

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

8(156)
2009

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Авдошин С. М., Савельева А. А. О новом подходе к проблеме анализа эффективности криптосистем 2
Перегида А. И., Тимашов Д. А. Математическая модель надежности систем защиты информации 10

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Ахремчик О. Л. Информационная база для автоматизированного проектирования схем систем управления технологическими объектами 17
Шумарин С. В., Талицкий Е. Н. Оптимизация виброзащиты электронной аппаратуры методом частотной отстройки. 21

ОПТИМИЗАЦИЯ

- Карпенко А. П., Овчинников В. А., Семенхин А. С. Программная система PRADIS//FRONT для построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельного генетического алгоритма 27
Казakov П. В. Кластерное расширение генетического алгоритма для решения многоэкстремальных задач оптимизации. 33
Антонов В. А. Информационная технология оптимизации новым численным методом и построения на его основе полиномиальных степенных трендов с самоопределяющимися показателями степени 39

МОДЕЛИРОВАНИЕ

- Потапов А. П., Петров И. Б. Моделирование высокоскоростных соударений методом сложенных частиц (SPH) 46
Воловиков В. В., Увайсов С. У. Модель тепломассопереноса в конструкциях радиоэлектронных устройств 50

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Швецов А. Н., Сергушичева М. А. Проектирование прикладных мультиагентных систем с использованием пакета DISIT 54
Арзамасцев А. А., Зенкова Н. А., Неудахин А. В. Технология построения медицинской экспертной системы на основе аппарата искусственных нейронных сетей 60

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Пекунов В. В. Процедуры с планированием повторного входа в языках высокого уровня при традиционном и параллельном программировании 63

WEB-ТЕХНОЛОГИИ

- 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 68
Пшинник К. В., Козицын А. С., Елкин Е. В. Выделение значимой информации из Web-ресурсов 74
Contents 78
Приложение. Шилов В. В. Игры, в которые играли автоматы. Исторические очерки.

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

АВДОШИН С. М.
АНТОНОВ Б. И.
БАТИЩЕВ Д. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ГОРБАТОВ В. А.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
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.
НАРИНЬЯНИ А. С.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.

Редакция:

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

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 004.056.5, 330.322.54

С. М. Авдошин, канд. техн. наук, проф.,
А. А. Савельева, аспирант,
Государственный университет —
Высшая школа экономики,
e-mail: savdoshin@hse.ru

О новом подходе к проблеме анализа эффективности криптосистем

Предлагается комплексный подход к оценке эффективности защиты информационных ресурсов предприятия, обеспечиваемой криптографическими средствами. В основе подхода лежит формализованный процесс анализа надежности криптосистемы в определенном контексте использования, математическая модель угроз безопасности защищаемых информационных ресурсов и автоматизированные средства для оценки стойкости криптосистем. Разработанный подход позволяет провести экономическое обоснование расходов организации на обеспечение информационной безопасности и сделать обоснованный выбор мер и средств криптографической защиты.

Ключевые слова: моделирование угроз, криптосистема, управление рисками, метод дисконтирования денежных потоков.

Введение

Средства криптографической защиты информации (СКЗИ) представляют собой средства вычислительной техники, осуществляющие криптографическое преобразование информации для обеспечения ее конфиденциальности и контроля целостности [1].

При оценке эффективности СКЗИ важнейшим критерием считается криптостойкость. Такой подход не учитывает других важных требований к криптосистемам. Обоснованный выбор методов криптографической защиты информации для конкретных информационных систем должен опираться не только на криптостойкость, но и на другие критерии эффективности, такие как (см. [2]):

- минимальный объем используемой ключевой информации;
- минимальная сложность реализации (в количестве машинных операций);
- стоимость;
- высокое быстроедействие.

Анализ публикаций в открытом доступе показал, что подходящие методики оценки эффективности криптографических систем до сих пор не разработаны (к аналогичным выводам пришли авторы работы [2]). Исключением является статья В. П. Иванова [3], в которой эффективность криптографических средств защиты предлагается оценивать с использованием математического аппарата теории массового обслуживания и теории катастроф на основе *вероятностно-временной группы показателей*, в числе которых:

- среднее время безопасного функционирования защищаемой системы;
- время безопасного функционирования защищаемой системы с вероятностью несанкционированного доступа (НСД) не выше заданной;
- экономическая эффективность созданной системы защиты информации.

Выбор показателей эффективности представляет интерес, однако методика имеет ряд критических недостатков, которые делают невозможным ее применение на практике для оценки современных СКЗИ. В первую очередь — это границы применимости: методика подходит только для оценки криптосистем, принадлежащих по классификации Ж. Brassara [4] к классу *криптосистем ограниченного использования*, стойкость которых основывается на сохранении в секрете алгоритмов зашифрования и расшифрования. Однако, согласно фундаментальному допущению Кирхгоффа [5], стойкость криптосистемы должна основываться не на секретности алгоритмов зашифрования и расшифрования, а на секретности некоторого значения, которое называется ее *ключом*. Все современные криптосистемы построены по этому принципу, и исследования их надежности всегда должны проводиться в предположении, что потенциальному противнику о криптосистеме известно все, за исключением используемого ключа.

Еще одним недостатком описанной в работе [3] методики является то, что она не учитывает зависимости эффективности криптосистемы от условий ее использования. Очевидно, эффективность одной и той же криптосистемы в разных контекстах может существенно различаться. Среда функционирования системы накладывает определенные ограничения на возможные сценарии атак. Простая модель сценария атаки представлена на рис. 1 [6]. Она включает три элемента, каждый из которых оказывает влияние на общую картину угроз. Типы атак, которым подвержена криптосистема, зависят от навыков, уровня дос-



Рис. 1. Модель сценария атаки

тупа, бюджета и других характеристик потенциальных злоумышленников [7]. Информация, подлежащая защите, определяет возможных злоумышленников, которые могут осуществлять попытки взлома в целях нарушения конфиденциальности, целостности или доступности (во избежание избыточности из модели исключен элемент "Защищаемые ресурсы", который задается неявно — через элемент "Злоумышленник").

Использование СКЗИ, обеспечивающих устойчивость к взлому ниже некоторой "фоновой" вероятности, является экономически неоправданным [8]. Например, если вероятность выхода компании из бизнеса равна 2^{-30} (менее чем один из миллиона), то есть ли смысл для защиты информации, которая может нанести компании ущерб, сопоставимый с кризисом рынка, использовать алгоритм, вероятность вскрытия которого за приемлемое время составляет 2^{-200} ?

Наиболее эффективным при выборе и оценке криптографической системы считается использование экспертных оценок и имитационное моделирование [2]. Разработаны методы и средства, позволяющие построить модели угроз и уязвимостей информационных систем и на основе анализа рисков получить количественную оценку соотношения потерь от угроз безопасности и затрат на создание системы защиты. В работах Н. Кукановой [9, 10] описаны методы и инструменты анализа и контроля информационных рисков: британский CRAMM [11] (компания *Insight Consulting*, подразделение *Siemens*), американский *Risk-Watch* [12] (компания *RiskWatch*) и российский ГРИФ [13] (компания *Digital Security*). Эти инструментальные средства полезны специалисту при проведении аудита систем обеспечения безопасности предприятия, однако они не учитывают специфики СКЗИ.

Для оценки эффективности СКЗИ необходимы методики, позволяющие принимать во внима-

ние взаимосвязь типов криптосистемы, потенциальных злоумышленников и возможных атак на защищаемые информационные ресурсы.

Постановка задачи

Задача состоит в разработке методики анализа эффективности криптографической системы с учетом того, каким угрозам защищаемая информация будет подвергаться со стороны злоумышленников.

В качестве исходных данных для проведения оценки эффективности криптографической защиты информационных ресурсов используются данные об особенностях реализации исследуемой криптосистемы и типах потенциальных злоумышленников. Анализ эффективности осуществляется на основе финансово-экономических показателей СКЗИ. Оценка эффективности создает предпосылку для определения соответствия криптосистемы потребностям организации, принятия решений и осуществления мер по отказу от СКЗИ, не обеспечивающих необходимый уровень защиты, и внедрения систем, достигших наилучших значений показателей.

Для решения поставленной задачи необходимо:

- формализовать процесс оценки эффективности криптографической защиты;
- разработать математическую модель угроз безопасности информационных ресурсов, защищенных с использованием криптографических средств;
- провести анализ существующих методов оценки СКЗИ с экономических позиций;
- выбрать финансово-экономические показатели, подходящие для экономической оценки инвестиций в СКЗИ.

Процесс оценки эффективности криптографической защиты

Процесс оценки эффективности криптографической защиты можно представить в виде схемы (рис. 2) [6].

Цель каждого этапа — получение ответа на вопрос:

- Этап 1: Какая криптосистема является объектом атаки?
- Этап 2: Кто будет атаковать эту криптосистему?
- Этап 3: Какие методы криптоанализа с наибольшей вероятностью будут использованы при осуществлении попыток взлома криптосистемы?
- Этап 4: Способна ли криптосистема противостоять таким атакам?



Рис. 2. Процесс оценки эффективности криптографической защиты

- Этап 5: Является ли использование исследуемой криптосистемы экономически выгодным в данном контексте?

Этапы 1—3 направлены на моделирование угроз, которым подвергаются информационные ресурсы, защищаемые с использованием исследуемой криптосистемы. Первый этап — определение объекта исследования. Здесь описываются конкретные характеристики криптосистемы. На втором этапе задаются параметры, определяющие тип потенциальных взломщиков криптосистемы. Между характеристиками криптосистемы и возможными типами атак существует связь. Аналогично, навыки, уровень доступа, бюджет в распоряжении злоумышленника определяют типы атак, которые он может предпринять [7]. Таким образом, при наличии формальных представлений исследуемой криптосистемы и потенциальных злоумышленников мы можем перейти к этапу 3 на рис. 2, т. е. определить типы атак, которым подвержена криптосистема, а также вероятность их реализации.

Этап 4 представляет собой анализ устойчивости криптосистемы к атакам, определенным на этапе 3. Цель этапа 4 — оценить риск нарушения безопасности информационных ресурсов, защищенных с использованием исследуемой криптосистемы. Необходимо обеспечить специалиста набором инструментальных средств для проведения исследований.

Наконец, этап 5 предполагает использование различных подходов к оценке экономической эф-

фективности инвестиций в СКЗИ на основании данных, полученных на этапах 1—4.

Математическая модель угроз безопасности информационных ресурсов

Задача оценки криптографической защищенности информационных ресурсов сводится к выделению подмножества атак, которым может подвергаться криптосистема в данном контексте использования, и определению устойчивости системы к этим атакам. Под устойчивостью системы будем понимать ее *криптостойкость*, т. е. способность противостоять атакам криптоаналитика [16].

Определение криптостойкости является научной задачей, для решения которой необходим набор инструментальных средств, позволяющих моделировать поведение злоумышленника при попытках взлома системы с использованием различных методов криптоанализа (см., например, [17]). Описание реализации программного комплекса, предназначенного для исследования стойкости асимметричных криптосистем с использованием математических методов, выходит за рамки данной статьи и приведено в [6].

Для определения эффективности криптосистемы имеет смысл проверять ее устойчивость не ко всем возможным атакам, а к тем, которые представляют для нее наибольшую угрозу. Состав множества потенциально опасных атак зависит от типа криптосистемы и условий использования криптосистемы.

Для выделения набора атак, которым подвержена криптосистема, построим математическую модель угроз безопасности защищаемых информационных ресурсов, основываясь на следующих предпосылках:

- один взломщик может предпринять атаки различного типа, а одна и та же атака может исходить от разных взломщиков;
- к одной и той же криптосистеме применимы атаки различного типа, а одна и та же атака позволяет взломать различные криптосистемы;
- злоумышленник с наибольшей вероятностью выберет ту атаку, которая обеспечит максимальный результат при фиксированных затратах, либо наименее затратный вариант из множества атак, приводящих к одинаковому результату.

Пусть $A \subseteq A_1 \times A_2 \times \dots \times A_8$ — множество параметрических моделей атак, где A_j ($j = \overline{1, 8}$) — множество значений j -го параметра модели атаки, определяющего тип атаки в соответствии с критериями разработанной классификации (см. [18]):

- A_1 — по доступу к открытому коду;
- A_2 — по контролю над процессом шифрования;
- A_3 — по исходу атаки;

A_4 — по объему необходимых ресурсов;
 A_5 — по степени применимости к различным шифрам;
 A_6 — по используемым средствам;
 A_7 — по последствиям атаки;
 A_8 — по возможности распараллеливания.

Каждая модель $\mathbf{a} \in A$ представляет собой вектор (a_1, a_2, \dots, a_8) , где $a_j \in A_j, j = \overline{1, 8}$. Заметим, что поскольку множества значений параметров модели атаки конечны, то мощность множества моделей атак $|A| \leq \prod_{j=1}^8 |A_j|$.

Пусть $B \subseteq B_1 \times B_2 \times \dots \times B_6$ — множество параметрических моделей злоумышленников, где B_j ($j = \overline{1, 6}$) — множество значений j -го параметра модели злоумышленника в соответствии с критериями описанной в [18] классификации:

B_1 — по технической оснащенности;
 B_2 — по конечной цели;
 B_3 — по доступу к шифрующим средствам;
 B_4 — по уровню подготовки;
 B_5 — по первичной информации о средстве шифрования;
 B_6 — по возможности кооперации.

Каждая модель $\mathbf{b} \in B$ задана в виде вектора (b_1, b_2, \dots, b_6) , где $b_j \in B_j, j = \overline{1, 6}, |B| \leq \prod_{j=1}^6 |B_j|$.

Пусть $C \subseteq C_1 \times C_2 \times \dots \times C_5$ — множество параметрических моделей криптосистем, где C_j ($j = \overline{1, 5}$) — множество значений j -го параметра модели криптосистемы в соответствии с многокритериальной классификацией, описанной в [18]:

C_1 — по доступности информации о криптоалгоритме;
 C_2 — по числу ключей;
 C_3 — по стойкости криптоалгоритма;
 C_4 — по используемым средствам;
 C_5 — по наличию сертификата;

Каждая модель $\mathbf{c} \in C$ представляет собой вектор (c_1, c_2, \dots, c_5) , где $c_j \in C_j, j = \overline{1, 5}$; мощность множества моделей криптосистем $|C| \leq \prod_{j=1}^5 |C_j|$.

При дальнейшем изложении для краткости слово "модель" применительно к модели атаки, модели злоумышленника и модели криптосистемы будем опускать.

С каждой атакой будем связывать значение риска, вычисляемое по общеизвестной формуле на основе двух факторов — вероятности происшествия и тяжести возможных последствий:

$$\text{Риск} = \text{Влияние} \cdot \text{Вероятность}.$$

Обозначим через $\mathfrak{R}: A \times B \times C \rightarrow [0; 1]$ функцию, задающую уровень риска, связанного с атакой $\mathbf{a} \in A$ в условиях, когда она может быть применена злоумышленником $\mathbf{b} \in B$ для взлома криптосистемы $\mathbf{c} \in C$.

Пусть $I: C \times A \rightarrow [0; 1]$ — функция влияния (от англ. *impact* — влияние, воздействие). Под влиянием мы будем понимать степень ущерба от применения атаки $\mathbf{a} \in A$ к криптосистеме $\mathbf{c} \in C$.

Пусть $P: B \times A \rightarrow [0; 1]$ — вероятность того, что злоумышленник $\mathbf{b} \in B$ предпримет атаку $\mathbf{a} \in A$, т. е. обладает ресурсами для ее осуществления и сочтет эту атаку целесообразной.

Тогда функция риска \mathfrak{R} выражается следующим образом:

$$\mathfrak{R}(\mathbf{a}, \mathbf{b}, \mathbf{c}) = I(\mathbf{c}, \mathbf{a}) \cdot P(\mathbf{b}, \mathbf{a}).$$

Определим функцию $I(\mathbf{c}, \mathbf{a})$. Для этого рассмотрим семейство функций $I_{gh}: C_g \times A_h \rightarrow \mathbb{R}_+$, $g = \overline{1, 5}, h = \overline{1, 8}$, где \mathbb{R}_+ — множество неотрицательных действительных чисел. Здесь функция I_{gh} задает уровень взаимного влияния параметра криптосистемы c_g и параметра атаки a_h :

- $I_{gh}(c, a) = 0$, если атака со значением параметра $a \in A_h$ не применима к криптосистеме со значением параметра $c \in C_g$;
- $0 < I_{gh}(c, a) < 1$, если значение параметра криптосистемы $c \in C_g$ снижает вероятность успешного применения атаки со значением параметра $a \in A_h$;
- $I_{gh}(c, a) = 1$, если значение параметра криптосистемы $c \in C_g$ не влияет на применимость атаки с параметром $a \in A_h$;
- $I_{gh}(c, a) > 1$, если значение параметра криптосистемы $c \in C_g$ указывает на то, что атака с параметром $a \in A_h$ применима для ее взлома.

Уровень взаимного влияния параметров криптосистемы и атаки определяется на основе экспертных оценок.

Обозначим через $\overline{I}_{gh}: C_g \times A_h \rightarrow [0; 1]$ нормированную функцию:

$$\overline{I}_{gh}(c, a) = \frac{I_{gh}(c, a)}{\sum_{\xi \in C_g} I_{gh}(\xi, a)}.$$

Тогда уровень ущерба от применения атаки $\mathbf{a} \in A$ к криптосистеме $\mathbf{c} \in C$ вычисляется по следующей формуле:

$$I(\mathbf{c}, \mathbf{a}) = \min_{h = \overline{1, 8}} \prod_{g = \overline{1, 5}} \overline{I}_{gh}(c_g, a_h),$$

где атака и криптосистема заданы параметрами (a_1, a_2, \dots, a_8) и (c_1, c_2, \dots, c_5) соответственно. Заметим, что уровень влияния всех параметров криптосистемы на применимость атаки с заданным значением h -го параметра в этой формуле вычисляется по мультипликативному критерию:

$\prod_{g=1}^5 \overline{I}_{gh}(c_g, a_h)$. Если значение хотя бы одного из параметров криптосистемы противоречит возможности применения атаки, то результатом оценки применимости атаки к криптосистеме будет нулевое значение, что соответствует нулевому уровню ущерба от атаки.

Определим функцию $P(\mathbf{b}, \mathbf{a})$. Для этого рассмотрим семейство функций $P_{th}: B_t \times A_h \rightarrow \mathbb{R}_+$, $t = \overline{1, 6}$, $h = \overline{1, 8}$. Здесь функция P_{th} задает уровень взаимного влияния параметра злоумышленника b_t и параметра атаки a_h :

- $P_{th}(b, a) = 0$, если злоумышленник со значением параметра $b \in B_t$ ни при каких обстоятельствах не будет использовать атаку со значением параметра $a \in A_h$;
- $0 < P_{th}(b, a) < 1$, если значение параметра злоумышленника $b \in B_t$ снижает вероятность использования атаки со значением параметра $a \in A_h$;
- $P_{th}(b, a) = 1$, если значение параметра злоумышленника $b \in B_t$ не влияет на вероятность использования атаки со значением параметра $a \in A_h$;
- $P_{th}(b, a) > 1$, если злоумышленник со значением параметра $b \in B_t$ с большой вероятностью будет использовать атаку со значением параметра $a \in A_h$.

Уровень взаимного влияния параметров злоумышленника и атаки также определяется экспертами.

Обозначим через $\overline{P}_{th}: B_t \times A_h \rightarrow [0; 1]$ нормированную функцию:

$$\overline{P}_{th}(b, a) = \frac{P_{th}(b, a)}{\sum_{\beta \in B_t} P_{th}(\beta, a)}.$$

Тогда вероятность того, что злоумышленник $\mathbf{b} \in B$ предпримет атаку $\mathbf{a} \in A$, вычислим по формуле

$$P(\mathbf{a}, \mathbf{b}) = \min_{h = \overline{1, 8}} \prod_{t = \overline{1, 6}} \overline{P}_{th}(b_t, a_h),$$

где атака и злоумышленник заданы параметрами (a_1, a_2, \dots, a_8) и (b_1, b_2, \dots, b_6) соответственно.

Таким образом, общая формула для определения уровня риска, связанного с применением атаки $\mathbf{a} \in A$ в условиях, когда эта атака может быть применена злоумышленником $\mathbf{b} \in B$ для взлома криптосистемы $\mathbf{c} \in C$, имеет вид

$$\mathfrak{R}(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \min_{h = \overline{1, 8}} \prod_{g = \overline{1, 5}} \overline{I}_{gh}(c_g, a_h) \times \times \min_{h = \overline{1, 8}} \prod_{t = \overline{1, 6}} \overline{P}_{th}(b_t, a_h).$$

Будем считать, что криптосистема $\mathbf{c} \in C$ подвержена атаке $\mathbf{a} \in A$ в условиях, когда ей угрожает злоумышленник $\mathbf{b} \in B$, если $\mathfrak{R}(\mathbf{a}, \mathbf{b}, \mathbf{c}) > \theta$, т. е. связанный с ней уровень риска превышает заданное пороговое значение θ , где $\theta \in [0; 1]$. Допустимый уровень риска θ является настраиваемым параметром модели угроз криптосистемы. Значение θ задается с учетом двух критериев:

- критичности защищаемых данных;
- временных и других ресурсов, доступных специалисту, который осуществляет аудит системы. В общем случае:
- криптосистема может включать несколько подсистем (например, генератор ключей и симметричный шифратор), к каждой из которых применим свой набор атак;
- на криптосистему может нападать несколько злоумышленников.

Множество атак, которым подвержена криптосистема, состоящая из подсистем $\mathbf{c} \in C'$ ($C' \subseteq C$), в условиях, когда ей угрожают злоумышленники $\mathbf{b} \in B'$ ($B' \subseteq B$), будем определять по формуле $\Lambda = \bigcup_{\mathbf{b} \in B'} \bigcup_{\mathbf{c} \in C'} \lambda(\mathbf{b}, \mathbf{c})$, где $\lambda(\mathbf{b}, \mathbf{c}) = \{\mathbf{a} \in A: \mathfrak{R}(\mathbf{a}, \mathbf{b}, \mathbf{c}) > \theta\}$ при заданном уровне риска. Для оценки защищенности криптосистемы необходимо с использованием инструментальных средств оценить ее способность противостоять атакам, входящим в множество Λ .

В описанной математической модели сделаны следующие допущения:

- не учитывается зависимость параметров атаки от сочетания параметров криптосистемы, хотя влияние каждого параметра принимается во внимание;
- не учитывается возможность совместных действий со стороны взломщиков различных типов, хотя можно задать модель нападения со стороны однородного коллектива злоумышленников.

Исправление модели с учетом указанных допущений привело бы к ее значительному усложнению. Вопрос о том, насколько эти допущения снижают точность моделирования угроз безопасности, подлежит дальнейшим исследованиям.

Инструментальные средства криптоанализа

После того, как выделен набор атак, представляющих наибольшую угрозу для защищаемых данных, необходимо оценить способность криптосистемы противостоять этим атакам.

Базой для получения таких оценок может служить статистика взлома и успешных атак на криптосистемы. Например, известно, что стартовавший в 1997 г. на сайте www.distributed.net проект "распределенного взлома" RC5-64 (блочного

шифра компании RSA, использующего 64-битный ключ) [19], в котором на добровольной основе приняли участие более 300 тыс. пользователей глобальной сети, был успешно завершён за пять лет (1757 дней) — за это время было перебрано 85 % всего пространства ключей. Однако такая информация, во-первых, не всегда доступна, а, во-вторых, со временем теряет актуальность, так как повышение производительности вычислительной техники и появление новых видов атак на шифры ведет к понижению стойкости известных криптографических алгоритмов. Для проверки надежности шифров, используемых в криптосистеме, специалисту необходим набор инструментальных средств, позволяющих осуществлять разработку, повышать эффективность, оценивать быстродействие и проводить сравнительный анализ алгоритмов криптоанализа. Рассмотрим преимущества и недостатки доступных решений, позволяющих решать задачи криптоанализа систем шифрования.

Математические пакеты Maple [20] и Mathematica [21] отличаются простотой кодирования алгоритмов и не имеют встроенных ограничений на разрядность операндов. Тем не менее, помимо платформенной зависимости они обладают критическим недостатком — низкой эффективностью теоретико-числовых операций.

Высокой эффективности можно добиться, используя встроенные средства низкоуровневого языка программирования для разработки функций, необходимых для исследования криптосистем. Однако важно отметить, что реализация примитивов для конструирования современных методов криптоанализа асимметричных шифров оперирует числами в длинной арифметике. Встроенные числовые типы языков C и C++ имеют ограниченную разрядность:

- long — 32 бит;
- long long — 64 бит;
- double: 53 бит — мантисса, 11 бит — экспонента;
- long double: в зависимости от реализации языка может быть определен как double (см. выше) либо как extended double: 64 бит — мантисса, 15 бит — экспонента [22].

В реализации языков на платформе .NET отсутствует тип `extended double`: он доступен только неявно при выполнении промежуточных вычислений (например, где умножение дает результат, выходящий за пределы диапазона значений `double`, но последующее деление возвращает промежуточный результат обратно в этот диапазон). Кроме того, существует встроенный 128-битный тип данных `decimal`, позволяющий представлять целые числа разрядностью до 96 бит (в соответствии с размером мантиссы), однако он

реализуется в режиме эмуляции, поскольку аппаратная поддержка этого типа на сегодняшний день отсутствует [23].

Java поддерживает возможность работы с длинными числами и обладает переносимостью, однако недостатком является низкая эффективность реализации.

Рассмотрим специализированные библиотеки функций для работы с длинной арифметикой и теоретико-числовыми задачами, находящиеся в открытом доступе: LIP, CLN, LiDIA, GMP, NTL.

Библиотека для работы с длинной арифметикой **LIP** (Long Integer Package) [24] является одной из первых таких библиотек. Она была разработана на языке ANSI C известным специалистом Арженом Ленстрой (Arjen K. Lenstra) и поддерживается Полом Лейлендом (Paul Leyland). При хорошей переносимости эта библиотека обладает низкой эффективностью. Кроме того, в ней отсутствует поддержка высокоуровневых теоретико-числовых алгоритмов.

Библиотека **CLN** (a Class Library for Numbers) [25] реализует элементарные арифметические, логические и трансцендентные функции. Авторами библиотеки являются Бруно Хейбл (Bruno Haible) и Ричард Крекел (Richard Kreckel). CLN содержит большой набор классов, реализованных на C++, в частности, классы для поддержки операций с целыми, рациональными и комплексными числами, числами с плавающей запятой, модулярной арифметики. Поскольку числовая библиотека задумывалась как универсальная, это привело к ее ограниченной применимости для решения узкоспециализированных задач.

Библиотека теоретико-числовых алгоритмов **LiDIA** [26], предложенная Томасом Папаниколау (Thomas Papanikolaou, Technical University of Darmstadt), написана на C++, поддерживает различные пакеты для работы с целыми числами (GMP, CLN, LIP) и характеризуется высокоэффективными реализациями типов данных с увеличенной точностью и алгоритмов с большой временной сложностью. Недостатком библиотеки LiDIA является невозможность сборки в операционных системах Windows, что очень существенно в связи с широким использованием продуктов Microsoft и необходимостью проверки их защищенности.

При разработке **GMP** (GNU Multiple Precision arithmetic library) [27] был сделан упор на скорость. Эффективность от использования библиотеки теоретико-числовых алгоритмов GMP растет при увеличении разрядности операндов. Часть функций реализована на языке C, часть — на ассемблере. Автором является Торбжорд Гранланд (Torbjord Granlund). Помимо несовместимости с платформой Windows, недостатком GMP является отсутствие алгоритмов формирования факторной

Сравнительный анализ программных решений для решения задач криптоанализа

Критерии оценки	Решение						
	Mathematica	LIP	CLN	LiDIA	GMP	NTL	КРИПТО
Эффективность вычислений	–	–	–	–	+	+	+
Возможность сборки в ОС Windows	+	+	+	–	–	+	+
Наличие алгоритмов работы с разреженными матрицами	–	–	–	+	+	–	+
Наличие алгоритмов создания факторной базы, решета и разложения на множители	–	–	–	+	–	–	+
Удобство пользовательского интерфейса	+	–	–	–	–	–	+

базы, разложения на множители и ряда других, необходимых для реализации современных методов криптоанализа.

Известная математическая библиотека **NTL** (a Library for doing Number Theory) [28] разработана Виктором Шаупом (Victor Shoup) для поддержки теоретико-числовых алгоритмов. Функции, реализованные на языке C++, характеризуются переносимостью. Библиотеку можно использовать совместно с GMP в целях повышения эффективности. NTL имеет большое число преимуществ по сравнению с рассмотренными аналогами (табл. 1), однако для решения поставленной задачи реализованных в библиотеке NTL алгоритмов недостаточно. Кроме того, для ее использования в криптоанализе специалист должен обладать квалификацией программиста.

В рамках данной работы нами был реализован программный комплекс "Инструментальные средства криптоанализа асимметричных шифров" (обозначение в табл. 1 — КРИПТО) [14, 15]. Программный комплекс состоит из библиотеки КОНСТРУКТОР, включающей необходимые примитивы для конструирования современных методов криптоанализа асимметричных шифров, и приложения АНАЛИТИК, имеющего графический интерфейс пользователя для доступа алгоритмам факторизации и дискретного логарифмирования с использованием функций библиотеки КОНСТРУКТОР. Для выполнения операций с длинными числами использована библиотека NTL. Выбор базовой библиотеки обусловлен ее функциональностью, скоростью (на ней были реализованы некоторые криптоатаки), компактностью (исходный код занимает чуть более 600 Кбайт) и переносимостью. Библиотека КОНСТРУКТОР написана на языке C++ и содержит компоненты, реализующие следующие основные функции:

- дискретное логарифмирование;
- факторизация целых чисел;
- тестирование чисел на простоту;
- вычисление коэффициентов Безу;
- выполнение базовых операций над матрицами в кольцах вычетов и конечных полях;

- решение систем линейных уравнений в кольцах вычетов и конечных полях;
- восстановление общего решения системы линейных уравнений в кольце вычетов по решениям систем линейных уравнений в конечных полях с использованием Китайской теоремы об остатках.

Расчет эффективности капитальных вложений в использование СКЗИ

Оценки вероятности взлома криптосистемы за определенный период позволяют определить сокращение риска НСД к данным от использования криптосистемы, например, за один год — на 95 %, за два года — на 70 %, за три года — на 35 %. При наличии достоверных оценок объема потерь от реализации угроз нарушения конфиденциальности, целостности или доступности защищаемых данных можно получить математические ожидания потерь и использовать их для определения эффективности криптосистемы с экономических позиций.

На основании анализа преимуществ и недостатков методом оценки эффективности инвестиций в средства обеспечения информационной безопасности (ИБ) (табл. 2) был сделан вывод, что оптимальным является *метод дисконтированных денежных потоков* [29], позволяющий получить наиболее полное представление о целесообразности капитальных вложений, хотя и требующий много времени и усилий на расчет экономических показателей.

Определим денежные потоки, связанные с использованием СКЗИ, за период t (где $t = 0, 1, 2, \dots, T$ — периоды, T — горизонт расчета).

Затраты $Cost_t$ на приобретение, установку и эксплуатацию СКЗИ могут быть определены очень точно, так как основной объем затрат составляет оплата труда персонала службы безопасности.

С защищаемой информацией связаны значения дохода $Profit_t$ и ущерба $Loss_t$ от НСД к защищаемой информации в течение указанного промежутка времени t . Пусть результаты оценки способности криптосистемы противостоять атакам,

Таблица 2

Сравнительный анализ методов оценки эффективности инвестиций в средства обеспечения ИБ

Методика оценки	Преимущества	Недостатки
Коэффициент возврата инвестиций	<ul style="list-style-type: none"> Показатель, понятный финансистам 	<ul style="list-style-type: none"> Отсутствие достоверных методов расчета в области ИТ «Статичный» показатель
Совокупная стоимость владения	<ul style="list-style-type: none"> Позволяет оценить целесообразность реализации проекта на основании оценки только затрат Предполагает оценку затрат на различных этапах всего жизненного цикла системы 	<ul style="list-style-type: none"> Не учитывает качество системы безопасности «Статичный» показатель Показатель, специфичный для ИТ
Дисконтированные показатели эффективности инвестиций	<ul style="list-style-type: none"> Показатель, понятный финансистам Учитывает зависимость потока денежных средств от времени Учитывает все потоки денежных средств, связанные с реализацией проекта 	<ul style="list-style-type: none"> Сложность расчета

представляющим для нее угрозу, с использованием инструментальных средств показали, что с вероятностью P_t в t -м периоде злоумышленник получит доступ к защищаемой информации. Затраты $Cost_t$ на приобретение, установку и эксплуатацию СКЗИ могут быть определены очень точно. Тогда математическое ожидание дохода R_t , связанного с использованием оцениваемой СКЗИ, вычисляется по формуле

$$R_t = -Cost_t + Profit_t \cdot (1 - P_t) - Loss_t \cdot P_t.$$

На основании этих данных о притоках и оттоках денежных средств вычисляются финансово-экономические показатели эффективности инвестиций в криптосистему и делаются выводы о ее соответствии потребностям организации.

Анализ разработанной методики

К достоинствам разработанного подхода к оценке эффективности криптосистем можно отнести следующее:

- возможность экономического обоснования расходов организации на обеспечение информационной безопасности и непрерывности бизнеса с использованием СКЗИ;
- возможность оценить не только риски, связанные с защищаемыми информационными ресурсами, но и выгоду, которую может принести внедрение СКЗИ;
- наличие формализованного пятиэтапного процесса оценки эффективности криптосистемы и математической модели угроз;

- наличие инструментальных средств, позволяющих аналитику оценить стойкость исследуемой криптосистемы к различным видам атак.

Можно выделить следующие направления дальнейшего развития темы:

- разработка программного обеспечения для автоматизации процесса построения модели угроз на основе:
 - данных об особенностях реализации исследуемой криптосистемы;
 - данных о потенциальных злоумышленниках;
- подготовка базы знаний, в том числе:
 - поиск и систематизация опубликованных данных об осуществленных криптоатаках;
 - привлечение экспертов, способных дать достоверные оценки зависимостей между параметрами элементов математической модели;
- поддержание базы знаний в актуальном состоянии, так как
 - с ростом вычислительных мощностей, изменением цен на аппаратные и программные средства и под влиянием других факторов уровень взаимного влияния параметров может меняться;
 - с появлением новых видов атак может возникнуть необходимость дополнения разработанных классификационных схем новыми критериями, что потребует введения новых зависимостей для соответствующих параметров моделей.

Кроме того, планируется расширить функциональность инструментальных средств криптоанализа путем добавления возможностей оценки стойкости алгоритмов генерации хеш-значений, псевдослучайных генераторов и симметричных шифров.

Список литературы

1. **Руководящий** документ Гостехкомиссии России "Защита от несанкционированного доступа к информации. Термины и определения". М.: ГТК РФ, 1992. — 13 с.
2. **Яковлев А. В., Безбогов А. А., Родин В. В., Шамкин В. Н.** Криптографическая защита информации: Учеб. пос. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2006. 140 с.
3. **Иванов В. П.** Математическая оценка защищенности информации от несанкционированного доступа // Специальная техника. 2004, N 1. С. 58—64.
4. **Brassard J.** Modern Cryptology. Springer-Verlag, Berlin — Heidelberg, 1988. 107 p. (Русский перевод: Брассар Ж. Современная криптология. Полимед, 1999. 176 с.).
5. **Kerckhoffs A.** La cryptographie militaire // Journal des sciences militaires. 1883. Vol. IX. P. 5—38, Jan. 1883 (P. 161—191, Feb. 1883).
6. **Savelieva A.** Formal methods and tools for evaluating cryptographic systems security // St. Petersburg, ISP RAS, In Proceedings of the Second Spring Young Researchers Colloquium on Software Engineering (SYRCoSE'2008), 2008. Vol 1. P. 33—36.
7. **Schneier B.** Modeling security threats // Dr. Dobb's Journal, December, 1999.
8. **Баричев С. Г.** Основной вопрос криптографии // Chief Information Officer — руководитель информационной службы. 2005, № 5 (37). С. 93—95.

9. **Куканова Н.** Методы и средства анализа рисков и управление ими в ИС // Byte/Россия. 2005. № 12. С. 69–73.

10. **Куканова Н.** Современные методы и средства анализа и управления рисками информационных систем компаний // http://www.dsec.ru/about/articles/ar_compare/2006.03.17

11. **CRAMM V Official website** // Siemens Enterprise Communications Limited 2006. Available at: www.cramm.com

12. **RiskWatch Official website** // Risk Watch, Inc. Available at: <http://www.riskwatch.com/>

13. **Digital Security: ГРИФ.** Система анализа и управления информационными рисками // Digital Security, 2002–2008. <http://www.dsec.ru/products/grif/>

14. **Авдошин С. М., Савельева А. А.** Инструментальные средства криптоанализа асимметричных шифров. М.: ВНИИЦ, 2008. № 50200800603.

15. **Авдошин С. М., Савельева А. А.** Инструментальные средства криптоанализа асимметричных шифров. Свидетельство о государственной регистрации в Реестре программ для ЭВМ № 2005612258 от 22.05.08.

16. **Ростовцев А. Г., Михайлова Н. В.** Методы криптоанализа классических шифров // 1998. <http://crypto.hotbox.ru/download/cryptoan.zip>

17. **Авдошин С. М., Савельева А. А.** Криптоанализ: современное состояние и перспективы развития // Информационные технологии, 2007. № 3. Приложение. 24 с.

18. **Авдошин С. М., Савельева А. А.** Проблемы оценки криптозащищенности информационных систем // "Новые информационные технологии". Тез. докл. XVI Международной студенческой школы-семинара. М.: МИЭМ, 2008. С. 15–29.

19. **www.distributed.net:** Проект RC5. <http://www.distributed.net/rc5/01.04.2008>

20. **Дьяконов В. П.** Maple 9.5/10 в математике, физике и образовании (+ CD-ROM). М.: Солон-Пресс. 2006. 720 с.

21. **Шмидский Я. К.** Mathematica 5. Самоучитель. М.: Диалектика, 2004. 592 с.

