибровкой;
- снижение возможности возникновения ошибок при расчете показателей калибровочной функции до минимума.

Обобщенная функциональная схема структуры пакета программ представлена на рис. 2.

Реализованный пакет программ представляет собой совокупность основных подсистем: "Интерфейсная подсистема" и "Подсистема расчетов".

"Интерфейсная подсистема" программного продукта выполняет функции взаимодействия с пользователем и "Подсистемой расчетов". Для взаимодействия с пользователем разработан пользовательский интерфейс, описание которого приведено ниже.

Для хранения данных на жестком диске разработан особый метод, в его основе использование динамически формируемой иерархической структуры, позволяющей хранить результаты измерений и расчетов, а также аттестованные характеристики СО. Принцип организации и работы структуры данных не содержит лимитов на объем хранимой информации, ограничения устанавливает только файловая система. Сохранение и считывание информации с внешних носителей осуществляется по запросу пользователя посредством соответствующих элементов управления интерфейса.

"Генератор запросов" формирует входной блок данных для "Подсистемы расчетов". Блок данных содержит аттестованные значения стандартных образцов, результаты их измерения, а также все необходимые параметры способа расчета калибровочной функции: вид модели, информацию о включении точки начала координат, будет ли использован интервальный алгоритм поиска или требуется построить градуировочный график по средним точкам доверительных интервалов.

Функциональная структура "Подсистемы расчетов" представлена на рис. 3.

"Подсистема расчетов" выполняет расчет параметров калибровочных функций в зависимости от запроса. Алгоритмы расчета градуировочных функций в соответствии с различными методиками реализованы расчетными модулями. Расчетные модули ISO и РМГ реализуют расчеты в соответствии с нормативной документацией: ISO 11095 и РМГ 54-2002.

"Менеджер расчетов" с единой точкой входа анализирует запрос "Интерфейсной подсистемы". В зависимости от заданной модели данный блок преобразует входной массив данных, осуществляя таким образом учет варианта точки (0, 0) в расчетах. Например, если необходимо преобразовать стандартную модель в соответствии с предположением о точке ноль — ноль как точка, то входной блок измерительной информации пополняется дополнительной точкой с координатами 0, 0.

При наличии соответствующего запроса "Менеджер расчетов" реализует алгоритм поиска оптимальной мо-

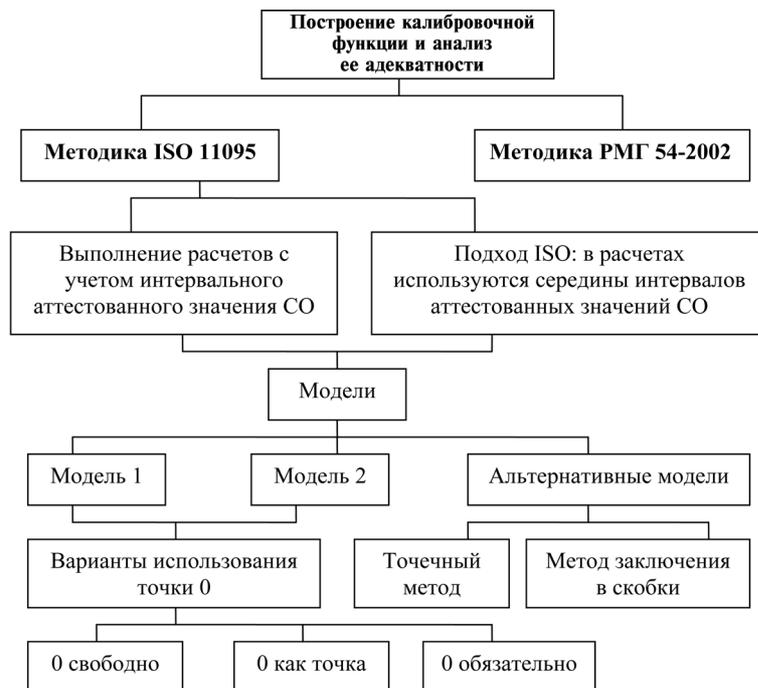


Рис. 3. Функциональная структура "Подсистемы расчетов"

дели с учетом интервальных значений стандартных образцов. В основе алгоритма лежит перебор точек внутри доверительного интервала с заданным шагом, на каждой итерации алгоритма формируется массив измерений и вызывается соответствующий модуль, результатом работы каждой итерации является расчетное значение критерия Фишера. Оптимальным набором входных измерений будет массив точек, полученный из доверительных интервалов.

Пакет программ имеет интуитивно понятный, удобный и универсальный интерфейс. Главная экранная форма ПП для калибровки измерительных приборов (рис. 4) состоит из четырех основных панелей: "Измерения", "Расчеты", "Дополнительные данные" и "График".

Панель "Измерения" позволяет вводить, просматривать и редактировать входной массив данных: аттестованные характеристики стандартных образцов или аттестованных смесей, результаты измерений.

Панель "Расчеты" позволяет просматривать и сравнивать параметры калибровочных функций, полученные в соответствии с различными методиками. Для каждой рассчитанной модели на экране отображаются коэффициенты линейного уравнения, расчетные и табличные значения критерия Фишера, заключение об адекватности модели. Если расчеты выполнялись в соответствии со стандартной методикой РМГ 54-2002, отображается заголовок выбранного метода.

Интерфейс пользователя включает визуализацию всех промежуточных данных, полученных на различных этапах использования стандартных методик ISO 11095: таблицы дисперсионного анализа "Анова", графика остатков, экспериментальных точек и градуировочного графика. Таблица дисперсионного анализа содержится на панели "ANOVA" вкладки "Расчеты". Градуировочный график представлен на соответствующей вкладке "График".

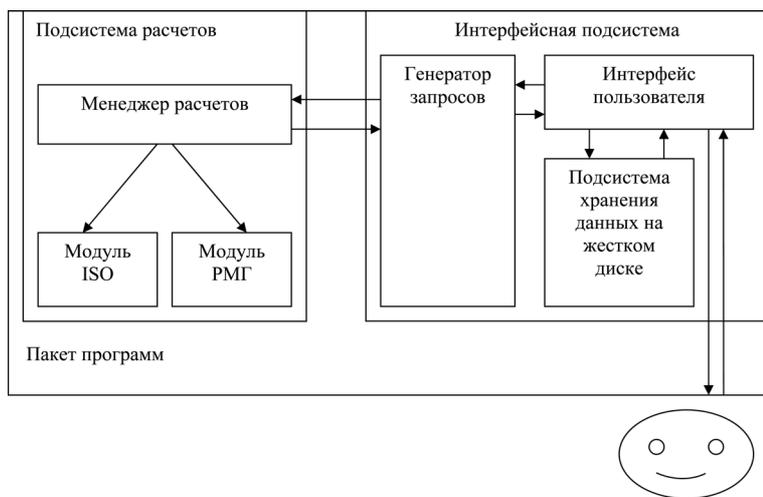


Рис. 2. Функциональная структура пакета программ

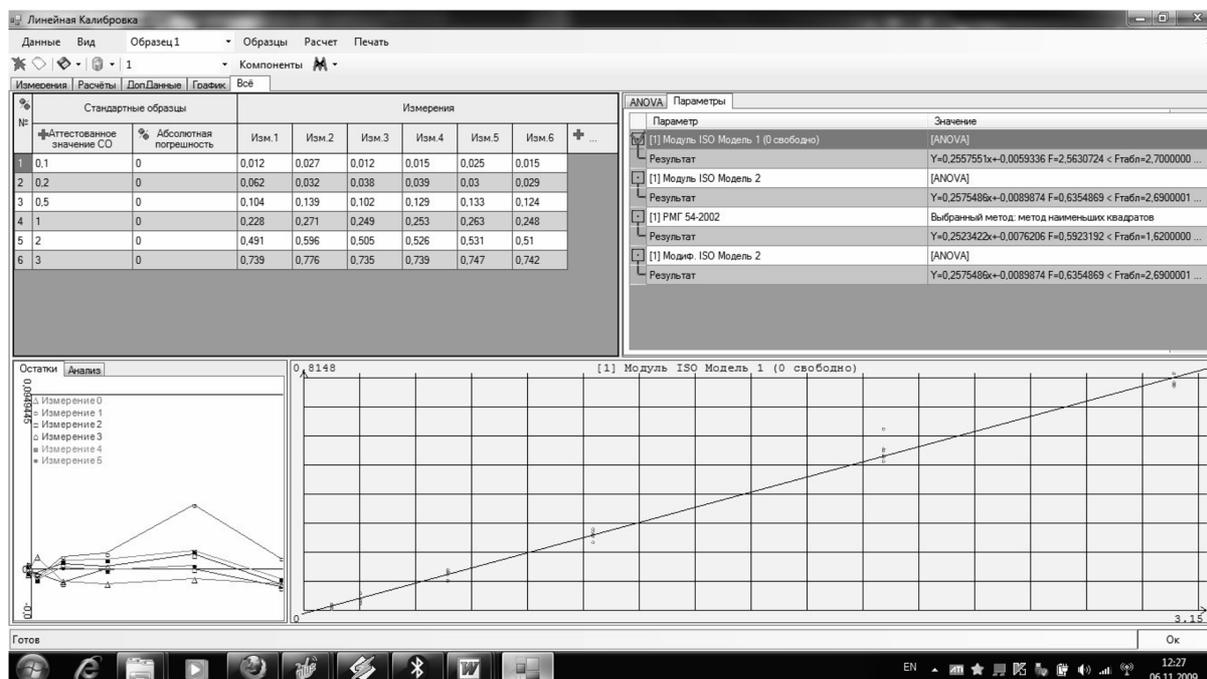


Рис. 4. Главная экранная форма пакета программ

Вкладка "Дополнительные данные" содержит панель "Остатки", на ней представлен график остатков, являющийся основным средством выбора одной из двух моделей ISO: "Модель 1", если остаток стандартного отклонения постоянен; "Модель 2", если точки графика образуют некоторый тренд, например линейный, или, если дисперсия возрастает с увеличением аттестованного значения в измеренных СО [5, 6].

Дополнительная функциональная возможность пакета программ состоит в визуализации зависимости расчетного критерия Фишера от выбранной точки внутри доверительного интервала аттестованного значения СО, представленной на вкладке "Дополнительные данные" панели "Анализ".

Результатом калибровки прибора с применением ПП является градуировочная зависимость, выраженная алгебраически и графически, из данной информации формируется отчет, который при необходимости можно распечатать.

Главными отличительными особенностями представленного программного продукта являются заложенные в основу его разработки принципы выбора калибровочной модели в соответствии с особенностями измерительного средства и объекта измерений, а также унификация процесса расчета градуировочной функции, направленные на снижение погрешностей измерений.