22. **American National Standards Institute and Institute of Electrical and Electronic Engineers** // IEEE standard for binary floating-point arithmetic. ANSI/IEEE Standard, Std 754-1985, New York, 1985.

23. **ECMA international.** Common Language Infrastructure (CLI) Partitions I to VI. ECMA-335, June 2006. <http://www.ecma-international.org/publications/standards/Ecma-335.htm>

24. **LIP.** <http://www.win.tue.nl/~klenstra/06.02.2007>

25. **CLN.** <http://www.ginac.de/CLN/06.02.2007>

26. **LiDIA.** <http://www.cdc.informatik.tu-darmstadt.de/TI/LiDIA/06.02.2007>

27. **GMP.** <http://gmplib.org/06.02.2007>

28. **NTL.** <http://www.shoup.net/ntl/06.02.2007>

29. **Старик Д. Э.** Расчеты эффективности инвестиционных проектов. М.: Финстатинформ, 2001.

УДК 519.873:004.732

А. И. Перегуда, д-р техн. наук, проф.,

Д. А. Тимашов, аспирант,

e-mail: i18n@yandex.ru,

Обнинский государственный технический университет атомной энергетики (ИАТЭ),

г. Обнинск

Математическая модель надежности систем защиты информации

Рассмотрена математическая модель надежности систем защиты информации, имеющих сложную структуру. Предложены соотношения для оценки таких параметров надежности, как среднее время наработки до первой аварии системы и функция распределения времени наработки до первой аварии системы.

Ключевые слова: надежность, защита информации, структура системы.

1. Постановка задачи

Использование различных автоматизированных систем, построенных на базе вычислительной техники, сделало актуальной проблему информационной безопасности. Для обеспечения защиты информации от несанкционированного доступа требуется обеспечить определенный уровень надежности таких систем, включая надежность их

аппаратных средств и программного обеспечения. Математические модели надежности систем защиты информации рассматривались в различных источниках, например в работе [1].

Рассмотрим вначале структуру предлагаемой модели. Будем считать, что информационная система состоит из *объекта защиты* и *системы безопасности*. Имеются различные виды атак в целях получения несанкционированного доступа к объекту защиты. Система безопасности призвана парировать эти атаки, тем самым обеспечивая защиту от несанкционированного доступа. Поскольку система безопасности не является абсолютно надежной, то необходимо рассматривать два вида ее отказов. Во-первых, это скрытые отказы, т. е. такие отказы системы безопасности, которые не могут быть обнаружены без проведения профилактических мероприятий. Во-вторых, это ложные отказы, т. е. ложные срабатывания системы безопасности. Таким образом, несанкционированный доступ к информации с использованием определенного класса атак возможен только тогда, когда атака приходится на период неработоспособности соответствующей подсистемы системы безопасности. Данное событие будем называть *аварией*.

Поскольку в такой сложной технической системе, какой является система безопасности, невозможно сразу обнаружить отказы отдельных подсистем, то для целей обнаружения неисправностей применяются различные диагностические процедуры, которые контролируют состояние

подсистем в заданные моменты времени. Таким образом, необходимо рассматривать процедуру контрольных профилактик, которые выполняются не мгновенно. Структура подсистем системы безопасности, как правило, достаточно сложна, поэтому необходимо также включить в рассмотрение структуры этих подсистем.

Поставленную задачу будем решать, используя методы, аналогичные тем, которые применяются в работах [2, 3].

2. Решение задачи

Введем следующие обозначения. Считаем, что имеется N типов атак, и при этом атаки разных видов статистически независимы. Предполагаем, что атаки на систему и их парирование можно описывать с помощью альтернирующего процесса восстановления [4]. То есть будем считать, что после отражения атаки, либо после ложного срабатывания системы безопасности случайный процесс как бы возвращается в исходное состояние с теми же статистическими характеристиками, что и в начальный момент времени. Также предполагаем, что совпадение двух атак является практически невероятным событием. Следует отметить, что хотя данные ограничения и являются, вообще говоря, достаточно строгими, но указанные предположения выполняются, например, в случае экспоненциально распределенных случайных интервалов времени до начала атаки и до ложного срабатывания системы безопасности (в силу свойства нестарения). Так что сфера применимости предлагаемой модели, по крайней мере, не уже, чем у моделей на основе полумарковских процессов. Пусть случайные времена до очередной атаки i -го вида χ_i на каждом цикле рассматриваемого случайного процесса независимы и одинаково распределены с функцией распределения $F_{\chi_i}(t)$. Соответственно, случайные времена отражения атаки i -го вида γ_i на каждом цикле рассматриваемого случайного процесса независимы и одинаково распределены с функцией распределения $F_{\gamma_i}(t)$. Все наработки i -й подсистемы системы безопасности до ложного отказа φ_i на каждом цикле рассматриваемого случайного процесса также независимы и одинаково распределены с функцией распределения $F_{\varphi_i}(t)$. И наконец, все времена восстановления после ложного срабатывания i -й подсистемы системы безопасности ψ_i на каждом цикле рассматриваемого случайного процесса независимы и одинаково распределены с функцией распределения $F_{\psi_i}(t)$.

Структуру подсистем системы безопасности по отношению к ложным отказам будем учитывать следующим образом. Пусть i -я подсистема системы безопасности состоит из M_i элементов. Счи-

таем, что все наработки j -го элемента i -й подсистемы системы безопасности до ложного отказа φ_i^j на каждом цикле рассматриваемого случайного процесса независимы и одинаково распределены с функцией распределения $F_{\varphi_i^j}(t)$. При этом восстановление осуществляется только после ложного отказа. Тогда время наработки подсистемы системы безопасности до ложного отказа можно записать в виде $\varphi_i = \varphi_i(\varphi_i^1, \dots, \varphi_i^{M_i})$. Вид функции будем определять с помощью следующей процедуры. Ложным сечением назовем такое множество элементов подсистемы системы безопасности, ложный отказ которых приводит к ложному отказу подсистемы. Минимальным ложным сечением назовем такое ложное сечение, ни одно из подмножеств элементов которого не является ложным сечением. Тогда время наработки до ложного отказа минимального ложного сечения определяется как максимум из всех времен наработки до ложного отказа его элементов. Для заданной структуры подсистемы системы безопасности необходимо выделить все минимальные ложные сечения. Время наработки до ложного отказа подсистемы в таком случае представляет собой минимум из времен наработок до ложного отказа всех имеющихся минимальных ложных сечений.

При построении модели примем еще несколько допущений. Во-первых, предполагаем, что скрытые и ложные отказы системы безопасности статистически независимы. То есть считаем, что скрытые отказы и восстановления подсистемы системы безопасности после скрытого отказа никак не влияют на его ложные отказы. Во-вторых, придется допустить, что ложные срабатывания возможны как во время профилактики системы безопасности, так и во время ее восстановления после скрытого отказа. Поскольку данные промежутки времени, как правило, незначительны по сравнению со временем исправного функционирования, то данное предположение не будет сколько-нибудь значительно исказить результат.

Рассмотрим теперь подробнее процесс функционирования системы безопасности. Как уже упоминалось ранее, i -я подсистема системы безопасности состоит из M_i элементов. Каждый из них может находиться только в одном из двух состояний — исправном и неисправном, а также и подсистема системы безопасности может находиться только в одном из этих двух состояний. Время безотказного функционирования ξ_i^j j -го элемента i -й подсистемы системы безопасности есть случайная величина. Будем считать, что восстановление подсистемы системы безопасности проводится только после того, как будет обнаружен от-

каз этой подсистемы. Время наработки подсистемы системы безопасности до скрытого отказа с учетом ее структуры можно записать в виде $\xi_i = \xi_i(\xi_i^1, \dots, \xi_i^{M_i})$. Вид этой функции будем определять следующим образом. Скрытым сечением назовем такое множество элементов подсистемы системы безопасности, скрытый отказ которых приводит к скрытому отказу подсистемы. Минимальным скрытым сечением назовем такое скрытое сечение, ни одно из подмножеств элементов которого не является скрытым сечением. Тогда время наработки скрытого сечения до скрытого отказа определяем как максимум из времен наработок до скрытого отказа его элементов. Исходя из структуры подсистемы системы безопасности, необходимо выделить все минимальные скрытые сечения. Тогда время наработки подсистемы системы безопасности до скрытого отказа определяется как минимум из времен наработок до скрытого отказа всех имеющихся минимальных скрытых сечений. После обнаружения отказа i -й подсистемы системы безопасности проводится восстановление ее работоспособности в течение времени η , также случайного. Предполагаем, что восстановление полное, т. е. все времена функционирования элемента после каждого восстановления независимы и одинаково распределены, а также что все времена восстановления подсистемы также независимы и одинаково распределены. Кроме того, рассмотрим процедуру периодических профилактик подсистем системы безопасности. Пусть промежуток времени между двумя профилактиками i -й подсистемы системы безопасности равен T_i , а длительность профилактики этой подсистемы равна θ_i . Таким образом, все время функционирования i -й подсистемы системы безопасности распадается на отдельные циклы, в каждом из которых i -я подсистема системы безопасности часть времени проводит в исправном состоянии (множество интервалов времени Q_i^+), а остальное время затрачивается на устранение неисправностей (множество интервалов времени Q_i^-).

При данных предположениях процесс функционирования автоматизированной системы можно описывать с помощью наложения альтернирующих процессов восстановления. Один из этих процессов описывает атаки на объект защиты, которые либо парируются системой безопасности, либо приводят к аварии, и ложные отказы подсистем системы безопасности. Прочие же процессы описывают скрытые отказы элементов подсистем системы безопасности, восстановле-

ние после этих отказов и периодическую профилактику системы безопасности.

Рассмотрим первый из упомянутых процессов. Представим длительность цикла регенерации этого процесса, на котором не было аварии, в следующем виде:

$$\begin{aligned} \tau = & (\chi_1 + \gamma_1) J_{\chi_1 < \chi_2 \wedge \dots \wedge \chi_N \wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})} + \dots + \\ & + (\chi_N + \gamma_N) J_{\chi_N < \chi_1 \wedge \dots \wedge \chi_{N-1} \wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})} + \\ & + (\varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) + \psi_1) \times \\ & \times J_{\varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) < \chi_1 \wedge \dots \wedge \chi_N \wedge \varphi_2(\varphi_2^1, \dots, \varphi_2^{M_2}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})} + \dots + \\ & + (\varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N}) + \psi_N) \times \\ & \times J_{\varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N}) < \chi_1 \wedge \dots \wedge \chi_N \wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \varphi_{N-1}(\varphi_{N-1}^1, \dots, \varphi_{N-1}^{M_{N-1}})}, \end{aligned}$$

где $J_A = J(w \in A) = \begin{cases} 1, & w \in A, \\ 0, & w \notin A. \end{cases}$ — индикатор события $w \in A$, $\varphi \wedge \chi = \min(\varphi, \chi)$ — обозначение для операции взятия минимума от двух случайных величин. Это означает, что осуществляется один из $2N$ вариантов развития процесса. Либо начинается одна из N возможных атак, которая затем парируется системой безопасности, либо происходит ложный отказ одной из N подсистем системы безопасности. Если же происходит авария, то время работы автоматизированной системы от начала цикла до аварии записывается так:

$$\tau' = \chi_1 \wedge \dots \wedge \chi_N.$$

Время работы автоматизированной системы до аварии представим в виде

$$\omega = \sum_{k=1}^{v-1} \tau_k + \tau'_v,$$

где v — номер цикла регенерации, на котором произошла авария. Функция распределения времени наработки ω автоматизированной системы до аварии тогда определяется следующим образом:

$$F_\omega(t) = P(\omega \leq t) = P\left(\sum_{k=1}^{v-1} \tau_k + \tau'_v \leq t\right).$$

Найдем вероятность того, что авария произошла на n -м цикле регенерации $P(v = n)$. Поскольку рассматриваемый процесс является альтернирующим процессом восстановления, то вероятность того, что авария произошла на n -м цикле регенерации, можно представить в виде $P(v = n) = r_1^{n-1} r_2$, где r_1 — вероятность того, что на цикле регенерации рассматриваемого процесса аварии не было, r_2 — вероятность того, что на цикле регенерации рассматриваемого процесса произошла авария.

Запишем теперь преобразование Лапласа — Стилтеса функции распределения $F_{\omega}(t)$:

$$\tilde{F}_{\omega}(s) = \int_0^{\infty} e^{-st} dF_{\omega}(t) = Me^{-s\omega}.$$

Воспользуемся методом условных распределений и выразим вероятность сложного события через условные вероятности этого события при соответствующих условиях. Если условия несовместны и представляют собой полную группу событий, то имеем

$$\begin{aligned} Me^{-s\omega} &= \sum_{n=1}^{\infty} M(e^{-s\omega} | v = n) P(v = n) = \\ &= \sum_{n=1}^{\infty} M\left(e^{-s\left(\sum_{k=1}^{v-1} \tau_k + \tau'_v\right)} | v = n\right) P(v = n) = \\ &= \sum_{n=1}^{\infty} M\left(e^{-s\left(\sum_{k=1}^{n-1} \tau_k + \tau'_n\right)}\right) P(v = n). \end{aligned}$$

Поскольку в силу приведенных выше условий все τ_k и τ' независимы и одинаково распределены, то с учетом того, что $\tilde{F}_{\tau}(s) = Me^{-s\tau}$ и $\tilde{F}_{\tau'}(s) = Me^{-s\tau'}$, получим

$$\begin{aligned} M(e^{-s\omega} | v = n) &= Me^{-s(n-1)\tau} Me^{-s\tau'} = \\ &= (\tilde{F}_{\tau}(s))^{n-1} \tilde{F}_{\tau'}(s). \end{aligned}$$

Следовательно,

$$\begin{aligned} \tilde{F}_{\omega}(s) &= \sum_{n=1}^{\infty} (\tilde{F}_{\tau}(s))^{n-1} \tilde{F}_{\tau'}(s) r_1^{n-1} r_2 = \\ &= \frac{\tilde{F}_{\tau'}(s) r_2}{1 - \tilde{F}_{\tau}(s) r_1} = \frac{Me^{-s\tau'} r_2}{1 - Me^{-s\tau} r_1}. \end{aligned}$$

Найти обратное преобразование для $\tilde{F}_{\omega}(s)$ в общем случае не представляется возможным, но можно вычислить среднее время наработки автоматизированной системы до аварии $M\omega$. Поскольку $M\omega = \left. \frac{d\tilde{F}_{\omega}(s)}{ds} \right|_{s=0}$, то, опуская несложные преобразования, запишем

$$M\omega = \frac{r_2}{1-r_1} M\tau' + \frac{r_1 r_2}{(1-r_1)^2} M\tau.$$

Учтем, что $r_1 + r_2 = 1$, тогда

$$M\omega = M\tau' + \frac{1-r_2}{r_2} M\tau.$$

Нужно теперь найти $M\tau$ и $M\tau'$. Вычислим вначале функцию распределения $F_{\tau'}(t) =$

$= P(\tau' \leq t) = P(\chi_1 \wedge \dots \wedge \chi_N \leq t)$. Запишем ее в следующем виде:

$$\begin{aligned} F_{\tau'}(t) &= P(\chi_1 \wedge \dots \wedge \chi_N \leq t) = \\ &= 1 - P(\chi_1 \wedge \dots \wedge \chi_N > t) = \\ &= 1 - P(\chi_1 > t) \dots P(\chi_N > t) = \\ &= 1 - \prod_{i=1}^N (1 - P(\chi_i \leq t)). \end{aligned}$$

Тогда

$$\begin{aligned} M\tau' &= \int_0^{\infty} (1 - F_{\tau'}(t)) dt = \\ &= \int_0^{\infty} \left(\prod_{i=1}^N (1 - F_{\chi_i}(t)) \right) dt. \end{aligned}$$

Получим теперь выражение для $M\tau$. Преобразуем его так:

$$\begin{aligned} M\tau &= M((\chi_1 + \gamma_1) J_{\chi_1 < \chi_2 \wedge \dots \wedge \chi_N \wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})} + \dots + \\ &+ (\chi_N + \gamma_N) J_{\chi_N < \chi_1 \wedge \dots \wedge \chi_{N-1} \wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})} + \\ &+ (\varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) + \psi_1) \times \\ &\times J_{\varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) < \chi_1 \wedge \dots \wedge \chi_N \wedge \varphi_2(\varphi_2^1, \dots, \varphi_2^{M_2}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})} + \dots + \\ &+ (\varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N}) + \psi_N) \times \\ &\times J_{\varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N}) < \chi_1 \wedge \dots \wedge \chi_N \wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \varphi_{N-1}(\varphi_{N-1}^1, \dots, \varphi_{N-1}^{M_{N-1}})} = \\ &= M(\chi_1 \wedge \dots \wedge \chi_N \wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})) + \\ &+ M\gamma_1 P(\chi_1 < \chi_2 \wedge \dots \wedge \chi_N \wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \\ &\wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})) + \dots + M\gamma_N P(\chi_N < \chi_1 \wedge \dots \wedge \chi_{N-1} \wedge \\ &\wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})) + \\ &+ M\psi_1 P(\varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) < \chi_1 \wedge \dots \wedge \chi_N \wedge \\ &\wedge \varphi_2(\varphi_2^1, \dots, \varphi_2^{M_2}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})) + \dots + \\ &+ M\psi_N P(\varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N}) < \chi_1 \wedge \dots \wedge \chi_N \wedge \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \\ &\wedge \varphi_{N-1}(\varphi_{N-1}^1, \dots, \varphi_{N-1}^{M_{N-1}})). \end{aligned}$$

Опуская некоторые несложные преобразования, запишем

$$\begin{aligned} M\tau &= \int_0^{\infty} \left(\prod_{i=1}^N (1 - F_{\chi_i}(t)) \times \right. \\ &\times \prod_{i=1}^N (1 - F_{\varphi_i(\varphi_i^1, \dots, \varphi_i^{M_i})}(t)) \Big) dt + \\ &+ \sum_{i=1}^N M\gamma_i \int_0^{\infty} \prod_{\substack{j=1 \\ j \neq i}}^N (1 - F_{\chi_j}(t)) \times \\ &\times \prod_{j=1}^N (1 - F_{\varphi_j(\varphi_j^1, \dots, \varphi_j^{M_j})}(t)) dF_{\chi_i}(t) + \end{aligned}$$

$$+ \sum_{i=1}^N M \Psi_i \int_0^{\infty} \prod_{j=1}^N (1 - F_{\chi_j}(t)) \times \\ \times \prod_{\substack{j=1 \\ j \neq i}}^N (1 - F_{\varphi_j(\varphi_j^1, \dots, \varphi_j^{M_j})}(t)) dF_{\varphi_i(\varphi_i^1, \dots, \varphi_i^{M_i})}(t).$$

Осталось найти r_2 — вероятность того, что на цикле регенерации рассматриваемого процесса произошла авария. Авария происходит тогда, когда атака на объект защиты началась до ложного отказа системы безопасности, и атака на объект защиты пришлась на неработоспособное состояние системы безопасности. Как можно заметить, эти условия независимы. Вероятность того, что атака на объект защиты произошла до ложного отказа системы безопасности, можно представить так:

$$P(\chi_1 \wedge \dots \wedge \chi_N \leq \varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \\ \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})) = \int_0^{\infty} (1 - \\ - F_{\varphi_1(\varphi_1^1, \dots, \varphi_1^{M_1}) \wedge \dots \wedge \varphi_N(\varphi_N^1, \dots, \varphi_N^{M_N})}(t)) dF_{\chi_1 \wedge \dots \wedge \chi_N}(t) = \\ = \int_0^{\infty} \prod_{i=1}^N (1 - F_{\varphi_i(\varphi_i^1, \dots, \varphi_i^{M_i})}(t)) dF_{\chi_1 \wedge \dots \wedge \chi_N}(t).$$

Вычислим теперь $F_{\chi_1 \wedge \dots \wedge \chi_N}(t)$:

$$F_{\chi_1 \wedge \dots \wedge \chi_N}(t) = 1 - \prod_{i=1}^N (1 - F_{\chi_i}(t)).$$

Далее запишем выражение для вероятности того, что атака на объект защиты пришлась на неработоспособное состояние системы безопасности. Воспользуемся формулой полной вероятности и получим

$$\sum_{i=1}^N q_i P(\chi_1 < \chi_1 \wedge \dots \wedge \chi_{i-1} \wedge \chi_{i+1} \wedge \dots \wedge \chi_N),$$

где q_i — вероятность того, что атака i -го типа пришлась на неработоспособное состояние i -й подсистемы системы безопасности. Преобразуем это выражение и получим

$$\sum_{i=1}^N q_i \int_0^{\infty} \prod_{\substack{j=1 \\ j \neq i}}^N (1 - F_{\chi_j}(t)) dF_{\chi_i}(t).$$

Далее необходимо вычислить q_i . Для этого будем рассматривать процесс функционирования подсистемы системы безопасности.

Вероятность аварии вследствие несрабатывания i -й подсистемы системы безопасности в момент времени t очевидно запишется в виде $P(t \in Q_i^-)$. То-

гда $q_i = \int_0^{\infty} P(t \in Q_i^-) dF_{\tau_i}(t)$. Найдем сначала $P_i(t) = P(t \in Q_i^+) = 1 - P(t \in Q_i^-)$. В соответствии с формулой полной вероятности имеем

$$P_i(t) = \int_0^{\infty} \int_0^{\infty} P(t \in Q_i^+ | \xi_i(\xi_i^1, \dots, \xi_i^{M_i}) = \\ = x, \eta_i = y) dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) dF_{\eta_i}(y),$$

где $\xi_i(\xi_i^1, \dots, \xi_i^{M_i})$ и η_i — случайные времена до первого отказа и первого восстановления соответственно.

Процесс функционирования периодически контролируемой i -й подсистемы системы безопасности распадается на циклы регенерации длительностью

$\tau_i(\xi_i(\xi_i^1, \dots, \xi_i^{M_i}), \eta_i) = \left(\left[\frac{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}{T_i + \theta_i} \right] + 1 \right) \times \\ \times (T_i + \theta_i) + \eta_i$, где в квадратных скобках — целая часть величины x . Отметим, что возможны два варианта: $\tau_i(\xi_i(\xi_i^1, \dots, \xi_i^{M_i}), \eta_i) > t$ и $\tau_i(\xi_i(\xi_i^1, \dots, \xi_i^{M_i}), \eta_i) \leq t$, в соответствии с которыми представим $P_i(t)$ так:

$$P_i(t) = \iint_{\tau_i(x,y) \leq t} P(t \in Q_i^+ | \xi_i(\xi_i^1, \dots, \xi_i^{M_i}) = \\ = x, \eta_i = y) dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) dF_{\eta_i}(y) + \\ + \iint_{(\tau_i(x,y) > t)} P(t \in Q_i^+ | \xi_i(\xi_i^1, \dots, \xi_i^{M_i}) = \\ = x, \eta_i = y) dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) dF_{\eta_i}(y) = I_1 + I_2.$$

Вычислим сначала I_2 . Условие $\tau_i(x, y) > t$ означает, что момент регенерации i -й подсистемы системы безопасности наступил после момента времени t . Тогда I_2 преобразуем так:

$$I_2 = \iint_{\tau_i(x,y) > t} P(t \in Q_i^+ | \xi_i(\xi_i^1, \dots, \xi_i^{M_i}) = \\ = x, \eta_i = y) dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) dF_{\eta_i}(y) = \\ = \iint_{\tau_i(x,y) > t} (J_{t \in [0, x]}) dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) dF_{\eta_i}(y) = \\ = \int_0^{\infty} \int_0^{\infty} (J_{t \in [0, x]}) \times \\ \times t - \left(\left[\frac{x}{T_i + \theta_i} \right] + 1 \right) (T_i + \theta_i) \\ \times dF_{\eta_i}(y) dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) =$$

$$= \int_0^{\infty} (J_{t \in [0, x]} \left(1 - F_{\eta_i} \left(t - \left(\left[\frac{x}{T_i + \theta_i} \right] + 1 \right) \times (T_i + \theta_i) \right) \right) dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x).$$

Нетрудно заметить, что если $J_{t \in [0, x]} \neq 0$, то получаем $t - \left(\left[\frac{x}{T_i + \theta_i} \right] + 1 \right) (T_i + \theta_i) < 0$, и тогда $F_{\eta_i} \left(t - \left(\left[\frac{x}{T_i + \theta_i} \right] + 1 \right) (T_i + \theta_i) \right) = 0$. Следовательно, этот интеграл можно переписать в следующем виде:

$$I_2 = \int_0^{\infty} (J_{t \in [0, x]} dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) = \int_t^{\infty} F_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) = \bar{F}_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(t).$$

Вычислим теперь I_1 . При рассмотрении ситуации $\tau_i(\xi_i(\xi_i^1, \dots, \xi_i^{M_i}), \eta_i) \leq t$ учтем, что в момент времени $\tau_i(\xi_i(\xi_i^1, \dots, \xi_i^{M_i}), \eta_i)$ имеет место момент регенерации, тогда запишем

$$I_1 = \iint_{\tau_i(x, y) \leq t} P(t \in Q_i^+ | \xi_i(\xi_i^1, \dots, \xi_i^{M_i})) = x, \eta_i = y) dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) dF_{\eta_i}(y) = \iint_{\tau_i(x, y) \leq t} P_i(t - \tau_i(x, y)) dF_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(x) dF_{\eta_i}(y) = \int_0^t P_i(t - z) dF_{\tau_i}(z).$$

Сложим теперь I_1 и I_2 и получим интегральное уравнение для $P_i(t)$:

$$P_i(t) = f_i(t) + \int_0^t P_i(t - y) dF_{\tau_i}(y),$$

где $f_i(t) = \bar{F}_{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}(t)$, $F_{\tau_i}(y) = P(\tau_i(\xi_i(\xi_i^1, \dots, \xi_i^{M_i}), \eta_i) \leq y) = P\left(\left(\left[\frac{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}{T_i + \theta_i}\right] + 1\right)(T_i + \theta_i) + \eta_i \leq y\right)$.

Так как полученное уравнение является уравнением в свертках, то будем искать его решение, используя преобразование Лапласа — Стильтеса [5]:

$$\tilde{P}_i(s) = \tilde{f}_i(s) + \tilde{P}_i(s) \tilde{F}_{\tau_i}(s),$$

где $\tilde{F}_{\xi}(s) = \int_0^{\infty} e^{-st} d\tilde{F}_{\xi}(t) = M e^{-s\xi}$. Тогда решение можно записать в виде

$$\tilde{P}_i(s) = \frac{\tilde{f}_i(s)}{1 - \tilde{F}_{\tau_i}(s)}.$$

Найти обратное преобразование в общем виде не представляется возможным, поэтому найдем асимптотическое соотношение для $P_i(t)$, применив Тауберovu теорему $\lim_{t \rightarrow \infty} P_i(t) = \lim_{s \rightarrow 0} \tilde{P}_i(s)$. Тогда получим

$$\lim_{s \rightarrow 0} \tilde{P}_i(s) = \lim_{s \rightarrow 0} \frac{\tilde{f}_i(s)}{1 - \tilde{F}_{\tau_i}(s)}.$$

Заметим, что и числитель и знаменатель стремятся к нулю. Применив правило Лопиталья, получим

$$\lim_{s \rightarrow 0} \tilde{P}_i(s) = \frac{M \xi_i(\xi_i^1, \dots, \xi_i^{M_i})}{M \tau_i},$$

тогда

$$P_i = \lim_{t \rightarrow \infty} P_i(t) = \frac{M \xi_i(\xi_i^1, \dots, \xi_i^{M_i})}{M \eta_i + (T_i + \theta_i) + (T_i + \theta_i) M \left[\frac{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}{T_i + \theta_i} \right]} = K_{\Gamma i}^{CB}.$$

Таким образом, вероятность того, что i -я подсистема системы безопасности находится в исправном состоянии при $t \rightarrow \infty$, совпадает с коэффициентом готовности этой подсистемы $K_{\Gamma i}^{CB}$. Следовательно, асимптотическое соотношение для вероятности аварии вследствие несрабатывания i -й подсистемы системы безопасности, очевидно, запишется в виде

$$q_i \approx 1 - K_{\Gamma i}^{CB} = K_{\text{нг}i}^{CB},$$

где $K_{\text{нг}i}^{CB}$ — коэффициент неготовности i -й подсистемы.

Теперь можем выписать вероятность того, что на цикле регенерации рассматриваемого процесса функционирования произошла авария r_2 :

$$r_2 = \int_0^{\infty} \prod_{i=1}^N (1 - F_{\varphi_i(\varphi_i^1, \dots, \varphi_i^{M_i})}(t)) dF_{\chi_1 \wedge \dots \wedge \chi_N}(t) \times \\ \times \sum_{i=1}^N q_i \int_0^{\infty} \prod_{\substack{j=1 \\ j \neq i}}^N (1 - F_{\chi_j}(t)) dF_{\chi_i}(t),$$

где $F_{\chi_1 \wedge \dots \wedge \chi_N}(t) = 1 - \prod_{i=1}^N (1 - F_{\chi_i}(t))$ и $q_i \approx 1 - \frac{M\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}{M\eta_i + (T_i + \theta_i) + (T_i + \theta_i)M\left[\frac{\xi_i(\xi_i^1, \dots, \xi_i^{M_i})}{T_i + \theta_i}\right]}$.

Получить точное выражение для вероятности аварии не представляется возможным, но можно записать экспоненциальную оценку для функции распределения времени до первой аварии. Для этого применим следующую теорему [6]:

$$\lim_{\frac{r_2\tau}{M\tau} \rightarrow 0} P\left\{\frac{r_2\omega}{M\tau} > x\right\} = e^{-x},$$

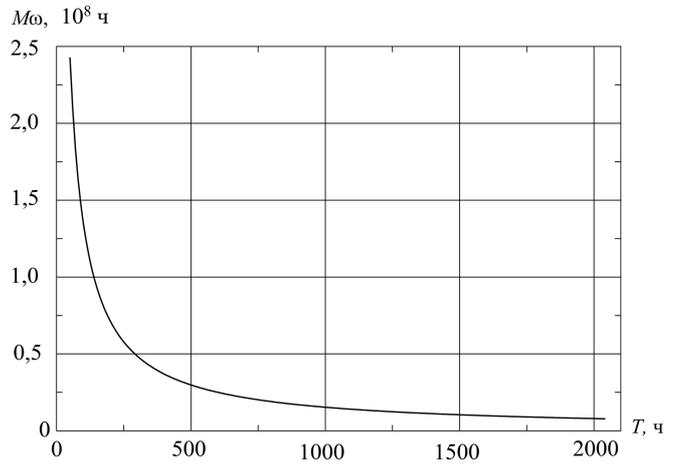
где r_2 — вероятность аварии на полуинтервале $[t_n, t_{n+1}]$, являющемся периодом регенерации; τ — длительность периода регенерации при условии, что на этом периоде аварии не произошло; ω — момент первой аварии. Здесь $\xi \xrightarrow{x} 0$ означает сходимость ξ к нулю по Хинчину. Кроме того, из-

вестно [6], что если $\frac{M\left(\frac{r_2\tau}{M\tau}\right)^2}{M\left(\frac{r_2\tau}{M\tau}\right)} = \frac{M\tau^2 r_2}{(M\tau)^2} \rightarrow 0$, то

$\frac{r_2\tau}{M\tau} \xrightarrow{x} 0$. Тогда можно записать

$$F_{\omega}(t) \approx 1 - e^{-\frac{r_2 t}{M\tau}} \text{ при условии } \frac{M\tau^2 r_2}{(M\tau)^2} \rightarrow 0, \\ \text{т. е. при } r_2 \ll 1.$$

Рассмотрим теперь пример, когда все случайные величины распределены экспоненциально. Тогда $F_{\chi_i}(t) = 1 - e^{-\lambda_{\chi_i} t}$, $F_{\gamma_i}(t) = 1 - e^{-\lambda_{\gamma_i} t}$, $F_{\varphi_i}(t) = 1 - e^{-\lambda_{\varphi_i} t}$, $F_{\psi_i}(t) = 1 - e^{-\lambda_{\psi_i} t}$, $F_{\xi_i}(t) = 1 - e^{-\lambda_{\xi_i} t}$, $F_{\eta_i}(t) = 1 - e^{-\lambda_{\eta_i} t}$. Будем считать, что имеются два вида атак и две подсистемы системы безопасно-



Зависимость среднего времени до первой аварии от периода профилактики системы безопасности

сти. При этом каждая подсистема системы безопасности состоит из двух параллельно соединенных элементов. Для простоты считаем, что все случайные величины распределены экспоненциально, обе подсистемы системы безопасности одинаковы, а также одинаковы параметры обоих видов атак. Примем следующие значения параметров: $\lambda_{\chi} = 10^{-6} \text{ ч}^{-1}$, $\lambda_{\gamma} = 1 \text{ ч}^{-1}$, $\lambda_{\varphi} = 10^{-5} \text{ ч}^{-1}$, $\lambda_{\psi} = 10^{-1} \text{ ч}^{-1}$, $\lambda_{\xi} = 10^{-4} \text{ ч}^{-1}$, $\lambda_{\eta} = 1 \text{ ч}^{-1}$, $\theta = 10 \text{ ч}$. Опуская несложные преобразования, запишем сразу результат:

$$M\left[\frac{\xi}{T+\theta}\right] = \frac{2e^{-\lambda_{\xi}(T+\theta)}}{1 - e^{-\lambda_{\xi}(T+\theta)}} - \frac{e^{-2\lambda_{\xi}(T+\theta)}}{1 - e^{-2\lambda_{\xi}(T+\theta)}};$$

$$q = 1 - \frac{\frac{3}{2} \frac{1}{\lambda_{\xi}}}{\frac{1}{\lambda_{\eta}} + (T+\theta) + (T+\theta)M\left[\frac{\xi}{T+\theta}\right]},$$

$$M\tau' = \frac{1}{2\lambda_{\chi}}; M\tau = \frac{1}{(\lambda_{\chi} + 2\lambda_{\varphi})} \left(\frac{1}{2} + \frac{\lambda_{\chi}}{\lambda_{\gamma}} + \frac{2\lambda_{\varphi}}{\lambda_{\psi}}\right);$$

$$r_2 = \frac{\lambda_{\chi}}{\lambda_{\chi} + 2\lambda_{\varphi}} q;$$

$$M\omega = M\tau' + \frac{1 - r_2}{r_2} M\tau.$$

Приведем теперь график зависимости среднего времени до первой аварии от периода профилактики системы безопасности. При построении графика отметим, что оптимальный период контрольных профилактик отсутствует.

Заключение

В данной работе удалось получить асимптотические оценки для показателей надежности систем

защиты информации. Предложенные оценки достаточно просто вычисляются и учитывают большое число различных параметров процесса функционирования автоматизированной системы.

Список литературы

1. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа. СПб.: Наука и техника, 2004. 384 с.
2. Перегуда А. И., Тимашов Д. А. Моделирование процесса функционирования АТК "ОЗ-СБ" с периодически контролиру-

руемой системой безопасности // Надежность. 2007. № 2. С. 38—48.

3. Перегуда А. И., Тимашов Д. А. Моделирование процесса функционирования АТК "ОЗ-СБ" с учетом аварий, вызванных обесточиванием // Надежность. 2008. № 1. С. 48—63.

4. Кокс Д. Р., Смит В. Л. Теория восстановления.: Пер. с англ. М.: Сов. радио, 1967. 299 с.

5. Беллиман Р., Кук К. Л. Дифференциально-разностные уравнения: Пер. с англ. М.: Мир, 1967. 548 с.

6. Барзилович Е. Ю., Беляев Ю. К., Каштанов В. А. и др. Вопросы математической теории надежности / Под ред. Б. В. Гнеденко. М.: Радио и связь, 1983. 376 с.

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

УДК [004.326 + 004.3].075.8

О. Л. Ахремчик, канд. техн. наук, доц.,
Тверской государственный
технический университет,
e-mail: axremchic@mail.ru

Информационная база для автоматизированного проектирования схем систем управления технологическими объектами

Рассматривается модель процесса создания схем систем управления технологическими объектами. Приводится иерархическая декомпозиция системы управления в виде теоретико-множественных описаний подсистем, цепей и элементов систем управления. Выделяются свойства сигналов входов-выходов элементов как свойства всех уровней иерархии, позволяющие осуществить автоматический синтез принципиальных схем.

Ключевые слова: система управления, свойства, структура, элемент.

Постановка и актуальность задачи

Задача интеграции систем автоматизированного проектирования (САПР) и автоматизированного управления технологическими процессами (АСУ ТП) является особо актуальной в настоящее время в связи с необходимостью непрерывной поддержки жизненного цикла сложных технических систем [1]. Использование моделей открытых систем заставляет проводить многоаспектное рассмотрение развития связки САПР — АСУ ТП [2].

К методологическим аспектам относятся:

- достижение возможностей концептуальных описаний АСУ ТП с необходимой степенью

детализации (преобразование концепции структуризации на высшем уровне абстракции по С. В. Емельянову);

- предоставление правил и средств автоматического синтеза системы по ее неполному описанию (преобразование концепции формализации с учетом алфавитов и семантики проблемной области в авторском понимании);
- обеспечение открытости и расширяемости концепций структуризации для разных классов систем управления.

К программно-технологическим аспектам относятся:

- организация условий для обеспечения машинной независимости продукта;
- выбор модели для представления данных;
- обеспечение коллективного и распределенного доступа к ресурсам;
- открытость для взаимодействия с другими приложениями;
- возможности реализации повторных запусков;
- визуализация и печать результатов в соответствии с требованиями нормативно-технической документации.

Значительный объем ручных операций при проектировании схем и алгоритмов функционирования систем управления технологическими объектами определяется недостаточно подробным рассмотрением первых двух выделенных методологических аспектов, особенно при генерации проектных решений. Это не позволяет осуществить синтез и анализ многообразия вариантов технической реализации выбранных алгоритмов управления технологическими объектами в сжатые сроки и с достаточной точностью расчета критериальных оценок. Результаты проводимых в данном направлении исследований показывают, что современные программные системы и комплексы используются для согласования решений [3, 4].

В области схмотехнического проектирования средств автоматизации и электронных приборов задача автоматического перехода от математического описания в виде системы логических уравнений к принципиальным электрическим схемам и печатным платам вполне успешно решается на основе использования цепочек САПР [5].