Заключение

Подводя итог необходимо отметить, что использование разработанного программного продукта метрологами аттестационных лабораторий позволяет исключить погрешности, связанные с неправильным выбором вида калибровочной функции и ошибками при ее расчете, снизить затраты времени на выполнение калибровки приборов посредством автоматизации расчетов, унифицировать процесс калибровки приборов за счет использования стандартных алгоритмов. Пакет программ может быть использован для калибровки измерительных систем в таких сферах деятельности, как медицина, экология, легкая и пищевая промышленность, парфюмерия и т. п. Внедрение описанного программного продукта в различные аттестационные лаборатории позволит получить измерительную информацию с заданной точностью для принятия адекватных решений при управлении производством, для поддержания экологической обстановки, оценки качества пищевых продуктов, питьевой воды и т. д.

Список литературы

1. **Свидетельство** об официальной регистрации программы для ЭВМ № 2006613662. Программа линейной калибровки измерительных комплексов / Беднаржевский С. С., Коптева Е. А. и др. (RU). Заяв. 06.07.2006. Зарег. 20.10.2006.
2. **Свидетельство** об официальной регистрации программы для ЭВМ № 2007613086. Градуировка средств измерений на основе рекомендаций по межгосударственной стандартизации / Шевченко Н. Г., Коптева Е. А. и др. (RU). Заяв. 23.04.2007. Зарег. 20.07.07.
3. **ISO 11095 International Standard**. Liner calibration using reference materials. ISO, 1996.
4. **РМГ 54-2002 ГСИ**. Характеристики градуировочные средств измерений состава и свойств веществ и материалов. Методика выполнения измерений с использованием стандартных образцов. — Введ. 26-01-04.
5. **Коптева Е. А., Бушмелева К. И.** Моделирование калибровочных функций с использованием пакета программ для анализа качества питьевой воды // *Материалы Межд. форума "Новые информационные технологии и менеджмент качества" (NIT & QM) / Под ред. В. Н. Азарова.* — М.: Фонд "Качество", 2009. С. 82—85.
6. **Яценко Е. А., Бушмелева К. И.** Линейная калибровка с применением специализированного программного средства // *Метрология.* 2009. № 11. С. 41—47.

С. Ю. Курсков, канд. физ.-мат. наук, доц.,
Н. Ю. Ершова, канд. физ.-мат. наук, доц.,
О. А. Мурсалимов, инженер,
 Петрозаводский государственный университет,
 г. Петрозаводск,
 e-mail: kurskov@psu.karelia.ru

Организация удаленного доступа к микроконтроллерам MCS-96 на базе распределенной информационно-измерительной и управляющей системы поддержки научно-образовательного процесса

Описана организация удаленного доступа к микроконтроллерам Intel MCS-96 на базе распределенной информационно-измерительной и управляющей системы, обеспечивающей доступ к аппаратным и информационным ресурсам лабораторного комплекса в сетях интранет/Интернет.

Ключевые слова: удаленный доступ к оборудованию, информационно-измерительная и управляющая система, сервер оборудования, сервер микроконтроллеров MCS-96

Введение

Рост интереса к высшему образованию неизбежно приводит к увеличению числа студентов в учебных группах и, следовательно, к нехватке рабочих мест в лабораториях университета, в которых используется дорогостоящее оборудование. Наиболее эффективным решением этой проблемы является организация многопользовательского доступа к приборам лабораторного комплекса через компьютерную сеть.

Применение Web-технологий позволяет обеспечить доступ к физическому оборудованию (см., например, [1, 2]), однако значительное число измерительных и исполнительных устройств, сложность современных аппаратных средств, а также необходимость протоколирования хода работы и результатов измерений требуют развертывания в лаборатории распределенной информационно-измерительной и управляющей системы.

Распространенные инструментальные пакеты (например пакет LabVIEW), а также системы визуализации измерительной информации (SCADA-системы), написанные с использованием инструментальных пакетов, обеспечивают в той или иной степени удаленное взаимодействие с физической аппаратурой, однако в этих системах оборудование подключено к тому компьютеру, на котором запущен инструментальный пакет. Это затрудняет использование различных приборных интерфейсов, связанных с отдельными подсистемами лабораторного комплекса и подключенных к разным компьютерам. Кроме того, хотя подобные пакеты имеют дружественный интерфейс и обычно обладают средствами визуального программирования, они не обеспечивают гибкости в по-

строении системы и ее расширяемости. Поэтому использование подобных пакетов не всегда оправдано для создания систем сбора данных, контроля и управления.

Цель данной работы — обеспечение удаленного доступа к микроконтроллерам семейства Intel MCS-96 в рамках лабораторных практикумов по курсам "Микропроцессорные средства" и "Организация удаленного доступа к АИИС", которые читают студентам физико-технического факультета Петрозаводского государственного университета.

Структура распределенной системы

Распределенная информационно-измерительная и управляющая система построена по модульному принципу, который реализуется как в структуре системы, так и в программном обеспечении [3]. Элементы системы встроены в систему по унифицированным правилам и взаимодействуют по единому протоколу в соответствии с принципами открытых систем.

Отличительной особенностью системы является то, что она позволяет объединить различные приборные интерфейсы (систему PXI, систему КАМАК, канал общего пользования GPIB и др.) [4] с выделенными для них компьютерами в единую сеть, функционирующую на базе стека протоколов TCP/IP.

В качестве основы для построения системы использована технология "клиент—сервер", причем для доступа к физическому оборудованию лабораторного комплекса был разработан собственный прикладной протокол поверх TCP/IP [5], что обеспечивает функционирование системы в сетях Интернет/интранет. Необходимость разработки прикладного протокола обусловлена тем, что используемые обычно Web-технологии не обеспечивают гибкого управления ходом измерений, поскольку сбором данных, в этом случае, управляют не программы-клиенты, работающие на удаленных компьютерах, а исполняемый код, хранящийся на компьютере, который непосредственно подключен к лабораторной установке.

В состав гетерогенной системы входят: программы-клиенты, осуществляющие непосредственное управление оборудованием; коммуникационный сервер, являющийся ключевым звеном всей системы; серверы оборудования (КАМАК-сервер [6]; сервер канала общего пользования (КОП) [7]; сервер Ethernet-устройств [8] и др.); измерительные и исполнительные устройства лабораторного комплекса, а также сервер базы данных [9] (рис. 1).

Кадры прикладного протокола обмена данными между коммуникационным сервером и серверами оборудования и кадры протокола обмена данными между коммуникационным сервером и клиентами приведены на рис. 2.

Кадр запроса к серверу оборудования содержит адрес ресурса, функцию, тип данных, ключ и данные. Если значение поля "Тип данных" равно нулю, то в поле "Данные" находятся сами данные. Если "Тип данных" равен единице, то поле "Данные" содержит длину передаваемых данных в байтах (данные идут после основного кадра непрерывным потоком). Поле "Ключ" предназначено для администрирования системы. Ответ сервера оборудования содержит в себе номер ошибки, тип данных, данные и информацию о состоянии системы. Поле "Состояние" используется сервером оборудования для передачи пользователю содержимого регистров состояния контроллера приборного интерфейса. Кадры протокола обмена данными между коммуникационным сервером и клиентами содер-

жат дополнительное поле "Сервер", в котором указывается псевдоним запрошенного сервера оборудования.

Сервер оборудования представляет собой сервер последовательной обработки запросов. В его задачу входит определение допустимости для данного оборудования запрошенной функции и указанного адреса, передача запроса оборудованию, а также пересылка клиенту ответа или номера ошибки при возникновении исключительной ситуации. Таким путем достигается универсальность коммуникационного сервера, который работает с серверами оборудования по единому прикладному протоколу.

Сервер оборудования имеет типовую структуру и для разных приборных интерфейсов отличается лишь библиотеками методов, реализующих взаимодействие с конкретным приборным интерфейсом.

Центральное звено системы — коммуникационный сервер, являющийся посредником между клиентами и серверами оборудования. В его функции входят анализ переданной программой-клиентом информации (команд управления физической аппаратурой) и формирование на ее основе запроса к тому или иному серверу оборудования в соответствии с прикладным протоколом, а также аутентификация пользователей, что повышает безопасность при работе с системой. Кроме того, структура коммуникационного сервера обеспечивает одновременный доступ нескольких пользователей к исследовательскому комплексу и его подсистемам; при этом устройства, используемые одним клиентом, на время обмена данными защищаются от других клиентов.

Основу коммуникационного сервера составляют следующие классы:

- StartCServer — инициализация сервера. В этом классе осуществляется опрос всех серверов оборудования и установление связи с работающими серверами, после чего коммуникационный сервер переходит в режим ожидания связи с клиентами;
- ServerThread — класс, реализующий алгоритм обслуживания определенного клиента. В этом классе обрабатываются команды клиента, формируются запросы к серверам оборудования, обновляется информация об используемых модулях (приборах), подключенных к соответствующему приборному интерфейсу;
- CserverProtocol — интерфейс, который определяет коды операций, ошибок и другие константы протоколов;
- MainServInf — класс для хранения информации о серверах оборудования, входящих в информационную систему. Данные содержатся в виде набора за-

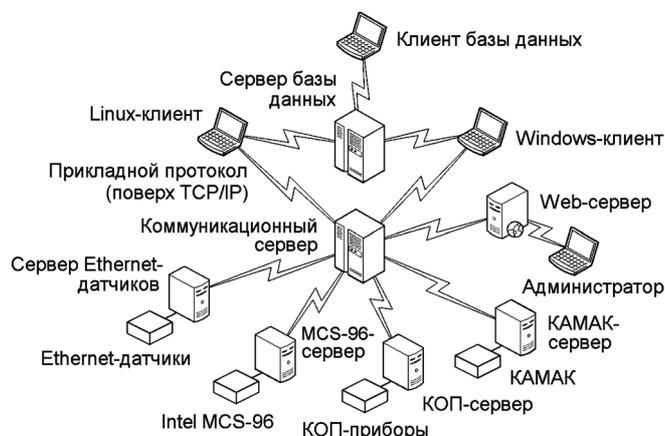


Рис. 1. Схема распределенной информационно-измерительной и управляющей системы

Запрос к серверу оборудования

адрес ресурса	функция	т	ключ	данные
1	8 9	12 13 14	17 18	21

Ответ сервера оборудования

ошибка	т	данные	состояние
1	4 5 6	9 10	13

Запрос к коммуникационному серверу

сервер	запрос к серверу оборудования
1	4 5

Ответ коммуникационного сервера

сервер	ответ сервера оборудования
1	4 5 17

Рис. 2. Кадры протокола обмена данными между коммуникационным сервером и серверами оборудования (а) и кадры протокола обмена данными между коммуникационным сервером и клиентами (б): т — тип данных

писей, включающих следующие поля: IP-адрес сервера оборудования; номер порта, на котором сервер оборудования ожидает связи с коммуникационным сервером, его псевдоним, сокет (если связь установлена) и состояние сервера оборудования;

- MainClientInfo — класс, отвечающий за мониторинг системы. Используется для хранения информации о пользователях, работающих в данный момент с исследовательской аппаратурой. Информация включает в себя IP-адрес клиента, идентификационный номер клиента и выделенные ему ресурсы (псевдоним сервера оборудования и адрес ресурса).

В классах ReplyFromCServer, QueryToCServer, ClientReply, ClientQuery определены кадры запроса и кадры ответа прикладного протокола для обмена с серверами оборудования и клиентами, а также методы работы с ними.

Система работает следующим образом. После запуска коммуникационный сервер считывает информацию о доступных серверах оборудования из конфигурационного файла (IP-адрес, номер порта, псевдоним). Далее последовательно устанавливаются постоянные соединения с серверами оборудования. Сервер, с которым соединение не было установлено, помечается как недоступный в данный момент. Попытка установления связи с ним будет повторена при обращении к нему любого клиента. После инициализации серверов оборудования коммуникационный сервер переходит в режим ожидания связи с клиентами. При установлении связи сервер генерирует параллельный процесс обслуживания клиента и присваивает этому процессу уникальный номер CID (Client ID), не равный нулю. Обмен информацией с клиентом осуществляется в режиме "запрос—ответ".

В системе предусмотрен режим администратора [10]. Доступ администратора к коммуникационному серверу осуществляется с помощью стандартного браузера и Web-сервера. Взаимодействие Web-сервера с коммуникационным сервером реализовано посредством сервлета. Сервлет используется для организации сетевого обмена с коммуникационным сервером в соответствии с протоколом системы и динамической генерации HTML-страниц. Сервлет предоставляет администратору как средство удаленного доступа к системе, так и механизм самого доступа.

Данные, получаемые клиентами от физического оборудования, сохраняются в базе данных, которой управляет СУБД Oracle 9i. Программы-клиенты обращаются к

ней напрямую, минуя коммуникационный сервер, поскольку последний не обладает информацией о типе проводимых измерений.

Безопасность системы достигается путем шифрования данных, которыми обмениваются коммуникационный сервер, серверы оборудования и клиенты [11]. Это представляется необходимым, так как существует вероятность подмены кадров неавторизованным пользователем и, следовательно, получение несанкционированного доступа к оборудованию лабораторного комплекса, а также к информации клиентов, работающих с системой.

Программное обеспечение системы реализовано на языке Java. Выбор языка Java обусловлен тем, что код, написанный на нем, может работать на любой программно-аппаратной платформе, для которой существует виртуальная Java-машина. Методы записи-чтения портов ввода-вывода для доступа к контроллерам приборных интерфейсов написаны на языке Си и оформлены в виде динамической библиотеки. При создании программного обеспечения был использован пакет разработчика Java — Java Standard Edition Development Kit v.6.0.

Сервер микроконтроллеров Intel MCS-96

Лабораторный практикум по курсу "Микропроцессорные средства" построен на базе макетов, разработанных в центре "Микропроцессорные технологии" при Санкт-Петербургском ГЭТУ им. Ульянова (Ленина). Лабораторный макет содержит шестнадцатиразрядный микроконтроллер 80C196KC семейства MCS-96 фирмы Intel, блок внешней памяти, состоящий из микросхем ПЗУ и ОЗУ, драйвер последовательного канала RS-232, блоки световой и звуковой индикации, разъемы расширения.

Удаленный доступ к микропроцессорным платам организован с помощью разработанного авторами сервера, который позволяет подключать к системе до четырех микроконтроллеров 80C196KC. Микроконтроллеры подключаются к персональному компьютеру через последовательные порты COM1—COM4 стандарта RS-232.

Сервер микроконтроллеров представляет собой программу, которая создает до четырех отдельно работающих потоков, каждый из которых управляет своим микроконтроллером, подключенным к соответствующему COM-порту. Коммуникационный сервер различает микроконтроллеры по адресу COM-порта, который включается в поле адреса ресурса кадра запроса.

В состав сервера микроконтроллеров входят следующие классы:

- MCSServer — основной класс сервера — он реализует "прослушивание сети", подключение коммуникационного сервера;
- MCSServerThread, который реализует алгоритм обслуживания клиента, в частности, обеспечивает проверку корректности значений входных параметров запроса клиента, прием HEX-файла, загрузку его во внешнюю память на лабораторном макете и выполнение команд управления микроконтроллером;
- CserverProtocol, в данном интерфейсе определены коды операций, ошибок и другие константы прикладного протокола (общий для системы);
- QueryToEServer, в данном классе определен объект "кадр запроса" к серверу оборудования и методы для работы с этим объектом (общий для системы);
- ReplyFromEServer, в этом классе определен объект "кадр ответа" от сервера оборудования и методы для работы с этим объектом (общий для системы);

- MCS96Lib, данный класс содержит библиотеку основных методов для работы с микроконтроллерами, а также ряд дополнительных методов, выполняющих вспомогательные функции. Основные методы класса: Reset — сброс микроконтроллера в исходное состояние; Init — инициализация микроконтроллера; Load_Hex — загрузка откомпилированной программы пользователя во внешнее ОЗУ микроконтроллера; TestLine — проверка связи с микроконтроллером и его инициализация; Run — выполнение загруженной программы пользователя; ReceiveData — прием данных от микроконтроллера. Для выполнения операций чтения-записи в порты ввода-вывода в данном классе объявлено шесть внешних методов, реализованных на языке Си: InitCOM — установка параметров COM-порта; InCOM — чтение байта из COM-порта; OutCOM — запись байта в COM-порт; FIFOon — включение FIFO-буфера; FIFOoff — выключение FIFO-буфера; DirectIO — доступ к портам ввода-вывода в защищенном режиме.

Ход каждой лабораторной работы контролирует программа-клиент распределенной информационно-измерительной и управляющей системы. Программа-клиент для работы с микроконтроллерами состоит из библиотеки методов, формирующих запросы к MCS-серверу на выполнение команд, и класса-оболочки, позволяющего пользователям создавать программы для управления аппаратными средствами лабораторного комплекса. В классе-библиотеке также "спрятаны" от пользователя методы, реализующие взаимодействие клиента приложения с распределенной системой.

Управляет обменом данными с информационно-измерительной и управляющей системой класс-оболочка. Класс-оболочка имеет специальное окно для ввода студентами команд, пересылаемых серверу оборудования.

При изменении алгоритма работы с устройством, подключенным к микроконтроллеру, программа-клиент позволяет изменить программу во внешней оперативной памяти, с которой работает микроконтроллер. При выборе соответствующей команды клиент запускает компилятор с Ассемблера для MCS-96 с указанием имени компилируемого файла, затем компоновщик, и, далее полученный HEX-файл пересылает по сети на сервер. Если адресованный микроконтроллер не занят, то сервер загружает исполняемый файл во внешнюю оперативную память на лабораторном макете и дает команду на его выполнение.

Возможность удаленной замены исполняемого файла позволяет пользователю изменять алгоритм работы микроконтроллера с внешними устройствами в соответствии с поставленной задачей.

На данный момент лабораторный практикум "Микропроцессорные средства" содержит работы по изучению способов адресации памяти, особенностей системы и формата команд микроконтроллеров семейства MCS-96, программированию шестнадцатиразрядных счетчиков, широтно-импульсного модулятора, аналого-цифрового преобразователя, блока быстрого ввода-вывода, программируемого контроллера прерываний и сервера периферийных транзакций.

В лабораторном практикуме "Организация удаленного доступа к АИИС" микропроцессорные платы используются в качестве физического оборудования, для управления которым студенты разрабатывают клиент-серверные приложения с доступом к устройствам через сеть Интернет.

Заключение

В заключение отметим, что основным результатом работы стало создание программного обеспечения, реализующего удаленный доступ к микроконтроллерам семейства

MCS-96 и расширяющего функциональные возможности распределенной информационно-измерительной и управляющей системы. Данное программное обеспечение поддерживает многопользовательский режим, т. е. обеспечивает одновременный доступ нескольких студентов к микроконтроллерным платам, при этом модуль, используемый одним студентом, на время обмена данными защищается от других студентов. Кроме того, это программное обеспечение позволяет организовать индивидуальную работу студентов с лабораторным оборудованием со стандартных рабочих мест в компьютерных классах и лабораториях.

Также следует отметить, что использование распределенной информационно-измерительной и управляющей системы как основы для построения лабораторных практикумов особенно выгодно тем, что система позволяет упростить подключение нового исследовательского и лабораторного оборудования. В отличие от распределенных систем добавление к системе нового приборного интерфейса с различной измерительной и исполнительной аппаратурой сводится к регистрации соответствующего сервера оборудования в коммуникационном сервере, после чего клиентские программы получают доступ к этой аппаратуре. Кроме того, поскольку алгоритм работы распределенной системы задается клиентской программой, выполняющейся на удаленном компьютере, а не программой-сервером, можно отказаться от лабораторных работ с жестко заданным алгоритмом выполнения. Другими словами, обучаемый может сам модифицировать управляющую программу в соответствии с выполняемым заданием.

Список литературы

1. **Зимин А. М., Букеткин Б. В., Почувев А. П. и др.** Учебная Интернет-лаборатория "Испытания материалов" // Информационные технологии. 2006. № 10. С. 58—65.
2. **Зимин А. М., Аверченко В. А., Лабзов С. Ю. и др.** Лабораторный практикум по спектральной диагностике плазмы с удаленным доступом через Интернет // Информационные технологии. 2002. № 3. С. 37—42.

3. **Гаврилов С. Е., Жиганов Е. Д., Кипрушкин С. А., Курсков С. Ю.** Распределенная информационно-измерительная система для удаленного управления экспериментом в области оптической спектроскопии // Научный сервис в сети Интернет: Тр. Всерос. науч. конф. М.: Изд-во Моск. ун-та, 2002. С. 157—159.

4. **Эрглис К. Э.** Интерфейсы открытых систем. М.: Горячая линия Телеком, 2000.

5. **Gavrilov S. E., Kiprushkin S. A., Kurskov S. Yu.** Distributed information system with remote access to physical equipment // Proceedings of the International Conference on Computer, Communication and Control Technologies (CCCT '03) and the 9th International Conference on Information Systems Analysis and Synthesis (ISAS I03), July 31 — August 2, 2003, Orlando, Florida, USA. Orlando, 2003. V. 1. P. 288—291.

6. **Zhiganov E. D., Kiprushkin S. A., Kurskov S. Yu.** CAMAC Server for Remote Access to Physical Equipment // Learning and Teaching Science and Mathematics in Secondary and Higher Education. Joensuu: University of Joensuu, 2000. P. 170—173.

7. **Кашуба А. С., Кипрушкин С. А., Курсков С. Ю.** Сервер канала общего пользования распределенной информационной системы поддержки научных исследований в области оптической спектроскопии // Технологии информационного общества — Интернет и современное общество 2002 (IST/IMS 2002): Материалы V Всерос. объединенной конф. — СПб.: Изд-во С.-Петербург. ун-та, 2002. С. 104—105.