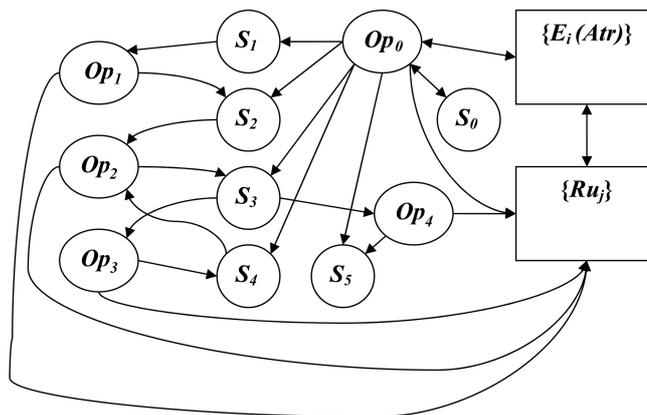
При проектировании принципиальных схем систем автоматизации и управления решение задачи связано с трудностями, определяемыми нарушением морфизмов при смене алфавита языка проектного описания. Применение агрегативно-декомпозиционной технологии позволяет добиться значительных успехов в развитии концепций структуризации и формализации в области автоматизированных систем [6]. Поэтому постановка задачи заключается в распространении данной технологии на область проектирования схем технической реализации АСУ ТП, что позволит создать модели, обеспечивающие ускорение сроков проектирования систем промышленной автоматизации за счет внедрения методов и средств автоматизированного проектирования на стадии перехода от обобщенной функциональной структуры системы управления к проектной схемной документации. Выделение правил преобразования структуры системы управления и детализация описаний классов элементов систем управления с определением базовых свойств позволяют осуществить связь верхнего уровня САПР с физическим уровнем АСУ ТП.

Модель процесса построения схемных описаний АСУ ТП

Схемные описания АСУ ТП отражают ее функциональную, техническую и топологическую структуры. Рассматриваемые результаты в первую очередь касаются описаний технической структуры в виде функциональной схемы в заданном элементном базисе, принципиальной электрической схемы, схемы соединений. Исходными данными для разработки схем являются:

- техническое задание;
- типовые решения;
- алгоритмы управления;
- вид и значения целевой функции и базовых ограничений как процесса проектирования, так и создаваемой системы управления;
- требования нормативной документации для представления проектных описаний;
- топологическая структура объекта управления;
- описания технических средств.

Учет специфики объекта осуществляется проектировщиком и заключается в выборе типа датчиков и исполнительных устройств, алгоритма управления, топологической структуры.



Модель процесса построения схемных описаний системы управления

Модель процесса построения схем систем управления технологическими объектами представляет собой граф с двумя типами вершин (см. рисунок). Вершины типа S_i соответствуют моделям технической структуры. Вершины типа Op_k соответствуют операторам преобразования моделей S_j . Оператор задается в виде

$$Op = \langle M, Alg \rangle,$$

где M — описание вектора признаков (Atr) рассматриваемой структуры и ее элементов $\{E_i(Atr)\}$; Alg — алгоритм преобразования структуры, использующий определенные правила Ru_j .

Среди моделей технической структуры выделяются классы: S_0 — модель системы, построенная по техническому заданию; S_1 — модель базовой структуры, определяющая порядок следования элементов и основные элементы; S_2 — модель технической структуры, соответствующая функциональной схеме автоматизации; S_3 — модель, соответствующая принципиальной схеме; S_4 — модель, соответствующая монтажной схеме (соединений); S_5 — модель программной конфигурации технических средств для реализации функций в соответствии с моделями S_1 — S_4 .

Работа операторов базируется на наборах правил:

$$\begin{aligned} Op_0 &\rightarrow Alg_0 \rightarrow \{Ru_0\}, Op_1 \rightarrow Alg_1 \rightarrow \{Ru_1, Ru_2\}, \\ Op_2 &\rightarrow Alg_2 \rightarrow \{Ru_3\}, Op_3 \rightarrow Alg_3 \rightarrow \{Ru_4\}, \\ Op_4 &\rightarrow Alg_4 \rightarrow \{Ru_5\}, \end{aligned}$$

где Ru_0 — правила формирования технического задания; Ru_1 — правила проверки области поиска элементов (базовыми свойствами для поиска являются функциональное назначение и свойства, определяемые спецификой объекта, принадлежность группам стоков или истоков в графовой модели структуры); Ru_2 — правила проверки ограничений (возможности выполнения операций,

непосредственной совместимости элементов; учет дополнительных ограничений); Ru_3 — правила преобразования функциональной структуры в принципиальную схему; Ru_4 — правила преобразования принципиальной схемы в схему соединений (учет топологической структуры, условий размещения технических элементов); Ru_5 — правила создания алгоритмической и программной конфигураций.

Иерархическая декомпозиция системы управления

На высшем уровне иерархии теоретико-множественное описание технической реализации системы управления представляется в виде [2]

$$Su = \langle C, F, S, Atr, H \rangle,$$

где Su — система управления; C — связи с окружающей средой; F — выполняемые функции; S — модель структуры; Atr — совокупность свойств; H — описание жизненного цикла системы.

Реализация набора функций осуществляется на основе выделения в составе системы управления контуров управления и сигнализации.

Для выполнения сравнительного анализа вариантов технических структур предлагается введение в состав информационной базы системных свойств [7]. В качестве данных свойств автором выделяются:

- способность к модернизации;
- однородность состава технических средств;
- удобство проектирования (изменение составляющих системы в процессе проектирования, наличие типовых проектных решений, наличие цепочек САПР для проектирования, наличие обучающих сред по проектированию).

Концептуальное представление модели технической структуры для информационной базы имеет вид

$$S = E \cup C, E = \{E_{HW} \cup E_{SW} \cup E_D(C_{ij})\}, \\ C_{ij} = \{C_i, C_j\}, \quad (1)$$

где S — модель технической структуры; E — набор элементов для ее построения; C — набор связей; E_{HW} , E_{SW} — модули аппаратного и программного обеспечения; E_D — модули проектного описания и технической документации; C_{ij} — набор связей между техническими элементами; C_i — связи информационные; C_j — связи энергетические (питания).

Включение модулей E_D предлагается автором и отражает рассмотрение элементов технической структуры S в системном единстве с функциональной и топологической структурами, выраженном через набор схемных описаний.

На втором уровне система управления рассматривается в виде набора подсистем, выделенных по функциональному назначению:

$$Su_i R_{cv} Sub_C \cup Sub_I \cup Sub_R \cup Sub_{DT} \cup \\ \cup Sub_{PS} \cup Sub_D,$$

где R_{cv} — отношение включения; Sub_C — подсистема измерения; Sub_I — подсистема контроля, сигнализации и индикации; Sub_R — подсистема регулирования и логического управления; Sub_{DT} — подсистема передачи данных; Sub_{PS} — подсистема энергопитания; Sub_D — подсистема аварийной защиты.

Набор функций в отличие от предложенных ранее [7] в части управляющих функций более подробно представлен отдельно функциями регулирования и логического управления, что очень важно при формировании описаний технической реализации на базе современных управляющих устройств с многообразием входов-выходов и отсутствием их инъективности. Среди свойств подсистем выделяются:

$$Atr_1: Sub \in Su; Atr_2: \forall (E_i \in Sub, E_j \in Sub) \rightarrow \\ \rightarrow (E_i \in Su, E_j \in Su), i \neq j;$$

$$Atr_3: \forall Sub: F_{SUB} = \text{const}, F_{SUB} \in \{F\}; \\ Atr_4: \forall Sub: I \in \{I\}, O \in \{O\},$$

где E — элементы технического обеспечения в смысле (1); F_{SUB} — функция подсистемы, определяемая решаемой задачей; $\{F\}$ — множество функций; $\{I\}$, $\{O\}$ — множества входов, выходов подсистем.

Свойства Atr_3 подсистемы технического обеспечения наследуют свойства АСУ ТП. Свойства Atr_2 подсистемы отличаются друг от друга вариациями программно-технической реализации.

Каждая подсистема представляется совокупностью цепей. Область значений для множества типов цепей задается в виде

$$\{N\} = \{Ch_1, Ch_2, Ch_3, Ch_4, Ch_5\},$$

где N — тип цепи; $Ch_1, Ch_2, Ch_3, Ch_4, Ch_5$ — цепи измерения, управления, сигнализации, передачи данных, питания.

Свойства цепи формально определяются:

$$Atr_{Ch} = \{Name, Atr_{Ch}^0, Atr_{Ch}^1, \dots, Atr_{Ch}^6\}, \\ Atr_{Ch}^0: Ch = \{Ch_a, Ch_d\}, Atr_{Ch}^1: Ch R_c, E_{T1} \& \dots \& E_{Tk}, \\ Atr_{Ch}^2: F_{Ch} = \text{const}, Atr_{Ch}^3: F_{Chi} \neq F_{Chj}, i \neq j, \\ Atr_{Ch}^4: \forall Ch \exists I, O, I \in \{I\}, O \in \{O\},$$

$$Atr_{Ch}^5: Ch = \{Ch_{ac}, Ch_{dc}\},$$

$$Atr_{Ch}^6: Ch_{ac} = \{Ch_{ac1}, Ch_{ac3}\}, \quad Atr_{Ch}^7: Ch \in Co,$$

где *Name* — имя цепи; Ch_a, Ch_d — цепи с аналоговым и дискретным информационными сигналами; k — число типов элементов в цепи; F_{Ch} — функция цепи; E_{T_i} — элементы из класса T_i классификации элементов; Ch_{ac}, Ch_{dc} — цепи переменного и постоянного тока; Ch_{ac1}, Ch_{ac3} — однофазные и трехфазные цепи; Co — фиксация принадлежности цепи определенному контуру, связывающая цепь с функциональной структурой.

Принципиально новым в предлагаемой модели информационной базы является учет свойств передаваемых в цепи сигналов ($Atr_{Ch}^0, Atr_{Ch}^5, Atr_{Ch}^6$) при формировании связей и выполнении оператора Op_1 (на уровне синтеза функциональной схемы автоматизации).

Свойства сигналов наследуются в модели технических элементов в виде свойств входов-выходов элементов. Области значений сигнала, учитываемые при построении цепей, отражают детализацию информационных сигналов и сигналов питания при их физической реализации с использованием как постоянного (DC), так и переменного (AC) тока (см. таблицу).

Используемая для построения информационной базы модель технических элементов и представляется в виде иерархической структуры, верхние уровни которой занимают абстрактные объекты, отражающие типы и подтипы элементов, а на нижних уровнях располагаются конкретные элементы и их свойства. Переход между уровнями осуществляется на основе отношений "род — вид", "иметь свойства", "иметь значения свойств". Формально элемент представляется в виде

$$E = \langle Name, T_i, Atr \rangle,$$

где *Name* — имя; T_i — класс таксономии, к которому относится элемент; *Atr* — свойства элемента.

Для построения цепей питания в информационной базе представляются следующие характеристики элементов: энергозависимость, номер и назначение контактов разъемов для подключения питания, значения электрических параметров энергетических сигналов.

Для анализа вариантов технической реализации системы управления в информационной базе предусматривается учет свойств, являющихся составляющими многокритериальной оценки (стоимость, надежность, метрологические характеристики, энергопотребление и др.), а также системных свойств [7].

Для преобразования моделей технической структуры S_1 — S_3 (см. рисунок) все модели эле-

Область значений сигналов входов-выходов элементов систем управления

Сигнал входа/выхода	Область значений	Сигнал входа/выхода	Область значений
Аналоговый	Потенциальный (+) Потенциальный (-)	Аналоговый	Токовый (+) Токовый (-)
Аналоговый	Частотный (fr)	Дискретный	Релейный (AC)
Дискретный	Релейный (DC+) Релейный (DC-)	Дискретный	Цифровой (+) Цифровой (-)
Аналоговый	Напряжение переменного тока (AC)	Дискретный	Импульсный (B-DC) Импульсный (0-DC) Импульсный (M-DC)
Дискретный	Импульсный (B-AC) Импульсный (0-AC) Импульсный (M-AC)	От параметрического датчика	0, выход 1, выход 2
Питание	Постоянный ток (+) Постоянный ток (-)	Питание	Переменный ток (0, фазы A, B, C)

ментов имеют три обязательных параметра: информационный вход, функция преобразования информационного входа в информационный выход, информационный выход. Параметрам входа выхода на уровне значений свойств присваиваются значения сигналов в цепи (см. таблицу), составной частью которой является данный элемент.

Общий алгоритм автоматизированного проектирования схем систем управления технологическими объектами с учетом представленного описания основан на операциях агрегации и включает следующие шаги.

Шаг 1. Создание начальной модели структуры системы с определением базовых элементов и порядка их следования.

Шаг 2. Создание цепей и определение их свойств.

Шаг 3. Определение типа цепи и вида сигналов в ней.

Шаг 4. Определение свойств входов-выходов элементов для организации связей и поиск элементов.

Шаг 5. Создание связей элементов.

Шаг 6. Объединение цепей в контуры и определение их свойств.

Шаг 7. Объединение контуров в подсистемы и определение их свойств.

Итерационно после каждого шага осуществляется анализ полученных свойств и коррекция результатов в ходе повторения предыдущего шага при отклонении значений свойств от заданных.

Анализ алгоритма показывает, что для автоматизированного проектирования схем систем управления необходимы свойства составляющих более нижних уровней описания системы управления, чем непосредственно сопряженного с рассматриваемым уровнем.

Результаты работы и их апробация

Построение схем систем управления технологическими объектами основывается на последовательном преобразовании моделей технической структуры системы. Информационная база для автоматизированного проектирования схем включает: описания классов элементов и их свойств; связанные с классами правила преобразования моделей технической структуры. Выбор базовых свойств составляющих систем управления технологическими объектами в виде сигналов входов-выходов позволяет перейти к автоматическому синтезу принципиальных схем на основе однозначного определения аргументов и значений функции преобразования входов-выходов элементов.

Программная реализация предложенных моделей и алгоритмов выполнена на базе языка С и SQL-запросов. Апробация проводилась для набора управляющих и измерительных цепей подсистем управления тепловыми объектами разного назначения. За критерий адекватности представления элементов в информационной базе принимается сходимость алгоритмов синтеза новых технических структур из заданного набора элементов, являющихся базовыми при построении функциональных цепей систем управления.

По тридцати заданным вариантам измерительных и исполнительных устройств в цепях с аналоговыми и релейными сигналами обеспечивался автоматический поиск управляющих устройств и синтез схемных описаний на основе оценки совместимости элементов системы управления при переходе от функциональной к технической структуре. Все полученные варианты при экспертной оценке отмечены как правильные и соответствующие проектным решениям, полученным в ходе выполнения реальных проектов.

Полученные результаты обеспечивают взаимосвязь прикладного и представительского уровней средств и систем автоматизированного проектирования и физического и интерфейсного уровней АСУ ТП, что выражается в детализации концептуальных описаний составляющих систем управления в САПР до свойств сигналов элементов, учитываемых на физическом уровне АСУ ТП.

Список литературы

1. **Норенков И. П.** Основы автоматизированного проектирования. М.: МГТУ им. Н. Э. Баумана, 2006. 448 с.
2. **Технология** системного моделирования / Под ред. С. В. Емельянова. М.: Машиностроение, Берлин: Техник, 1988. 520 с.
3. **Трахтенгерц Э. А.** Компьютерная поддержка принятия решений в САПР // Автоматизация проектирования. 1998. № 5. С. 27–38.
4. **Трахтенгерц Э. А.** Компьютерная поддержка принятия решений в САПР // Автоматизация проектирования. 1999. № 1. С. 16–26.
5. **Курейчик В. М.** Математическое обеспечение конструкторского и технологического проектирования с применением САПР. М.: Радио и связь, 1990. 348 с.
6. **Цвиркун А. Д.** Основы синтеза структуры сложных систем. М.: Наука, 1982. 200 с.
7. **Методы** анализа и синтеза структур управляющих систем / Под ред. Б. Г. Волика. М.: Энергоатомиздат, 1988. 296 с.

УДК 621.396.6:658.512.011.56

С. В. Шумарин, аспирант;

Е. Н. Талицкий, д-р техн. наук, проф.,
Владимирский государственный университет,
e-mail: ental@vpti.vladimir.ru

Оптимизация виброзащиты электронной аппаратуры методом частотной отстройки

Анализируются возможные способы виброзащиты электронной аппаратуры методом частотной отстройки. Предлагается алгоритм оптимизации массогабаритных показателей аппаратуры при частотной отстройке.

Ключевые слова: электронная аппаратура, виброзащита, частотная отстройка, алгоритм.

Введение

Многие виды электронной аппаратуры (ЭА), устанавливаемой на подвижных объектах — самолетах, бронетранспортерах, ракетах и т. д., подвергаются воздействию вибраций в широком диапазоне частот. Это приводит к возникновению резонансных колебаний таких широко распространенных элементов конструкций ЭА, как ячейки, состоящие из электромонтажных плат с установленными на них электрорадиоэлементами (ЭРЭ). Виброперегрузки ЭРЭ увеличиваются при этом в десятки раз, что вызывает значительное возрастание интенсивности отказов как за счет механических разрушений элементов конструкций, так и за счет искажения параметров электрических сигналов. В этом случае виброзащита ЭА часто сводится к устранению резонансных колебаний ячеек.

В работе [1] рассмотрен алгоритм проектирования виброзащиты ЭА, в соответствии с которым, если верхняя граница диапазона воздействующей вибрации не превышает 500 Гц, устранять резонансные колебания предпочтительнее частотной отстройкой (ЧО), так как она конструктивно и технологически проста по сравнению с остальными способами виброзащиты. Применение ЧО характерно для многих видов ЭА, используемой на высокоскоростных транспортных средствах на гусеничном ходу.

Для обеспечения ЧО первая собственная частота колебаний (СЧК) ячеек ЭА должна не менее чем на 30 % превышать максимальную частоту возбуждения [2]. Так как это достигается повышением жесткости конструкции за счет увеличения толщины плат, применения ребер жесткости, масса ячеек значительно увеличивается и часто превосходит допустимые нормы, что вызывает необходимость ее минимизации.

Особенности применения способов частотной отстройки

Повышать СЧК ячеек ЭА можно усилением жесткости крепления, например, установкой дополнительных точек крепления (ДТК), уменьшением площади ячейки, увеличением толщины электромонтажной платы, применением ребра жесткости или дополнительного слоя из легкого, но жесткого пенопласта типа ППУ-305А. Как показано в работе [3], последний способ позволяет существенно повышать СЧК. Если одним способом не удастся устранить резонансы ячеек, то используют несколько способов одновременно.

Каждый способ, повышая СЧК, увеличивает суммарную массу ячеек и в определенной степени ухудшает их конструктивные параметры, а следовательно, имеет свои ограничения. Так, уменьшение площади электромонтажных плат ограничивается электрической принципиальной схемой и часто необходимостью применения типовых конструкций блоков ЭА, а толщина плат — номенклатурой используемых материалов. Нанесение пенопластов ухудшает ремонтпригодность и теплообмен в ЭА.

Установка ребер жесткости усложняет конструкции ЭА и зависит от применяемого способа крепления. Повышение жесткости крепления достигается также введением ДТК, что расширяет возможные варианты установки ребер, но уменьшает площадь для электромонтажа и часто невозможно ввиду конструктивных особенностей блоков.

В работе [4] на примере связанной аппаратуры показано, что с учетом большинства факторов частотную отстройку целесообразно начинать с применения ребер жесткости, затем использовать

ДТК. Если и в этом случае обеспечить ЧО не удастся, то нужно увеличивать толщину платы, потом уменьшать ее площадь и применять дополнительный слой из жесткого пенопласта. Последние три способа часто неприемлемы ввиду различных ограничений, указанных выше. Поэтому наилучших результатов можно добиться, используя ребра жесткости и ДТК, часто совместно.

Применение ребер жесткости и дополнительных точек крепления

Проектирование ЧО рекомендуется начинать с определения собственных форм колебаний (СФК) ячейки ЭА. Находить СФК прямоугольных конструкций с простейшими моделями креплений целесообразно аналитическими методами, так как они обеспечивают наибольшее быстрое действие вычислений, а сложных конструкций — с помощью систем конечно-элементного анализа или специально разработанных препостпроцессорных сред на их основе.

С этой целью нами разработана программа *VibroDefence* [5], которая имеет в своем составе пре- и постпроцессор, использующие мощную систему конечно-элементного анализа ANSYS в качестве вычислителя. Это не требует от разработчика ЭА изучения встроенного интерфейса и в то же время позволяет анализировать СФК ячеек любой сложности и выбирать те формы, которые конструктивно обеспечивают требуемую частоту, а также оптимизировать массогабаритные показатели.

Метод ЧО с помощью ребер жесткости и ДТК заключается в модификации ячейки таким образом, чтобы СФК исходной конструкции, частота которой выше максимальной частоты вибрации, примерно равнялась первой для модифицируемой конструкции.

Расположение ребер жесткости и ДТК определяются из анализа СФК ячеек. Первоначально рассматриваются те СФК, обеспечение которых не требует изменения крепления и может быть осуществлено только установкой ребер жесткости в местах образования узловых линий [4]. При

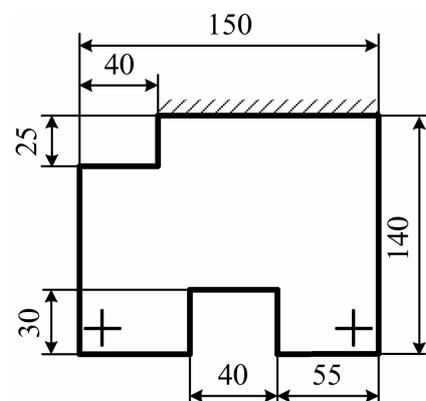


Рис. 1. Конструкция ячейки:
 ▨ — жесткое защемление;
 — свободный край;
 + — точка крепления

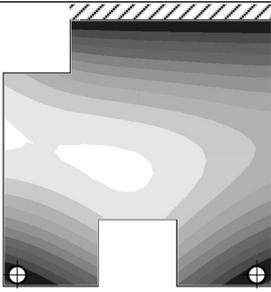
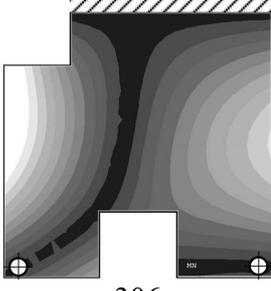
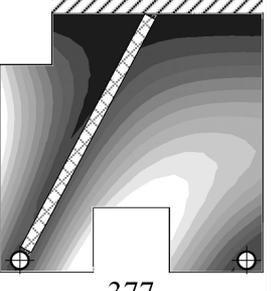
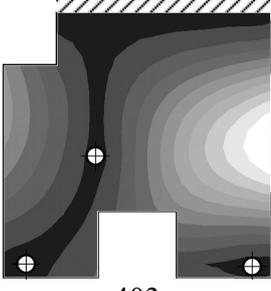
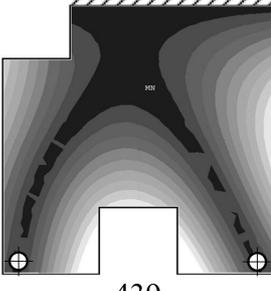
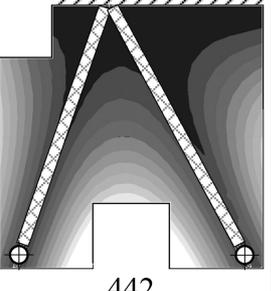
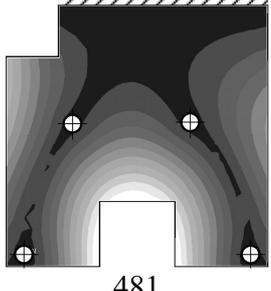
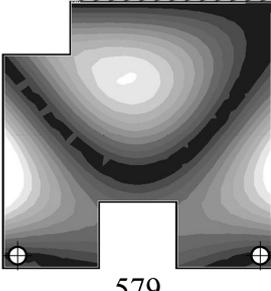
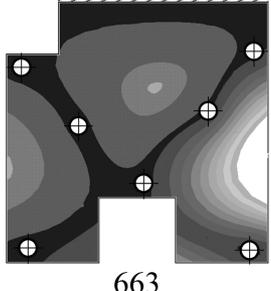
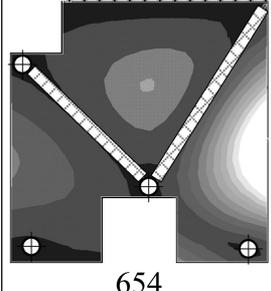
этом ребра должны проходить через места крепления ячейки. Если СЧК таких конструкций ниже требуемых значений, то применяются ДТК для обеспечения остальных СФК. Сочетание ДТК и ребер жесткости позволяет создать конструкции практически с любыми первыми СФК.

В качестве примера на рис. 1 представлена ячейка ЭА сложной конструкции из стеклотек-

столита СТЕФ (ГОСТ 12652—74) толщиной 1,5 мм, а в таблице — возможные конструктивные решения для ее ЧО, полученные с использованием программы *VibroDefence*.

В столбце 2 приведены первые четыре СФК исходной конструкции ячейки (черным цветом изображены минимумы амплитуд колебаний, белым — максимумы), в столбцах 3—5 — варианты

Собственные формы и частоты колебаний

№ СФК	Формы колебаний и СЧК, Гц	СФК и СЧК только при установке ребер жесткости, Гц	СФК и СЧК только при установке ДТК, Гц	СФК и СЧК при установке ребер жесткости и ДТК, Гц
1	2	3	4	5
1	 245	—	—	—
2	 306	 377	 403	—
3	 439	 442	 481	—
4	 579	— Не применимы, так как ребра не будут проходить через места крепления	 663	 654

конструкций, включающие ребра и ДТК, и соответствующие им СЧК.

В таблице под точкой крепления подразумевается крепление винтом В.М2.5-6gx10.32ЛС59-1.133 (ГОСТ 17476—80) массой 0,8 г — 1 шт. с шайбой изоляционной А2.5.04.013 (ГОСТ 11371—78) массой 0,1 г — 2 шт. и стойкой с крепежной массой 9 г — 1 шт. Каждая ДТК уменьшает полезную площадь ячеек на 59 мм². Считалось, что ребра жесткости сечением 3 × 10 мм выполнены из материала платы. Для наглядности полученных результатов ЭРЭ на плату не устанавливались.

Так как СЧК должна превышать верхнюю границу частоты возбуждения, выбор той или иной СФК для проектирования ЧО определяется диапазоном воздействующей вибрации. Например, для виброзащиты ячейки ЭА, используемой на железнодорожном транспорте (частоты возбуждения до 300 Гц по ГОСТ РВ 20.39.304—98) необходимо выбрать СФК не ниже второй (306 Гц), в составе высокоскоростных транспортных средств на гусеничном ходу (частоты до 500 Гц) — не ниже четвертой (579 Гц) и т. д.

В столбце 3 показаны варианты конструкции только с ребрами жесткости, в столбце 4 — только с ДТК, в столбце 5 — с ребрами и ДТК совместно. Для каждой из них должно выполняться условие отсутствия резонансов:

$$f_1 > 1,3f_b, \quad (1)$$

где f_1 — первая СЧК ячейки; f_b — верхняя граница диапазона воздействующей вибрации.

В данном случае при эксплуатации до 300 Гц только вариант с ДТК (403 Гц) удовлетворяет этому условию. Для виброзащиты ячейки только ребрами жесткости необходимо обеспечивать следующую СФК, третью, на частоте 439 Гц; при эксплуатации до 500 Гц подходят обе предложенные конструкции, так как их частоты (663 и 654 Гц) превышают $1,3 \cdot 500$ Гц = 650 Гц. Окончательный выбор определяется разработчиком с учетом возможных массогабаритных показателей, конструкций блока ЭА и других факторов.

Применение слоев жесткости

Повысить СЧК возможно также применением слоев из жесткого пенопласта, но это может привести к существенному увеличению массы конструкции (рис. 2) [3]. Этот недостаток можно устранить, если применить для электромонтажной платы стеклотекстолит меньшей толщины. Например, вместо толщины 1,5 мм применить стеклотекстолит толщиной 0,5 мм. Так как плотность жестких пенопластов в 10 или более раз меньше плотности стеклотекстолита, нанесение слоя из такого материала толщиной 10 мм или даже боль-

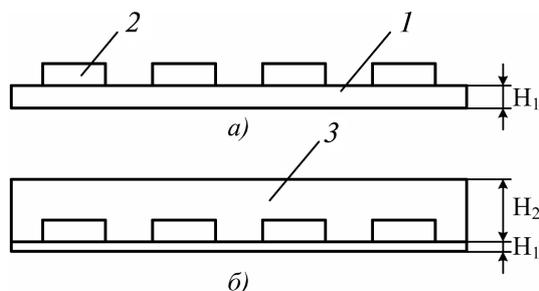


Рис. 2. Пример нанесения слоя из жесткого пенопласта: а — до нанесения; б — после нанесения; 1 — плата; 2 — ЭРЭ; 3 — слой пенопласта

ше не приведет к существенному увеличению массы конструкции, а жесткость ее и, следовательно, СЧК возрастут.

Максимально допустимую толщину слоя, наносимого по всей поверхности ячейки, можно найти из выражения [3]

$$H_2 = \frac{N\rho_1(h_{1н} - h_{1к})}{\rho_2}, \quad (2)$$

где N — допустимый коэффициент увеличения начальной массы конструкции; $h_{1н}$ и $h_{1к}$ — начальная и конечная толщина электромонтажной платы соответственно; ρ_1 и ρ_2 — плотности материала платы и слоя соответственно.

Если слой пенопласта наносится заливкой поверх ЭРЭ, то его толщина должна быть увеличена пропорционально уменьшению объема, занимаемого ЭРЭ.

Целевая функция

Основной задачей при ЧО является обеспечение условия (1). При этом необходимо, чтобы увеличение массы конструкции ячейки было минимальным. Поэтому в качестве целевой функции целесообразно выбрать минимум массы при обеспечении основного условия эффективности ЧО в виде (1). Также нужно учитывать, что и минимальная масса не должна превышать значение, которое может быть указано в ТЗ на устройство. Исходя из этого, наряду с параметром оптимизации масса конструкции ячеек ЭА должна входить и как ограничение: $m \leq m_{\text{доп}}$. В массу ячейки ЭА (m) входит суммарная масса ребер жесткости, которая зависит от числа ребер n , их плотности ρ_p , высоты H_p , длины l и ширины s_p . Также массу конструкции увеличивают точки крепления $m_{\text{ДТК}}$, массогабаритные параметры электромонтажной платы (S , $H_{\text{п}}$, ρ) и слоя из пенопласта (H_c , ρ_c).

Габаритные размеры ячейки — максимальная длина и ширина определяются исходя из конструкции блоков, требований схемы электрической

принципиальной и не изменяются в процессе применения ЧО, а общая высота ячейки может существенно повыситься и превысить допустимое значение вследствие применения ребер жесткости или слоя пенопласта. Поэтому она тоже должна быть ограничена и не превышать значение $H_{доп}$, определяемое конструкцией блоков ЭА.

В общем виде целевая функция имеет вид:

$$m \rightarrow \min(S, H_p, \rho, m_{ДТК}, n, \rho_p, H_p, l_1 \dots l_n, s_p, H_c, \rho_c),$$

при условии, что $f_1 \geq 1,3 f_b$; $m \leq m_{доп}$; $H \leq H_{доп}$.

Количество, длина ребер жесткости и число ДТК определяются, исходя из конструкции ячейки ЭА. Затем в зависимости от выбранного способа ЧО оптимизируются высота ребер или площадь ячейки, или высота платы, или толщина слоя. Ширина ребер жесткости варьируется в диапазоне 1,5...3 мм, так как ребро больше 3 мм приводит к существенному увеличению массы, а при значениях ниже 1,5 мм могут возникнуть резонансные колебания самих ребер вследствие недостаточной их жесткости. Высота ребер жесткости должна быть достаточна для образования узловых линий вдоль ребер. Дальнейшее увеличение высоты ребер не повышает существенно СЧК конструкции, но значительно увеличивает ее массу [4]. Обычно ребра жесткости высотой меньше 5 мм не приводят к повышению СЧК плат толщиной 1...1,5 мм.

Поиск оптимального значения перечисленных факторов может быть проведен простым приращением значения или любым методом одномерной оптимизации после определения максимальных и минимальных значений: дихотомического деления, золотого сечения, чисел Фибоначчи, полиномиальной аппроксимации и т. д.

Алгоритм оптимизации

Блок-схема алгоритма оптимизации виброзащиты ячеек ЭА методом частотной отстройки показана на рис. 3. Она может быть встроена в общий алгоритм проектирования виброзащиты, опубликованный в работе [1], вместо соответствующего этапа или применяться автономно.

В работе [4] описан алгоритм оптимизации ЧО только простейших конструкций, что позволяло

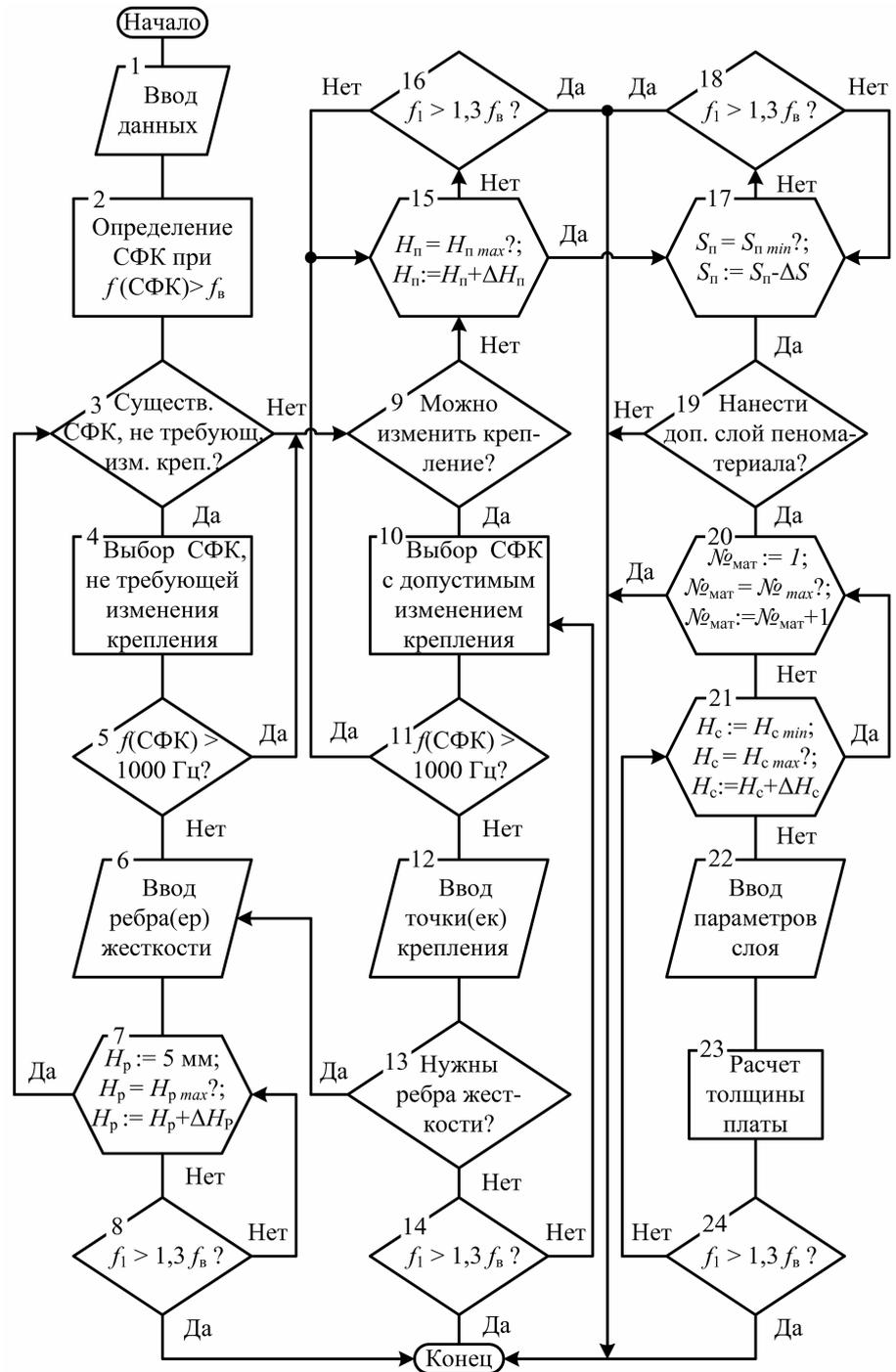


Рис. 3. Алгоритм оптимизации частотной отстройки

представлять соответствующую блок-схему в виде линейного последовательного применения рассмотренных выше способов ЧО. Алгоритм на рис. 3 предназначен для проектирования ЧО практически любых ячеек ЭА и представляет собой оптимизационную функцию, полученную декомпозицией цикла увеличения жесткости конструкции [1]. В нем выполняются структурный синтез конструкции на основе пяти основных способов ЧО с последующей параметрической оптимизацией выбранного варианта в целях поиска минимальной массы конструкции при обеспечении основного условия (1).

Первые два этапа — установка ребер жесткости и изменение крепления ячейки представляют собой вложенные циклы и проводятся по представленному выше методу, основанному на анализе и обеспечении СФК ячейки. Операции определения СЧК в алгоритме условно не показаны. Эти характеристики находятся при обращении к ним в блоках проверки условия (1).

Исходными данными (блок 1) для проектирования ЧО являются размеры ячейки ЭА, способ ее крепления и частотный диапазон воздействующих вибраций. В блоке 2 определяются несколько СФК конструкции, частота которых превышает верхнюю границу диапазона воздействующих вибраций. Далее (блоки 3—5, 9—11, 13) проводится анализ СФК в целях их возможного обеспечения с помощью ребер жесткости, ДТК или совместным использованием этих способов.

Выбор СФК зависит от конкретного решения, но рекомендуется первоначально рассматривать те СФК, обеспечить которые возможно только ребрами жесткости без установки ДТК (блок 4). Располагать ребра жесткости необходимо в местах образования узловых линий (блок 6).