8. **Kiprushkin S., Kurskov S., Sukharev E.** Connection of network sensors to distributed information measurement and control system for education and research // International Journal "Information Technologies & Knowledge". 2007. V. 1, N 2. P. 171—175.

9. **Kiprushkin S., Kurskov S., Semin V.** Development of database for distributed information measurement and control system // International Journal "Information Technologies & Knowledge". 2008. V. 2, N 3. P. 283—286.

10. **Kiprushkin S. A., Kurskov S. Yu., Nosovich N. G.** Resources Control in Distributed Information Measurement System // Proceedings of the 3rd International Conference on Computing, Communication and Control Technologies (CCCT '05), July 24—27, 2005, Austin, Texas, USA. —Austin, 2005. V. 1. P. 255—258.

11. **Kiprushkin S. A., Korolev N. A., Kurskov S. Yu.** Sharing of Instrument Resources on the Basis of Distributed Information Measurement System // Proc. of the Second IASTED International Multi-Conference on Automation, Control, and Information Technology — Automation, Control, and Applications: ACIT-ACA 2005, June 20—24, 2005, Novosibirsk, Russia. — Novosibirsk: ACTA Press, 2005. P. 170—175.

ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

УДК 004.94

В. Г. Гетманов, д-р техн. наук, проф.,

С. Е. Орлов, аспирант, Национальный исследовательский ядерный университет "МИФИ"
e-mail: vgetm@starnet.ru

Программный комплекс для спектрально-временного анализа нестационарных колебательных сигналов

Рассматривается программный комплекс SPECTRANS, предназначенный для спектрально-временного анализа нестационарных колебательных сигналов, основанный на использовании локальных и сплайновых моделей. Приводятся сведения о применении и особенностях структуры ПК SPECTRANS. Анализируются этапы работы комплекса, связанные с просмотром и редактированием, локальной и сплайновой аппроксимацией сигналов. Описываются примеры проведения спектрально-временного анализа нестационарных колебательных сигналов на основе программного комплекса SPECTRANS.

Ключевые слова: спектрально-временный анализ, нестационарные колебательные сигналы, локальная и сплайновая аппроксимация

Введение

Программный комплекс SPECTRANS позволяет решать многочисленные задачи спектрально-временного анализа (СВАН) — оценивания нестационарных параметрических функций колебательных сигналов (амплитудных, фазовых и частотных функций). На основе данного программного комплекса могут обрабатываться наблюдения одно- и многочастотных амплитудно- и частотно-модулированных сигналов, наблюдения сигналов с низкочастотными аддитивными трендами и наблюдения систем одночастотных амплитудно- и частотно-модулированных сигналов для задач оценивания нестационарных разностей фаз. SPECTRANS ориентирован на СВАН нестационарных колебательных сигналов для радиоэлектроники, измерительной техники, систем связи, экспериментальной механики и т. д.

Программный комплекс SPECTRANS базируется на применении систем локальных и сплайновых моделей, используемых для аппроксимации нестационарных колебательных сигналов [1]. Ранняя версия данного комплекса описана в работах [2, 3]. Технология цифровой обработки нестационарных сигналов на основе SPECTRANS составляет предмет изобретения [4]. SPECTRANS состоит из набора взаимосвязанных программных модулей, осуществляющих вычисления, графический вывод результатов обработки и управление программным комплексом с помощью систем интерфейсных окон и полей.

Программный комплекс SPECTRANS отличается от комплексов программ для аппроксимации Curves Fitting Toolkit для LabVIEW 8.5 и Curves Fitting Toolbox для MatLab2008 благодаря применению специальных модельных аппроксимационных функций и специальных, реализующих большое быстродействие оптимизационных процедур, обеспечивающих решения задач аппроксимации.

Главная особенность программного комплекса SPECTRANS — возможность проведения СВАН на коротких интервалах времени и при значительных нестационарностях в сигналах по параметрам. Этот комплекс может быть применен к таким сигналам, для которых ДПФ (дискретное преобразование Фурье) является неэффективным, например, на интервалах наблюдений, состоящих из единиц — десятков периодов колебаний сигнала со значительной внутрипериодной амплитудной или частотной модуляцией. Вследствие использования подгонки для аппроксимационных моделей, комплекс обеспечивает разрешающую способность предлагаемого СВАН, на порядок превосходящую разрешающую способность ДПФ.

В настоящее время СВАН для нестационарных колебательных сигналов проводятся, главным образом, тремя группами методов; методы Short Time Fourier Transform (STFT) на основе ДПФ; методы на основе wavelet-преобразований; методы на основе ЧВП (частотно-временных распределений).

СВАН может быть осуществлен с помощью целого ряда достаточно распространенных программных систем, к которым можно отнести:

- систему LabVIEW 8.6 фирмы National Instruments (США), включающую программный комплекс NI Spectrum Measurement Toolkit [5, 6];
- систему программируемого анализатора сигналов модели Pulse 3560 В & К фирмы Bruel & Kjer (Дания), включающую программные модули Pulse Software System Data и Time Capture STFT 7705 для анализа временных выборок [7, 8];

- программную систему FlexPro 8 фирмы Weisang GmbH (Германия) [9];
- программную систему SIG-VIEW фирмы Intel (США) [10];
- систему программ The Time-Frequency Toolbox for MatLab, разработанную в CNRS — Centre National Recherche Scientifique (Франция) [11];
- библиотеку программ для ЧВП разработки Rice University DSP Group (Техас, США) [12].

Среди близких по назначению отечественных программных систем необходимо отметить:

- систему WinПОС фирмы НПП "Мера" (г. Королев) [13];
- систему ZETLab Studio фирмы ЗАО "Электронные технологии и метрологические системы" (г. Зеленоград, Моск. обл.) [14];
- систему для ЧВП, разработанную в Нижегородском университете [15].

Методы, примененные в программном комплексе SPECTRANS, являются в определенной степени альтернативными и дополнительными по отношению к методам STFT (ДПФ), wavelet-преобразований и ЧВП (частотно-временных распределений).

Проведение СВАН на основе предлагаемого программного комплекса реализуется с участием оператора. Работа SPECTRANS включает несколько этапов.

Этап № 0 — просмотр и редактирование файлов сигналов. Данный этап является предварительным. В результате просмотра экспертно проверяется качество записей сигналов, например на наличие аномальных наблюдений, уточняется соответствие записей сигналов условиям эксперимента и проводится выбор конкретного сигнала для последующего анализа. В результате редактирования в сигнале выделяется определенный участок, который предполагается подвергнуть дальнейшей цифровой обработке. На данном этапе задается значение интервала дискретизации исходя из априорных сведений о дискретизации экспериментального сигнала.

Этап № 1 — применение алгоритмов локальной аппроксимации. На этапе локальной аппроксимации выбирается подходящая локальная модель для аппроксимации сигнала, сформированного на этапе № 0, осуществляется разбиение основного интервала наблюдения на локальные интервалы, настройка параметров алгоритмов локальной аппроксимации и построение системы локальных моделей с помощью алгоритмов локальной аппроксимации.

Этап № 2 — применение алгоритма сплайновой аппроксимации. На этапе сплайновой аппроксимации проводится разбиение основного интервала наблюдения на сплайновые интервалы, выбор порядка сплайнового полинома и применение алгоритма сплайновой аппроксимации системы локальных моделей от этапа № 1.

В процессе работы SPECTRANS переход от этапа к этапу может осуществляться в произвольном порядке. Реализуются два режима работы комплекса — с экспериментальными сигналами и с модельными сигналами. Режим модельных сигналов используется для оценивания возможных точностных характеристик алгоритмов СВАН и определения настроек аппроксимационных алгоритмов.

Этап просмотра и редактирования

На этапе № 0 обеспечивается просмотр и редактирование экспериментальных сигналов [16]. На вход комплекса SPECTRANS подаются файлы исходных экспе-

риментальных сигналов в форматах .txt, .wav, .xls или .mat. Сигналы в файлах размещаются в прямоугольных матрицах и располагаются либо по строкам, либо по столбцам. В программном комплексе выполняется автоматическое приведение матриц сигналов к унифицированной форме с размещением сигналов по строкам. Все необходимые операции по просмотру и редактированию файлов сигналов осуществляются в едином интерфейсном окне с использованием системы интерфейсных полей. На рис. 1 (см. вторую сторону обложки) представлено изображение интерфейсного окна для просмотра и редактирования с обозначением интерфейсных полей.

В интерфейсном поле "Файлы" выбираются имена файлов сигналов для просмотра и редактирования. Программный комплекс может работать с одним или одновременно с двумя файлами сигналов. В указанном поле может быть осуществлен выбор сигналов в следующих трех вариантах: выбор файла № 1; выбор файла № 2; выбор одновременно файлов № 1 и № 2. Случай выбора файлов № 1 или № 2 соответствует задачам цифровой обработки сигналов типа вычисления разностей фаз.

В интерфейсном поле "Параметры файла" для выбранных файлов № 1 и № 2 показываются число строк в матрицах сигналов, равное числу сигналов, записанных в файлах, и числу столбцов в матрицах сигналов, равное числу точек N_{j1} , N_{j2} в сигналах.

В интерфейсном поле "Число сигналов" для случая работы с одним файлом назначается число сигналов (1 или 2), равное числу строк матрицы сигнала, предполагаемых для цифровой обработки.

В интерфейсном поле "Номер сигнала" для выбранного файла с № 1 или № 2 проводится выбор номеров сигналов (номеров строк в соответствующих матрицах) для последующего просмотра и редактирования.

Этап № 0 включает случаи принятия решения по обработке одного и двух сигналов.

Случай одного сигнала. В интерфейсном поле со слайдерами выполняется редактирование выбранного сигнала — назначение граничных точек N_1 и N_2 , $0 \leq N_1 \leq N_2 \leq N_f - 1$, N_f — число точек в сигнале. В двух интерфейсных графических полях показываются графические изображения сигнала в двух вариантах — с длиной, соответствующей исходному сигналу с числом точек N_f и отредактированному сигналу с сокращенной длиной от точки N_1 до точки N_2 , $0 \leq N_1 \leq N_2 \leq N_f - 1$.

Случай двух сигналов. В интерфейсном поле со слайдерами выполняется редактирование выбранных сигналов — назначение для каждого из сигналов одинаковых граничных точек N_1 и N_2 , $0 \leq N_1 \leq N_2 \leq N_{f0} - 1$, $N_{f0} = \min(N_{f1}, N_{f2})$. В четырех интерфейсных графических полях с названиями файлов сигналов и их номерами показываются изображения исходных сигналов № 1 и № 2 с длинами N_{f0} и изображения отредактированных сигналов с длинами от точек N_1 до N_2 , $0 \leq N_1 \leq N_2 \leq N_{f0} - 1$.

В результате этапа № 0 в дисковую память записываются файлы сигналов № 1 или № 2, № 1 и № 2 с исходными и сокращенными длинами.