Затем осуществляется поиск оптимальной высоты установленных ребер жесткости от 5 мм до максимально допустимого значения. В алгоритме это изображено в виде циклического перебора вариантов конструкции (блок 7). На каждой итерации выполняется приращение высоты ребра на определенное значение (2—5 мм) и проверяется, не превышает ли СЧК ячейки верхнюю границу вибрации на 30 % (блок 8). Если да (резонансные колебания устранены), работа алгоритма заканчивается. В противном случае необходимо вернуться к анализу других СФК (блоки 3 и 4) с другими вариантами расположения ребер жесткости. Обыч-

но высшие СФК требуют более сложных вариантов установки ребер.

Если не удастся обеспечить ни одной СФК в диапазоне от f_B до 1000 Гц (блоки 5 и 11) только установкой ребер жесткости, то при возможном изменении крепления конструкции (блок 9) рассматривается обеспечение прочих СФК (блок 10) с помощью одних ДТК, размещаемых также по узловым линиям (блок 12), или одновременно ребрами и ДТК (блок 13).

Если и эти меры не приводят к повышению СЧК, то рассматриваются варианты увеличения толщины (блоки 15, 16) и уменьшения площади ячеек (блоки 17, 18), представленные в виде соответствующих этапов параметрической оптимизации.

Заключительной мерой (блоки 19—24) может быть нанесение слоя из пенопласта. В этом случае (блок 19) выбирается материал слоя из базы данных (блок 20), определяется его высота (блок 21, 22) и рассчитывается толщина электромонтажной платы (блок 23) по формуле (2).

Заключение

Представленный алгоритм и разработанная на его основе программа, входящая в состав программного комплекса [5], позволяют разработчику быстро и эффективно применять способы частотной отстройки ячеек электронной аппаратуры, что значительно сокращает время разработки новых изделий и уменьшает объем их экспериментальной отработки.

Список литературы

1. **Талицкий Е. Н.** Алгоритм проектирования виброзащиты электронной аппаратуры // Информационные технологии. 2009. № 1. С. 34—36.
2. **Ухин В. А., Талицкий Е. Н.** Определение собственных частот колебаний ячеек радиотехнических устройств со сложным способом закрепления // Перспективные технологии в средствах передачи информации: Материалы VI международной научно-технической конференции. Владимир, 2005. С. 156—158.
3. **Талицкий Е. Н.** Оценка эффективности антирезонансных покрытий субблоков микроэлектронной аппаратуры // НТС "Техника средств связи". 1982. Вып. 1. С. 43—52.
4. **Ухин В. А., Талицкий Е. Н.** Алгоритм проектирования виброзащиты электронной аппаратуры методом частотной отстройки // Проектирование и технология электронных средств. 2006. № 3. С. 61—64.
5. **Талицкий Е. Н., Шумарин С. В.** Программный комплекс проектирования виброзащиты ячеек электронной аппаратуры // Программные продукты и системы. 2008. № 3. С. 103—104.

УДК 519.6

А. П. Карпенко, д-р физ.-мат. наук, проф.,
МГТУ им. Н. Э. Баумана,

В. А. Овчинников, канд. техн. наук,
ген. директор, ООО "Ладуга",

А. С. Семенихин, магистр, IBM,
e-mail: karpenko@nog.ru

Программная система PRADIS//FRONT для построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельного генетического алгоритма

Рассматривается распределенная программная система PRADIS//FRONT для приближенного построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельных генетических алгоритмов. В качестве среды моделирования динамических систем используется программный комплекс PRADIS.

Ключевые слова: задача многокритериальной оптимизации, множество Парето, фронт Парето, динамическая система, генетический алгоритм, параллельный генетический алгоритм, программный комплекс PRADIS.

Введение

Проектирование сложной технической системы обычно включает в себя этап параметрической оптимизации. В настоящее время при этом чаще всего используется многокритериальная оптимизация, при которой качество функционирования объекта проектирования определяется некоторым набором критериев оптимальности.

Методы решения задачи многокритериальной оптимизации чрезвычайно разнообразны. Для пользователя наиболее удобны методы, основанные на непосредственном использовании множества Парето (а тем самым, и фронта Парето). В этом случае лицо, принимающее решение (ЛПР), выбирает компромиссное решение на фронте Парето неформальными методами исходя только из

своих предпочтений. Основным недостатком методов этого класса, сдерживающим их широкое применение, является высокая вычислительная сложность.

Известно значительное число методов и алгоритмов приближенного построения множества Парето. Относительно новым и высокоэффективным классом таких методов являются методы на основе генетических алгоритмов.

Генетические алгоритмы принадлежат к классу эволюционных алгоритмов и обладают рядом характеристик, делающих их более предпочтительными, чем классические методы оптимизации [1]. Так, генетические алгоритмы применимы к задачам большой размерности и способны захватить Парето-оптимальные точки даже при однократном запуске алгоритма.

Основными операциями генетического алгоритма при приближенном построении множества Парето являются операции скрещивания (кроссовера, рекомбинации), мутации, вычисления пригодности индивида и отбора (селекции), из которых специфическими для задачи построения множества Парето являются только две последние операции.

Выделяют следующие основные способы селекции в генетических алгоритмах приближенного построения множества Парето:

- селекция по переключающимся частным критериям оптимальности;
- агрегирующая селекция;
- селекция, основанная на понятии Парето-доминирования [1].

Для того чтобы построить репрезентативную аппроксимацию множества Парето, необходимо обеспечить разнообразие популяций. Эта задача является одной из важнейших в проблематике приближенного построения множества Парето с помощью генетических алгоритмов. Для решения данной задачи могут быть использованы следующие подходы:

- формирование популяционных ниш (подпопуляций);
- ограниченное скрещивание;
- переопределение;
- перезапуск процесса эволюции;
- уплотнение (объединение) [2].

Наряду с поддержанием разнообразия популяций важнейшую роль в обеспечении эффективности генетических алгоритмов приближенного построения множества Парето играет элитизм — ме-

ханизм включения лучших индивидов данной популяции в следующую популяцию, т. е. клонирование лучших индивидов.

Построение множества Парето в задаче многокритериальной оптимизации динамических систем требует многократного моделирования исследуемой динамической системы при различных значениях варьируемых параметров. Моделирование динамических систем является самостоятельной проблемой, для решения которой разработано значительное число программных комплексов. В программной системе PRADIS//FRONT моделирование выполняется средствами программного комплекса PRADIS [3].

Многократное моделирование сложной динамической системы требует больших вычислительных ресурсов и в приемлемое для исследователя время может быть выполнено только на параллельных вычислительных машинах. Система PRADIS//FRONT ориентирована на использование в качестве таких машин распределенных ЭВМ, например вычислительных кластеров.

Существует большое число программных средств, обеспечивающих эффективное использование ресурсов распределенной ЭВМ. Программная система PRADIS//FRONT построена на основе коммуникационной библиотеки MPI, точнее — ее версии MPICH. Данное решение определено следующими факторами: большим числом реализаций MPICH, его широким распространением; наличием свободных, регулярно обновляемых имплементаций; поддержкой операционными системами семейства Windows (программный комплекс PRADIS функционирует в среде Windows).

Работоспособность программной системы PRADIS//FRONT проверена на стандартных тестовых задачах многокритериальной оптимизации [2]. Практическое использование системы рассмотрено на примере приближенного построения множества Парето для задачи многокритериальной оптимизации автомобильной коробки переключения передач.

Постановка задачи

Совокупность частных критериев оптимальности (векторный критерий оптимальности) обозначим $\Phi(X) = (\phi_1(X), \phi_2(X), \dots, \phi_m(X))$, где $X \subset R^n$; R^n — n -мерное арифметическое пространство. Положим, что ставится задача максимизации каждого из указанных критериев в одной и той же области допустимых значений $D_x \in \Pi \cap D$, где $\Pi = \{X | x_i^- \leq x_i \leq x_i^+, i \in [1, n]\}$ — "технологический" параллелепипед, $D = \{X | g_1(X) \geq 0, g_2(X) \geq 0, \dots\}$. Здесь $g_1(X), g_2(X), \dots$ — ограничи-

вающие функции. Задача многокритериальной оптимизации условно записывается в виде

$$\max_{X \in D_X} \Phi(X) = \Phi(X^*). \quad (1)$$

Векторный критерий оптимальности $\Phi(X)$ выполняет отображение множества D_X в некоторое множество D_Φ пространства критериев, которое называется множеством достижимости. Введем на множестве D_Φ отношение предпочтения. Будем говорить, что вектор $\Phi^1 \in D_\Phi$ предпочтительнее вектора $\Phi^2 \in D_\Phi$ или вектор Φ^1 доминирует вектор Φ^2 , и писать $\Phi^1 > \Phi^2$, если среди равенств и неравенств $\phi_k(X^1) \geq \phi_k(X^2)$, $k \in [1:m]$ имеется хотя бы одно строгое неравенство. Выделим из множества D_Φ подмножество точек D_Φ^* (фронт Парето), для которых нет более предпочтительных точек. Множество $D_\Phi^* \in D_\Phi$, соответствующее множеству D_Φ^* , называется множеством Парето. Таким образом, если $X \in D_X^*$, то $\Phi(X) \in D_\Phi^*$.

Пусть I — бинарный вектор индивида (генотипа), а D_I — множество всех возможных его значений. Тогда популяцией P называется некоторое мультимножество векторов $I \in D_I$. Качество индивида определяется скалярной функцией пригодности $f(I)$, вычисляемой на основе значений частных критериев оптимальности $\phi_1(X), \phi_2(X), \dots, \phi_m(X)$. Введем в рассмотрение также функцию $M(I)$, осуществляющую отображение индивида $I \in D_I$ в его фенотип $X = M(I)$. Таким образом, для вычисления функции пригодности некоторого индивида $I \in D_I$ необходимо выполнить цепочку вычислений

$$X = M(I) \rightarrow \Phi = \Phi(X) \rightarrow f = f(\Phi). \quad (2)$$

Генетический алгоритм приближенного построения множества Парето

В настоящее время в вычислительной практике наиболее часто используются четыре метода приближенного построения множества Парето на основе генетических алгоритмов: метод VEGA (Vector Evaluated Genetic Algorithm); метод FFGA (Fonseca and Fleming's Multiobjective Genetic Algorithm); метод NPGA (Niche Pareto Genetic Algorithm); метод SPEA (Strength Pareto Evolutionary Algorithm) [1].

Существует два подхода к проектированию параллельных генетических алгоритмов — однопопуляционный и многопопуляционный. В однопопуляционном подходе параллельно вычисляются только функции пригодности, а генетические операции выполняются централизованно на host-процессоре. Такой подход, очевидно, ориентирован на master-slave парадигму параллельного программирования.

С точки зрения параллелизации более естественным является, конечно, многопопуляционный подход. Суть этого подхода состоит в разделении главной популяции на части и реализации каждым из процессоров собственного генетического алгоритма над той частью популяции, которая хранится в его памяти. Для повышения эффективности поиска процессоры могут обмениваться между собой генетическим материалом.

Известно большое число параллельных генетических алгоритмов, ориентированных на различные классы параллельных вычислительных систем и на различные критерии качества алгоритмов. В соответствии с одним из известных вариантов классификации этих алгоритмов выделяются следующие пять классов таких алгоритмов [4].

GPGA (Global Parallel Genetic Algorithm) представляет собой однопопуляционный алгоритм. Host-процессор содержит все поколение в своей памяти и выполняет над ним операции селекции, кроссовера и мутации. Функции пригодности индивидов вычисляются на slave-процессорах. При высокой вычислительной сложности функций пригодности основной проблемой при реализации данного алгоритма является проблема балансировки загрузки slave-процессоров. Очевидно, что рассматриваемый алгоритм ориентирован на MIMD-вычислительные системы.

DGA (Distributed Genetic Algorithm) — многопопуляционный алгоритм, также ориентированный на MIMD-вычислительные системы.

MPGA (Massively Parallel Genetic Algorithm) называют также Cellular Algorithm (клеточный алгоритм). Данный алгоритм ориентирован на вычислительные машины класса SIMD.

Hierarchical PGA представляют собой иерархические алгоритмы, в которых на разных уровнях иерархии реализованы различные генетические алгоритмы (рассмотренные выше и другие). Обычно используются двухуровневые иерархии: DGA-модель — на верхнем уровне; GPGA- или MPGA-модель — на нижнем уровне.

Hybrid PGA — класс алгоритмов, использующих сочетание параллельных генетических алгоритмов и классических оптимизационных методов.

В работе в качестве базового метода выбран последовательный метод NPGA. Это объясняется тем обстоятельством, что метод NPGA существенно более прост в реализации, чем метод SPEA, и, в то же время, обеспечивает достаточно высокую эффективность [1]. Использована модификация метода, предложенная в работах [2, 5] и заключающаяся в следующем:

- использование метода ранжирования индивидов на основе Парето-доминирования [2];

- использование аутбридинга (дальнеродственного скрещивания) вместо турнирной селекции [5];
- использование операции клонирования.

По определению ранга, индивиду, для которого ни один из индивидов текущей популяции не лучше него по всем частным критериям оптимальности, присваивается ранг 1. Ранг остальных индивидов определяется по формуле

$$\text{rank}(I_k) = 1 + a_k, \quad (3)$$

где a_k — число индивидов текущей популяции, лучших по всем частным критериям оптимальности.

Используемые механизмы формирования индивидов обеспечивают выполнение условия $X = M(I) \in \Pi$, но не обеспечивают, в общем случае, выполнение условия

$$X = M(I) \in D. \quad (4)$$

Ранг индивидов, которые нарушают ограничение (4), назначается в зависимости от того, в какой мере эти ограничения нарушены. Ранг любого из индивидов из числа тех, для которых ограничение (4) нарушено, выше ранга любого из индивидов, для которых это ограничение выполнено.

Функция пригодности строится на основе функции

$$\varphi(I) = 1 + \sum_{k=1}^{\text{rank}(I)-1} \eta(k), \quad (5)$$

где $\eta(k)$ — число индивидов ранга k . Точнее говоря, в качестве функции пригодности используется функция

$$f(I_1) = \mu(I_1)\varphi(I_1), \quad (6)$$

где $\mu(I_1)$ — нишевое число индивида I_1 , вычисляемое по формуле

$$\mu(I_1) = \sum_{I_k \in P_t} \text{Sh}(d(I_1, I_k)). \quad (7)$$

Здесь функция

$$\text{Sh}(d) = \begin{cases} 1 - d/\sigma_{\text{share}}, & d < \sigma_{\text{share}}; \\ 0, & d \geq \sigma_{\text{share}} \end{cases} \quad (7)$$

есть функция разделения; $d = d(I_1, I_k)$ — расстояние между индивидами I_1, I_k в некоторой метрике; σ_{share} — параметр разделения.

В системе PRADIS//FRONT индивиды для скрещивания (индивиды-родители) выбираются по следующему правилу. Первый индивид выбирается случайно (из промежуточной популяции P') вне зависимости от значения его функции пригодности. Второй индивид выбирается из той же популяции на некотором расстоянии от данного индивида (исходя из принципов поддержания разнообразия популяции). В случае, если нельзя

найти второго индивида на указанном расстоянии, в качестве этого индивида из множества P' выбирается случайный индивид.

Важнейшим оператором в любом генетическом алгоритме является оператор скрещивания. Существует множество разновидностей этого оператора: одноточечный, двухточечный, равномерный и пр. [5]. В системе PRADIS//FRONT скрещивание выполняется по следующему правилу: выбирается ведущий родитель; к значениям его генов прибавляется разность между соответствующими генами родителей, умноженная на коэффициент скрещивания CrossoverRate; индивид, полученный в результате скрещивания, записывается во множество P'' .

Известно также множество операторов мутации индивидов, различающихся числом индивидов, подверженных мутации, алгоритмами мутации, числом генов, участвующих в мутации, и пр. [5]. В системе PARET//FRONT используется оператор мутации, основанный на стандартном операторе, но со следующими отличиями:

- чтобы усилить эффект от выбранного правила скрещивания, все произведенное потомство подвергается мутации в одном гене хромосомы;
- ген модифицируется с помощью стохастического параметра мутации MutationRate.

Нами была сделана попытка использовать также классическую схему оператора мутации — инвертирование заданного числа битов в хромосоме. Однако тестирование показало, что такая схема обеспечивает существенно худшие результаты, чем схема, рассмотренная выше.

Распараллеливание вычислений в работе реализовано по схеме алгоритма GPGA. Основания для такого выбора следующие:

- ориентация работы на распределенные вычислительные системы. В результате из рассмотрения исключаются алгоритмы, ориентированные на SIMD-системы, а также алгоритмы, требующие большого числа коммуникаций между процессорами;
- высокая вычислительная сложность функции пригодности индивидов, обусловленная тем, что каждое вычисление такой функции требует интегрирования соответствующей системы обыкновенных дифференциальных уравнений и вычисления значений всех частных критериев оптимальности;
- простота реализации модели параллельных вычислений master-slave.

Организация программной системы PRADIS//FRONT

Для балансировки загрузки процессоров в системе PRADIS//FRONT используется равномер-

ная статическая балансировка [6], при которой каждый из slave-процессоров обрабатывает одинаковое число индивидов, назначаемых ему host-процессором.

Программная система PRADIS//FRONT состоит из серверной и клиентской частей. *Серверная часть* системы реализует следующие основные функции:

- инициализация приложения;
- поддержка файлов, с помощью которых осуществляется обмен данными с клиентской частью;
- поддержка очереди заданий для клиентов;
- синхронизация работы;
- выдача клиентам заданий для расчета;
- прием от клиентов результатов расчета;
- реализация генетического алгоритма;
- формирование файла результата.

Клиентская часть системы реализует следующие функции:

- поддержка работы программного комплекса PRADIS;
- запуск PRADIS для расчета;
- получение от PRADIS результатов расчета;
- прием от серверной части заданий на расчет;
- передача серверной части результатов расчета.

В системе PRADIS//FRONT определено два типа сообщений. Сообщения, посылаемые сервером и принимаемые клиентами, имеют формат

<Номер задачи для расчета> ,

а сообщения, посылаемые клиентами и принимаемые сервером, — формат

<Номер рассчитанной задачи> +
+ <ID процессора> + <Результаты расчета> .

Для синхронизации серверной и клиентских частей перед началом обмена сообщениями используется барьерная синхронизация (с помощью вызова MPI_Barrier (MPI_COMM_WORLD)).

Серверная часть системы. Сервер назначает задания клиентам по одному. Благодаря этому при сбое в работе одного из клиентов пропадает лишь то задание, которое данный клиент обрабатывал в момент сбоя.

Для формирования заданий сервер использует шаблон задания, в котором варьируемые параметры заменены специальными символами. Формирование заданий в серверной части системы выполняет функция replaceFiles(). Функция создает необходимое число заданий, сохраняя каждое из них в отдельном файле. Имена файлов формируются следующим образом: к имени файла, в котором находится шаблон задания, добавляется постфикс "_multi_" и <номер задания>. Если, например, имя файла шаблона есть test_multi.psl, то

идентификатор файла задания номер 97 выглядит как test_multi_multi_97_psl.

После того как все файлы с заданиями сформированы, сервер начинает выполнять только диспетчерские функции: раздает номера заданий клиентам; получает от них результаты расчетов и сохраняет их в буфере FunctionValues.

После завершения обработки клиентами всех заданий и приема сервером всех результатов обработки сервер с помощью функции SetMainObject осуществляет обработку данных из буфера FunctionValues: отбрасывает номера заданий и ID процессоров, упорядочивает результаты расчета в соответствии с номерами заданий и записывает результаты в массив funcSets, представляющий собой поле класса FunctionStudy. Затем сервер реализует одну итерацию генетического алгоритма.

Клиентская часть системы. Основными функциями клиентской части системы являются обмен данными с серверной частью и взаимодействие с программным комплексом PRADIS (решателем). Как указывалось выше, задания на расчет решателю, записанные на одном из его внутренних языков, готовит серверная часть системы, а клиентской части сообщается только номер задания.

В памяти каждого клиента содержится объект многовариантного анализа FunctionStudyMPI, у которого есть поля параметров и поля результатов расчета.

Запуск PRADIS осуществляет функция системы RunSingle(int n), где n — номер задания для расчета. В первую очередь функция RunSingle(...) формирует имя соответствующего файла задания. Сформированное имя файла служит аргументом для вызова функции решателя solver → Run(filename).

Программный комплекс PRADIS дает возможность использовать задания, написанные на языках PSL, PPL (Python), а также задания в формате схем предпроцессора Qucs [3]. Для каждого из этих способов требуется запускать свою часть решателя. Функция solver → Run(filename) автоматически запускает требуемый решатель (в зависимости от расширения имени файла, указанного в ее аргументе). В настоящей версии программной системы PRADIS//FRONT реализован запуск заданий только на языке PSL.

Результаты своей работы PRADIS записывает в выходной файл. Для примера, рассмотренного в п. "Серверная часть системы" этот файл имеет имя test_multi_multi_97.psl.dat.

Тестирование алгоритма приближенного построения множества Парето

Работоспособность алгоритма, рассмотренного в предыдущем разделе, проверялась на тестовых задачах ZDT₃, ZDT₆, DTLZ₄ из стандартного

Тестовые задачи

Задача	Частные критерии оптимальности	Размерность и множество допустимых значений D
ZDT ₃	$\phi_1(X) = x_1; \phi_2(X) = g \left[1 - \sqrt{\frac{\phi_1(X)}{g}} - \frac{\phi_1(X)}{g} \sin(10\pi\phi_1(X)) \right];$ $g(x_2, \dots, x_n) = 1 + 9 \sum_{i=2}^n \frac{x_i}{n-1}$	$n = 30, 0 \leq x_i \leq 1$
ZDT ₆	$\phi_1(X) = 1 - \exp(-4x_1)\sin^6(6\pi x_1);$ $\phi_2(X) = g \left[1 - \left(\frac{\phi_1(X)}{g} \right)^2 \right];$ $g(x_2, \dots, x_n) = 1 + 9 \left(\sum_{i=2}^n \frac{x_i}{n-1} \right)^{0,25}$	$n = 10, 0 \leq x_i \leq 1$
DTLZ ₄	$\phi_1(X) = (1 + g) \cos\left(\frac{\pi x_1^\alpha}{2}\right) \cos\left(\frac{\pi x_2^\alpha}{2}\right);$ $\phi_2(X) = (1 + g) \cos\left(\frac{\pi x_1^\alpha}{2}\right) \sin\left(\frac{\pi x_2^\alpha}{2}\right);$ $\phi_3(X) = (1 + g) \sin\left(\frac{\pi x_1^\alpha}{2}\right);$ $g(x_1, \dots, x_n) = \sum_{i=3}^n (x_i - 0,5)^2, \alpha = 1$	$n = 12, 0 \leq x_i \leq 1$

набора тестов для непрерывных многокритериальных задач [2] (см. таблицу). При решении всех тестовых задач использовались следующие значения параметров алгоритма:

- параметр деления $\sigma_{share} = 0,01$;
- доля отбираемых для скрещивания индивидов $T_r = 0,3$;
- параметр рекомбинации CrossoverRate = 0,7;
- мутационный параметр MutationRate = 1,0;
- число индивидов $N_p = 1000$;
- число поколений $N_{gen} = 1000$.

Результаты тестирования показывают, что используемый в системе алгоритм приближенного построения множества Парето позволяет получить хорошие решения для тестовых задач ZDT₃, ZDT₆ и удовлетворительные результаты — для задачи DTLZ₄.

Приближенное построение множества Парето для задачи многокритериальной оптимизации автомобильной коробки переключения передач

Программная система PRADIS//FRONT использована для построения фронта Парето в задаче многокритериальной оптимизации автомобильной коробки переключения передач. Рассматривается участок разгона заднеприводного



Рис. 1. Структурная схема модели

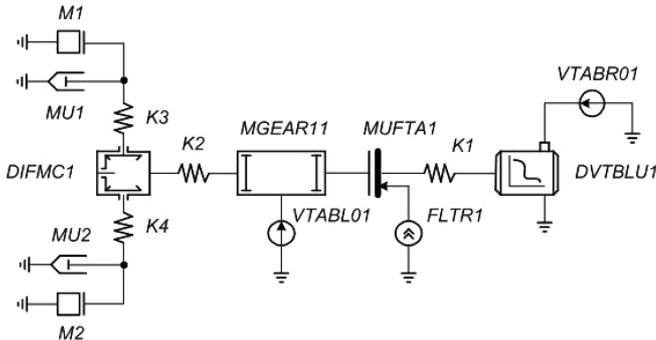


Рис. 2. Схема модели в редакторе Qucs

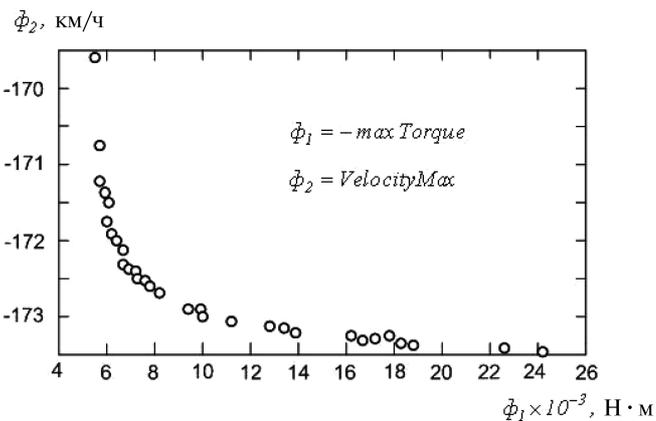


Рис. 3. Аппроксимация фронта Парето для двухкритериальной задачи

автомобиля массой 1500 кг с пятиступенчатой коробкой переключения передач при заданной характеристике двигателя. Разгон осуществляется из состояния покоя в течение 40 с. Полагается, что в начальный момент времени включена первая передача. При движении автомобиля учитывается сопротивление воздуха, трение качения, инерционные и упругие характеристики трансмиссии. Общая структура модели автомобиля представлена на рис. 1.

Упрощенная модель автомобиля в виде схемы редактора Qucs системы PRADIS приведена на

рис. 2. Основными элементами модели являются: двигатель DVTBLU1; муфта MUFTA1; коробка переключения передач MGEAR11; дифференциал DIFMC1; колеса M1, M2. Характеристика двигателя (зависимость момента от частоты вращения вала) задана таблично в виде кусочно-линейной функции.

Варьируемыми параметрами модели являются следующие десять параметров ($n = 10$):

- N_1, N_2, N_3, N_4, N_5 — передаточные отношения для всех пяти передач;
- T_1 — время сброса сцепления;
- T_2, T_3, T_4, T_5 — времена включения 2-, 3-й и т. д. передач.

Параллелепипед D допустимых значений вектора варьируемых параметров определяется неравенствами

$$1 \leq N_i \leq 20, N_i \in Z, i \in [1:5],$$

$$0 \leq T_1 \leq 1, T_1 \in R^1,$$

$$1 \leq T_i \leq 20, T_i \in R^1, i \in [2:5],$$

где Z — множество натуральных чисел; R_1 — пространство вещественных чисел. Множество допустимых значений D формируется неравенствами $T_2 < T_3 < T_4 < T_5$.

Выходные переменные модели: Time60 — время разгона до 60 км/ч; Time100 — время разгона до 100 км/ч; Velocity — скорость автомобиля; MaxTorque — максимальный момент в трансмиссии; Torque — момент в трансмиссии; Torque_Engine — момент на вале двигателя; V_Engine — частота вращения вала двигателя; Out_Gear — частота вращения выходного вала коробки переключения передач; Efficiency — КПД трансмиссии.

На основе указанных выходных параметров модели сформировано четыре следующих критерия оптимальности:

- максимальный момент в трансмиссии — MaxTorque (подлежит минимизации);
- максимальная скорость автомобиля — VelocityMax (подлежит максимизации);
- время разгона до 100 км/ч — Time100 (подлежит минимизации);
- эластичность автомобиля (время разгона с 60 до 100 км/ч) — Time60_Time100 (подлежит минимизации).

С помощью программной системы PRADIS//FRONT получены аппроксимации множества Парето для трех следующих задач многокритериальной оптимизации:

1) двухкритериальная задача ($\phi_1 = -\text{MaxTorque}$, $\phi_2 = \text{VelocityMax}$);

2) трехкритериальная задача ($\phi_1 = -\text{MaxTorque}$, $\phi_2 = \text{VelocityMax}$, $\phi_3 = -\text{Time100}$);

3) четырехкритериальная задача ($\phi_1 = -\text{MaxTorque}$, $\phi_2 = \text{VelocityMax}$, $\phi_3 = -\text{Time100}$, $\phi_4 = -\text{Time60_Time100}$).

Результаты расчетов для двухкритериальной задачи приведены на рис. 3.

Заключение

Работа впервые позволила использовать программный комплекс PRADIS в параллельном режиме.

Результаты работы показывают, что программная система PRADIS//FRONT является удобным и надежным средством решения задач многокритериальной оптимизации сложных динамических систем на основе построения соответствующих множеств Парето.

В развитие работы планируется реализовать запуск заданий на распределенной вычислительной системе не только на входном языке PSL системы PRADIS, но также на языке PPL и в формате схем предпроцессора Qucs. Кроме того, планируется реализация многопоточной серверной части приложения. Идея состоит в том, чтобы организовать дополнительные потоки, которые будут осуществлять обмен сообщениями с разными клиентами одновременно. Планируется также использовать

другие методы балансировки загрузки узлов распределенной вычислительной системы [6].

Список литературы

1. **Гуменникова А. П.** Адаптивные поисковые алгоритмы для решения сложных задач многокритериальной оптимизации: Дис. ... канд. техн. наук. Красноярск, 2006. 129 с.
2. **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.
3. **PRADIS.** Руководство к программе. <http://www.laduga.ru>
4. **Cantú-Paz E.** A Survey of Parallel Genetic Algorithms // Calculateurs Paralleles. Paris: Hermes, 1998. Vol. 10. N 2. available via ftp from: <ftp://ftpilligal.ge.uiuc.edu/pub/papers/Publications/cantupaz/survey.ps.Z>.
5. **Филиппов С. Ж.** Параметрическая идентификация систем поддержки принятия решений на основе параллельных генетических алгоритмов: Дис. ... канд. техн. наук. Санкт-Петербург, 2003. 152 с.
6. **Карпенко А. П., Федорук В. Г., Федорук Е. В.** Исследование эффективности балансировки загрузки многопроцессорной системы при распараллеливании одного класса вычислительных задач // Наука и образование: электронное научное издание. Инженерное образование, № гос. регистрации 0420700025. 2007. <http://technomag.edu.ru>, август.
7. **Карпенко А. П., Овчинников В. А., Семенихин А. С.** Распределенная программная система для построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельного генетического алгоритма // Наука и образование: электронное научное издание. Инженерное образование, № гос. регистрации 0420700025. август, 2008. <http://technomag.edu.ru>.

УДК 004.89 + 004.021

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

Кластерное расширение генетического алгоритма для решения многоэкстремальных задач оптимизации

Рассматривается новый подход к расширению функциональности стандартного генетического алгоритма на основе использования кластеров решений. Данный подход позволяет эффективно решать многоэкстремальные задачи оптимизации в многомерных поисковых пространствах. Представлены экспериментальные результаты применения данной модификации генетического алгоритма.

Ключевые слова: генетический алгоритм, кластер, многоэкстремальная оптимизация, кластеризация популяции, разнообразие популяции.

Введение

Многие практические задачи оптимизации имеют не одно, а множество оптимальных решений, т. е. являются многоэкстремальными. Определение таких решений одновременно трудоемкая и важная задача, так как предоставляет исследователю возможность выбирать среди множества наиболее приемлемые для себя оптимальные решения без потери качества результата. Однако автоматизация решения многоэкстремальных задач достаточно сложна ввиду отсутствия соответствующих методов, являющихся инвариантными к математическим моделям задач оптимизации и одновременно гарантирующими локализацию более одного экстремума. Поэтому актуальными являются разработка и исследование новых методов оптимизации, обладающих такими свойствами, например на основе генетических алгоритмов (ГА).

Стандартный генетический алгоритм является высокоэффективным средством глобальной оптимизации, но позволяет локализовать за один запуск лишь одно экстремальное решение, что су-

щественно снижает эффективность его использования при решении многоэкстремальных задач. Однако, обладая свойством параллелизма, ГА имеет полноценные возможности к нахождению всех оптимальных решений, распределенных в поисковом пространстве. Попытки реализации этого, как правило, связаны с интеграцией в ГА средств распределенного поиска, а также выделения и группировки по специальным признакам хромосом в популяции. В настоящее время можно выделить несколько подходов [1–4] разной эффективности для адаптации ГА к решению задач оптимизации с множеством экстремумов на основе:

- принципов имитационного моделирования;
- обработки локальных подпопуляций;
- обработки распределенных популяций.

Первый подход предполагает многократные (по предварительно подготовленному плану) запуски ГА с варьированием значений управляющих параметров и позволяет получить "хорошие" результаты лишь при малом числе экстремумов и "предсказуемом" пространстве решений.

Второй подход подразумевает выделение в основной популяции подпопуляций хромосом, имеющих определенные свойства. Как правило, для этого изменяются, реже добавляются новые, генетические операторы. Представителями таких алгоритмов являются ГА с логистической моделью [1], дифференциальной эволюции [2], а также алгоритмы, где в оператор отбора добавлено использование так называемых нишевых механизмов [3]. Они являются компьютерной (в рамках ГА) имитацией существующих в живой природе экосистем. Последние поддерживают естественное разнообразие и развитие живых организмов в зависимости от их роли (биологических ниш) в экосистемах. В отличие от искусственной эволюции единственной популяции в ГА, в результате которой получается множество одинаковых оптимальных решений, в экосистемах эволюция направлена на развитие разных непохожих особей в целях заполнения соответствующих ниш. В работе [5] обобщение этого принципа было названо *crowding* (уплотнение) и заключалось в формировании случайного набора подпопуляций, взаимного сравнения решений и исключения из них похожих (по их оптимальности) хромосом. Для реализации нишевого механизма в ГА используется на этапе селекции так называемая разделяющая *fitness*-функция (*fitness-sharing*). С ее помощью определяется влияние наличия группы близких хромосом в подпопуляции на повышение оптимальности решений.

Отмеченные модификации ГА позволяют достаточно эффективно поддерживать и сохранять разнообразие в популяции для предотвращения ее преждевременной сходимости к единственному

решению. Однако для своего функционирования данные ГА используют дополнительные к основным управляющие параметры, которые оказывают существенное влияние на эффективность генетического поиска, но их настройка представляет определенные трудности.

Последний подход объединяет алгоритмы, в которых используются разделенные популяции, эволюционирующие изолированно, либо взаимодействуя друг с другом, например для передачи хромосом, кодирующих перспективные решения. Группу алгоритмов, функционирующих по этим принципам, составляют параллельные ГА, а также ГА с моделью островов [1, 4]. В них популяции развиваются независимо друг от друга, но между ними по определенным правилам организуется миграция, которая способствует увеличению разнообразия генетического материала в них. Для эффективного использования этих алгоритмов также требуется настройка дополнительных управляющих параметров и возможность выполнения в распределенных вычислительных системах.

Предлагаемое кластерное расширение генетического алгоритма (КГА) сочетает в себе основные принципы последних двух подходов с привлечением минимального числа дополнительных управляющих параметров. В нем используется единственная подпопуляция, содержащая кластеры хромосом, формируемых по принципу фенотипического отличия друг от друга. Это отличает КГА от других модификаций ГА, в частности кластерно-ориентированного генетического алгоритма [1], где пространство поиска разбивается с помощью нишевых механизмов на участки в зависимости от границ значений *fitness*-функции.

1. Принципы кластерного расширения генетического алгоритма

В целях обеспечения всестороннего исследования пространства решений и предотвращения сходимости к единственному решению в КГА используется понятие кластера хромосом.

Под *кластером хромосом* понимается группа решений, имеющих похожие свойства, т. е. кодирующие их хромосомы с похожим фенотипом. Для определения меры близости могут быть использованы вещественная (Евклидова), бинарная (Хемминга) метрика D , а также другие, специальные относительно выбранного алфавита для кодирования хромосом, метрики [6]. Число кластеров зависит от задаваемого в качестве дополнительного управляющего параметра ГА радиуса гиперсферы кластера R_c . Хромосомы, располагающиеся в пределах R_c до центроида кластера Z_i , рассматриваются как похожие и принадлежащие Z_i .

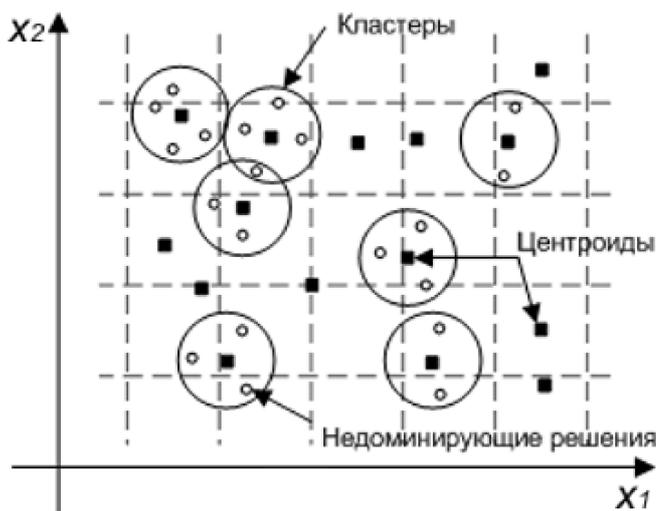


Рис. 1. Выделение кластеров для двумерного пространства решений

Множество центроидов кластеров Z_c образуют отдельную подпопуляцию хромосом. Для кластеризации элементов популяции используется принцип доминирования [3]. Пусть Z_1, Z_2, \dots, Z_k — k фрагментов популяции P_n , представляющих собой кластеры. Хромосома $C^* \in Z_i$ является доминирующей в кластере, если для $\forall C' \in Z_i: f(C^*) \leq f(C')$. Здесь предполагается, что решается задача минимизации. Знак нестрогого неравенства означает, что в кластере может быть более одного доминирующего решения. Поэтому хромосома C^* является центроидом кластера Z_i тогда и только тогда, если для $\forall C' \in Z_i: D(C^*, C') \leq R_c$. Важно, что отдельные хромосомы, соответствующие этому неравенству могут принадлежать одновременно и другим кластерам, а также доминировать сразу в нескольких кластерах (рис. 1).