Этап локальной аппроксимации

На этапе № 1 решается задача локальной аппроксимации наблюдений нестационарных сигналов, которые представляются функциональными моделями $x_M(p(Ti), Ti)$, где $p(Ti)$ — векторные параметрические функции,

оценивание которых требуется провести в результате СВАН. При этом для сигнала с дискретными точками $i = 0, 1, \dots, N_f - 1$ вводятся локальные интервалы, которым соответствуют точки, удовлетворяющие неравенствам $N(j - 1) \leq i \leq N_j - 1$, j — номер локального интервала, $j = 1, \dots, m$, m — число локальных интервалов. На основе локальных моделей сигналов $x_M(f(\beta_j, Ti), Ti)$ и локальных наблюдений $y_j(Ti)$ формируются локальные функционалы вида

$$S(\beta_j, y_j) = \sum_{i=N(j-1)}^{i=N_j-1} (y_j(Ti) - x_M(f(\beta_j, Ti), Ti))^2.$$

Решается последовательность задач локальной аппроксимации и нахождения локальных моделей параметрических функций

$$\beta_j^{\circ} = \arg\{\min_{\beta_j} S(\beta_j, y_j)\}, p_{1j}^{\circ}(Ti) = f(\beta_j^{\circ}, Ti), Ti),$$

$$p_{1j}^{\circ}(Ti) = 0, \text{ для } i < N(j - 1), i > N_j - 1.$$

Аппроксимационная модель, соответствующая этапу № 1, для параметрической функции представляется в виде суммы локальных моделей

$$p_1^{\circ}(Ti) = \sum_{j=1}^m p_{1j}^{\circ}(Ti).$$

Этап № 1 состоит из трех подэтапов. На подэтапе 1.1 назначаются системы подходящих локальных моделей для локальной аппроксимации сигнала, выбранного на этапе № 0. Целесообразно здесь привести различные варианты локальных моделей, которые используются в данном ПК для задач локальной аппроксимации нестационарных колебательных сигналов.

1. Узкополосные сигналы могут быть аппроксимированы последовательностями локальных модельных кусочно-синусоидальных функций с постоянными амплитудами и частотами вида

$$x_M(f(\beta_j, Ti), Ti) = a_j \cos \omega_j Ti + b_j \sin \omega_j Ti,$$

где $\beta_j^T = (a_j, b_j, \omega_j)$ и a_j, b_j — локальные амплитуды, ω_j — локальные частоты, $j = 1, \dots, m$. В этом случае локальные модели для амплитудных и частотных параметров представляются последовательностями кусочно-постоянных функций $f^T(\beta_j, Ti) = (f_1(\beta_j, Ti), f_2(\beta_j, Ti))$, $f_1(\beta_j, Ti) = (a_j^2 + b_j^2)^{1/2}$, $f_2(\beta_j, Ti) = \omega_j$, $f(\beta_j, Ti) = 0$ для $i < N(j - 1)$, $i > N_j - 1$.

2. Узкополосные сигналы с медленными аддитивными трендами могут быть аппроксимированы последовательностями локальных моделей в виде сумм кусочно-синусоидальных функций и кусочно-линейных функций, $\beta_j^T = (a_j, b_j, \omega_j, c_{1j}, c_{2j})$ — вектор параметров локальной модели,

$$x_M(f(\beta_j, Ti), Ti) = a_j \cos \omega_j Ti + b_j \sin \omega_j Ti + c_{1j} + c_{2j} Ti.$$

3. Узкополосные сигналы с частотной модуляцией могут быть аппроксимированы последовательностями локальных модельных кусочно-синусоидальных функций с постоянными амплитудами и линейно изменяющимися частотами, $\beta_j^T = (a_{1j}, a_{2j}, \omega_j, \vartheta_j, \varphi_j)$ — вектор параметров локальной модели,

$$x_M(f(\beta_j, Ti), Ti) = a_j \cos(\omega_j Ti + \vartheta_j(Ti)^2/2) + b_j \sin(\omega_j Ti + \vartheta_j(Ti)^2/2).$$

4. Одночастотные сигналы со значительными амплитудно-частотными модуляциями на ограниченном интервале могут быть аппроксимированы последовательностями кусочно-синусоидальных функций с линейно изменяющимися амплитудами и частотами, $\beta_j^T = (a_{1j}, a_{2j}, \omega_j, \vartheta_j, \varphi_j)$ — вектор параметров локальной модели,

$$x_M(f, \beta_j, Ti), Ti) = (a_{1j} + a_{2j}Ti)\cos(\omega_j Ti + \vartheta_j Ti)^2/2 + \varphi_j.$$

5. Нестационарные многочастотные сигналы могут быть аппроксимированы локальными полигармоническими моделями с постоянными амплитудами и частотами, $\beta_j^T = (a_{1j}, \dots, a_{L_{0j}}, b_{1j}, \dots, b_{L_{0j}}, \omega_{1j}, \dots, \omega_{L_{0j}})$ — вектор параметров локальной модели,

$$x_M(f, \beta_j, Ti), Ti) = \sum_{l=1}^{L_0} (a_{lj}\cos\omega_{lj}Ti + b_{lj}\sin\omega_{lj}Ti).$$

Перечень вариантов локальных моделей 1—5 может быть расширен на основе рассмотрения особенностей нестационарных колебательных сигналов для различных предметных областей.

На **подэтапе 1.2** на основном интервале наблюдения с N_f точками назначается число локальных интервалов m . Число точек N на локальном интервале определяется соотношением $N = \text{ent}(N_f/m)$. При этом обработке подвергается сигнал с $N_f' = Nm$ точек.

На **подэтапе 1.3** проводится настройка параметров алгоритмов локальной аппроксимации, необходимая для формирования поисковых процедур. Так, алгоритм локальной аппроксимации на основе локальных модельных кусочно-синусоидальных функций с постоянными амплитудами и частотами требует задания начальной и конечной частот диапазона поиска по частоте f_{\min}, f_{\max} и число точек частотного поиска L_f .

Этап сплайновой аппроксимации

На этапе № 2 последовательности локальных оценок параметрических функций $p_1^\circ(Ti)$ сглаживаются сплайновыми функциями для повышения точности оценивания. Для этого среди точек $i = 0, 1, \dots, N_f - 1$ вводятся точки стыковки сплайновых интервалов $i_k, i_0 = 0, i_n = N_f - 1$ и сплайновые интервалы $i_{k-1} \leq i \leq i_k, k = 1, \dots, n$. Определяются сплайновые функции $g(\alpha_k, Ti)$ для $i_{k-1} \leq i \leq i_k, g(\alpha_k, Ti) = 0$ для $i < i_{k-1}, i > i_k, \alpha_k$ — векторы. В ПК SPECTRANS в качестве сплайновых функций используются взвешенные суммы дискретных ортогональных полиномов $q_r(Ti)$ на сплайновых интервалах, r — порядок полинома, $r = 0, 1, \dots, L_0$,

$$g(\alpha_k, Ti) = \sum_{r=0}^{L_0} \alpha_{kr} q_r(Ti).$$

Сплайновые модельные функции представлены следующим образом:

$$g(\alpha, Ti) = \sum_{k=1}^n g(\alpha_k, Ti), \alpha^T = (\alpha_1^T, \alpha_2^T, \dots, \alpha_n^T).$$

Условия гладкости сплайновых модельных функций в виде равенств для производных сплайновых функций

в точках стыковки $\tau_k, s = 0, 1, \dots, s_0, s$ — порядок производных, представляются с помощью функции $h^{(s)}(\alpha)$

$$h^{(s)T}(\alpha) = (g^{(s)}(\alpha_1, i_1) - g^{(s)}(\alpha_2, i_1), \dots, g^{(s)}(\alpha_{n-1}, i_{n-1}) - g^{(s)}(\alpha_n, i_{n-1})) = 0.$$

Для оценки $p_1^\circ(Ti)$ и сплайновой функции $g(\alpha, Ti)$ вводится функционал

$$S(\alpha, p_1^\circ) = \sum_{i=0}^{N_f-1} (p_1^\circ(Ti) - g(\alpha, Ti))^2.$$

Векторы-параметры α принадлежат ограничивающему множеству A , образованному системой равенств $A = \{\alpha : (h^{(s)}(\alpha) = 0, s = 0, \dots, s_0)\}$. Построение сплайновой аппроксимационной модели и оценка параметрической функции на втором этапе базируется на решении задачи условной оптимизации

$$\alpha^\circ = \arg\{\min_{\alpha \in A} S(\alpha, \bar{p}_1^\circ)\}, p_2^\circ(Ti) = g(\alpha^\circ, Ti).$$

Искомая оценка $p_0(Ti)$ параметрической функции нестационарного сигнала определяется на основе второго этапа и служит в качестве результата СВАН:

$$p^\circ(Ti) = p_2^\circ(Ti).$$

На **подэтапе 2.1** задаются число сплайновых интервалов n и порядок сплайновых полиномов L_0 .

Примеры работы ПК SPECTRANS

Работа ПК SPECTRANS иллюстрируется на примерах цифровой обработки модельных сигналов.

Пример 1. СВАН одночастотных узкополосных зашумленных сигналов, реализующий оценивание их амплитудных и частотных параметрических функций.

На рис. 2 изображено интерфейсное окно для формирования модельных наблюдений узкополосных сигналов с обозначением интерфейсных полей.

Модельные наблюдения узкополосных сигналов $y_M(Ti)$ в дискретных точках $i = 0, 1, \dots, N_f - 1$ представляются соотношениями вида

$$y_M(Ti) = E_M(Ti)\cos\varphi_M(Ti) + w_M(i), \quad (1)$$

где модельные амплитудные и фазовые (частотные) функции $E_M(Ti), \varphi_M(Ti)$ и $f_M(Ti)$ определяются формулами

$$\begin{aligned} E_M(Ti) &= E_0(1 + \mu_a \cos(2\pi f_a Ti + \varphi_a)); \\ \varphi_M(Ti) &= 2\pi f_0 Ti + (\mu_f f_0 / f_m) \cos(2\pi f_m Ti + \varphi_m) + \varphi_0; \\ f_M(Ti) &= \dot{\varphi}_M(Ti) = f_0(1 - \mu_f \sin(2\pi f_m Ti + \varphi_m)). \end{aligned} \quad (2)$$

Модельные погрешности $w_M(i)$ представляются некоррелированными нормальными числами с нулевым математическим ожиданием и дисперсией σ^2 . Параметры для (1), (2) принимали значения: $E_0 = 1,0, \mu_a = 0,55, f_a = 0,14$ Гц; $\varphi_a = 0,0, f_0 = 5,87$ Гц, $\mu_f = 0,6, f_m = 0,13$ Гц, $\varphi_m = 2,5, \varphi_0 = 0,0, T = 0,0065$ с, $N_f = 1024, \sigma = 0,5$. Интервал наблюдения в данном случае составлял $t_f = N_f T = 6,656$ с. На рис. 3 представлена отдельная реализация модельного узкополосного сигнала со значительными амплитудными и частотными модуляциями.

После формирования модельного наблюдения проводится выбор подходящих локальных моделей. В дан-

ном случае для сигнала рис. 3 выбираются локальные кусочно-синусоидальные модельные функции.

Далее выполняется настройка алгоритма локальной аппроксимации. На рис. 4 представлено интерфейсное окно для настроек соответствующего алгоритма локальной аппроксимации.

Число локальных интервалов было принято равным $m = 16$, число точек на локальном интервале $N = 64$ и длительность локального интервала, равной $NT = 0,416$ с, на котором в среднем помешалось $NTf_0 = 2,44$ периодов исходного сигнала. Диапазон поиска по частоте определялся с помощью задания нижней и верхней частот $f_{\min} = 2$ Гц, $f_{\max} = 10$ Гц и числа точек поиска по частоте — $L_f = 100$.

После проведения этапа локальной аппроксимации осуществлялся переход к этапу сплайновой аппроксимации. С использованием соответствующего интерфейсного окна число сплайновых интервалов принималось равным $n = 8$, порядок сплайновых полиномов $L_0 = 2$.

Результаты СВАН для данного модельного сигнала-оценки переменных во времени модельных амплитудной и частотной функций представлены на рис. 5, а, б. Кривые 1 (штриховая линия) соответствуют модельной синусоидальной амплитудной $E_M(t)$ и частотной $f_M(t)$ функциям, кусочно-постоянные линии 2 представляют собой локальные оценки параметрической амплитудной $E_{M,1}^\circ(t)$ и частотной $f_{M,2}^\circ(t)$ функции, которые получены на этапе № 1, кривые 3 являются сплайновыми оценками параметрической амплитудной $E_{M,1}^\circ(t)$ и частотной $f_{M,2}^\circ(t)$ функций, полученных на этапе № 2.

Пример 2. СВАН двухчастотных зашумленных сигналов, реализующий оценивание амплитудных и частотных параметрических составляющих функций.

С помощью соответствующего интерфейсного окна были сформированы наблюдения двухчастотного сигнала. Модельная функция наблюдений для нестационарного двухчастотного сигнала $y_M(Ti)$ в точках $i = 0, 1, \dots, N_f - 1$ представлялась выражением

$$y_M(Ti) = \sum_{l=1}^L E_{Ml}(Ti) \cos(\varphi_{Ml}(Ti)) + w_{Ml}(i), \quad (3)$$

где L_0 — задаваемое число составляющих сигналов. Модельные амплитудные и фазовые (частотные) функции $E_{l,M}(Ti)$, $\varphi_{l,M}(Ti)$ и $f_{l,M}(Ti)$ для составляющих с номерами $l = 1, \dots, L$, $L = 2$ определялись формулами (4)

$$\begin{aligned} E_{Ml}(Ti) &= E_{0l}(1 + \mu_{al} \cos(2\pi f_{al} Ti + \varphi_{al})), \\ \varphi_{Ml}(Ti) &= 2\pi f_{0l} Ti + (\mu_{fl} f_{0l} / f_{ml}) \cos(2\pi f_{ml} Ti + \varphi_{ml}) + \varphi_{0l}, \\ f_{Ml}(Ti) &= \dot{\varphi}_{l,M}(Ti) = \\ &= f_{0l} (1 - \mu_{fl}) \sin(2\pi f_{ml} Ti + \varphi_{ml}). \end{aligned} \quad (4)$$

Погрешности $w_M(i)$ представлялись в виде некоррелированных нормальных чисел с нулевым математическим ожиданием и дисперсией σ^2 . Параметры для выражений (3), (4) помещены в таблице.

Моделирование для (3, 4) проводилось при значениях $T = 0,0041$ с, $N_f = 8192$, $\sigma = 1,25$. Длительность интерва-

l	E_{0l}	μ_{al}	f_{al} , Гц	φ_{al}	f_{0l} , Гц	μ_{fl}	f_{ml} , Гц	φ_{ml}	φ_{0l}
1	7,0	0,3	0,026	0,0	50,2	0,005	0,03	0,0	0,4
2	2,5	0,42	0,033	1,0	49,2	0,012	0,03	0,5	3,0

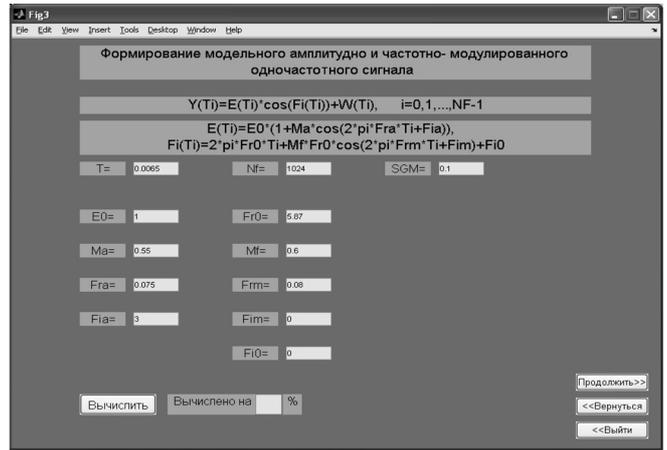


Рис. 2. Интерфейсное окно для формирования модельных наблюдений

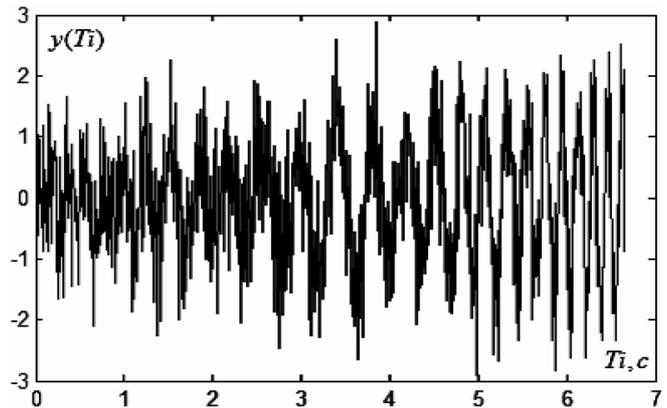


Рис. 3. Реализация модельного узкополосного сигнала

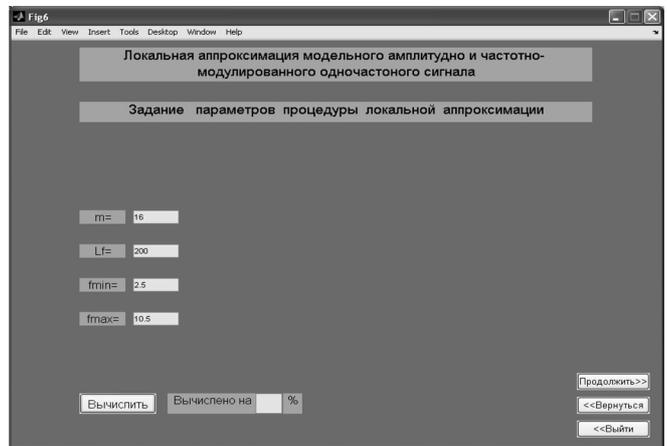


Рис. 4. Интерфейсное окно для формирования настроек алгоритма локальной аппроксимации

ла наблюдения составила $t_f = 33,58$ с, выбрано $m = 64$, длительность локальных интервалов $NT = 0,524$ с.

Этап локальной аппроксимации проводится с настройками, которые задаются в соответствующем интерфейсном окне. Этап сплайновой аппроксимации не проводится.

Результаты СВАН нестационарного двухчастотного сигнала-оценки модельных переменных во времени амплитудных и частотных функций для составляющих представлены на рис. 6, а, б.

Список литературы

1. Гетманов В. Г. Технология спектрально-временного анализа нестационарных сигналов на основе локальных и сплайновых аппроксимационных моделей // Труды 11-й международной конференции "Цифровая обработка сигналов и ее приложения" // М.: РНТО-РЭС им. А. С. Попова. 2009. Т. 1. С. 144—147.
2. Гетманов В. Г., Царева Е. Б. Применение программного комплекса Vibrans-Matlab для спектрального анализа нестационарных колебательных сигналов // Информационные технологии в проектировании и производстве. 2006. № 2. С. 47—53.
3. Гетманов В. Г., Царева Е. Б. Разработка программного комплекса Vibrans-Matlab / Научная сессия МИФИ-2005 // Сб. научн. трудов. Т. 12. Информатика и процессы управления. М.: МИФИ. 2005. С. 120—121.
4. Гетманов В. Г. Способ и устройство спектрально-временного анализа нестационарных сигналов. Патент на изобретение № 2365981, приоритет изобретения 12.02.2008.

5. Федосов В. П., Нестеренко А. К. Цифровая обработка сигналов в LabVIEW. М.: ДМК-Пресс. 2007. 472 с.
6. NI Spectral Measurements Toolkit / User Guide. National Instruments Corporation, 2008. — 35 p.
7. Кисин А. Анализатор Pulse 3560 от Bruel & Kjaer // Компоненты и технологии. 2004. № 4. С. 35—37.
8. Программное обеспечение анализатора сигналов Pulse 3560. URL: <http://www.bruel.ru>
9. FlexPro 8 Software. URL: <http://www.weisang.com>
10. DSP Intel Software VIEW // www.sigview.com
11. Auger F., Flandrine P., Lemoine O. The Time — Frequency Toolbox Matlab / Reference Guide. CNRS, 2005. 196 p.
12. Rice DSP Group [сайт]. URL: <http://www.dsp.rice.edu>
13. WinПОС. Пакет обработки сигналов [сайт]. URL: <http://www.nppmera.ru>
14. ZETLabstudio [сайт]. URL: <http://www.zetms.ru>
15. Шкелёв Е. И., Лунов С. Ю. Объектно-ориентированная система для спектрально-временного анализа сигналов в базовой полосе частот // Вестник ННГУ. Серия Радиофизика. 2004. Вып. 1. С. 55—61.
16. Гетманов В. Г., Орлов С. Е. Автоматизация просмотра и редактирования файлов экспериментальных сигналов для MatLab-2007 // Материал научной сессии МИФИ-2009. XII Московская международная телекоммуникационная конференция студентов и молодых ученых "Молодежь и наука". В 2-х ч. Ч. 2. М.: МИФИ. 2009. С. 151—153.