Подпопуляция найденных центроидов кластеров представляет собой механизм поддержания разнообразия популяции для параллельного исследования всех областей поискового пространства. Для реализации этого кластеры по особым правилам копируются в следующее поколение. Поскольку все кластеры связаны с различными участками поискового пространства, то соответствующие им хромосомы в последующих поколениях станут основой для формирования шаблонов, содержащих коды участков пространства решений с другими экстремумами.

Известно, что при работе стандартного ГА популяция стягивается к единственному оптимальному решению. Предположим, что после t числа поколений распределение популяции в пространстве решений имеет вид,

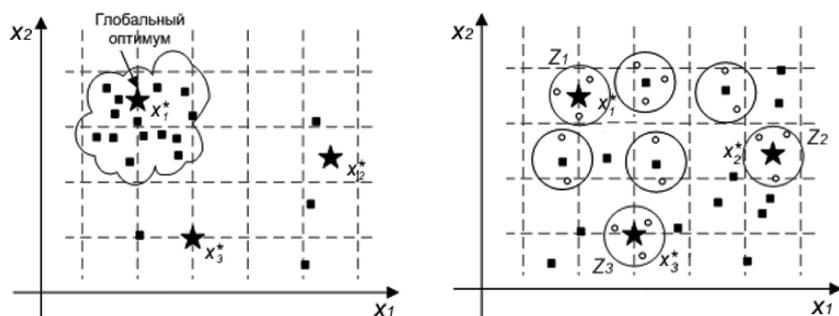


Рис. 2. Ограниченность стандартного ГА и возможности КГА при локализации группы экстремумов

приведенный на рис. 2. Тогда единственным оптимальным решением станет X_1^* , в то время как остальные, X_2^* и X_3^* , будут потеряны. Однако благодаря кластерам Z_1, Z_2, Z_3 сохраняется возможность исследования и локализации решений, находящихся в их окрестностях, несмотря на то, что решения, входящие в эти кластеры, могут и не быть оптимальными.

Такая возможность реализуется с помощью процедур выделения и копирования кластеров. Они определяются в текущей популяции, а затем сохраняются в отдельной подпопуляции. Представляя, в определенном смысле, лучшие среди "непохожих" между собой хромосом, центроиды кластеров копируются в новую популяцию, моделируя этим распределенную стратегию элитизма. Число кластеров может быть произвольным и зависит от свойств задачи или потребностей пользователя КГА.

2. Процедуры выделения и копирования кластеров

Процедура выделения кластеров состоит в определении N_z числа кластеров, определяемых координатами центроидов, каждый из которых соответствует хромосоме, которая доминирует над другими хромосомами-членами кластера, находящимися на расстоянии, не превышающем R_c , от центроида. Ниже приведен псевдокод алгоритма, выполняющего кластеризацию текущей популяции. В нем используются следующие обозначения: Z_c — множество кластеров, находящихся в поколении $G(t)$; алгоритм строит множество Z_c , последовательно исследуя хромосомы популяции $G(t)$ в порядке уменьшения их оптимальности и добавляя уже выбранные их индексы к множеству $MarkC$. Каждая выбранная хромосома C^* проверяется на принадлежность одному из уже найденных кластеров. Если такие кластеры в Z_c не обнаруживаются, то в Z_c добавляется новый кластер с центроидом в C^* .

Procedure LocateClusters

```
Begin
  Zc = ∅;
  MarkC = ∅;
  Foreach (C ∈ G(t))
    Begin
      Search the best C+: C+ ∈ G(t) ∧ C+ ∉
      MarkC
      MarkC = MarkC ∪ {C+}
      InCluster = False
      Foreach (C' ∈ Zc)
        If (d(C', C*) ≤ Rc) then
          InCluster = True;
          Break;
        End (If)
      End (Foreach)
      If (not InCluster) then
        Zc = Zc ∪ {C+}
      End (Foreach)
    End (LocateClusters)
```

Центроиды найденных кластеров Z_c сохраняются в отдельной подпопуляции $G(Z_c)$. Основная популяция подвергается применению генетических операторов и следовательно, претерпевает изменения, поэтому найденные кластеры теряются. Для предотвращения этого предварительно сохраненные центроиды по изложенному ниже алгоритму копируются в новую популяцию, тем самым восстанавливая "присутствие" ГА в соответствующих участках поискового пространства.

Procedure CopyClusters

```
Begin
  MarkC = ∅;
  Foreach (z ∈ Zc)
    Begin
      Select the worst C-: C- ∈ G(t+1) ∧ C- ∈
      z ∧ C- ∉ MarkC
      If (C- exist) then
        If (f(C-) < f(z)) then
          C- = z
        end (If)
      else
        Select the worst C- ∈ G(t+1);
        C- = z
      end (If)
      MarkC = MarkC ∪ {C-}
    End (CopyClusters)
```

Алгоритм процедуры описывает следующую последовательность действий для копирования кластеров в новую популяцию.

Для каждого ранее найденного кластера Z_{c_i} в новой популяции $G(t+1)$ определяется множество хромосом, входящих в этот кластер. Среди этого множества определяется "худшая" хромосома C^- , которая заменяется центроидом кластера. Если

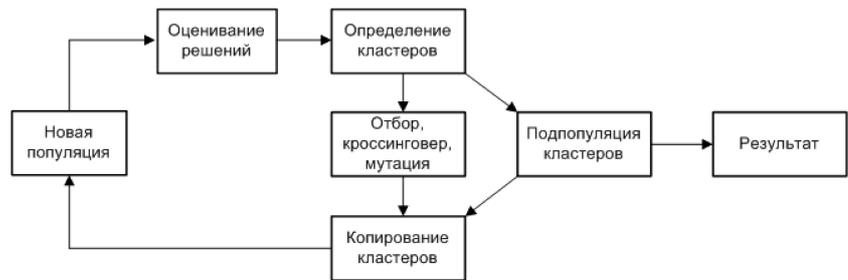


Рис. 3. Структура кластерной модификации генетического алгоритма

ли же для кластера Z_{c_i} не обнаруживаются соответствующие хромосомы в $G(t+1)$, то центроид этого кластера заменяет худшую хромосому новой популяции среди непомеченных хромосом, не принадлежащих ни одному из Z_{c_i} . Учитывая, что $N_z < N_p$, где N_p — размер популяции, такие хромосомы всегда будут в ней присутствовать.

Таким образом, все кластеры либо дополняют собой новую популяцию, либо заменяют в ней аналогичные хромосомы, но с меньшей оптимальностью. Возможен крайний случай, когда ни один кластер не будет сохранен в новой популяции, если она будет содержать, по крайней мере, одну хромосому для каждого кластера Z_c и если каждая из них будет иметь значение *fitness*-функции лучше, чем у центроида соответствующего кластера.

Таким образом, вычислительно КГА отличается от стандартного генетического алгоритма в выполнении двух процедур и настройке одного дополнительного параметра (рис. 3).

Использование дополнительных операций определения и копирования кластеров увеличивает общую временную сложность ГА с кластерным расширением. Основные временные затраты здесь связаны с необходимостью вычисления меры близости между хромосомами для отнесения их к тому или иному кластеру. Аналитическая оценка добавочной временной сложности указанных операций показала, что общая временная сложность вычисления меры близости варьируется от линейной $O(N_p)$ до квадратичной $O(N_p^2)$. Второй случай соответствует ситуации, когда R_c мало и, следовательно, обрабатывается большое число кластеров.

3. Вычислительные особенности кластерной модификации генетического алгоритма

Параметр R_c — радиус гиперсферы кластера играет ключевую роль при поиске решений. Уменьшение значения этого параметра приводит к увеличению числа кластеров, и наоборот. Частный случай, когда $R_c \rightarrow 1$, соответствует стратегии элитизма, когда из популяции в популяцию копируется единственный кластер, соответствующий лучшему решению.

Подобно другим управляющим параметрам генетического алгоритма значение Rc определяется экспериментально, например, с учетом того, чтобы его значение было больше меры близости между двумя наиболее различными решениями $Rc \geq 2d$. Также для определения крайней верхней границы этого параметра может быть использована

$$Rc^+ = \frac{1}{2} \sqrt{\sum_{i=1}^n (x_i^+ - x_i^-)^2}.$$

Экспериментальный анализ динамики изменения числа кластеров в процессе работы КГА показал сильную зависимость скорости и результативности поиска от значений Np и Rc . Причем выбор значений для последнего параметра, в общем случае, может быть выполнен лишь экспериментально с учетом следующих рекомендаций:

- кластеры должны охватывать все пространство поиска;
- значение Rc должно быть достаточно для генерации кластеров, покрывающих все оптимальные решения;
- интенсивную мутацию имеет смысл применять только при большом числе поколений.

После достижения заданного числа поколений подпопуляция кластеров содержит разнообразные решения с различной оптимальностью. Для выделения из них группы решений, определяющих глобальный экстремум (на данном уровне решения задачи) используется следующий критерий $|f(Zc_i) - f(C^*)| \leq \varepsilon$, где $f(Zc_i)$ — оптимальность i -го центра кластера; $f(C^*)$ — оптимальность лучшей хромосомы последней популяции; $\varepsilon > 0$ — параметр, определяющий верхнюю границу "глобального" оптимума.

Задавая значение параметра ε , можно изменять уровень оптимальности решений от глобального до рационального. Последнее особенно важно для многих практических задач, когда необходимо обеспечить выбор среди рациональных решений, найденных за короткое время с минимальными затратами на настройку КГА.

4. Экспериментальное исследование кластерного расширения генетического алгоритма

Экспериментальное исследование КГА связано не только с оценкой его эффективности при решении мультимодальных оптимизационных задач, но и определением влияния ключевых для него управляющих параметров Np , Rc на результативность поиска. Одновременно должно быть уменьшено число вычислений, необходимых для определения решений.

Возможности КГА исследовались совместно с другими ГА при решении тестовой многоэкстре-

мальной оптимизационной задачи (*Shubert's function*) следующего вида [7]:

$$f(x_1, x_2) = \prod_{i=1}^2 \sum_{j=1}^5 j \cos[(j+1)x_i + j],$$

где $-10 \leq x_i \leq 10$.

Данная тестовая функция (рис. 4, см. третью сторону обложки) имеет достаточно сложный рельеф, содержит 743 локальных, 18 глобальных минимумов со значением в них целевой функции равной $-186,73$.

Для сравнительной оценки вместе с КГА использовались стандартный ГА, метод дифференциальной эволюции [2] и ГА с нишевой надстройкой [3]. Эффективность алгоритмов определялась в соответствии с числом найденных экстремумов, а также по общему числу вычислений *fitness*-функции при оценке решений. Учитывая, что основная временная трудоемкость генетических алгоритмов связана с необходимостью вычисления *fitness*-функции, эти алгоритмы существенно отличаются по сложности их программной реализации. Для каждого генетического алгоритма экспериментально определялся индивидуальный (кроме числа поколений и размера популяций — для всех алгоритмов они были одинаковы) набор значений управляющих параметров, соответствующий лучшему результату, полученному в процессе оптимизации. Затем выполнялась серия запусков всех ГА для окончательной оценки их эффективности, полученные результаты усреднялись в том числе и в процентном отношении (см. таблицу).

Из таблицы видно, что КГА во всех случаях нашел все экстремумы. В отличие от остальных "нестандартных" генетических алгоритмов КГА сумел сделать это быстрее и с необходимостью настройки меньшего числа дополнительных управляющих параметров.

На рис. 5 (см. третью сторону обложки) представлена плотность исследования пространства решений при локализации экстремумов. Видно, что результат достигнут не за счет хаотичного ис-

Сравнение эффективности различных модификаций ГА при решении многоэкстремальных задач оптимизации

Алгоритм	Найденные экстремумы (процент от общего числа), %	Число вычислений <i>fitness</i> -функции (среднее по числу запусков ГА)
Стандартный генетический алгоритм	10	30 200
Алгоритм дифференциальной эволюции	80	97 372
ГА с нишевой надстройкой	50	81 312
КГА	100	62 137

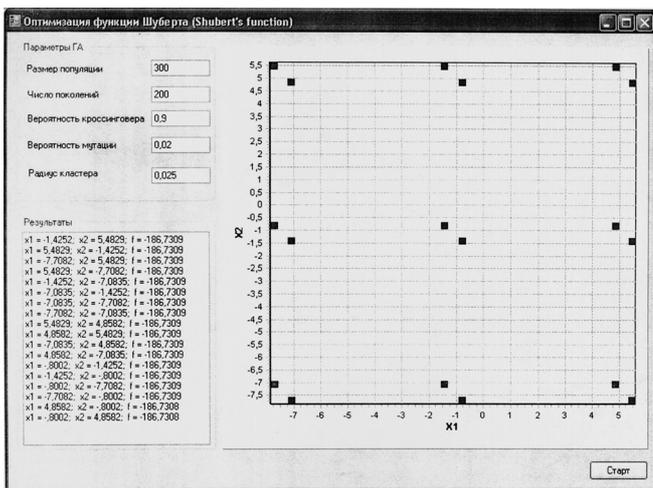


Рис. 6. Результаты оптимизации КГА мультимодальной функции

следования поискового пространства, а благодаря направленному синтезу решений в наиболее перспективных его участках.

Результаты оптимизации представлены на рис. 6. В ней приведены значения управляющих параметров, при которых алгоритм надежно находит все оптимальные решения, а также их координаты и относительное распределение в поисковом пространстве.

Также анализировалось влияние значений параметров, размер популяции и радиус кластера на результативность поиска. Было выявлено, что при несложной настройке можно определить такие их значения, при которых КГА находит все оптимумы за минимальное число поколений. Однако такая настройка должна быть индивидуальной для каждой задачи. В частности, для тестовой функции результаты показали, что увеличение R_c приводит к уменьшению числа глобальных оптимумов. Очевидно, что при больших R_c некоторые решения, соответствующие глобальным оптимумам могут размещаться в одном кластере и тем самым теряться. Граничное значение R_c , при кото-

ром определялись все 18 глобальных оптимумов за установленное время, составило 0,025, при большем значении их число уменьшалось. При $R_c > 0,4$ определялся только один глобальный оптимум, что соответствовало классической стратегии элитизма.

Заключение

Таким образом, использование в КГА принципов кластеризации позволяет всесторонне исследовать пространство решений благодаря сохранению в подпопуляции кластеров и дальнейшему тиражированию генетически разных хромосом. Варьируя значениями управляющего параметра, радиуса гиперсферы кластера, можно настраивать КГА на локализацию как группы глобальных, так и множества различных субоптимальных решений. Последнее особенно важно для многих практических задач, когда важно обеспечить выбор среди рациональных решений, найденных за короткое время с минимальными затратами на настройку ГА. Экспериментальные исследования КГА показали его эффективность относительно других средств мультимодальной оптимизации в сочетании с меньшими вычислительными затратами.

Список литературы

1. Гладков Л. А., Зинченко Л. А., Курейчик В. В., Курейчик В. М., Лебедев Б. К., Нужнов Е. В., Сорокин С. Н. Методы генетического поиска. Таганрог: Изд-во ТРТУ, 2002. 122 с.
2. Пупков К. А., Феоктистов В. А. Алгоритм дифференциальной эволюции для задач технического проектирования // Информационные технологии. 2004. № 8. С. 25—31.
3. Bäck T., Fogel D., Michalewicz Z. Evolutionary computation 2. Advanced algorithms and operators. IOP Publishing, 2000. 304 p.
4. Chambers L. Practical handbook of genetic algorithms: complex coding systems. CRC Press, Inc., 1999. 659 p.
5. De Jong K. Analysis of behavior of class of genetic adaptive systems: Ph D. Thesis / K. De Jong. — University of Michigan, 1975.
6. Mattiussi C., Waibel M., Floreano D. Measures of diversity for populations and distances between individuals with highly reorganizable genomes // Evolutionary computation. 2004. N 12 (4). P. 495—515.
7. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. Berlin: Springer-Verlag, 1996. 387 p.

Новая книга

Критически важные объекты и кибертерроризм. Часть 1. Системный подход к организации противодействия. / О. О. Андреев и др. Под ред. В. А. Васенина. — М.: МЦНМ. 2008. — 398 с.

В первой части коллективного учебно-научного издания излагаются методические основы и практические аспекты перехода к разработке национальной системы мер и мероприятий, к созданию механизмов, моделей и средств защиты объектов критически важных для государства инфраструктур от деструктивных информационных воздействий на них с террористическими целями. На основе анализа международного опыта на этом направлении и российских реалий предлагаются концептуальные положения такой системы. Описываются способы идентификации и систематизации критических объектов, субъектов информационного взаимодействия, элементы среды окружения, характеризующие предметную область.

Рассматриваются подходы к разработке политики безопасного использования информационных активов, вычислительных и телекоммуникационных ресурсов критически важных объектов на всех этапах их жизненного цикла, в том числе — к формированию профилей защиты.

В. А. Антонов, д-р техн. наук, вед. науч. сотр.,
Институт горного дела УрО РАН,
г. Екатеринбург,
e-mail: antonov@igd.uran.ru

Информационная технология оптимизации новым численным методом и построения на его основе полиномиальных степенных трендов с самоопределяющимися показателями степени

Изложена суть нового численного метода приближенной параболической вершины (МППВ), предназначенного для нахождения экстремума заданной функции. Рассмотрена развитая на основе МППВ методология и информационная технология построения по случайно распределенным значениям зависимой величины полиномиальных степенных трендов с самоопределяющимися показателями степени. Предложена классификация трендов такого типа. Представлена система методических и технологических приемов их получения. Приведены практические примеры построения указанных трендов.

Ключевые слова: экстремум, оптимизация, численный метод, аппроксимирующая функция, информационная технология, регрессия, тренд, коэффициент детерминации.

Введение

В информационных технологиях для решения задач оптимизации, сводящихся к нахождению экстремума заданной функции, в большинстве случаев применяют численные методы, среди которых представляет интерес метод имитации, в частности, аппроксимации заданной функции в области ее экстремума более простой в аналитическом смысле функцией. В разделе 1 данной статьи изложен предлагаемый численный метод нахождения экстремума заданной функции с применением параболической аппроксимирующей функции, имеющей единственный экстремум в виде максимума или минимума, который условно будем называть вершиной. Поскольку аппроксимация проводится с применением последовательно повторяющихся и сходящихся расчетных процедур, то методу присвоено соответствующее название — метод приближений параболической вершины (метод ППВ, или сокращенно МППВ). Показана возможность применения МППВ в решении уравнений регрессии, сводящихся к линейному виду относительно оцениваемых коэф-

фициентов. В разделе 2 статьи на основе частного приложения МППВ рассмотрена методология и информационная технология построения серии трендов случайно распределенных значений зависимой величины в виде степенных полиномов, где показатели степени не назначаются априори, а самоопределяются в области рациональных чисел.

1. Оптимизационный метод ППВ

Допустим, что заданная функция $U(X^m)$, где m — не степень, а число аргументов X , т. е.

$$U(X^m) \equiv U(X_1, X_2, X_3, \dots, X_j, \dots, X_m),$$

однозначно определена в ограниченной области, имеет единственный экстремум, обладает свойствами гладкости и непрерывности. Задача состоит в отыскании совокупности значений аргументов, соответствующих экстремуму. В примененном для ее решения методе ППВ искомая совокупность аргументов находится путем последовательных приближений экстремальной области (вершины) также гладкой и непрерывной аппроксимирующей функции к области экстремума заданной функции.

Суть метода ППВ состоит в следующем. В каждом последовательном k -приближении экстремальную область заданной функции $U(X^m)$ аппроксимируют m -мерным параболоидом $P_k(X^m)$ вида

$$P_k = C_k + \sum_{j=1}^m (C_{j1k} X_j + C_{j2k} X_j^2),$$

где C_k, C_{j1k}, C_{j2k} — коэффициенты. Аппроксимацию проводят по таким опорным точкам, принадлежащим заданной функции и взятым в количестве $t = 2m + 1$, для которых при поиске в ней максимума или минимума выполняются соответствующие неравенства параболических коэффициентов $C_{j2k} < 0$ или $C_{j2k} > 0$. Далее рассчитывают совокупность аргументов $X_{1Bk}, X_{2Bk}, \dots, X_{mBk}$, соответствующих вершинной точке параболоида. Затем, принимая найденную совокупность вершинных аргументов и рассчитанное с их учетом значение функции $U(X_{Bk}^m)$, получают вновь образованную опорную точку. В следующем $(k + 1)$ -приближении заменяют одну из опорных точек с наименьшим значением U в случае поиска максимума заданной функции или с наибольшим значением U в случае поиска ее минимума на вновь образованную опорную точку. После замены опорной точки проверяют выполнение отмеченных неравенств параболических коэффициентов. В случае невыполнения условия $C_{j2k} < 0$ опорные точки с наименьшим значением функции U пере-

мещают в направлении ее увеличения, а при невыполнении $C_{j2k} > 0$ опорные точки с наибольшим значением функции U перемещают в направлении ее уменьшения до восстановления требуемых неравенств. В процессе k - или $(k + 1)$ -приближения опорные точки, наиболее удаленные от экстремума заданной функции, заменяются опорными точками, менее от него удаленными. Поэтому с ростом номера приближения опорные точки располагаются все ближе к экстремуму заданной функции, и качество аппроксимации экстремума вершиной параболоида улучшается. На стадии приближений $k > t$ все опорные точки становятся расчетными. При дальнейшем росте номера приближения область опорных точек уменьшается, и они все в большей мере концентрируются вокруг точки экстремума заданной функции. При этом совокупность аргументов, рассчитанных по вершине параболоида, все меньше отличается от совокупности искомым аргументов, соответствующих экстремуму функции $U(X^m)$.

Достаточность приближений оценивается сравнением сколь угодно малой допустимой, т. е. заданной, простой средней относительной погрешности δ определения аргументов экстремума функции $U(X^m)$ с ее частным значением δ_{jk} , рассчитанным в текущем приближении для каждого аргумента по формуле

$$\delta_{jk} = \frac{\sum_{f=1}^t |X_{j(k+1)} - X_{jkf}|}{t|X_{j(k+1)}|},$$

где X_{jkf} — значение j -аргумента опорной f -точки в k -приближении; $X_{j(k+1)}$ — значение j -аргумента новой опорной точки, рассчитанной для $(k + 1)$ -приближения. Здесь принимаем, что совокупность аргументов, рассчитанных для $(k + 1)$ -приближения, наиболее вероятная. Приближения с номером k считаются достаточными, если для каждого аргумента выполняется неравенство

$$\delta_{jk} \leq \delta.$$

Если заданная функция $U(X^m)$ имеет не один, а несколько экстремумов, то для определения данным методом одного из них следует на область опорных точек наложить дополнительное ограничение по их положению в областях функциональной зависимости, примыкающих к искомому экстремуму, который обеспечивает локализацию вершины аппроксимирующего параболоида в обозначенных ограниченных интервалах изменения аргументов. При определении глобального (наибольшего или наименьшего) экстремума такие интервалы могут быть весьма широкими. В наиболее простых случаях аппроксимирующий

параболоид имеет геометрическое толкование. Например, в случае $m = 1$ ($t = 3$) функция $P(X)$ — обычная парабола, а в случае $m = 2$ ($t = 5$) функция $P(X_1, X_2)$ — эллиптический параболоид.

Одним из важных информационных приложений МППВ является решение уравнения регрессии случайной зависимой величины, которое с учетом дополнительных детерминированных аргументов μ_j сводится к линейной стохастической зависимости следующего вида:

$$Y_i = A_1 Z_{1i}(X_{1i}, \mu_1) + A_2 Z_{2i}(X_{2i}, \mu_2) + \dots + A_j Z_{ji}(X_{ji}, \mu_j) + \dots + A_m Z_{mi}(X_{mi}, \mu_m) + B + \varepsilon_i, \quad (1)$$

где Y_i — заданное случайное значение зависимой величины, $i = 1, 2, 3, \dots, n$; n — число случайных значений зависимой величины ($n > m$); X_{ji} — заданное детерминированное i -значение независимой j -величины (основного j -аргумента); ε_i — случайное отклонение зависимой величины от линии регрессии, т. е. от ее тренда; A_1, A_2, \dots, A_m, B — коэффициенты тренда; $Z_{ji}(X_{ji}, \mu_j)$ — частное i -значение непрерывной монотонно изменяющейся функции $Z_j(X_j, \mu_j)$. Функция $Z_j(X_j, \mu_j)$ по отношению к дополнительному аргументу может быть, например, степенной $Z = X^\mu$ ($X > 0$), показательной $Z = \mu^x$ ($\mu > 0$), экспоненциальной $Z = e^{\mu x}$.

При заданных фиксированных значениях μ_j коэффициенты тренда оцениваются методом наименьших квадратов (МНК), где сумма квадратов

$$Q = \sum_{i=1}^n \varepsilon_i^2$$

имеет единственный экстремум в виде минимума [1]. Допуская изменение μ_j и учитывая оценку коэффициентов тренда МНК, приходим к тому, что функция $Q(\mu^m)$ обладает свойствами гладкости и непрерывности, имеет в пространстве μ^m также единственный минимум, положение которого определяется МППВ. Следовательно, в результате совместного применения МНК и МППВ может быть определена такая совокупность дополнительных аргументов $\mu_1, \mu_2, \dots, \mu_m$, при которой сумма квадратов случайных отклонений Q является наименьшей из возможных минимальных сумм, оцениваемых МНК.

2. Полиномный степенной тренд с самоопределяющимися показателями степени

В экспериментальных исследованиях каких-либо природных процессов, проводящихся с применением регрессионного анализа выборки случайных значений зависимой величины Y_i и детерминированных значений аргументов, наиболее сложная закономерная связь между ними часто

выражается полиномными степенными трендами с априори назначенными целочисленными показателями степени. Однако природные закономерности редко совпадают с такими трендами. Поэтому выявление достоверных закономерностей распределения зависимой величины, построение детерминированных моделей исследуемых процессов и решение на их основе задач формирования и распространения соответствующей информации в широкой области интерполяции, особенно в области экстраполяции, становятся весьма затруднительными.

При решении отмеченной проблемы, очевидно, необходимо исходить из того, что показатели степени должны не назначаться априори, т. е. искусственно подбираться, а объективно определяться в пространстве рациональных чисел по характерным признакам выборки заданных значений зависимой и независимой величин. Рассмотрим соответствующие методологию и технологию создания такого тренда, включающую концепцию его построения, классификацию типов тренда и систематизацию методических приемов определения показателей степени. Основные положения концепции состоят в следующем.

2.1. Руководящий принцип математической формализации тренда состоит в том, что наличие каждого интервала, в пределах которого при частном изменении какого-либо j -аргумента величина тренда Y_T изменяется непрерывно и монотонно, т. е. возрастает или убывает, обусловлено объективно существующим закономерным влиянием и отображается в математической формуле трендового полинома соответствующим степенным членом вида $A_j X_j^{\mu_j}$. Число степенных членов, зависящих от выделенного j -аргумента, равно сумме, состоящей из числа экстремумов v_j в распределении зависимой величины вдоль данного аргумента и единицы ($v_j + 1$). Число всех степенных членов в трендовом полиноме определяется суммой $m + \sum_{j=1}^m v_j$.

2.2. Математическая формула трендового полинома с учетом постоянного коэффициента B и всех степенных членов имеет вид

$$Y_T = B + \sum_{j=1}^m \sum_{q=1}^{v_j+1} A_{jq} X_j^{\mu_{jq}}, \quad (2)$$

где q — номер интервала монотонности тренда вдоль j -аргумента; A_{jq} — коэффициент при степенной функции; μ_{jq} — показатель степени в виде рационального числа.

2.3. Коэффициенты и показатели степени, входящие в математическое выражение тренда, опре-

деляются совместным применением МНК и МППВ.

Построенные по указанному принципу тренды называются полиномными степенными с самоопределяющимися показателями степени (тренды СПС). В соответствии с принятой концепцией формирования трендов СПС предлагается классифицировать их, разделяя на типы по признакам числа аргументов и экстремумов в распределении зависимой величины. Тип тренда представляется перечнем его аргументов с указанием в скобках после каждого из них $v_i + 1$ обозначений показателей степени, соответствующих числу интервалов монотонности тренда вдоль данного аргумента. Например, если аргумент один и экстремумов нет, то тип тренда $X(\mu)$; если аргумент один, а экстремумов два, то тип тренда $X(\mu_1, \mu_2, \mu_3)$; если аргументов два и вдоль одного из них нет экстремума, а вдоль второго аргумента имеется один экстремум, то тип тренда $X_1(\mu_{11})X_2(\mu_{21}, \mu_{22})$.

Для оценки достоверности тренда определяется коэффициент его детерминации R^2 по формуле

$$R^2 = 1 - \frac{\sum_{i=1}^n \varepsilon_i^2}{\sum_{i=1}^n (Y_i - Y_T)^2}.$$

Поскольку коэффициенты тренда B и A_{jq} оцениваются МНК, то R^2 зависит от дополнительных аргументов — показателей степени. Отметим условие единственности экстремума функции $R^2(\mu^m)$, которое состоит в том, что $v_j = 0$, т. е. тренд (2) не должен содержать экстремумов. Тогда функция $R^2(\mu^m)$ с учетом подстановки в (1) $Z_{ji} = X_{ji}^{\mu_{ji}}$ аналогична функции $Q(\mu^m)$ с той лишь разницей, что имеет единственный максимум. При $v_j > 0$ в тренде есть экстремумы, и его полином содержит два или больше членов с одним и тем же основным j -аргументом. В этом случае функция $R^2(\mu^m)$ вследствие перестановок значений показателей степени в отмеченных членах полинома содержит $(v_{jM} + 1)!$ одинаковых по значениям максимумов, где v_{jM} — максимальное число экстремумов тренда, выбранное из числа имеющихся экстремумов вдоль каждого основного аргумента. Если в повторяющихся с j -аргументом членах полинома придать показателям степени фиксированные значения, то из функции $R^2(\mu^m)$ получается ее ортогональное сечение, в котором функция R^2 содержит единственный максимум.

Построение трендов СПС имеет некоторые принципиальные особенности. Показатели степени, соответствующие максимуму функции $R^2(\mu^m)$, определяются МНК и МППВ непосредственно в случае $v_j = 0$ или путем последователь-

ных s -приближений в случае $v_j > 0$. Каждое s -приближение содержит цикл последовательно проводимых кратных приближений, осуществляемых МНК и МППВ в ортогональных сечениях. Для выделения очередного сечения показателя степени, не задействованным в нем, придают фиксированные значения, ранее полученные для них в предыдущих сечениях. При чередовании сечений в выделенном s -приближении и увеличении номера s показатели степени, определяемые в текущем сечении, сходятся к искомым значениям. Коэффициент R_{\max}^2 , увеличиваясь через дискретные интервалы, постепенно приближается к какой-либо одной (из числа существующих) точке максимума функции $R^2(\mu^m)$. Приближение, после которого знак разности $R_{\max}^2(s+1) - R_{\max}^2(s)$ изменяется с плюса на минус, принимается достаточным. Чем меньше погрешность δ , заданная в сечениях, тем точнее s -приближениями определяются искомые показатели степени.

В общем случае выбор и последовательность чередования ортогональных сечений могут быть разными, и методика определения показателей степени в отдельных трендах становится многовариантной. С учетом соображений минимизации и компактности вычислительных операций предлагается преимущественно выбирать методики, содержащие в выделенных сечениях параболическую аппроксимацию с наибольшим числом аргументов, а следовательно, и с наибольшим числом опорных точек. Для сокращенного обозначения методики, применяемой в сечении и s -приближении, вводится соответствующая аббревиатура $MtT-(v_i + 1)$, где M — методика сечения; t — чис-

ло опорных точек; T — точки; $(v_i + 1)$ — кратность применения данной методики (по числу ортогональных сечений) в одном s -приближении. Например, для тренда типа $X(\mu_1, \mu_2)$ применяется методика МЗТ-2, содержащая в каждом s -приближении двукратно применяемую (в отдельных сечениях) методику с параболической аппроксимацией по трем опорным точкам, а для тренда $X_1(\mu_{11})X_2(\mu_{21}, \mu_{22})$ применяется методика М5Т-1 и МЗТ-1, состоящая в s -приближении из двух методик сечения, среди которых последовательно в одном сечении однократно применяется параболическая аппроксимация по пяти опорным точкам, а в другом сечении также однократно применяется аппроксимация по трем опорным точкам. Наиболее простые и часто используемые типы трендов, их математические формулы и обозначения методик расчетов приведены в таблице.

Отметим некоторые методические аспекты. Для обеспечения обусловленности матриц, по которым рассчитываются коэффициенты тренда и R^2 , модули фиксированных показателей степени, задающих сечение, и показателей степени опорных точек внутри сечения не должны совпадать. Процесс последовательных приближений внутри каждого сечения разделяется на начальную стадию локализации области максимума функции R^2 (заданной в данном сечении) и последующую стадию сходимости с ней вершины параболоида. На начальной стадии в нулевом приближении область показателей степени опорных точек выбирают такой, чтобы в ней содержались аргументы точки R_{\max}^2 , а также показатели степени новой опорной точки, рассчитанные для первого при-

Типы и соответствующие формулы трендов

m	v_i	Тип полиномно-степенного тренда СПС	Формула тренда	Методика определения показателей степени
1	0	$X(\mu)$	$A_{11}x_1^{\mu_{11}} + B$	МЗТ-1
1	1	$X(\mu_1, \mu_2)$	$A_{11}x_1^{\mu_{11}} + A_{12}x_1^{\mu_{12}} + B$	МЗТ-2
1	2	$X(\mu_1, \mu_2, \mu_3)$	$A_{11}x_1^{\mu_{11}} + A_{12}x_1^{\mu_{12}} + A_{13}x_1^{\mu_{13}} + B$	МЗТ-3
2	0; 0	$X_1(\mu_{11})X_2(\mu_{21})$	$A_{11}x_1^{\mu_{11}} + A_{21}x_2^{\mu_{21}} + B$	М5Т-1
2	0; 1	$X_1(\mu_{11})X_2(\mu_{21}, \mu_{22})$	$A_{11}x_1^{\mu_{11}} + A_{21}x_2^{\mu_{21}} + A_{22}x_2^{\mu_{22}} + B$	М5Т-1, МЗТ-1
2	1; 1	$X_1(\mu_{11}, \mu_{12})X_2(\mu_{21}, \mu_{22})$	$A_{11}x_1^{\mu_{11}} + A_{12}x_1^{\mu_{12}} + A_{21}x_2^{\mu_{21}} + A_{22}x_2^{\mu_{22}} + B$	М5Т-2
3	0; 0; 0	$X_1(\mu_{11})X_2(\mu_{21})X_3(\mu_{31})$	$A_{11}x_1^{\mu_{11}} + A_{21}x_2^{\mu_{21}} + A_{31}x_3^{\mu_{31}} + B$	М7Т-1
3	0; 0; 1	$X_1(\mu_{11})X_2(\mu_{21})X_3(\mu_{31}, \mu_{32})$	$A_{11}x_1^{\mu_{11}} + A_{21}x_2^{\mu_{21}} + A_{31}x_3^{\mu_{31}} + A_{32}x_3^{\mu_{32}} + B$	М7Т-1, МЗТ-1
3	0; 1; 1	$X_1(\mu_{11})X_2(\mu_{21}, \mu_{22})X_3(\mu_{31}, \mu_{32})$	$A_{11}x_1^{\mu_{11}} + A_{21}x_2^{\mu_{21}} + A_{22}x_2^{\mu_{22}} + A_{31}x_3^{\mu_{31}} + A_{32}x_3^{\mu_{32}} + B$	М7Т-1, М5Т-1
3	1; 1; 1	$X_1(\mu_{11}, \mu_{12})X_2(\mu_{21}, \mu_{22})X_3(\mu_{31}, \mu_{32})$	$A_{11}x_1^{\mu_{11}} + A_{12}x_1^{\mu_{12}} + A_{21}x_2^{\mu_{21}} + A_{22}x_2^{\mu_{22}} + A_{31}x_3^{\mu_{31}} + A_{32}x_3^{\mu_{32}} + B$	М7Т-2

ближения. В следующих приближениях начальной стадии в случаях нарушения условия $C_{j2k} < 0$, иногда возникающего при расположении части опорных точек в вогнутой области функции R^2 , опорные точки с наименьшим значением коэффициента R^2 передвигают в область повышенного его значений до тех пор, пока не устранится отмеченное нарушение. После замены всех заданных в нулевом приближении опорных точек на расчетные по вершине параболоида происходит их локализация около точки максимума функции R^2 , т. е. "захват" области максимума этой функции вершинной областью параболоида, и начальная стадия заканчивается. На последующей стадии вновь возникающие опорные точки рассчитывают исходя из рассчитанных точек в предыдущем приближении. Поэтому условие $C_{j2k} < 0$ выполняется, и с ростом номера приближения опорные точки сходятся к точке R^2_{max} .

В МЗТ для построения параболы задают три опорные точки. В нулевом приближении показатель степени первой точки μ_1 задают малым отрицательным числом. Показатель степени второй точки μ_2 задают также малым, но положительным числом. Показатель степени третьей точки μ_3 задают в случае $R^2_2 > R^2_1$ на интервале уменьшения R^2 , а в случае $R^2_2 < R^2_1$ на интервале увеличения R^2 таким, соответственно положительным или отрицательным числом, при котором положение вершины параболы находится в промежутке между наименьшим и наибольшим заданным показателем степени. В М5Т для построения эллиптического параболоида задают пять опорных точек.