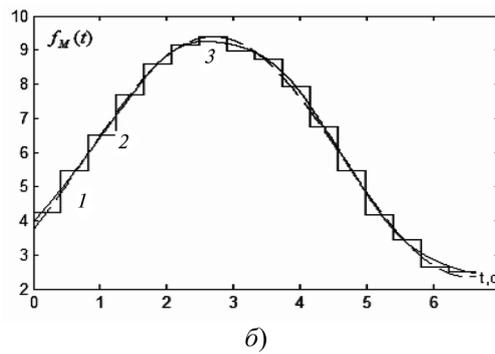
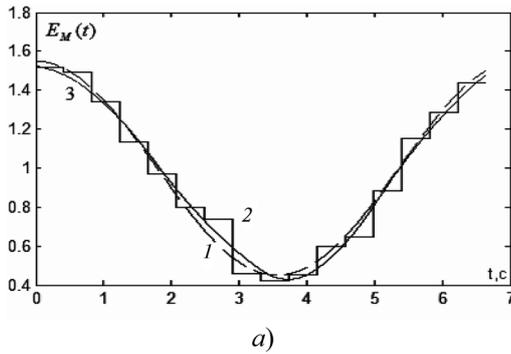


Рис. 5. Результаты СВАН одночастотного сигнала-оценки переменных во времени амплитудной (а) и частотной (б) функций: 1 — модельная функция; 2 — локальная оценка функции; 3 — сплайновая оценка функции

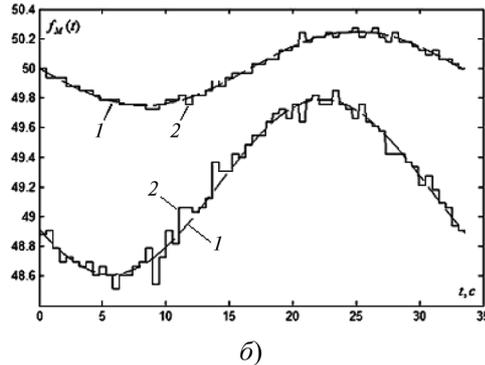
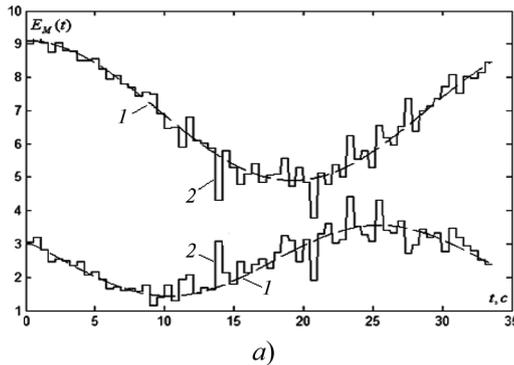


Рис. 6. Результаты СВАН нестационарного двухчастотного сигнала-оценки модельных переменных во времени амплитудных (а) и частотных (б) функций: 1 — модельные функции; 2 — локальные оценки функций

Штриховыми линиями 1 изображены модельные синусоидальные амплитудные $E_{M1}(t)$, $E_{M2}(t)$ и частотные $f_{M1}(t)$, $f_{M2}(t)$ функции составляющих модельного сигнала. Кусочно-постоянные линии 2 представляют собой локальные оценки $E_{M,1}^{\circ}(t)$, $E_{M,2}^{\circ}(t)$, $f_{M,1}^{\circ}(t)$, $f_{M,2}^{\circ}(t)$ для модельных амплитудных и частотных функций.

Заключение

Программный комплекс SPECTRANS может обеспечить решения большого числа задач цифровой обработки нестационарных колебательных сигналов для многих предметных областей.

Анализ результатов моделирования позволяет сделать вывод о потенциально удовлетворительной точности результатов СВАН — оценок параметрических функций нестационарных колебательных сигналов.

Программный комплекс SPECTRANS был использован для решения многих практических задач и опыт его применения показал, что данный комплекс является в достаточной степени удобным и эффективным инструментом цифровой обработки нестационарных колебательных сигналов.

Работа выполнена при поддержке РФФИ, проект 09-07-00086а.

УДК 002.53:002.55

Г. А. Тарнавский, д-р физ.-мат. наук,
Институт вычислительной математики
и математической геофизики СО РАН,
г. Новосибирск
E-mail: Gennady.Tarnavsky@gmail.com

Первый международный конгресс по "облачным" технологиям — Cloud Computing Congress ССС—2010

Приводится краткое сообщение о работе первого международного форума по проблемам развития современного направления информационных технологий — направления "Облачные вычисления".

Ключевые слова: информационные технологии, Интернет, облачные вычисления

22—23 июня 2010 года в Москве состоялся первый международный форум по "облачным" технологиям, Cloud Computing Congress СССР-2010 [1].

Интересно, что сам термин "облачные вычисления" еще буквально вчера был малоизвестен, хотя это словосочетание было впервые [2] использовано в 2001 году. Однако "идея становится силой, когда она овладевает массами". Эти слова классика политической мысли в полной мере могут быть отнесены к истории развития нового ИТ-тренда, направления Cloud Computing ("Облачные вычисления").

Идеология Cloud Computing. Направление Cloud Computing является быстроразвивающимся перспективным направлением современной информатики. Идеология "Облачных вычислений" заключается в переносе организации вычислений и обработки данных в существенной степени с персональных компьютеров на серверы Всемирной сети.

Концепция Cloud Computing основана на уверенности в том, что сеть Интернет в состоянии удовлетворить потребности пользователей в генерировании и обработке данных в широких диапазонах их запросов.

Технологии Cloud Computing. Направление "Облачные вычисления" содержит специализированный спектр технологий обработки и передачи данных, когда компьютерные ресурсы и мощности предоставляются пользователю как Интернет-сервисы. Пользователь имеет доступ к своей информации, которая постоянно хранится на Web-серверах, только как клиент во время Интернет-сеансов, с размещением этой информации (и результатов ее обработки) на персональных компьютерах, ноутбуках, нетбуках, смартфонах и т. п. К настоящему времени можно выделить несколько основных технологий (моделей) этого направления:

- инфраструктура как услуга (Infrastructure as a Service, IaaS);

- платформа как услуга (Platform as a Service, PaaS);
- данные как услуга (Data as a Service, DaaS);
- программное обеспечение как услуга (Software as a Service, SaaS);
- рабочее место как услуга (Workplace as a Service, WaaS);
- все как услуга (All as a Service, AaaS).

В рамках моделей IaaS, PaaS, DaaS, SaaS и WaaS заказчики платят не за владение программным продуктом как таковым, а за его аренду (модель AaaS), т. е. его использование через Web-интерфейс. Таким образом, в отличие от классической схемы лицензионной покупки программного продукта заказчику не требуется инвестировать большие средства для приобретения продукта и аппаратной платформы для его развертывания и обеспечения в дальнейшем работоспособности системы. Заказчик несет только сравнительно небольшие периодические затраты в виде абонентской платы, с возможностью ее прекращения или приостановления по истечении надобности в программном продукте, и ее возобновления при необходимости.

Облачные вычисления — новое явление, о котором говорят все, но о нем неизвестно почти ничего. Облачная концепция — современная модель, позволяющая представить информационные ресурсы как сервисы, и это может изменить отношение к ИТ-индустрии.

Цели конгресса. Cloud Computing Congress — актуальное масштабное событие, цель которого — ознакомить аудиторию с идеей облачных вычислений, представить многосторонний анализ облачных технологий, оценить перспективы развития и возможности использования в различных областях экономики, науки и инженерии.

Организатором конгресса являлась компания ITE — один из лидеров в сфере проведения конференций и выставок разнообразного профиля. Официальный партнер конгресса — корпорация Microsoft. Конгресс проводился при поддержке правительства Российской Федерации.

Ключевые темы конгресса:

- "Бизнес в облаках" — развитие Cloud Computing технологий в Европе и России и их значение для компаний;
- новая эра сотрудничества — взаимодействие между членами Cloud-сообщества;
- Cloud-экономика — финансовая сторона вопроса;
- анатомия Cloud Computing;
- безопасность и надежность "облачных" сервисов;
- *case-studies*: практические вопросы перехода на "облачную" инфраструктуру;
- виртуализация дата-центров — создание динамических ЦОД.

Программа конгресса была ориентирована, в основном, на бизнес-круги высшего (президенты и вице-президенты, генеральные и управляющие директора) и среднего звена (ИТ-директора, руководители проектов, бизнес-аналитики, топ-менеджеры по маркетингу) компаний со сферой деятельности (подробнее см. [1]):

- банки и страхования;
- инвестиционные и девелоперские компании;
- строительство и недвижимость;
- пищевая и фармацевтическая промышленность;

- оптовая и розничная торговля;
- нефтегазовый и энергетический сектор;
- металлургическая промышленность;
- транспорт и телекоммуникации;
- Интернет-провайдинг и IP-телефония;
- консалтинговые услуги и call-центры.

Пленарное заседание, с названием, специально подчеркивающим целенаправленность конгресса, — "Бизнес в облаках", ориентированное на то, чтобы дать VIP-участникам емкое и всесторонне представление о Cloud Computing, включало в себя следующие тренды:

- развитие технологий Cloud Computing в Европе и России;
- значение "облаков" для компаний;
- новый взгляд на создание и ведение бизнеса с помощью Cloud Computing;
- стратегия и модели перехода на "облачные вычисления";
- преимущества и недостатки "облачных" технологий для бизнес-процессов;
- советы экспертов по использованию "облачных" сервисов.

Тематика пленарного заседания дополнялась работой секций, где рассматривались разнообразные и многоплановые аспекты Cloud Computing. Перечислим темы заседаний каждой секции, поскольку это даст читателям данного пресс-релиза полностью почувствовать атмосферу первого международного форума.

Секция "Новая эра сотрудничества — взаимодействие между членами Cloud-сообщества":

- вендоры и интеграторы решений и технологий Cloud Computing;
- Cloud-провайдеры;
- юридические и консультационные компании;
- пользователи облачных вычислений.

Секция "Cloud-экономика — финансовая сторона вопроса":

- основные принципы экономики "облака";
- ценовые модели;
- расчет возврата инвестиций;
- сокращение затрат на внедрение "облачных" сервисов;
- сколько стоит "облако" построить?

Секция "Анатомия Cloud Computing":

- составные части "облака";
- IaaS (Infrastructure as a Service) — инфраструктура как услуга;
- PaaS (Platform as a Service) — платформы как услуга;
- SaaS (Software as a Service) — программное обеспечение как услуга.

Секция "Безопасность и надежность "облачных" сервисов":

- рекомендации и ключевые вопросы безопасности и надежности Cloud Computing;
- риски при переходе на "облачные" вычисления: политические и организационные, технические, юридические, экономические, а также риски, неспецифичные для этого направления.
- уязвимость технологий "облачных" вычислений;
- практика: что работает, а что нет?
- совместимость "облаков" различных провайдеров;
- выработка единых стандартов безопасности.

Секция "Практические вопросы перехода на "облачную" инфраструктуру":

- технические средства для построения "облаков", средства резервного копирования и дедубликации;
- решения для хранения "облаков";
- приступим-с!: секреты запуска Cloud Computing.

Секция "Виртуализация дата-центров — создание динамических ЦОД":

- трансформация дата-центров и их развитие;
- сервисное обслуживание и организационная подготовка;
- результативность и эффективность управления ЦОД;
- защита виртуальных дата-центров;
- стратегии перевода ЦОД на "облачные" технологии;
- виртуализация ЦОД для обеспечения непрерывности бизнеса.