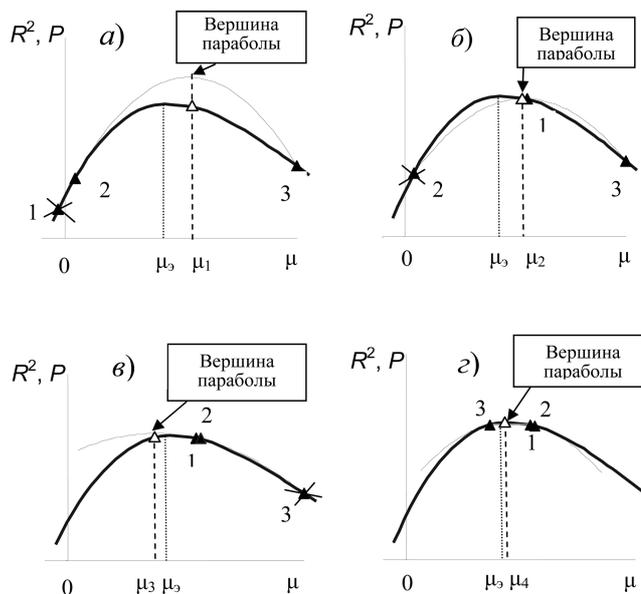


Рис. 1. Перемещение опорных точек в начальной стадии методики сечения МЗТ:

— график функции $R^2(\mu)$
 - - - график параболы $P(\mu)$

По наибольшему значению R^2 из трех точек, заданных около нуля в разных квадрантах системы координат (μ_{11}, μ_{21}) , выбирают квадрант, содержащий точку максимума R^2 . Оставшиеся две опорные точки располагают в выбранном квадранте по тем же соображениям в отношении каждого показателя степени, которые приведены в МЗТ.

Образование и перемещение опорных точек на начальных стадиях МЗТ в случае $R^2_2 > R^2_1$, а также М5Т в случае расположения максимума R^2 в первом квадранте показано, соответственно, на рис. 1 и 2. Опорные точки, по которым строится

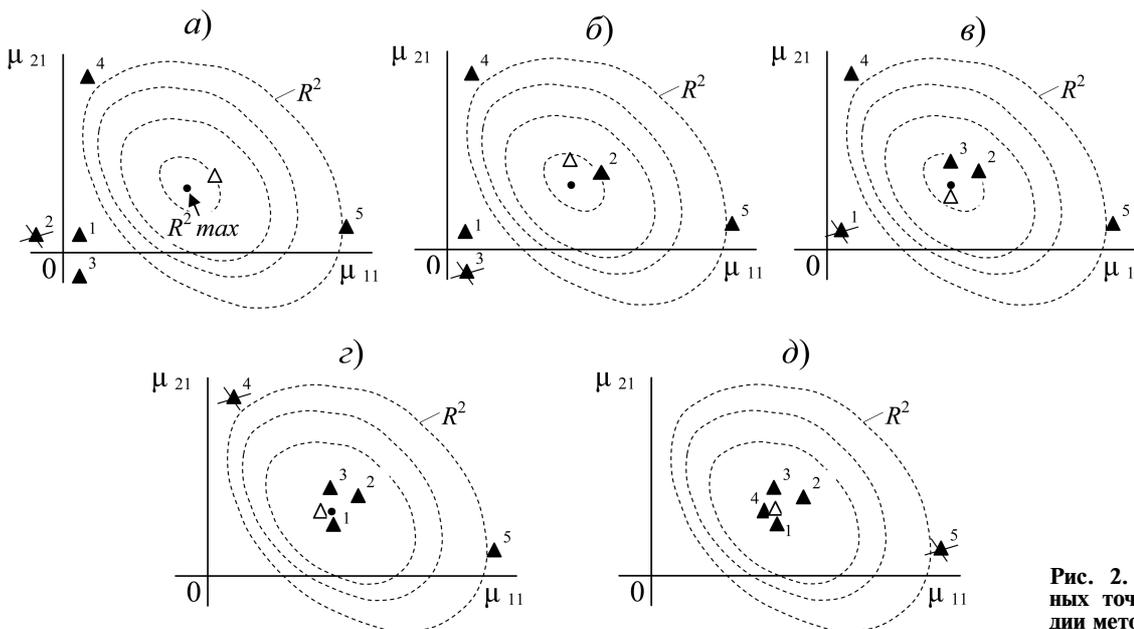


Рис. 2. Перемещение опорных точек в начальной стадии методики сечения М5Т

парабола в МЗТ и параболоид в М5Т, обозначены темными треугольниками и пронумерованы. Вновь образованные опорные точки, рассчитанные по вершине параболы или параболоида, обозначены треугольником с белым полем внутри. Выбывающая точка предыдущего приближения с наименьшим коэффициентом детерминации перечеркнута крестиком. На фрагментах *a*, *б*, *в* МЗТ показано образование расчетной опорной точки по итогам соответственно нулевого, первого и второго приближений. На фрагменте *г* показано образование следующей опорной точки в условиях локализации параболической вершины. С ростом номера приближения вновь образованная опорная точка располагается все ближе к точке максимума графика R^2 , имеющей искомым показателем степени μ_3 . На фрагментах *a*, *б*, *в*, *г*, *д* М5Т, начиная с нулевого по четвертое приближение, показан процесс концентрации опорных точек и последовательное перемещение их к точке R_{\max}^2 .

Некоторые практические результаты информационной технологии МППВ и построения полиномиальных степенных трендов СПС иллюстрированы рисунками. На рис. 3 представлены одномерно распределенные заданные случайные (фрагменты *a* и *б*) и детерминированные (фрагмент *в*) значения зависимой величины, а также графики соответствующих трендов. На фрагменте *a* представлен тренд типа $X(\mu)$ с отрицательным показателем степени $\mu = -0,8099$. Поэтому с ростом аргумента X график тренда приближается к асимптотическому значению 26,33, равному постоянному члену в его математической формуле. На фрагменте *б* приведен тренд типа $X(\mu_1, \mu_2)$, где $\mu_1 = 0,659$, $\mu_2 = -0,164$. График тренда, экстраполированный на интервал X от 160 до 400, имеет плавное, монотонное течение, без отклонений в сторону резкого возрастания или убывания, так характерных для полиномов с целочисленными показателями степени. На фрагменте *в* показано, как по нескольким заданным значениям, принадлежавшим неизвестной полиномиальной степенной функции $Y(X)$, трендом типа $X(\mu_1, \mu_2, \mu_3)$, где $\mu_1 = 1,67$, $\mu_2 = -0,73$, $\mu_3 = 0,81$, установлено с некоторой малой погрешностью ($R^2 = 0,999999$) ее математическое выражение.

На рис. 4 и 5 (см. третью сторону обложки) представлены двумерные распределения заданных случайных значений зависимой величины (фрагменты *a*) и поверхности трендов соответствующих типов $X_1(\mu_{11})X_2(\mu_{21})$ и $X_1(\mu_{11}, \mu_{12})X_2(\mu_{21})$ (фрагменты *б*). В тренде $X_1(\mu_{11})X_2(\mu_{21})$ определены показатели степени $\mu_{11} = 1,83$, $\mu_{21} = 0,52$. Его поверхность экстраполирована на внешнюю область, по площади в 4 раза превышающую ту, которую занимали исходные данные. В области экс-

траполяции поверхность тренда изменяется плавно и монотонно в соответствии с закономерностью, заданной исходными данными. Такая же ситуация и в тренде $X_1(\mu_{11}, \mu_{12})X_2(\mu_{21})$, где $\mu_{11} = 0,62$, $\mu_{12} = 1,58$, $\mu_{21} = 1,21$, с той лишь разницей, что в исходных данных случайные значения зависимой величины имеют разный алгебраический знак, и поверхность тренда значительно сложнее из-за наличия экстремума в направлении аргумента X_1 .

Достоверность интерполяции зависимой величины в широких интервалах и экстраполяции полиномиальными степенными трендами СПС значительно выше по сравнению с трендами, где показатели степени представлены целыми числами. Тем не менее, выбрать однозначно тип тренда можно, когда известны особенности и свойства искомой трендовой зависимости. Однако часто таких сведений нет, но требуется определить тренд, выражающий закономерность изменения зависимой величины с максимально возможной достоверностью. Если есть уверенность, что искомый тренд представлен степенным полиномом, то задача решается в случае, когда значения зависимой величины заданы в известных доверительных интервалах. Эти интервалы могут выражаться, например, относительной погрешностью тех средств или методов, с применением которых получены (косвенно рассчитаны или непосредственно измерены) заданные значения зависимой величины. Очевидно, что выбор искомого тренда должен быть ориентирован на соответствие его коэффициента детерминации R^2 фактически имеющимся доверительным интервалам в заданных значениях зависимой величины. С учетом таких интервалов рассчитывается коэффициент заданной детерминации R_3^2 . Наиболее достоверным трендом СПС из числа имеющихся типов можно считать такой, в котором абсолютная разность между коэффициентами R_3^2 и R^2 минимальная.

Заключение

Отметим преимущества и недостатки МППВ по сравнению с известными методами оптимизации. По исполнению и количеству численных расчетов МППВ, несомненно, намного проще многомерных методов прямого поиска, градиентных или второго порядка, например, итерационных методов Ньютона — Гаусса или Левенберга — Марквардта, в которых задача оптимизации решается с помощью сложных матричных расчетов и коррекцией направления поиска в каждой из многочисленных итераций. Для сравнения отметим, что в большинстве практических случаев применения МППВ при определении показате-

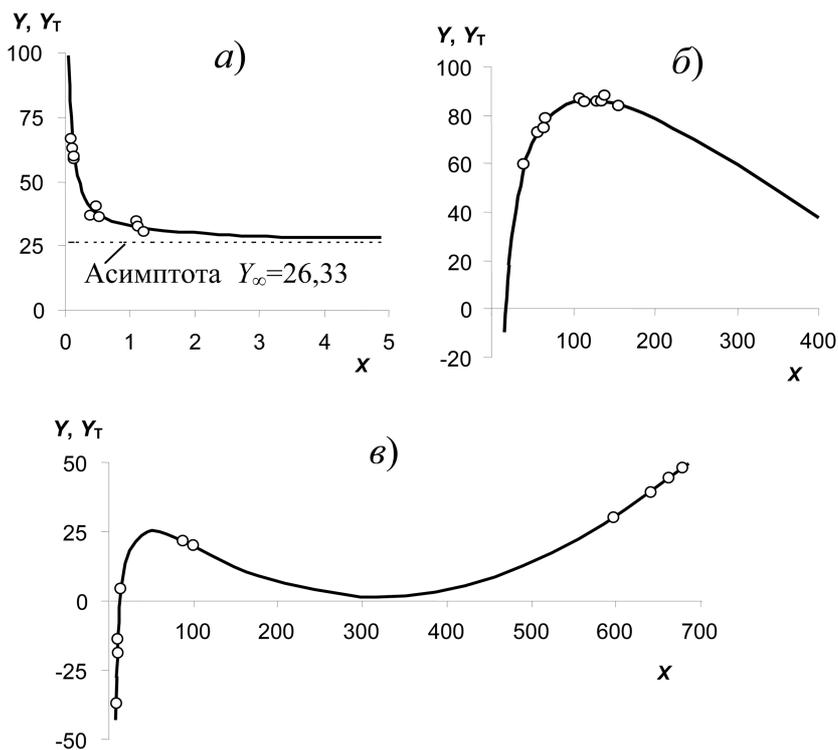


Рис. 3. Примеры построения одномерного полиномиального степенного тренда СПС
 ○ — заданное значение зависимой величины Y ; а — тренд типа $X(\mu)$, $Y_T = 6,439075X^{-0,809919} + 26,327691$, $\delta = 10^{-4}$, $R^2 = 0,9827$; б — тренд типа $X(\mu_1, \mu_2)$, $Y_T = 558,85 - 4,26X^{0,659} - 883,32X^{-0,164}$, $\delta = 10^{-4}$, $R^2 = 0,9901$; в — тренд типа $X(\mu_1, \mu_2, \mu_3)$, $Y_T = 0,004X^{1,67} - 450X^{-0,73} - 1,22X^{0,81} + 77,3$, $\delta = 10^{-6}$, $R^2 = 0,999999$

лей степени по методике сечения МЗТ или М5Т для достижения заданной погрешности расчетов требуется, соответственно, не больше 10 или 13 итераций. Тем не менее, к недостаткам МППВ следует отнести ограничения, связанные с требованиями к гладкости и непрерывности заданной функции, а к недостаткам построенных на его основе трендов СПС — то обстоятельство, что исходные данные необходимо приводить к услови-

выходу формул и графиков тренда.

Список литературы

1. Демиденко Е. З. Линейная и нелинейная регрессии. М.: Финансы и статистика, 1981.
2. Антонов В. А., Яковлев М. В. Полиномиальные степенные тренды с самоопределяющимися показателями степени (тренды ПС СПС): программа для ЭВМ, Российская Федерация; Ресурсы программ для ЭВМ, регистр. номер 2009611347. Екатеринбург. — ИГД УрО РАН. 2009.

Новая книга

Критически важные объекты и кибертерроризм. Часть 2. Аспекты программной реализации средств противодействия. / О. О. Андреев и др. Под ред. В. А. Васенина. — М.: МЦНМ. 2008. — 607 с.

Вторая часть коллективного учебно-научного издания посвящена изложению опыта и результатов, полученных авторами в ходе предпроектных исследований (эскизного проектирования), направленных на создание "линейки" программных средств защиты критически важных объектов от деструктивных информационных воздействий. Объективная оценка технологических трудностей, затрат ресурсов и времени на создание подобных средств является важным отправным фактором для планирования таких работ. В книге анализируются различные подходы к их проведению, описывается полигон, который использовался для тестовых испытаний прототипов программных средств.

Достаточно подробно в книге отражены результаты разработки представительного набора ("линейки") прототипов таких средств с учетом требований к ним, которые диктуются положениями соответствующих стандартов и нормативно-регулирующих документов.

УДК 519.633.6

А. П. Потапов, аспирант,
И. Б. Петров, д-р физ.-мат. наук, зав. каф.,
 Московский физико-технический институт
 (государственный университет),
 e-mail: potapov@7ka.mipt.ru

Моделирование высокоскоростных соударений методом сглаженных частиц (SPH)

Рассмотрены вопросы, связанные с моделированием высокоскоростных соударений различными вариантами метода сглаженных частиц при решении задач механики твердого деформируемого тела. Получены численные результаты решения задачи о распаде произвольного разрыва и проведен сравнительный анализ результатов.

Ключевые слова: метод сглаженных частиц, упругопластическая модель, распад разрыва, трехмерный алгоритм.

Введение

Современные проблемы безопасности жилищных и промышленных построек требуют решения трехмерных задач механики твердого деформируемого тела со сложной геометрией. Решения часто имеют области больших деформаций и разрушений, сопровождающихся разлетом материала. Численное решение такого рода задач сеточными методами сопряжено с большими трудностями, такими как построение трехмерной сетки, необходимостью ее периодической перестройки. Альтернативным вариантом решения такого класса задач является метод сглаженных частиц (*Smooth Particle Hydrodynamics* — SPH) [1, 2]. К достоинствам этого метода стоит отнести простоту реализации трехмерного варианта и возможность моделировать разлет материала без применения специальных техник.

Важным моментом моделирования является изучение волновой картины происходящих процессов. Рассмотрим решение задачи о распаде разрыва.

Метод сглаженных частиц (SPH)

Основная суть метода заключается в приближении тождества

$$a(x) = \int_R a(x') \delta(x' - x) dx'$$

(где $a(x)$ — интерполируемая физическая величина; x — координата точки, в которой необходимо вычислить значение) следующей цепочкой преобразований.

Вначале мы заменяем обобщенную функцию $\delta(x)$ аналитической функцией $\omega(x' - x, h)$, которую называют ядром сглаживания, а h — радиусом сглаживания. В результате получим

$$\bar{a}(x) = \int_R a(x') \omega(x' - x, h) dx'.$$

Ядро $\omega(x' - x, h)$ должно удовлетворять условиям

$$\int_R \omega(x, h) dx' = 1 \text{ и } \omega(x, h) \xrightarrow{h \rightarrow 0} \delta(x).$$

В работах Моногана доказано, что при соблюдении этих условий аппроксимация обеспечивает порядок $O(h^2)$. В расчетах используется функция вида $\omega(x, h) = e^{-(x/h)^2}$, которая удовлетворяет указанным условиям.

Следующее преобразование состоит в замене интегрирования суммированием по частицам-соседям:

$$\bar{a}(x) = \sum_i \frac{m_i a_i}{\rho_i} \omega(x_i - x, h),$$

где ρ_i — плотность частицы; m_i — ее масса.

Использование такой аппроксимации существенно упрощает вычисление градиента полевой функции $\frac{\partial a(x)}{\partial x_\alpha}$, так как достаточно аналитически продифференцировать ядро сглаживания, что даст

$$\frac{\partial \bar{a}(x)}{\partial x_\alpha} = \sum_i \frac{m_i a_i}{\rho_i} \frac{\partial \omega(x_i - x, h)}{\partial x_\alpha}.$$

В данной работе состояние вещества описывается следующими функциями:

ρ — плотность, u^α — вектор скорости, $\sigma^{\alpha\beta}$ — тензор напряжений, e — внутренняя энергия, $\varepsilon^{\alpha\beta}$ — тензор скоростей деформации.

Численные аппроксимации уравнений механики деформируемого тела для i -й частицы, по-

строенные с помощью метода SPH, выглядят следующим образом:

$$\frac{d\rho_i}{dt} = -\sum_k m_k (u_k^\alpha - u_i^\alpha) \frac{\partial \omega_{ik}}{\partial x_i^\alpha};$$

$$\frac{du_i^\alpha}{dt} = \sum_k m_k \left(\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} + \frac{\sigma_k^{\alpha\beta}}{\rho_k^2} \right) \frac{\partial \omega_{ik}}{\partial x_i^\beta};$$

$$\frac{de_i}{dt} = \sum_k m_k (u_i^\alpha - u_k^\alpha) \left(\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} + \frac{\sigma_k^{\alpha\beta}}{\rho_k^2} \right) \frac{\partial \omega_{ik}}{\partial x_i^\beta};$$

$$\frac{dS_i^{\alpha\beta}}{dt} = 2\mu \left(\varepsilon_i^{\alpha\beta} - \frac{1}{3} \delta^{\alpha\beta} \varepsilon_i^{\alpha\gamma} \right) +$$

$$+ S_i^{\alpha\gamma} R_i^{\beta\gamma} + S_i^{\gamma\beta} R_i^{\alpha\gamma};$$

$$\varepsilon_i^{\alpha\beta} = \frac{1}{2} \sum_k \frac{m_k}{\rho_k} \left[(u_k^\alpha - u_i^\alpha) \frac{\partial \omega_{ik}}{\partial x_i^\beta} + (u_k^\beta - u_i^\beta) \frac{\partial \omega_{ik}}{\partial x_i^\alpha} \right];$$

$$R_i^{\alpha\beta} = \frac{1}{2} \sum_k \frac{m_k}{\rho_k} \left[(u_k^\alpha - u_i^\alpha) \frac{\partial \omega_{ik}}{\partial x_i^\beta} - (u_k^\beta - u_i^\beta) \frac{\partial \omega_{ik}}{\partial x_i^\alpha} \right],$$

где k — номер соседней частицы; μ — коэффициент Лямэ; $S^{\alpha\beta}$ — девиатор тензора напряжений; $R^{\alpha\beta}$ — тензор скоростей деформации. Данные аппроксимации имеют порядок точности $O(h^2)$. Однако решения, найденные таким методом, обладают сильной немонотонностью, что мешает использовать данный вариант метода при решении задач с интенсивными взаимодействиями.

Искусственная вязкость

Для борьбы с нефизичными осцилляциями Моноган [1] предложил использовать искусственную вязкость. Численный эксперимент показал, что применение такой же формы вязкости в задачах механики деформируемого твердого тела дает удовлетворительные результаты.

При использовании искусственной вязкости в члены вида $\left(\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} + \frac{\sigma_k^{\alpha\beta}}{\rho_k^2} \right)$ добавляется $\frac{\alpha \xi_{ik} \bar{c}_{ik} + b \xi_{ik}^2}{\bar{\rho}_{ik}}$,

$$\text{где } \xi_{ik} = -\frac{(u_i^\alpha - u_k^\alpha)(x_i^\alpha - x_k^\alpha)h}{(x_i^\alpha - x_k^\alpha)^2 + 0,01h^2}; \bar{c}_{ik} —$$

средняя скорость звука; $\bar{\rho}_{ik}$ — средняя плотность i -й и j -й частиц; a и b

— коэффициенты искусственной вязкости.

Метод с искусственной вязкостью имеет порядок точности $O(h^2)$ и обладает приемлемой немонотонностью, что позволяет его использовать при решении практических задач.

На графиках показано решение в начальный момент времени, через 20, 40 и 60 шагов интегрирования. Слева показано напряжение $\sigma^{zz}(z)$, справа скорость $u^z(z)$. На рис. 1 показан распад разрыва по напряжению, а на рис. 2 распад разрыва по скорости. Четко видны осцилляции, которые с течением времени затухают.

Монотонная схема

В работе [3] предложен подход, использующий приближенное аналитическое решение задачи Римана. Идея заключается в замене всех $a_i + a_j$ и $a_i - a_j$ на $2a_{ij}^*$ и $2(a_{ij}^* - a_j)$ соответственно. Здесь a_i, a_j — значения полевой функции в i -й и j -й частицах, а a_{ij}^* — соответствующее решение задачи о распаде разрыва. В результате полученных изменений аппроксимации наши уравнения станут следующими:

$$\frac{d\rho_i}{dt} = -2 \sum_k m_k (u_{ik}^{\alpha*} - u_i^\alpha) \frac{\partial \omega_{ik}}{\partial x_i^\alpha};$$

$$\frac{du_i^\alpha}{dt} = \sum_k m_k \left(\frac{\sigma_i^{\alpha\beta*}}{\rho_i \rho_k} \right) \frac{\partial \omega_{ik}}{\partial x_i^\beta};$$

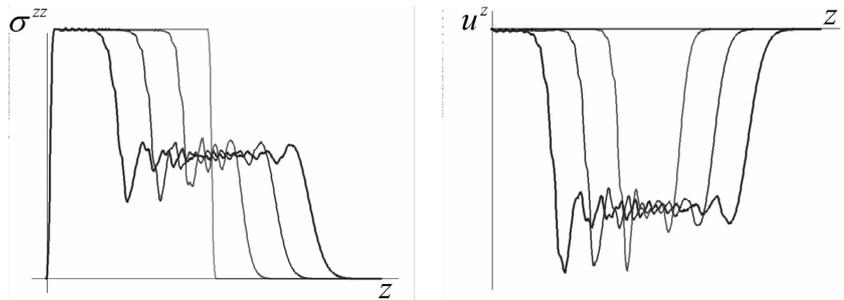


Рис. 1. Распад разрыва по напряжению (искусственная вязкость)

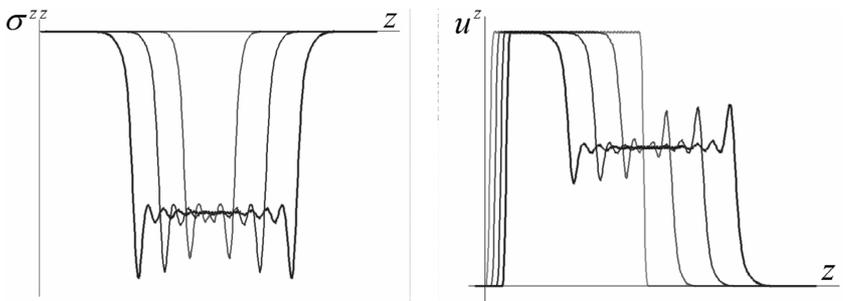


Рис. 2. Распад разрыва по скорости (искусственная вязкость)

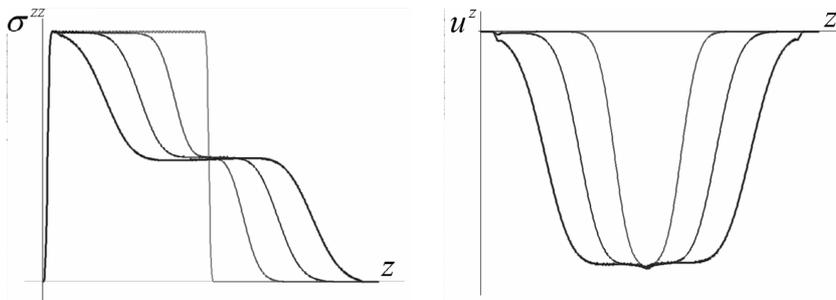


Рис. 3. Распад разрыва по напряжению (монотонная схема)

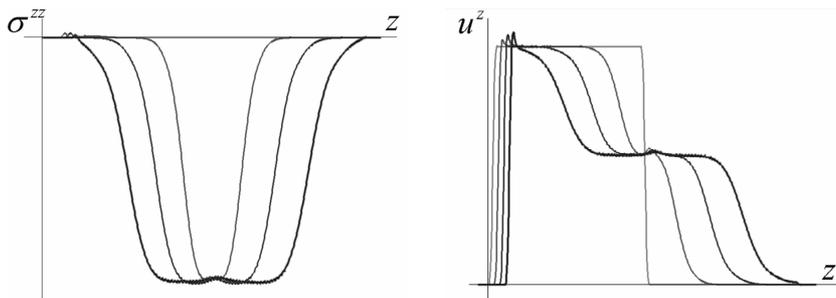


Рис. 4. Распад разрыва по скорости (монотонная схема)

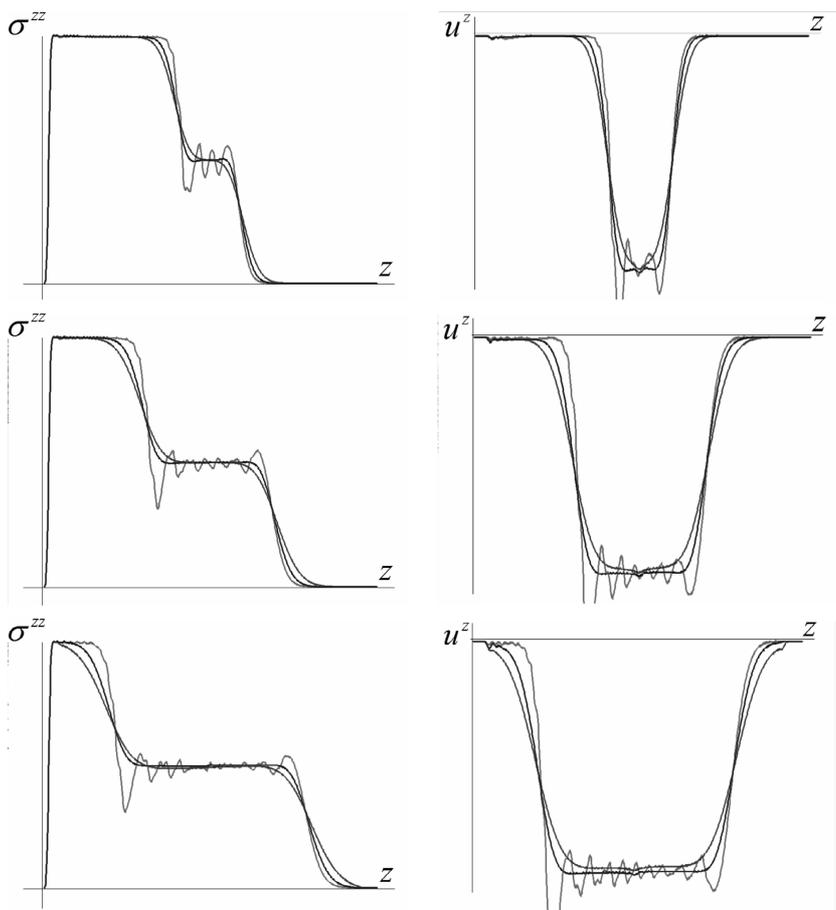


Рис. 5. Распад разрыва по напряжению (сравнительный график)

$$\frac{de_i}{dt} = \sum_k m_k (u_{ik}^{\alpha*} - u_k^\alpha) \left(\frac{\sigma_{ik}^{\alpha\beta*}}{\rho_i \rho_k} \right) \frac{\partial \omega_{ik}}{\partial x_i^\beta};$$

$$\frac{dS_i^{\alpha\beta}}{dt} = 2\mu \left(\varepsilon_i^{\alpha\beta} - \frac{1}{3} \delta^{\alpha\beta} \varepsilon_i^{\alpha\beta} \right) + S_i^{\alpha\gamma} R_i^{\beta\gamma} + S_i^{\gamma\beta} R_i^{\alpha\gamma};$$

$$\varepsilon_i^{\alpha\beta} = \sum_k \frac{m_k}{\rho_k} \left[(u_{ik}^{\alpha*} - u_i^\alpha) \frac{\partial \omega_{ik}}{\partial x_i^\beta} + (u_{ik}^{\beta*} - u_i^\beta) \frac{\partial \omega_{ik}}{\partial x_i^\alpha} \right];$$

$$R_i^{\alpha\beta} = \sum_k \frac{m_k}{\rho_k} \left[(u_{ik}^{\alpha*} - u_i^\alpha) \frac{\partial \omega_{ik}}{\partial x_i^\beta} - (u_{ik}^{\beta*} - u_i^\beta) \frac{\partial \omega_{ik}}{\partial x_i^\alpha} \right].$$

Использование приближенного аналитического решения Римана позволило создать монотонный алгоритм. Однако видно, что этот метод размывает фронт ударной волны, что характерно для подобных методов. Графики поведения решения задачи о распаде разрыва приведены на рис. 3, 4. На графиках показано решение в начальный момент времени, через 20, 40 и 60 шагов интегрирования.

Гибридная схема

Для сочетания высокого (второго) порядка точности и монотонности было предложено сделать гибридную схему. В данной работе приведены результаты вычислений с помощью линейной комбинации оригинального и монотонного методов с коэффициентом гибридности 0,5.

Как видно из сравнительного графика, на котором представлены решения задачи о распаде разрыва по напряжению (рис. 5) через 20, 40 и 60 шагов интегрирования, гибридная схема практически монотонна, в отличие от схемы с искусственной вязкостью, и размывает разрывы меньше, чем схема, использующая инварианты Римана.

Гибридная схема

Следующим улучшением алгоритма является использование коэффициента гибридности, который зависит от решения.

Для определения разрывов в решении используется аналог отношения второй и первой производных решения по координате.

$$\beta \sim \frac{a''}{a'} \sim \frac{(a_{i+1} - a_i) - (a_i - a_{i-1})}{(a_{i+1} - a_i) + (a_i - a_{i-1})}.$$

Однако в нашем случае мы не можем пронумеровать частицы вдоль координатной оси. И для определения порядка мы будем использовать само значение координаты

$$\beta = \frac{\sum_i (a^* - a_i) \frac{x^* - x_i}{|x^* - x_i|}}{\sum_i (a^* - a_i)}.$$

Здесь a^* , x^* — значение параметра и местоположение данной частицы, а суммирование ведется по всем ее соседям. Подставив в качестве a все компоненты скорости и напряжения, и просуммировав полученные β , мы получим коэффициент, который хорошо показывает разрывы в решении.

Далее приведены сравнительные результаты решения задачи о распаде разрыва по напряжению с помощью гибридной и гибридной схем через 20, 40 и 60 шагов интегрирования.

Из сравнительных графиков (рис. 6) видно, что гибридная схема меньше размывает разрывы.

Рассмотрим результаты расчетов для численного решения следующих задач.

1. Столкновение самолета со строительным сооружением. Данная модель представляет практический интерес с точки зрения террористической безопасности. При расчетах параметры ударника подбирались таким образом, чтобы наиболее реалистично имитировать падение легкого одномоторного самолета.

2. Падение деформируемого тела на слой грунта с большой скоростью. В данном случае речь идет о безопасности подземных сооружений, например, хранилища ядерных отходов. В работе рассмотрены случаи соударения со скоростями, меньшими и превышающими скорость звука в грунте, и показаны качественные различия волновых процессов.

Соударение самолета со зданием. В этой задаче основной интерес представляют волновые процессы в стенах и перекрытиях здания, а также разрушения в зоне соударения. На рис. 7 представлены распределения модуля скорости и давления. На рисунках четко виден фронт возмущений.

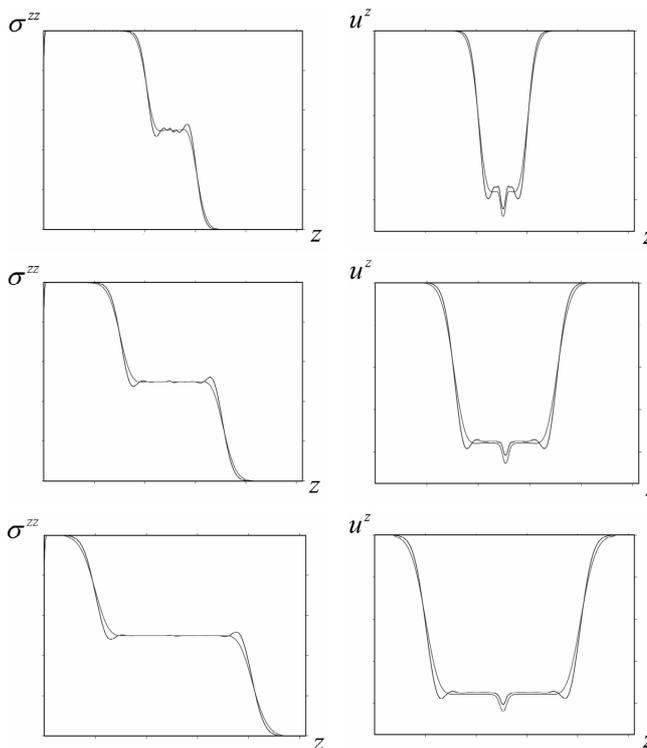


Рис. 6. Распад разрыва по напряжению (гибридная и гибридная схемы)

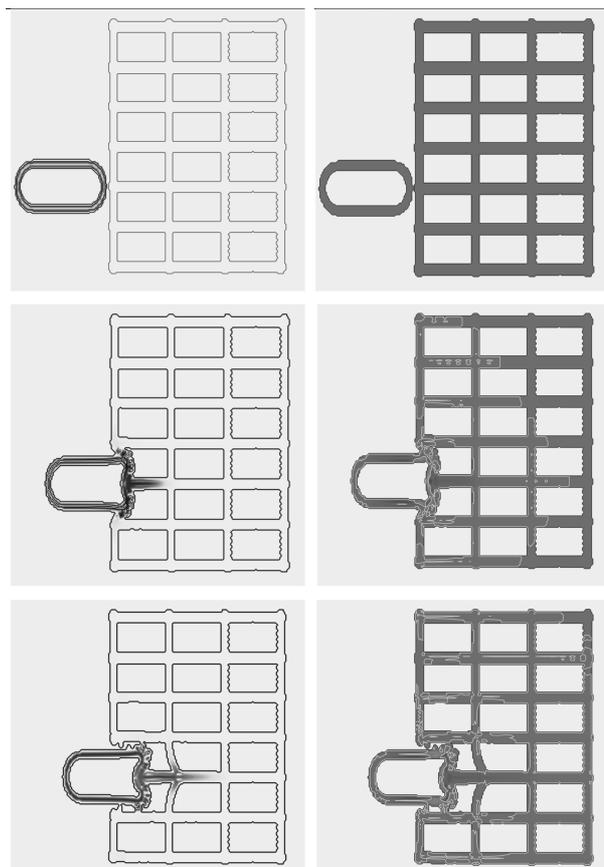


Рис. 7. Распределение давления (слева) и модуля скорости (справа)

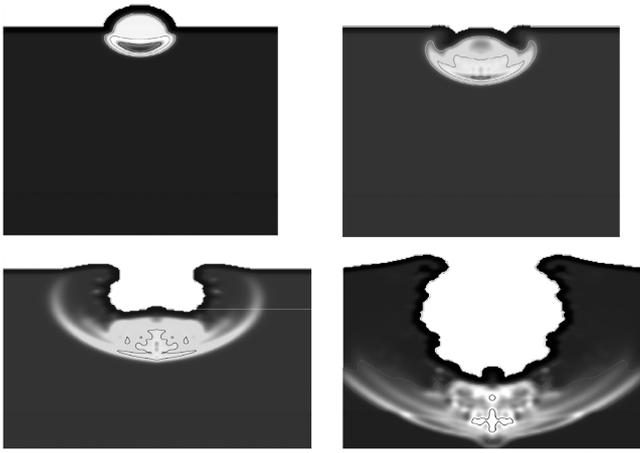


Рис. 8. Падение боида (железо, 10 000 кг, 10 000 м/с) на грунт (10 м грунта, снизу 2 м бетона)

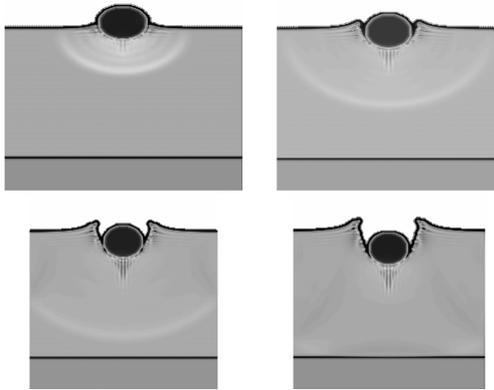


Рис. 9. Падение боида (железо, 10 000 кг, 1000 м/с) на грунт (10 м грунта, снизу 2 м бетона)

Также в области удара четко видны разрушения и большие деформации как здания, так и самолета. Представленные результаты свидетельствуют об адекватности качественных характеристик процесса соударения трехмерному моделированию на основе гибридного метода сглаженных частиц.

Соударение с грунтом. В данной модели интерес представляет разлет грунта при ударе и волновая картина, возникающая при ударе. На рис. 8 изображены волновые процессы и процессы разрушения при сверхзвуковых скоростях соударения. Скорость падения превосходит скорость звука в грунте (2400 м/с), поэтому виден сверхзвуковой конус. На рис. 9 отображены также волновые процессы и процессы разрушения для дозвуковых скоростей. В этом случае скорость меньше скорости звука, поэтому видна волна, обгоняющая ударник. Представленные результаты свидетельствуют о применимости гибридного метода сглаженных частиц для моделирования задач астероидной безопасности.

Список литературы

1. Monagan J. J. An introduction to SPH // Comput. Phys. Comm. 1988. Vol. 48. P. 89–96.
2. Блажевич Ю. В., Петров И. Б., Сабельников А. Е. Моделирование динамических процессов разрушения пористых конструкций в проблеме безопасности жилищных сооружений // Управление и обработка информации: модели процессов. М.: МФТИ, 2002. С. 72–78.
3. Parshikov A. N., Medin S. A., Loukashenko I. I., Milekhin V. A. Improvements in SPH methods by means of interparticle contact algorithm and analysis of perforation tests at moderate projectile velocities // International Journal of Impact Engineering. 2000. Vol. 24. P. 779–796.

УДК 621.396.6.049.75

В. В. Воловиков, канд. техн. наук, докторант,
С. У. Увайсов, д-р техн. наук, проф.,
 e-mail: Uvaysov@yandex.ru,
 Московский государственный институт
 электроники и математики

Модель тепломассопереноса в конструкциях радиоэлектронных устройств

Описаны результаты разработки модели тепломассопереноса, представленной в виде эквивалентной топологической цепи. Предложенная модель позволяет определить среднеобъемную и выходную температуру теплоносителя и используется как компонент топологических моделей тепловых процессов.

Ключевые слова: тепловые характеристики, математическое моделирование, радиоэлектронные устройства, тепломассоперенос, топологические модели.

Моделирование тепловых процессов в конструкциях радиоэлектронных устройств (РЭУ) с системами принудительного воздушного и жидкостного охлаждения связано с рассмотрением движения теплоносителя по каналам различных форм сечения. Например, каналами с круглой формой сечения являются трубки контуров жидкостного охлаждения, а с прямоугольной формой — каналы, образующиеся между печатными узлами и стенками конструкции. Одна из задач моделирования связана с определением температуры движущихся потоков теплоносителя по длине каналов или их участков внутри устройства. Понятие тепломассопереноса при моделировании РЭУ подразумевает, что у движущегося под действием внешней силы теплоносителя изменяется температура по длине канала в зависимости от теплоемкости, расхода и тепловой мощности.

Из теории теплообмена [3, 4] известно, что при постоянном массовом расходе и постоянной теп-

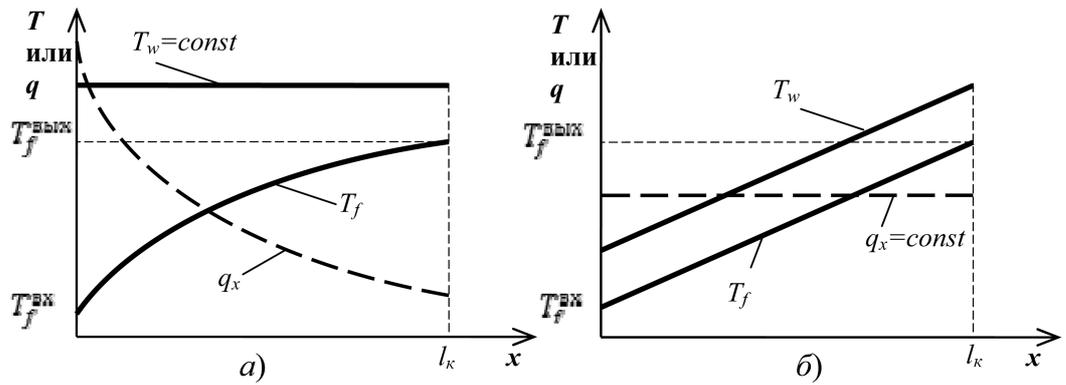


Рис. 1. Изменение температуры теплоносителя по длине канала: а — при $T_w = \text{const}$; б — при $q_x = \text{const}$

лостности теплоносителя (удельная теплоемкость теплоносителей, применяемых в РЭУ, в рабочем диапазоне температур практически неизменна) изменение его температуры T_f по длине канала при условии постоянной температуры стенки $T_w = \text{const}$ или постоянной плотности теплового потока $q_x = \text{const}$ происходит так, как это показано на рис. 1, а, б соответственно.

Среднее значение температуры теплоносителя по длине канала \bar{T}_f необходимо знать для вычисления коэффициента конвективной теплоотдачи между стенкой и теплоносителем.

Если $T_w = \text{const}$, то для определения \bar{T}_f применяется выражение [3, 4]

$$\bar{T}_f = T_w - \Delta T_{\text{In}} \quad (1)$$

Здесь среднелогарифмический температурный напор вычисляется по формуле

$$\Delta T_{\text{In}} = \frac{\Delta T_{\text{ВХ}} - \Delta T_{\text{ВЫХ}}}{\ln\left(\frac{\Delta T_{\text{ВХ}}}{\Delta T_{\text{ВЫХ}}}\right)}, \quad (2)$$

где $\Delta T_{\text{ВХ}} = T_w - T_f^{\text{ВХ}}$, $\Delta T_{\text{ВЫХ}} = T_w - T_f^{\text{ВЫХ}}$ — температурные напоры в начальном и конечном сечении рассматриваемого участка канала.

При $\Delta T_{\text{ВХ}}/\Delta T_{\text{ВЫХ}} < 2$ средняя температура теплоносителя \bar{T}_f может быть вычислена по более простой формуле, как среднее арифметическое:

$$\bar{T}_f = \frac{T_f^{\text{ВХ}} + T_f^{\text{ВЫХ}}}{2}. \quad (3)$$

Разница со среднелогарифмическим температурным напором в этом случае составит менее 4 %.

При постоянной плотности теплового потока $q_x = \text{const}$ на поверхности канала \bar{T}_f вычисляется по формуле (3).

В моделях, представляемых в виде эквивалентных топологических цепей, для определения \bar{T}_f используют диссипативную ветвь тепломассопереноса, представляющую собой тепловую проводимость, вычисляемую по формуле [1, 2]

$$\chi = G_p C_p = abwp\rho C_p, \quad (4)$$

где χ — тепловая проводимость; G_p — массовый расход теплоносителя в канале; C_p — удельная теплоемкость теплоносителя; a, b — геометрические размеры сечения канала, перпендикулярного направлению движения потока теплоносителя; w — скорость движения теплоносителя на входе в канал; ρ — плотность теплоносителя на входе в канал.

В настоящее время известно два варианта построения тепловых схем, описывающих процесс теплообмена между стенками и теплоносителем в канале, которые представлены на рис. 2.

В данных схемах узлы 1, 2 и 3 соответствуют теплоносителю на входе, теплоносителю на выходе и стенке. Тепловые сопротивления $R_{\text{КНВ}}$, $R1_{\text{КНВ}}$, $R2_{\text{КНВ}}$ моделируют конвективные связи стенок и потока теплоносителя, а сопротивления $R_{\text{ТМП}}$ — нагрев потока теплоносителя при движении вдоль канала. К узлу 3 подводится мощность тепловыделения.

Данные тепловые схемы имеют следующие недостатки.

1. При использовании тепловой схемы, приведенной на рис. 2, а, температура теплоносителя на выходе из канала ниже реального значения, что является следствием отвода части тепловой мощности от стенки через конвективное сопротивление $R1_{\text{КНВ}}$. В результате значение температуры

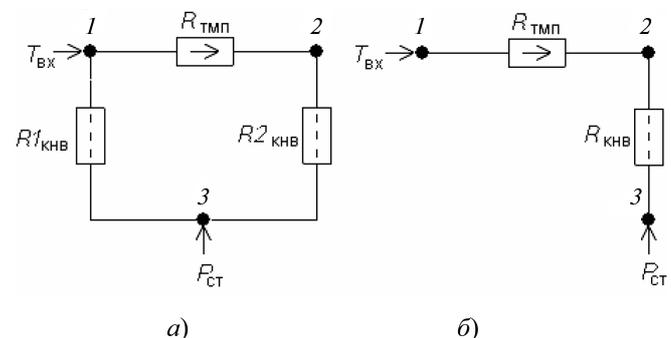


Рис. 2. Тепловые схемы включения ветвей конвекции и тепломассопереноса:

а — с разделением конвективных потоков; б — без разделения конвективных потоков

стенки канала также оказывается ниже реального значения.

2. При использовании тепловой схемы, приведенной на рис. 2, б, температура стенки оказывается выше реального значения, так как при вычислении конвективного теплового сопротивления в качестве температуры потока теплоносителя используется не усредненное значение, а максимальное — достигаемое на выходе из канала.

3. Приведенные схемы нельзя использовать в моделях, предполагающих разбиение канала на несколько частей, так как в этом случае изменение мощности тепловыделения на концевых участках канала приводит к изменению тепловых процессов (значений тепловых потоков и температур) на его начальных участках, что является нарушением физического смысла. Причина описанного эффекта состоит в том, что ветви, моделирующие тепломассоперенос в канале, "гальванически" связаны друг с другом. В последующем описании будем называть этот эффект *паразитным обратным влиянием*.

Таким образом, возникает задача разработки топологической модели тепломассопереноса, лишенной приведенных выше недостатков и отвечающей следующим требованиям:

- модель должна позволять вычислять температуру теплоносителя на входе, выходе, а также среднеобъемное значение для канала или его участка;
- модель должна исключать влияние через тепломассоперенос тепловых процессов на концевых участках канала на тепловые процессы на его начальных участках.

В соответствии с заданными требованиями была разработана топологическая модель тепломассопереноса, тепловая схема которой приведена на рис. 3, а. Рассмотрим ее подробнее.

Температура узла 1 соответствует температуре теплоносителя на входе моделируемого участка канала, 2 — среднеобъемной температуре теплоносителя в моделируемом участке канала, температура узлов 4 и 3 — одинаковая и соответствует температуре теплоносителя на выходе моделируемого участка канала.

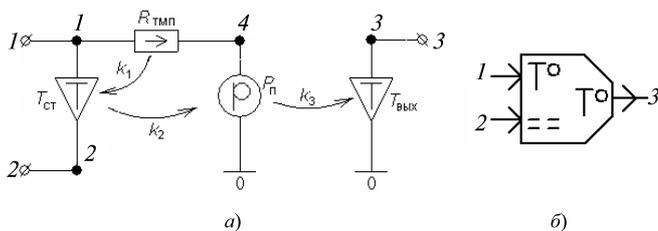


Рис. 3. Топологическая модель тепломассопереноса: а — тепловая схема замещения; б — условное изображение в модели тепловых процессов

Тепловое сопротивление $R_{\text{ТМП}}$ моделирует перепад температур теплоносителя между входом и выходом моделируемого участка канала. Для его расчета используется выражение (4).

Источник температуры $T_{\text{СТ}}$ является зависимым. Он определяет среднеобъемную температуру теплоносителя в моделируемом участке канала. Задаваемый им перепад температур определяется по формуле

$$T_{\text{СТ}} = -k_1 \Delta T(R_{\text{ТМП}}) = -k_1(T_4 - T_1), \quad (5)$$

где k_1 — коэффициент усиления температуры; $\Delta T(R_{\text{ТМП}})$ — перепад температур в ветви $R_{\text{ТМП}}$; T_4 , T_1 — температуры теплоносителя на выходе и входе моделируемого участка канала соответственно.

Источник тепловой мощности P_{II} является зависимым и определяет полную тепловую мощность, расходуемую на изменение энтальпии (теплосодержания) теплоносителя на моделируемом участке канала. Его значение определяется тепловой мощностью, проходящей через зависимый источник температуры $T_{\text{СТ}}$:

$$P_{\text{II}} = k_2 p(T_{\text{СТ}}), \quad (6)$$

где k_2 — коэффициент управления мощностью; $p(T_{\text{СТ}})$ — тепловой поток, проходящий через элемент $T_{\text{СТ}}$.

Источник температуры $T_{\text{ВЫХ}}$ также является зависимым и определяет температуру теплоносителя на выходе моделируемого участка канала. Значение источника $T_{\text{ВЫХ}}$ определяется по формуле

$$T_{\text{ВЫХ}} = k_3 \Delta T(P_{\text{II}}) = k_3(T_4 - T_0) = k_3 T_5, \quad (7)$$

где $k_3 = 1$ — коэффициент усиления температуры; ΔT — перепад температур в ветви P_{II} .

Разберем работу предложенной тепловой схемы при ее включении в модель тепловых процессов.

К узлу 1 подключается источник температуры, задающий температуру теплоносителя на входе моделируемого участка канала. К узлу 2 подключаются ветви, моделирующие конвективную теплоотдачу между стенками канала и теплоносителем.

Суммарный тепловой поток конвективных ветвей, подключенных к узлу 2, проходит через источник температуры $T_{\text{СТ}}$, который управляет источником тепловой мощности P_{II} с коэффициентом $k_2 = 1$. Это обеспечивает передачу полного значения теплового потока от конвективных ветвей в ветвь $R_{\text{ТМП}}$, что позволяет провести точный расчет перепада температур теплоносителя между входом и выходом. В свою очередь, перепад температур в ветви $R_{\text{ТМП}}$ управляет зависимым источником температуры $T_{\text{СТ}}$ с коэффициентом k_1 . Для случаев, когда среднеобъемная температура теп-

лоносителя вычисляется по формуле (3) значение $k_1 = -0,5$. Если для расчета используется соотношение (2), то

$$k_1 = \frac{1}{\ln\left(\frac{T_w - T_1}{T_w - T_2}\right)} - \frac{T_w - T_1}{T_2 - T_1}. \quad (8)$$

Данный источник поддерживает в узле 2 среднееобъемное значение температуры теплоносителя моделируемого участка канала.

Источник температуры $T_{\text{вых}}$ управляется перепадом температур в ветви $P_{\text{п}}$ с коэффициентом $k_3 = 1$. Этот источник введен в модель для обеспечения "гальванической развязки" элементов, моделирующих тепломассоперенос при их последовательном включении в модели тепловых процессов, и исключения, таким образом, паразитного обратного влияния.

Сравним предложенную модель тепломассопереноса с моделью на основе тепловой проводимости, включенную в соответствии с тепловыми схемами, приведенными на рис. 2. Для этого проведем моделирование канала круглого сечения (трубы), эскиз которого представлен на рис. 4. По каналу диаметром $d = 3$ мм и длиной $l = 600$ мм движется вода со скоростью $\omega = 0,3$ м/с. Температура воды на входе в трубу $T_{\text{вх}} = 10$ °С. Плотность теплового потока на поверхности трубы постоянна $q_x = \text{const}$, а количество передаваемой теплоносителю теплоты $Q = 200$ Вт.

Результатами моделирования, используемыми для сравнения, будут:

- среднеповерхностная температура стенки трубы;
- температуры воды на входе и выходе трубы, а также среднееобъемная температура потока теплоносителя.

Такой конструктивный элемент можно рассматривать как фрагмент конструкции РЭУ с системой принудительного жидкостного охлаждения.

Использованные в модели тепловых процессов (МТП) тепловые сопротивления конвективного теплообмена $R_{\text{кнв}}$ рассчитываются по формулам, приведенным в [3].

Результаты, полученные для моделей на основе теплопроводности (в соответствии со схемами, представленными на рис. 2, а, б), разработанной модели и аналитического решения задачи приведены в таблице.

Из представленных результатов видно, что применение тепловой

схемы, приведенной на рис. 2, а, приводит к занижению температуры теплоносителя и стенки канала по сравнению с аналитическим решением. Для схемы, показанной на рис. 2, б, температуры стенки и теплоносителя завышаются. Поэтому использование данных моделей в тех случаях, когда перепад температуры теплоносителя по длине канала превышает 10 %, приводит к чрезмерному завышению температуры стенки (при 10 %-ном перепаде температуры теплоносителя перегрев стенки завышается примерно на 5 %).

Применение данных схем в составе комплексных теплоаэродинамических (теплогидравлических) моделей может приводить к искажению значения аэродинамического сопротивления в канале вследствие неверного вычисления потерь на ускорение потока и сопротивления самотяги. В результате скорости течения и падения давления будут вычислены неверно, а так как скорости являются параметрами тепловых моделей, это приведет к дополнительному снижению точности расчета.

Использование предложенной топологической модели тепломассопереноса позволяет избежать погрешностей вычисления среднееобъемной и среднеповерхностной температур теплоносителя и стенки канала соответственно. В результате обеспечивается повышение точности вычисления скорости течения теплоносителя при проведении комплексного теплоаэродинамического моделирования.

Таким образом, разработанная топологическая модель тепломассопереноса обладает следующими достоинствами по сравнению с моделью на основе диссипативной ветви:

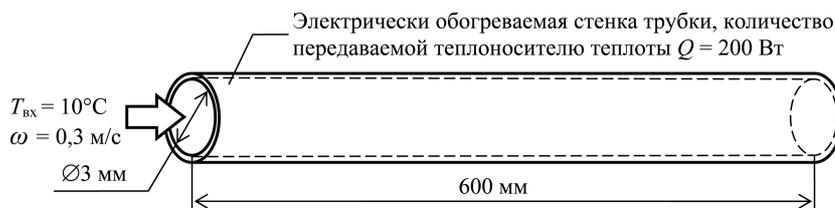


Рис. 4. Эскиз трубы с обогреваемой стенкой

Результаты анализа МТП в канале с равномерно нагретыми стенками

Параметр	По схеме рис. 2, а	По схеме рис. 3, б	Предложенная модель	Аналитическое решение
Температура воды на входе в трубу, °С	10	10	10	10
Среднееобъемная температура воды, °С	—	32,46	21,23	21,23
Температура воды на выходе из трубы, °С	18,98	—	32,46	32,46
Среднеповерхностная температура стенки трубы, °С	38,53	62,5	51,68	51,68

- позволяет повысить точность расчета температуры теплоносителя в канале, а также среднеповерхностной температуры стенки канала;
- обеспечивает возможность построения топологических моделей с разбиением канала на фрагменты по длине, исключая паразитное обратное влияние через ветвь теплопереноса тепловых процессов в конечных участках канала на начальные;
- полностью согласуется по выходным характеристикам с параметрами ветвей моделей аэродинамических и гидравлических процессов, что обеспечивает повышение точности расчета скоростей движения теплоносителя в конструкциях РЭУ.

К недостаткам предложенной модели можно отнести незначительное увеличение размерности модели тепловых процессов в РЭУ по сравнению с моделью тепломассопереноса на основе диссипативной ветви. Однако указанный недостаток не является критическим для применения разработанной модели в практике проектирования РЭУ, так как время расчета практически не увеличивается. Причина этого в том, что в РЭУ с системами принудительного воздушного и жидкостного охлаждения перепад температур теплоносителя относительно невелик. Следовательно, в моделях

допускается крупная дискретизация соответствующих участков пространства и размерность их невысока (менее 100 узлов).

Предложенная модель программно реализована в рамках подсистемы комплексного моделирования электрических, тепловых, аэродинамических, гидравлических и механических процессов АСОНИКА-П, входящей в состав автоматизированной системы обеспечения надежности и качества аппаратуры АСОНИКА.

Данная работа выполняется при поддержке Совета по грантам Президента Российской Федерации для государственной поддержки молодых российских ученых — кандидатов наук и их руководителей (№ гранта МК-3278.2008.8).

Список литературы

1. Дульнев Г. Н., Тарновский Н. Н. Тепловые режимы электронной аппаратуры: Учеб. пос. для студентов высших технических заведений. Л.: Энергия, 1971. 248 с.
2. Гольдин В. В. Исследование тепловых характеристик РЭС методами математического моделирования / В. В. Гольдин, В. Г. Журавский, Ю. Н. Кофанов и др. Под ред. А. В. Сарфанова. М.: Радио и связь, 2003. 456 с.
3. Михеев М. А., Михеева И. М. Основы теплопередачи. М.: Энергия, 1977. 344 с.
4. Bejan A., Kraus A. D. Heat transfer handbook / Adrian Bejan. New Jersey: John Wiley & Sons Inc., 2003. 1480 p.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УДК 007:681.51-007:001.33(075.8)

А. Н. Швецов, д-р техн. наук, проф., e-mail: smithv@mail.ru,
М. А. Сергушичева, аспирант, e-mail: marias@list.ru,
 Вологодский государственный технический университет

Проектирование прикладных мультиагентных систем с использованием пакета DISIT

Рассматриваются вопросы проектирования интеллектуальных мультиагентных систем. Описаны принципы построения и архитектура инструментального программного пакета DISIT. Применение пакета демонстрируется на примере системы организации технического обслуживания и ремонта на территориально распределенном предприятии энергоснабжения.

Ключевые слова: мультиагентные системы, модели предметной области, организация технического обслуживания и ремонта, структура агентов.

Введение

Автоматизацию современных бизнес-процессов трудно представить без использования возможностей корпоративных и глобальных сетей. Необходимость обработки огромных массивов ге-

терогенной информации в территориально распределенных организациях требует нового качества от корпоративных информационных систем: интеллектуального анализа данных, семантической обработки потоков сообщений, оперативной

Таблица 1

Характеристики интеллектуальности агентов

Сенсорика (чувства)	S_1	0	С одним чувством	
	S_2	1	С несколькими чувствами	
Память	M_1	0	Без памяти	
		1	С памятью на прошедшие события (конечной)	
		2	С потенциально неограниченной или бесконечной памятью	
	M_2	0	Только с кратковременной памятью	
		1	С кратковременной и долговременной памятью	
	M_3	0	Не может обучаться (не наращивает долговременную память)	
		1	Может обучаться (наращивает долговременную память)	
	M_4	0	Хранит сенсорную информацию по некоторому чувству	
		1	Хранит сенсорную информацию по всем чувствам	
	Моторика (действия во времени)	T_1	0	Действует только в реальном времени
			1	Может планировать действия
T_2		0	Не может визуализировать чувства	
		1	Может визуализировать некоторые чувства	
		2	Может визуализировать все чувства	
T_3		0	Не имеет модели среды существования	
		1	Имеет предопределенную модель	
		2	Может создавать ментальные модели среды	

поддержки принятия решений. В качестве основы для создания таких интеллектуальных информационных систем авторы предлагают использовать агентную парадигму [1–4].

Этот подход основан на рассмотрении распределенных систем как совокупности автономных агентов, взаимодействующих друг с другом в процессе решения проблемы. Поведение агентов определяется системными требованиями и ограничениями и является следствием их наблюдений, знаний и взаимодействий с другими агентами.

В общем случае агент должен иметь возможность взаимодействия со своим владельцем и другими агентами для получения заданий и возвращения результатов, ориентироваться в среде своего "существования" и принимать решения, необ-

ходимые для выполнения поставленных перед ним задач. Очевидно, что степень интеллектуальности агентов может быть различной. Для оценки различия можно вычислить интеллектуальное расстояние между их типами, рассматривая некоторые характеристики (табл. 1), присущие как живым существам, так и техническим системам, с использованием модульной метрики [3, 5].

Интеллектуальное расстояние (ИР) $ID = \sum_{i=S_1}^{T_3} |Par_{ai} - Par_{bi}|$, где Par_{ai} , Par_{bi} — значение признака для агентов a и b соответственно, а индекс i пробегает множество $\{S_1, S_2, M_1, M_2, M_3, M_4, T_1, T_2, T_3\}$.

Интеллектуальный вес (ИВ) агента можно определить как $IW = \sum_{i=S_1}^{T_3} Par_i$.

На основе этих понятий авторами выделяются уровни интеллектуальной иерархии (табл. 2) таким образом, что ИР в парах между программными объектами и информационными объектами (ИО), ИО и интеллектуальными агентами (ИА), ИА и интеллектуальными компонентами (ИК) равняется двум [3, 6].

Для мультиагентных систем (МАС) характерны три основных типа архитектуры: делиберативная, реактивная и гибридная [7]. Агенты с делиберативной архитектурой содержат символическую модель мира и принимают решения на основании логического вывода. Однако создание точной модели предметной области реального мира представляет существенные трудности в реализации. Поиск путей решения этой проблемы привел к концепции реактивных агентов, которая предполагает использование множества простых сценариев поведения в рамках реакции на установленные события окружающей среды. Очевидная ограниченность этого подхода состоит в невозможности описания всех возникающих ситуаций. Оптимальной представляется гибридная архитектура, в которой агент реагирует определенным образом на события, происходящие в окружающей среде, а в процессе реализации поведения использует накопленные знания о предметной области.

Таблица 2

Уровни интеллектуальной иерархии

По интеллектуальному весу и расстоянию	Характеристики разумности									По методу математического описания
	S_1	S_2	M_1	M_2	M_3	M_4	T_1	T_2	T_3	
Программные объекты	1	1	1	1	0	0	0	1	0	Конечные автоматы Контекстно-свободные грамматики Контекстно-зависимые грамматики Семиотические системы
ИО	1	1	1	1	0	1	0	1	1	
ИА	1	1	1	1	1	1	1	1	1	
ИК	1	1	2	1	1	1	1	1	2	

Модель предметной области

В соответствии с приведенной классификацией всем типам агентов выше уровня конечных автоматов требуются модели предметной области (МПО), которые должны обеспечить стыковку понятий и корректное взаимодействие всех компонентов. Бизнес-процессы должны быть описаны так, чтобы из этого описания можно было создать прототип МАС, способной их промоделировать.

Поскольку используемая методология ориентирована на аналитиков и разработчиков бизнес-приложений, она должна обеспечивать простоту и наглядность представления МПО. Поэтому в качестве базовой структуры представления знаний в данной методологии используются фрейм-концепты (ФК) [3].

Каждый ФК можно рассматривать как семантическую сеть, состоящую из выделенных вершин и связей. Средствами определения семантических отношений между фреймами являются слоты.

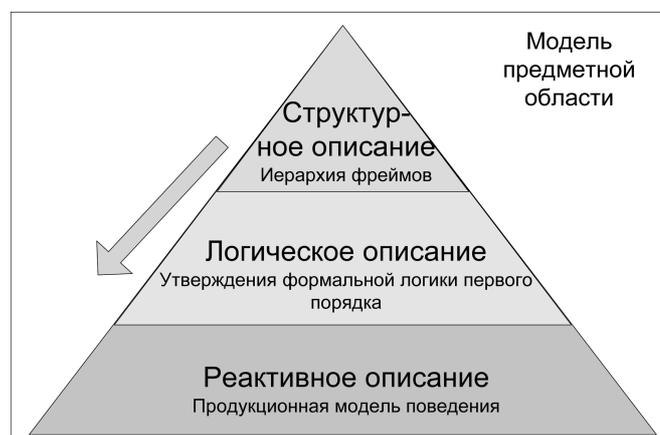


Рис. 1. Составляющие описания МПО

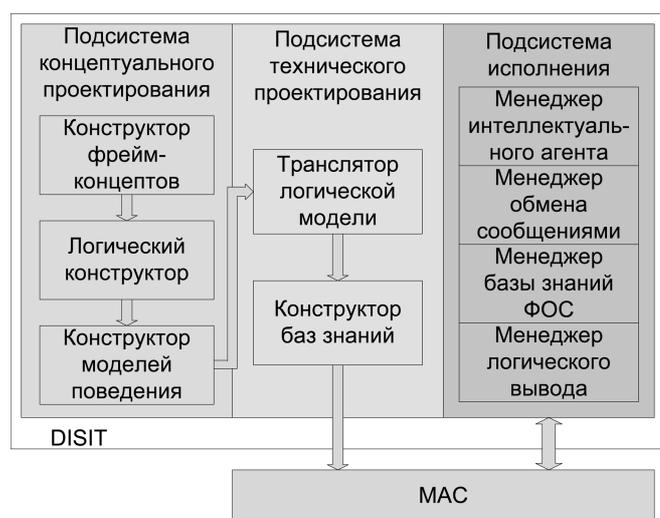


Рис. 2. Состав пакета DISIT

Благодаря слотам ФК обладают как процедурными (сценарии поведения при изменении значения какого-либо атрибута), так и декларативными (логические зависимости между ФК, имена и значения атрибутов и т. п.) свойствами.

Важнейшим свойством фреймов является наследование, которое происходит по так называемым АКО-связям ("a-kind-of — "это есть"). Слот АКО указывает на ФК более высокого уровня иерархии, откуда неявно наследуются значения аналогичных слотов. Продолжив объектно-ориентированную концепцию, выделяют два типа фреймов: фрейм-образец (аналог класса в ООП) и фрейм-экземпляр (аналог объекта). Переход от образца к экземпляру осуществляется путем указания значений атрибутов.

Далеко не все закономерности предметной области можно представить с помощью фреймов. Для этих случаев удобно применять давно известный и хорошо отработанный аппарат формальной логики первого порядка (для этого уровня характерны такие языки, как Пролог, KIF, дескриптивные XML-языки типа OWL). Фрейм при этом можно использовать как контейнер для связанных с ним утверждений.

Помимо логического и структурного описания МПО должна включать описание поведения фреймов-агентов (рис. 1), в качестве которого в гибридной архитектуре выступает реактивная продукционная модель. Процесс реализации поведения в такой архитектуре включает получение исходных данных (сообщения), сопоставление альтернатив (сравнение имеющихся правил с исходными данными и фактами из МПО), выбор (в случае наличия подходящего правила) и выполнение действий, указанных выбранным правилом.

Концепция DISIT

В соответствии с указанными принципами была разработана методология проектирования МАС, реализованная в виде программного пакета DISIT (*Distributed intellectual system integrated toolkit*). В данном инструментальном пакете последовательно выполняются следующие этапы методологии:

- представление МПО в виде структуры ФК;
- наполнение модели логикой взаимоотношений ФК и их атрибутов;
- выделение ИА и определение их поведения с учетом системных ограничений;
- трансляция полученной концептуальной МПО в структурно-логическую модель МАС;
- размещение ИК и ИА в корпоративной сети.

DISIT включает в себя подсистемы концептуального проектирования, технического проектирования и подсистему времени исполнения (рис. 2).

Подсистема концептуального проектирования

Конструктор фрейм-концептов (КФК) позволяет оформить структуру предметной области в виде иерархии фрейм-концептов, определить свойства объектов (имя, тип, структуру атрибутов, инициализировать значения атрибутов у конкретных экземпляров), отношения и связи между ними, а также следит за целостностью и непротиворечивостью данных, вводимых архитектором предметной области. Существует возможность декларативно описать сценарии поведения объекта и информацию о его применении, которые будут уточняться на последующих этапах.

В пакете DISIT предусматривается один ФК верхнего уровня иерархии, на остальные ФК наложено ограничение планарного дерева. Результатом этого этапа проектирования является сеть с определенной структурой по вложению концептов — ФК-проекция.

Логический конструктор (ЛК) дополняет ФК-проекцию до ФКМ-проекции путем добавления в нее модулей с логическими утверждениями. Под утверждениями понимаются предложения первого порядка в языке исчисления предикатов, где конкретные ФК понимаются как индивидуальные константы языка, а переменные принимают значения на множестве $\{FK\}$, мощность которого равна N_{FK} , где N_{FK} — общее число ФК в модели предметной области.

Утверждения вводятся в формате KIF (*knowledge interchange format*) [8], в который добавлены некоторые нелогические операторы. Все утверждения (аксиомы) начинаются с квантора и содержат оператор импликации:

Forall(<список перем. через запятую>)

(implies(<выр. 1>, <выр. 2>))

Exists(<список пост. через запя-

тую>)(implies(<выр. 1>, <выр. 2>))

Выражения внутри оператора импликации могут содержать заранее определенные предикатные символы.

Введенные утверждения распределяются по модулям автоматически, в соответствии с выбранным алгоритмом обхода ФКМ-проекции. Кроме того, для каждого модуля вводится некоторая база фактов, состоящая из атомарных утверждений с использованием предикатов.

ЛК поддерживает два режима работы: ввод утверждений в стандартном текстовом виде либо ввод с помощью визуального редактора, представляющего аксиому в виде дерева.

Конструктор моделей поведения (КМП) позволяет создавать необходимую иерархию ИА и ИО, редактировать их атрибуты и модели поведения.

Структура поведения ИА в КМП может быть как одноуровневая, так и многоуровневая. Пове-

дение ИО описывается путем задания реакции на входные воздействия, среди которых получение и отправка сообщения, изменение атрибута и таймера. Входное воздействие в общем случае может изменить состояние, инициировать выходное воздействие без изменения состояния, инициировать выходное воздействие и изменить состояние. Как именно будет реагировать агент на это воздействие, зависит от заданного сценария, в котором помимо самого события принимаются во внимание и включаются в условие атрибуты агента.

КМП имеет неполный компилятор для первичной проверки на ошибки созданных сценариев поведения и графически отображает иерархию ИО, атрибутов ИА и ИО. Кроме того, в него заложен механизм описания ИА и ИО с использованием контекстно-свободных языков, определяемых пользователем.

Подсистема технического проектирования

Транслятор логической модели (ТЛМ) выполняет первые четыре этапа построения структурно-логической модели:

- выделение интеллектуальных компонентов (задача ВИК);
- построение дерева интеллектуальных компонентов;
- декомпозиция ФК-проекции по дереву ИК;
- трансляция дерева фрейм-концептов в структурно-логическую модель МАС.

В результате решения задачи ВИК [3] получается декомпозированная ФКМ-проекция. Далее строится дерево, устанавливающее связи между ИК. При декомпозиции ФК-проекции по дереву ИК структура ФК-проекции связывается с деревьями T_{FK} . Для трансляции дерева фрейм-концептов T_{FK} в логическую модель МАС необходимо раскрыть структуру ИК, отображая дерево интеллектуальных компонентов в терминах ИК, ИА и ИО.

Конструктор баз знаний (КБЗ) для ИК, выделенных на предыдущем этапе, создает структуру баз знаний (БЗ), отображая в них модели ИА и ИО. В концепции DISIT была избрана смешанная стратегия построения БЗ, которая заключается в совмещении стратегий распределенного и централизованного подходов. Это обусловлено тем, что МАС корпоративного типа имеет иерархическую организацию, где ИК верхних уровней должны иметь возможность управления и слежения за состоянием компонентов подчиненных уровней. Поскольку отдельные ИК могут быть значительно удалены друг от друга пространственно, то большую часть знаний они должны хранить в своих локальных БЗ, а вышестоящим компонентам поставлять знания, необходимые для поддержания целостности и управляемости МАС.

Вместе с тем, аксиомы, устанавливаемые на ИК некоторого уровня, должны иметь возможность "распространяться вниз", т. е. быть известными нижестоящим компонентам.

Исходя из этого база знаний каждого ИК содержит:

- дерево декомпозиции;
- описания моделей ИА, которые принадлежат данному ИК;
- описания моделей ИО, которые принадлежат данному ИК;
- базу знаний формальной объектной системы (ФОС) данного ИК, которая включает в себя копию списков *GA* (множество глобальных аксиом) и *LAd* (распространяемые сверху вниз аксиомы), список *LA* (локальные аксиомы), список предикатных символов, базу фактов.

Средствами КБЗ инженеры по знаниям и эксперты предметной области устанавливают области действия наследуемых аксиом, т. е. формируют множества *LAd*. КБЗ формирует копии наследуемых аксиом, помещая их в базы знаний ФОС соответствующих ИК. Другой задачей КБЗ является распределение баз знаний ИК по сети. Вместе с базами по сети распределяются и менеджеры подсистемы исполнения, необходимые для функционирования МАС.

Подсистема исполнения

Менеджер формальной объектной системы (МФОС) позволяет изменять содержимое базы знаний ФОС. Редактирование аксиом можно проводить, как и в ЛК, в текстовом виде либо в графическом редакторе. При этом учитывается связь ее компонентов друг с другом: изменения, затрагивающие базы знаний подчиненных ИК (в списках *GA*, *LAd* и, возможно, в предикатных символах), запускают механизм сопоставления и обновления этих баз знаний.

Менеджер логического вывода (МЛВ) обслуживает запросы на осуществление операций логического вывода, поступающие от ИА и ИО с помощью менеджера обмена сообщениями. В качестве исходных данных МЛВ передается база знаний ИК, сформированная в результате работы подсистемы проектирования, и запрос на вывод — логическая формула, определяющая цель вывода. Выходные данные представляют собой отчет о результатах логического вывода, который может быть обработан другими компонентами системы или проанализирован пользователем.

МЛВ поддерживает два типа вывода: локальный и аддитивный, при котором базы знаний ИК, подчиненных данному ИК, объединяются (путем движения по дереву сначала горизонтально, затем вертикально). Выбор одного из них возложен на

механизмы реализации поведения. Для поддержки вывода используются средства языка Пролог. Текст программы на Прологе генерируется динамически, путем трансляции структур баз знаний, представленных в *KIF*-формате, в таблицы доменов, предикатов, фактов и правил Пролога. В дальнейшем используется компилятор командной строки для получения исполняемого файла.

Менеджер интеллектуального агента (МИА) и **менеджер обмена сообщениями (МОС)** в текущей версии комплекса объединены в схеме "клиент—сервер". При этом клиент непосредственно реализует поведение агентов, а сервер обслуживает все коммуникации среди подсоединенных приложений и вызывает клиента приложения.

Клиент позволяет задать параметры выполнения моделей поведения, а также параметры связи и передачи данных, наблюдать за ходом работы агента и динамически менять некоторые свойства. В его состав включен полный компилятор моделей для проверки сценариев и их компиляции, а также интерпретатор полученного объектного кода, для выполнения и анализа исполняемого агента. Дополнительно в этом модуле есть транспортный клиент для связи с сервером обмена сообщениями.

Сервер контролирует проходящие через него пакеты, создает соединения, централизует обмен сообщениями, задает параметры для подсоединенных клиентов, создает очереди для хранения передаваемых данных. Он отвечает за корректную пересылку сообщений, аутентификацию и авторизацию пользовательского доступа, а также за гарантию доставки сообщений. Каждый из клиентов-приложений может быть как приемником сообщений, так и их передатчиком, или тем и другим.

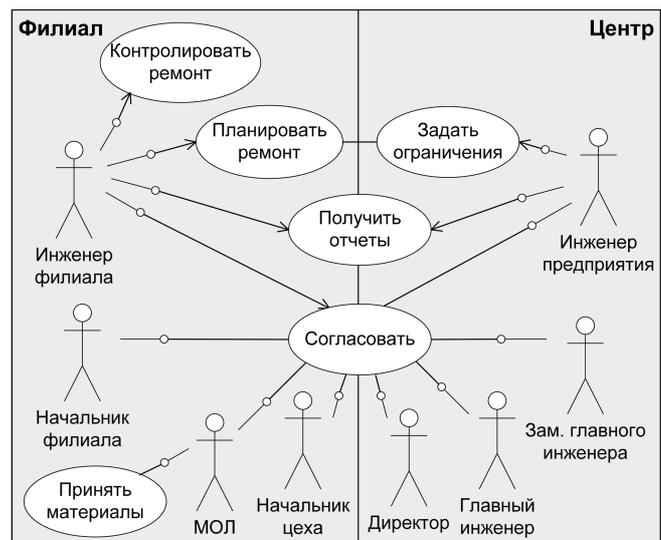


Рис. 3. Диаграмма прецедентов для системы организации ремонтных работ

С каждым подключенным клиентом-приложением связываются его идентификатор, очередь для простых сообщений, очередь для системных сообщений, очередь для событий. При реализации клиента использовался протокол *TCP/IP* с блокирующими соединениями и многопоточная архитектура (для каждого присоединившегося клиента создается отдельный поток).