На конгрессе состоялась **общая дискуссия "Вычисления в облаках — технология будущего?"** с обсуждением следующих вопросов:

- Готово ли наше общество к "облачным" технологиям?
- Единый взгляд на развитие: стремления и ожидания.
- Угрозы и препятствия на пути распространения технологий.
- Конкуренция на рынке: распределение сил.
- Роль и судьба "открытых продуктов".
- Прогнозы на текущий год.
- Q & A: ответ на любой вопрос!

Результаты конгресса. Двухдневная "информационная атака" позволила участникам конгресса, даже незнакомым с "облачными вычислениями":

— узнать, почему Cloud Computing столь важен для современного бизнеса;

— услышать мнения ключевых игроков на "облачном поле";

— увидеть, какой потенциал скрывает Cloud Computing, какие преимущества и опасности ожидают при переходе на "облако";

— обсудить дальнейшее распространение новой концепции в Европе и России;

— решить, наступит ли "эра облака";

— получить новый опыт и знания, которые помогут уменьшить операционные затраты на информационные технологии, снизить сложности информационных систем, а также повысить их гибкость, масштабируемость и доступность.

Подчеркнем, что в России уже существуют компании, оказывающие практически услуги по внедрению технологии Cloud Computing для бизнеса (например, услуга "Облачный хостинг", [3]). Однако следует констатировать, что проблемы создания "облачных" Web-ресурсов для компьютерного моделирования научных и инженерных задач на конгрессе не обсуждались, и "облачный" портал SciShop.ru [4] по-прежнему остается пионерным проектом этого направления.

Приводимая в настоящем сообщении достаточно полная и подробная информация ни в коей мере не является избыточной. Безусловно, заметка, содержащая лишь название форума, время и место его проведения, могла бы выполнить определенную функцию, однако в кратком сообщении невозможно очертить круг обсуждавшихся вопросов, понять и ощутить специфику атмосферы первого международного форума Cloud Computing Congress CCC-2010.

Список литературы

1. **Cloud Computing Congress CCC-2010.** URL: <http://Cloud-Congress.ru>
2. **Wikipedia.** Облачные вычисления. URL: http://ru.wikipedia.org/wiki/Облачные_вычисления
3. **Cloud Zone.** URL: <http://www.CloudZone.ru>
4. **Тарнавский Г. А., Алиев А. В., Анищик В. С., Тарнавский А. Г., Жибинов С. Б., Чесноков С. С.** Информационные технологии и проблемы создания Центра компьютерного моделирования в Интернете // Информационные технологии. 2009. № 8. С. 68—73.

CONTENTS

Karpenko A. P., Mukhlisullina D. T., Ovchinnikov V. A. Neural Network Approximation of Decision Maker's Utility Function in Multicriteria Optimization Problem	2
<p>We investigate an adaptive method for solving problem of multicriteria optimization based on neural network approximation of decision maker's utility function. We provide some results of method effectiveness research for solving two- and three-dimensional test problems. The method was implemented for solving two-dimensional optimization problem of combustion engine mechanical subsystem.</p>	
<p>Keywords: multicriteria optimization, direct adaptive method, approximation, neural networks, combustion engine</p>	
Halchenko V. Ya., Yakimov A. N., Ostapuschenko D. L. Global Optimum Search of Functions with Using of Multiagent Swarm Optimization Hybrid with Evolutional Composition Formation of Population	9
<p>A new hybrid of particle swarm optimization with genetic algorithm, allowed effective to optimize of high dimension multi-extremal functions is considered. It testing on objective functions with character features in behavior is realized.</p>	
<p>Keywords: global optimum, particle swarm optimization, genetic algorithm, hybrid, magnetic field source, synthesis</p>	
Mukhacheva E. A., Valiakmetova Yu. I., Khasanova E. I., Telitsky S. V. Projecting of Placing Orthogonal Objects on Polygons with Barriers	16
<p>While placing objects on polygons with barriers (building sites, vehicle platforms etc.) the problems of "through (guillotine) placing" and problems of "decomposition of free polygon area" on rectangle boxes emerge. For projecting the through (guillotine) placing we use some modification of "level" approach. The choice of the best placing is accomplished during the evolution with mutation combining. The problem of decomposition of free area is reduced to the problem of covering polygons with barriers. For solving the above problem a method of matrix representation of polygon is offered.</p>	
<p>Keywords: guillotine placing, evolutionary meta-heuristics, heuristics combining, polygon, covering of polygon</p>	
Zelepukhina V. A. Development of Virtual Labs of Mathematical Modelling for Internet by Division of Computing and Visual Tasks	22
<p>In article the way of development of virtual labs of mathematical modelling for the Internet grounded on the concept of division of computing process and visualization between the server and client sides is described. The offered technique is based on usage of two technologies: web tools of specialized mathematical packages for numeric computations and Java-technology for visualization of results of modelling.</p>	
<p>Keywords: virtual labs, visualization of data, mathematical modelling, computer simulation, web-technologies, Java, MATLAB</p>	
Kloss Yu. Yu., Martynov D. V., Tcheremissin F. G. The Development of the Methods of Computer Simulation and Analysis of a Knudsen Micropump	30
<p>Computer simulation of the gas movement in a Knudsen micropump on the basis of numerical decision of the Boltzmann equation is given. The transport operator of the kinetics equation is approximated by the finite-volume method. Tetrahedral meshes are used. The collision integral is calculated by projection method. Parametrical study of the pump is considered.</p>	
<p>Keywords: projection method, unstructured grids, finite-volume method, Boltzmann equation, Knudsen pump</p>	
Kalinina E. V., Retinskaya I. V., Serova D. A. Stochastic and Fuzzy Models of Communication Quality Estimation for IP Networks Monitoring	35
<p>The article deals with the continuous testing of the of IP-network's channels by the Ping command, which, basing on the open stream information, independently on data transmission technology or the type of communication channel, estimates its working capacity and quality of data transmission. The characteristics under study are the number of lost packages in sending echo-request series and the time of the return of the echo-response (a delay of a signal). There are offered stochastic models describing the behaviour and properties of the time of echo-response. There are constructed statistical criteria describing the working condition of channel and the quality of data transmission. The degree of the traffic quality (voice communication, video, the computer traffic) is defined by averaging and aggregating fuzzy sets operators, describing expert judgements of the system administrator about permissible distortions of the information. The efficiency of methods is show on the analysis of ISP INFORU.NET communications channels.</p>	
<p>Keywords: telecommunication, network, monitoring, communication quality, modeling, fuzzy set</p>	
Koryachko V. P., Shibanov V. A., Izhyanov Yu. L., Shibanov A. P. The Estimation of File Transmissions Time, Subject to Information Ageing	40
<p>The GERT-network model of transmission through communications channel with sliding window flow control is considered with the assumption of frame value loss with lapse of time. The frame number in file is specified as discrete random quantity. The expressions for file ageing probability estimation depended on time are offered.</p>	
<p>Keywords: GERT-network, file ageing, sliding window flow control, Multiprotocol label switching networks</p>	
Sotov L. S. The Model of the Device for Partitioning of Binary Set	46
<p>The model of the shaper of partitioning of binary sets is offered. The offered decision has effective hardware implementation and speed.</p>	
<p>Keywords: a partition of a set, a multistage interconnection network, a banyan switch</p>	

Davydov A. I., Shakhov V. G., Yadryshnikov I. B. Analysis of the IPv6 Protocol in Modern Operating Systems	52
The main disadvantage of the protocol IPv4 (and the main motive of the transition to IPv6) has been a limited number of IP-addresses, where there is a catastrophic shortage now. DHCP-server (issuing dynamic IP) and Network Address Translation (NAT) to some extent mitigate the problem, however, 32-bit field is reserved for developers of a protocol for IP-address provides 4 294 967 296 unique addresses, some of which are reserved for office needs, so the figures a bit overstated.	
Keywords: protocol, prefix, mask, security, session, server, client	
Gribova V. V., Cherkezishvili N. N. Ontology-Based Approach Evolution for Automated Development and Generation of User Interfaces Based on Dynamic Data	54
This paper describes a conception for automated development and generation of user interfaces based on dynamic data. This conception is evolution of ontology-based approach. User interfaces based on dynamic data allow users to form sets of input/output data during application run, as opposed to user interfaces based on static data for which input/output data must be specified in the interface development stage. User interfaces based on dynamic data are more flexible and provide new possibilities of application for software. However existing tools for user interface automatic development and generation are intended for user interfaces based on static data. This conception is aimed to eliminate existing tool limitations.	
Keywords: interface model, task model, automated generation of user interface, ontology	
Zuev A. S., Petrov Yu. I. Modernization of Context Menus of the Browser, Folders Tree and Desktop of Windows Operating System	59
The description of upgraded Windows OS browser, folders tree and desktop context menus is presented, results of comparison with similar interface elements are provided, ergonomics and efficiency of the given application are proved.	
Keywords: context menu, folders tree, browser, desktop, human-computer interaction, graphical interface, software ergonomics, graphical interfaces design, optimization of graphical interfaces, operating system window	
Bushmeleva K. I., Yatsenko E. A. Specialized Software Package for Measuring Gauges Calibration.	64
The description of specialized software product for calibration of measuring gauges is given: estimation of its capabilities, application field, information on structure of software package, user interface.	
Keywords: calibration, measuring gauges, adequacy estimation, linear dependence, software package, calibration models	
Kurskov S. Yu., Ershova N. Yu., Mursalimov O. A. Remote Access to Microcontrollers MCS-96 on Basis of Distributed Information Measurement and Control System for Scientific Research and Educational Process Support.	68
The present paper is devoted to remote access to MCS-96 microcontrollers on basis of the distributed information measurement and control system with remote access to informational and technical resources of the laboratory complex within the intranet/Internet networks.	
Keywords: remote access to labware, distributed information measurement and control system, equipment server, server of MCS-96 microcontrollers	
Guetmanov V. G., Orlov S. E. The Program Complex for the Spectral-Time Analysis of the Nonstationary Oscillation Signals	71
The program complex SPECTRANS for the spectral-time analysis of the nonstationary oscillation signals, based on the using of the local and spline models is considered. The information of the application end the features of the structure PC SPECTRANS are suggested. The stages of the operations of PC SPECTRANS, connected with the observation and editing, the local and spline approximation signals, are analysed. The examples of the realization of the spectral-time analysis of the nonstationary oscillation signals using the PC SPECTRANS are described.	
Keywords: spectral-time analysis, nonstationary oscillation signals, local and spline approximation	
Tarnavsky G. A. First International "Cloud Computing Congress CCC-2010"	77
The brief message about work of first international forum for development problems of modern information technologies trend, Cloud Computing, is given.	
Keywords: information technologies, Internet, cloud computing	

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *М. А. Зайцева*.

Корректор *Т. В. Пчёлкина*

Сдано в набор 09.08.2010. Подписано в печать 14.09.2010. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.

Усл. печ. л. 9,8. Уч.-изд. л. 11,33. Заказ 800. 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