Текущая версия комплекса *DISIT* занимает 13,32 Мбайт на жестком диске. Минимальные требования для нормального функционирования — процессор Intel Pentium II 266 МГц, 16 Мбайт видеопамати, 64 Мбайт ОЗУ. Для поддержания сетевого соединения достаточно модемной линии, однако для полноценной работы агентов могут потребоваться каналы связи с пропускной способностью в 10 Мбит/с.

Модель МАС поддержки организации ремонтов

Многим территориально распределенным предприятиям приходится координировать процесс планирования и организации ремонтных работ, проводящихся в их филиалах. Особенно важно это для компаний, прибыль которых зависит от безаварийной работы (например предприятий энергоснабжения). В таких компаниях стратегией технического обслуживания и ремонта (ТОиР) выбирается ремонт по техническому состоянию или планово-предупредительный ремонт. Обе стратегии предусматривают создание планов (перспективных, годовых, месячных и т. д.), которые в связи со спецификой многофилиальных предприятий часто вынуждены проходить длинный путь согласований и корректировок.

План ТОиР составляется на основе имеющихся ограничений и должен учитывать финансовые и трудовые затраты на проведение ремонта, надежность, ущерб от возможных аварийных ситуаций и другие факторы в условиях существенной

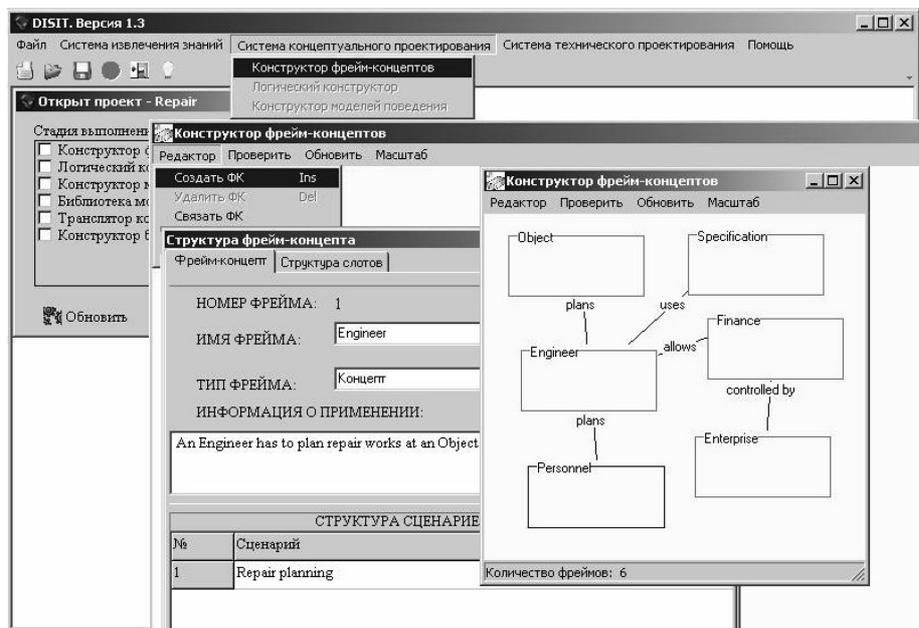


Рис. 4. FK-проекция в *DISIT*



Рис. 5. Конструктор моделей поведения ИА

неопределенности данных, так как поломки оборудования имеют вероятностный характер.

Обобщенная схема системы организации ремонтных работ на распределенном предприятии представлена на рис. 3. На ее основе построена *FK*-проекция МПО (рис. 4) средствами пакета *DISIT*. Уточняя далее *FKM*-проекцию предметной области, получаем описание необходимых групп агентов филиала и предприятия.

Эти группы включают в себя агентов, соответствующих разрабатываемым (инженер), согласо-

вывающим и утверждающим (материально ответственное лицо (МОЛ), начальник филиала, директор и т. д.) лицам. Виды агентов определяются в соответствии с типичными функциями действующих лиц, участвующих в процессе (см. рис. 3), по схеме "функция — агент". При этом инженеру филиала по ремонту в силу разнонаправленности его поведения соответствует несколько ИА: планировщик, агент по согласованию, составитель сменных заданий, контролер плана, генератор отчетов. МОЛ в этой системе соответствуют два агента — согласователь и менеджер материалов, инженеру предприятия по ремонту — три агента: агент верхнего уровня по планированию, генератор отчетов, согласователь. Остальным задействованным в процессе сотрудникам достаточно одного агента по согласованию.

Поведение агентов определяется в конструкторе моделей поведения (рис. 5).

При необходимости данная МАС легко масштабируется от двухуровневой системы (центр—филиал) до трех и более уровней за счет того, что функции вышестоящих уровней (суперцентров) дублирует второй уровень.

В настоящее время спроектированная система проходит этап отладки в подразделениях филиала ОАО "МРСК Северо-Запада" "Вологдаэнерго".

Работа поддержана грантом РФФИ № 08-01-00457-а.

Список литературы

1. **Wooldridge M.** An Introduction to Multi-Agent Systems. John Wiley and Sons, 2002. 376 p.
2. **Тарасов В. Б.** От многоагентных систем к интеллектуальным организациям: философия, психология, информатика (наука об искусственном). М.: Эдиториал УРСС, 2002. 352 с.
3. **Швецов А. Н., Яковлев С. А.** Распределенные интеллектуальные информационные системы. СПб.: Изд-во СПбГЭТУ "ЛЭТИ", 2003. 318 с.
4. **Стефанюк В. Л.** Локальная организация интеллектуальных систем. М.: Физматлит, 2004. 328 с.
5. **Берлекэмп Э.** Алгебраическая теория кодирования / Пер. с англ. под ред. С. Д. Бермана. М.: Мир, 1971. 480 с.
6. **Швецов А. Н.** Методология проектирования агентно-ориентированных корпоративных интеллектуальных систем // Интеллектуальные системы: Труды Шестого международ. симпозиума. — М.: Изд-во МГТУ им Н. Э. Баумана, 2004. С. 86—89.
7. **Гаврилова Т. А., Хорошевский В. Ф.** Базы знаний интеллектуальных систем. СПб: Питер, 2000. 384 с.
8. **Genesereth M. R.** Knowledge Interchange Format (KIF). Режим доступа: <http://logic.stanford.edu/kif/kif.html>.

УДК 519.95

А. А. Арзамасцев, д-р техн. наук, проф., зав. каф.,
Н. А. Зенкова, канд. психол. наук, доц.,
А. В. Неудахин, аспирант,
Тамбовский государственный университет
им. Г. Р. Державина,
e-mail: natulin@mail.ru

Технология построения медицинской экспертной системы на основе аппарата искусственных нейронных сетей

Предлагается автоматизированная технология построения экспертных систем, разработанных с использованием аппарата искусственных нейронных сетей. Реализация данной технологии осуществляется на примере построения медицинской экспертной системы.

Ключевые слова: автоматизированная технология, медицинская экспертная система, искусственные нейронные сети, интеллектуальное ядро, клиент-серверная технология доступа.

Актуальность проблемы

В различных сферах человеческой деятельности существуют объекты, решение задач управления которыми связано с анализом многофактор-

ных зависимостей, трудно поддающихся формализации ввиду недостаточности информации и ее значительной зашумленности. В некоторых случаях специалист в данной предметной области (эксперт) может решать такие задачи, пользуясь значительным опытом, интуицией, прогностической способностью в данной сфере. Однако такие решения являются в значительной степени субъективными. Например, при смене эксперта реализуются уже другие управленческие решения, которые, однако, могут быть не хуже предыдущих.

Таким образом, эксперт заключает в себе некоторую "модель" данной предметной области, реальная формализация которой часто бывает невозможна или значительно затруднена. Отчасти данная проблема, связанная с накоплением знаний об объекте, их обобщением и интерпретацией, может быть решена на основе разработки компьютерных экспертных систем (ЭС). Экспертные системы — специальные программы для ЭВМ, основанные на алгоритмах искусственного интеллекта и предполагающие использование соответствующей информации, полученной ранее от экспертов в заданной предметной области.

Основные идеи предлагаемой технологии

В данной статье рассмотрена технология, позволяющая разрабатывать ЭС с использованием в ка-

честве интеллектуального ядра ИНС-моделей, т. е. моделей, построенных на основе искусственных нейронных сетей (ИНС). Сделанный выбор обусловлен способностью ИНС к обучению и генерализации (обобщению) накопленных знаний [1].

Предлагаемая технология предполагает построение ЭС, определяющих решение задач, возникающих в различных предметных областях, на основе накопленной базы знаний о рассматриваемой задаче. Формирование базы знаний происходит при совместной работе пользователей и эксперта в исследуемой предметной области задачи. При последовательном вводе информации пользователями (входные данные задачи) происходит ее оценка экспертом, при этом определяются соответствующие выходные состояния задачи. Таким образом, происходит накопление базы знаний о рассматриваемой задаче до необходимого объема. Затем строится интеллектуальная модель задачи. Моделирование осуществляется с помощью построения и обучения ИНС на основе накопленных данных. Данный метод моделирования был рассмотрен в работе [2]. По мере поступления новой информации в базу знаний полученная интеллектуальная модель задачи совершенствуется. При накоплении некоторого критического объема базы знаний модель способна самостоятельно определять решение задачи. Таким образом, роль эксперта уменьшается по мере формирования интеллектуального ядра системы. В результате функционирования предложенной технологии строятся экспертные системы, позволяющие решать конкретную задачу в заданной предметной области.

Способ реализации технологии

Предлагаемая технология реализуется путем разработки целостной интерактивной системы, состоящей из взаимосвязанных компонентов, позволяющих осуществлять построение нейросетевых ЭС. Для каждого компонента системы определен круг решаемых им задач.

Система включает три основных компонента [3]:

- информационная часть;
- компонент импортирования данных (интегратор);
- модуль работы с ИНС.

Все компоненты являются взаимосвязанными частями единой целостной системы, функционирующей поэтапно, согласно схеме, представленной на рис. 1. Каждому этапу соответствует реализация определенных задач.

Первый этап — накопление информации. На первом этапе происходит формирование и накопление базы знаний объекта. Назначения связей,

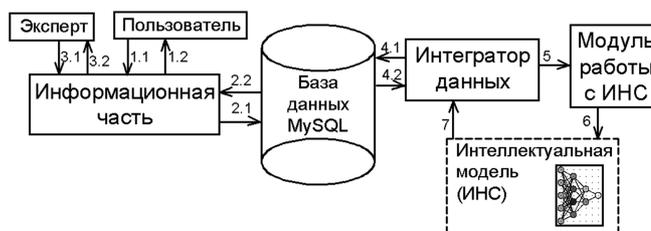


Рис. 1. Общая схема функционирования компонентов системы

представленных на рис. 1, на первом этапе функционирования:

- 1.1 — ввод пользователем параметрических характеристик объекта;
- 1.2 — вывод информации пользователю;
- 2.1 — запрос к базе данных (сохранение данных или запрос на данные);
- 2.2 — результаты запроса;
- 3.1 — запрос экспертом информации об объекте или ввод результатов анализа объекта;
- 3.2 — вывод информации эксперту.

Второй этап — построение модели. При получении определенного объема базы знаний объекта данные из базы импортируются в модуль работы с ИНС. На данном этапе происходит построение интеллектуальной модели объекта. Назначения связей, представленных на рис. 1, на втором этапе функционирования:

- 4.1 — осуществляется запрос на данные, хранящиеся в базе;
- 4.2 — в результате запроса данные поступают в интегратор;
- 5 — подготовленная обучающая выборка поступает в модуль работы с ИНС;
- 6 — результатом работы модуля является интеллектуальная модель объекта. Также на данном этапе происходит совершенствование модели объекта на основе вновь поступающих данных.

Третий этап — классификация. На данном этапе построенная интеллектуальная модель способна самостоятельно классифицировать поступающую в базу данных информацию. Назначения связей, представленных на рис. 1, на третьем этапе функционирования:

- 1.1 — ввод параметрических данных объекта;
- 2.1 — сохранение вновь поступившей информации в базе данных;
- 4.1 — запрос на вновь поступившую информацию;
- 4.2 — результаты запроса передаются в интегратор на обработку;
- 5 — обработанная информация передается в модуль работы с ИНС;
- 6 — подача данных на вход ИНС;
- 7 — рассчитанный моделью результат поступает в интегратор;

4.1 — сохранение результатов, рассчитанных моделью в базе данных;

2.2 — передача из базы данных результатов;

1.2 — представление результатов, рассчитанных моделью, пользователю.

Построение медицинской экспертной системы с интеллектуальным ядром с помощью предлагаемой технологии

В качестве примера реализации данной технологии при построении экспертной системы с интеллектуальным ядром на основе ИНС был выбран медицинский объект (начальная диагностика здоровья пациентов на основе результатов общего анализа крови).

Программная реализация модуля основана на алгоритме обучения ИНС, предложенном А. А. Арзамасцевым и др. в работе [4].

Общеклиническое исследование крови является одним из важнейших диагностических методов, которое тонко отражает реакцию кроветворных органов на воздействие на организм различных физиологических и патологических факторов. Во многих случаях оно играет большую роль в постановке диагноза.

В понятие "общеклиническое исследование крови" входят определение концентрации гемоглобина, подсчет числа эритроцитов, цветового показателя, лейкоцитов, скорости оседания эритроцитов и лейкоцитарной формулы.

Выборка для обучения ИНС представляла собой результаты общего анализа крови 400 пациентов. Входами обучающей выборки явились десять показателей общего анализа крови:

Нб-гемоглобин (г/л) — X_1 ;

скорость оседания эритроцитов (мм/ч) — X_2 ;

лейкоциты (10^9 /л) — X_3 ;

эритроциты (10^{12} /л) — X_4 ;

цветовой показатель — X_5 ;

лейкоцитарная формула (базофилы — X_6 , нейтрофилы палочкоядерные — X_7 , нейтрофилы сегментоядерные — X_8 , лимфоциты — X_9 , моноциты — X_{10}).

В качестве выходов ИНС были приняты различные состояния здоровья пациента, данные экспертом и соответствующие результатам анализа крови — Y :

1 — здоров;

2 — необходимо амбулаторное лечение;

3 — необходимо стационарное лечение;

4 — экстренная госпитализация.

Программа, реализующая модуль работы с ИНС, позволяет выбрать два типа настройки сети: ручной, когда пользователь сам настраивает сеть для обучения, или автоматический — сеть настраивается самостоятельно. В качестве метода

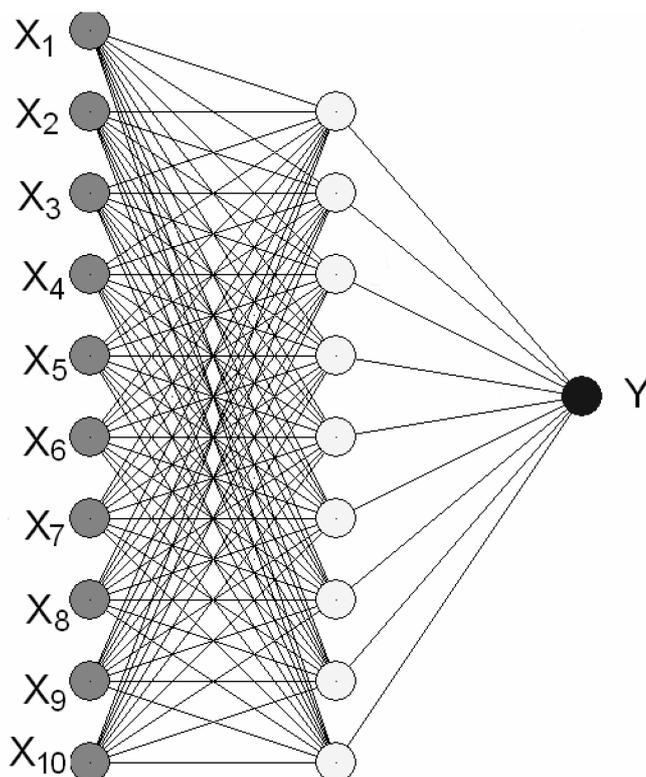


Рис. 2. Схематическое изображение структуры искусственной нейронной сети, подобранной в результате работы программы

минимизации ошибки используются метод полного сканирования, метод Монте-Карло, градиентный метод.

Предварительно обучающая выборка тестировалась на предмет выявления и устранения в ней различных противоречивых данных.

В результате обучения ИНС была подобрана структура искусственной нейронной сети, которая схематически представлена на рис. 2.

Данная сеть состоит из одного входного, одного выходного и одного скрытого слоя нейронов. Входной слой состоит из 10 нейронов, выходной — из 1 нейрона. Скрытый слой содержит 9 нейронов с линейной активационной функцией.

При оценке эффективности работы построенной медицинской экспертной системы было выявлено, что ЭС дает правильные рекомендации в 69 % случаев, близкие рекомендации — в 29 %, неверные рекомендации — в 2 % случаев.

Таким образом, с помощью предложенной автоматизированной технологии построения экспертных систем была разработана экспертная система на основе ИНС для медицинского объекта (диагностика здоровья пациентов на основе результатов общего анализа крови).

Достоинствами такой системы является способность к обучению на основе вновь поступивших данных. Навыки данной ЭС возрастают после каждого сеанса экспертизы.

Разработанная ЭС может найти свое применение в различных медицинских учреждениях, где ведется общеклиническое исследование крови. Она может быть полезна лаборантам и стажерам, а также врачам для более эффективной организации своей работы.

Кроме того, данная система может применяться в домашних условиях при наличии у человека результатов его общего анализа крови.

Заключение

Таким образом, в данной работе предложена автоматизированная технология построения экспертных систем, базирующаяся на аппарате искусственных нейронных сетей.

На основе предложенной технологии разработана целостная интерактивная система, позволяющая осуществлять проектирование экспертных систем, использующих в качестве интеллектуального ядра ИНС-модели.

Реализация данной технологии была рассмотрена на примере медицинского объекта. Результатом явилось создание медицинской экспертной системы, позволяющей осуществлять диагностику здоровья на основе результатов общего анализа крови пациентов.

Список литературы

1. **Осовский С.** Нейронные сети для обработки информации / Пер. с польск. И. Д. Рудинского. М.: Финансы и статистика, 2004. 344 с.
2. **Арзамасцев А. А., Зенкова Н. А.** Использование аппарата искусственных нейронных сетей для идентификации свойств личности в учебном процессе // Открытое образование. 2004. № 4. С. 61—64.
3. **Арзамасцев А. А., Зенкова Н. А., Неудахин А. В.** Автоматизированная технология построения экспертных систем с интеллектуальным ядром на основе ИНС-моделей // Открытое образование. 2008. № 3 (68). С. 35—39.
4. **Арзамасцев А. А., Крючин О. В., Азарова П. А., Зенкова Н. А.** Универсальный программный комплекс для компьютерного моделирования на основе искусственной нейронной сети с самоорганизацией структуры // Вестник Тамбовского университета. Сер. Естественные и технические науки. Тамбов, 2006. Т. 11. Вып. 4. С. 564—570.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

УДК 004.424.3::004.272.2::004.422.636

Введение

В. В. Пекунов, канд. техн. наук, доц.,
Ивановский государственный энергетический
университет, Региональный НОЦ
по наноматериалам "Жидкие кристаллы"
(г. Иваново),
e-mail: pekunov@mail.ru

Процедуры с планированием повторного входа в языках высокого уровня при традиционном и параллельном программировании

Предлагается новый подход к программированию алгоритмов, сводимых к порождению серии схожих подзадач с планированием последовательности их решения в соответствии со стратегиями постановки в очередь, стек или дек. Подход предполагает ввод в языки высокого уровня нового формализма: процедуры с планированием повторного входа. Описаны синтаксис и семантика таких процедур. Предложена стратегия распараллеливания процесса решения задач на многопроцессорных/многоядерных системах с общей памятью.

Ключевые слова: алгоритмы, очередь, стек, дек, процедура с планированием повторного входа, параллельное программирование, генератор.

Одним из традиционных приемов программирования является сведение задачи к схожим подзадачам меньшей размерности. Для алгоритмов, основанных на таком подходе, естественным решением является применение типовых линейных структур данных "дек", "очередь" и других. Назовем, например, алгоритмы:

- поиска в ширину в дереве [1];
- нумерации узлов дерева таким образом, чтобы в пределах каждого уровня нумерация была последовательной;
- нахождения в графе цепочки сетевых подграфов;
- поиска кратчайшего пути между электронными деталями на плате с применением волнового алгоритма Ли;
- нумерации узлов дерева по тройкам "родитель — потомки".

Первые четыре задачи решаются с помощью типовой структуры данных "очередь". Из множества возможных вариантов такого решения проанализируем наиболее значимые. В простейшем случае возможен явный ввод дополнительной переменной — очереди (реализованной с применением шаблонов STL [2]). Более перспективным решением может быть разработка соответствующего паттерна проектирования [3]. Оба подхода требуют ввода и контроля дополнительных структур данных, что чревато возникновением элементарных синтаксических и мелких логических ошибок программирования.

Пятая задача решается с помощью структуры "дек" (при обходе дерева по тройкам узлы ставятся в конец очереди, а при перенумерации узлов внутри троек — в начало).

Возможно решение указанных задач с помощью генераторов (разновидности сопрограмм [4]), реализованных, например, в языке Python [5], которое, однако, не является интуитивно понятным и не дает явных предпосылок для адекватного распараллеливания счета.

Предлагается новый, более компактный, очевидный и перспективный для параллельного решения подход к представлению алгоритмов упомянутого выше класса, сводимых к порождению серии схожих подзадач с планированием последовательности их решения в соответствии с одной из стратегий: очереди, стека, дека.

1. Планирование повторного входа в процедуру в языках высокого уровня

Предлагается ввести в алгоритмические языки высокого уровня формализм процедуры с планированием повторного входа. Такая процедура будет отличаться от обычной наличием плана исполнения, элементами которого (этапами) являются векторы значений параметров для очередного вызова процедуры. Рассмотрим два варианта реализации плана: динамический и статический.

1.1. Динамический план. В данном случае любой явный (в том числе рекурсивный) вызов процедуры с повторным входом создает *новый план*, который сначала содержит один элемент (начальный этап) — вектор значений параметров, указанных при вызове процедуры. В ходе исполнения процедура может включить в план один или несколько дополнительных этапов с указанием соответствующих значений параметров процедуры. Включение в план может проводиться в соответствии с одной из основных стратегий: "очередь", что соответствует вставке (планированию) в конец плана, или "стек", что соответствует вставке в начало плана¹. Потребность в тех или иных операциях вставки определяется реализуемым алгоритмом. При выходе из процедуры проводится проверка плана: если план пуст, то осуществляется возврат в вызывающую программу, в противном случае из начала плана извлекается очередной этап, соответствующие ему значения параметров помещаются в параметры процедуры и проводится повторный вход (переход в начало процедуры).

1.2. Статический план. Если процедура имеет *статический (постоянный) план*, то его исполнение может быть остановлено (с выходом из процедуры) и возобновлено при следующем входе в процедуру. Возможны две модификации статического плана: *глобальный* (единый на всех уровнях рекурсивного вызова), *локальный* (индивидуальный на каждом уровне рекурсии).

Существуют два способа входа в процедуру:

- обычный (характерный для динамического плана), с включением нового этапа в начало плана независимо от наличия или отсутствия в нем других этапов;
- с возобновлением исполнения, при котором значения параметров, указанные при вызове, игнорируются. Параметры процедуры получают значения, соответствующие первому этапу в плане. Если план пуст,

¹ План исполнения может быть реализован в виде очереди или стека (буфера с одним направлением включения данных), либо их комбинации — дека (буфера с двумя направлениями, например, кольцевого). Выбор реализации осуществляется компилятором в зависимости от операций с планом, задействованных в алгоритме.

то возникает ошибочная ситуация, которую можно индицировать, например, генерацией исключения.

Применение статического плана напоминает программирование с использованием сопрограмм с той разницей, что для рассматриваемых нами процедур точкой входа *всегда* является начало процедуры. Возможна *разработка простых или рекурсивных генераторов* (на базе глобального или локального статического плана). Простые генераторы позволяют компактно реализовать, например, стек и очередь для сложных типов данных без явного ввода дополнительных структур. Но в первую очередь статический план необходим при реализации алгоритмов, формирующих промежуточные результаты, требующие внешней обработки, по завершении которой уже запланированные работы должны быть продолжены. Сюда относятся схемы поиска с возобновлением, в частности, поиск кратчайших путей от одной исходной точки до нескольких конечных по волновому алгоритму Ли, где промежуточным результатом будет путь до очередной точки.

1.3. Передача параметров. Опишем особенности *основных правил передачи параметров* в процедуры с повторным входом. Параметры, передаваемые по значению, могут быть использованы тем же образом и по тем же правилам, что и при вызове обычных процедур. Изменения затрагивают лишь параметры, передаваемые по ссылке. В этом случае предлагается организовать "туннелирование по умолчанию" последних значений таких параметров между последовательными вызовами процедуры. В некоторых случаях может иметь смысл разрыв такой цепи передачи (*отсечение*) путем явного указания значений таких параметров на очередном этапе вызова.

Сформулируем простые правила:

- если при планировании этапа было затребовано отсечение, то при входе в процедуру, соответствующем данному этапу, параметр, передаваемый по ссылке, получит значение, явно указанное при планировании этапа;
- если при планировании этапа не предполагалось отсечение, то при соответствующем входе в процедуру параметр, передаваемый по ссылке, будет иметь значение, полученное при завершении предшествующего этапа (после очередного выхода из процедуры);
- при возврате в вызывающую программу фактический параметр, переданный по ссылке, будет иметь значение, полученное на последнем этапе исполнения процедуры с повторным входом.

2. Синтаксис процедур (void-функций) с планированием повторного входа в языке C/C++

Пусть возможность планирования повторного входа будет существовать лишь для процедур (точнее, void-функций в терминах C/C++). Предлагается ввести в язык несколько новых ключевых слов и функций, а также расширить трактовку служебного символа "!" и ключевых слов **static** и **continue**.

Заголовок процедуры с повторным входом имеет формат:

заголовок = "reenterable" [ограничение] пробелы ["static" пробелы [{"local"} {"global"}] пробелы] имя [пробелы] "(" [параметры] ")" ограничение = "[" максимальное_число_этапов "]"

Ключевое слово **reenterable** свидетельствует об отсутствии возвращаемого результата (процедура) и о возможности повторного входа. Наличие **static** (или **static global**) означает применение глобального статического плана, **static global** — локального статического, отсутствие **static** — динамического. Если максимальное число этапов не указано, то план может иметь произвольный размер.

Обычный вызов процедуры с повторным входом записывается в той же форме, что и вызов любой процедуры (void-функции) C/C++. Вызов с возобновлением исполнения отличается наличием ключевого слова **continue** перед именем процедуры. Способ вызова внутри процедуры идентифицируется функцией **plan_after_continue()**, которая возвращает **false** при обычном вызове и **true** при возобновлении.

Включение нового этапа в конец плана исполнения реализуется обращением к функции **plan_last**, которая существует в пределах текущей процедуры с повторным входом и имеет эквивалентный список формальных параметров. Значения, указанные в списке параметров при вызове **plan_last**, станут значениями параметров процедуры на этапе, спланированном при данном вызове, с вышеуказанными оговорками относительно параметров, передаваемых по ссылке.

Включение нового этапа в начало плана реализуется обращением к функции **plan_first** с теми же областью видимости, параметрами и правилами их передачи, что и при вызове **plan_last**.

Директива **clear_plan** очищает план исполнения в пределах текущей процедуры. Такая возможность может быть востребована в алгоритмах с досрочным завершением запланированных работ (например, при работе с волновым алгоритмом Ли прекращается анализ оставшихся элементов "волны", если кратчайший путь найден).

Прекращение исполнения плана реализуется директивой **plan_stop**. Ее исполнение не приводит к немедленному выходу из процедуры, но сигнализирует, что после завершения текущего этапа плана должен произойти возврат в вызывающую программу. При возврате динамический план очищается, а статический — сохраняется, что дает возможность повторного входа в процедуру с возобновлением исполнения плана.

Если при планировании этапа необходим разрыв (отсечение) цепи передачи параметра по ссылке, то предлагается указывать символ "!" непосредственно после значения соответствующего фактического параметра при обращении к функциям **plan_first** и **plan_last**.

3. Реализация некоторых алгоритмов

Во введении к данной статье перечислялись некоторые алгоритмы, которые могут быть решены с помощью очереди и дека. Более компактно, естественно и надежно эти алгоритмы могут быть записаны с применением процедур с планированием повторного входа (вставка в конец плана эквивалентна применению очереди, вставка как в начало, так и в конец эквивалентна использованию дека). Повышение надежности программирования (снижение числа потенциальных ошибок) объясняется устранением необходимости в явном вводе управляющей структуры (стека, дека, очереди), подразумевающей декларацию вспомогательных типов и переменных и организацию вставки и циклического извлечения элементов.

Рассмотрим последовательную нумерацию узлов дерева по уровням. Воспользуемся вспомогательной структурой, описывающей узел:

```
typedef struct TreeTag {
    int Data;
    struct TreeTag * Left;
    struct TreeTag * Right;
} TreeNode;
```

Запишем соответствующий код, в котором переменная **Root** — указатель на корневой элемент дерева:

```
int Number;
reenterable EnumerateByLevel (TreeNode * Cur) {
    Cur->Data = Number++;
    if (Cur->Left) plan_last (Cur->Left);
    if (Cur->Right) plan_last (Cur->Right);
}
/* Вызов: Number = 1; EnumerateByLevel (Root); */
```

Рассмотрим алгоритм нумерации узлов дерева по тройкам. Воспользуемся туннелированием параметра по ссылке **Number**:

```
reenterable EnumerateByFamilies (TreeNode * Cur,
    char Level, int &Number) {
    Cur->Data = Number++;
    if (Level) {
        if (Cur->Left) plan_last (Cur->Left, 0, Number);
        if (Cur->Right) plan_last (Cur->Right, 0,
            Number);
    }
    else {
        if (Cur->Right) plan_first (Cur->Right, 1,
            Number);
        if (Cur->Left) plan_first (Cur->Left, 1,
            Number);
    }
}
/* Вызов:
int Number = 1;
EnumerateByFamilies (Root, 0, Number); */
```

Проиллюстрируем применение статического плана, реализовав очередь², каждый элемент которой включает два значения (целочисленное и вещественное). При обычном обращении к процедуре **queue** значение помещается в очередь, а при вызове с возобновлением — извлекается:

```
reenterable static queue (int &Int, double &Dbl) {
    if (!plan_after_continue()) plan_last (Int, Dbl);
    plan_stop;
}
/* Помещение в очередь пары (A, B): queue (A, B); */
/* Извлечение пары (A, B): continue (A, B); */
```

Такая реализация несколько проще и потенциально дает более компактный и эффективный код по сравнению с шаблонами STL, так как не требует декларации (явной и/или неявной) новых классов и работы с объектами. Теоретически быстрее осуществляется и компиляция, поскольку не проводятся разбор и интерпретация шаблонов с выводом классов. Полученная реализация также компактнее и по сравнению с организацией очереди как простого массива структур.

² Аналогично можно организовать стек, заменив в реализации **plan_last** на **plan_first**.

4. Применение в параллельном программировании

Если рассматривать планирование очередного этапа исполнения процедуры как планирование новой, относительно независимой подзадачи, которая может быть исполнена на отдельном процессоре/ядре, то применение процедур с повторным входом может быть одним из вполне оправданных подходов к распараллеливанию алгоритма. Это особенно простой вариант распараллеливания для алгоритмов, реализация которых с применением процедур с планированием повторного входа выгодна изначально.

В эту группу входят, в частности, алгоритмы групповой обработки элементов, включенных в сложную нелинейную структуру данных (дерево, сеть), например, поиска минимального элемента дерева или суммирования таких элементов. Для эффективной обработки линейных структур данных (массивов) в параллельных языках программирования и параллельных расширениях стандартных языков обычно существуют стандартные средства, которые для повышения эффективности даже могут предусматривать балансировку загрузки процессоров системы, например, конструкция **parallel for** в расширении OpenMP [6] имеет специальный режим **dynamic**, регулирующий загрузку процессоров (методология "портфеля задач" [7]). Для параллельной обработки нелинейных структур данных стандартных средств пока нет. Однако можно "линеаризовать" задачи обработки таких структур, если для реализации алгоритма воспользоваться формализмом процедур с повторным входом, предполагающим исполнение линейного плана этапов обработки. Задача сведется к адекватному распараллеливанию процесса исполнения плана работ, что может быть реализовано внутренними средствами системы с применением стандартных приемов параллельной обработки линейных структур данных с балансировкой, например, той же идеи "портфеля задач". Разумеется, в таком случае необходимо четко выделять фрагменты плана, в пределах которого возможно одновременное исполнение работ.

Отметим, что для некоторых алгоритмов применение процедур с планированием повторного входа не дает существенного эффекта при работе с одноядерными системами³, но позволяет эффективно распараллелить расчет в многопроцессорных/многоядерных системах. Например, применение *повторного входа со вставкой в начало плана* при определенных условиях может быть эквивалентно программированию с применением рекурсии с точки зрения организации последовательности вызовов. В системах с общей памятью замена рекурсии на планирование в начало может быть *одной из стратегий*, позволяющих с минимальными затратами (или даже автоматически) распараллелить алгоритм. Наиболее эффективна такая стратегия для алгоритмов с полностью независимыми по данным ветвями дерева рекурсивных вызовов, например, для поиска оптимального хода в позиционных играх. Решение будет не менее простым, чем, например, полученное с применением концепции Т-параллелизма [6].

4.1. Параллельное исполнение групп запланированных этапов обработки. Дополнение синтаксиса C/C++. Планирование этапов обработки в общем случае, безусловно, подразумевает наличие детерминированной последовательности

исполнения этих этапов, которая для случая одноядерной системы определяется естественной последовательностью просмотра плана — от начала к концу. В случае многопроцессорной системы необходимо, придерживаясь, в целом, той же последовательности, обеспечить параллельность исполнения по меньшей мере некоторых этапов.

Предлагается ввести понятие *группы этапов* и представить план в виде последовательности непересекающихся групп. Этапы группы могут выполняться либо последовательно, либо параллельно. Всякая очередная группа по умолчанию исполняется в последовательном режиме, начиная с первого включенного в нее этапа. В процессе последовательного исполнения группы возможно *однократное переключение* в режим параллельного исполнения (обратное переключение недопустимо). В *параллельном режиме* группа рассматривается как динамически пополняемый "портфель задач": любые этапы, входящие в группу (в том числе и новые, попадающие в группу в результате планирования при исполнении уже включенных в группу этапов), могут исполняться параллельно и в любой последовательности на любом из процессоров системы.

Необходимо учитывать тот факт, что в параллельном режиме обеспечить "туннелирование по умолчанию" и корректность отсечения значений параметров, передаваемых по ссылке, в общем случае невозможно. Такие параметры будут доступны для одновременной модификации всем процессорам на любом этапе и, следовательно, могут рассматриваться как глобальные переменные, для корректной работы с которыми необходимо использовать семафоры или иные средства монопольного доступа.

План делится на группы с помощью *маркеров* — специальных пометок в плане, которые могут разделять два любых этапа, а также присутствовать до первого и/или за последним этапом. Группой является:

- весь план, если он не содержит маркеров;
- участок плана между двумя маркерами;
- участок, начинающийся с первого этапа и заканчивающийся этапом перед ближайшим справа маркером;
- участок, заканчивающийся на последнем этапе плана и начинающийся этапом после ближайшего слева маркера.

Позиция плана, соответствующая концу произвольной группы (маркеру или концу плана), фактически является точкой барьерной синхронизации. Перед началом работы (при явном вызове процедуры с повторным входом) план включает единственный начальный этап, не содержит маркеров и, следовательно, является группой.

Предлагается ввести новые ключевые слова (**plan_group_first**, **plan_group_last**, **plan_group_parallelize**), представляющие соответственно три новые базовые операции для работы с планом:

- вставку маркера в начало плана;
- вставку маркера в конец плана;
- переключение из последовательного режима исполнения группы в параллельный.

Также предлагается ввести новую функцию **plan_processor_id()**, возвращающую логический номер процессора (ядра) системы, на котором исполняется текущий этап плана. При работе с одноядерной системой новые ключевые слова игнорируются, а функция

³ Имеется в виду однопроцессорный компьютер, процессор с одним ядром.

plan_processor_id() всегда возвращает номер единственного процессора (ноль). Такая реализация следует основным принципам построения современных параллельных расширений языков программирования (OpenMP, DVM, HPF/HPC [7]).

4.2. Параллельная реализация некоторых простых алгоритмов. Приведем простой и малозатратный вариант параллельной реализации алгоритма поиска максимального элемента в дереве с корнем Root. Узлы дерева представляют собой элементы данных типа `TreeNode`, константа `nProcs` — число ядер в системе. Параллельный поиск⁴ представлен процедурой с повторным входом `_TreeMax`, функция `TreeMax` (с параметром — указателем на корень дерева) собирает частные результаты поиска по всем процессорам и возвращает конечный результат:

```
int MaxResult [nProcs];
reenterable _TreeMax (TreeNode * Cur) {
    int thread_id = plan_processor_id();
    if (Cut == Root) plan_group_parallelize;
    if (Cur->Left) plan_last (Cur->Left);
    if (Cur->Right) plan_last (Cur->Right);
    if (MaxResult[thread_id] < Cur->Data)
        MaxResult[thread_id] = Cur->Data;
}
int TreeMax (TreeNode * Root) {
    int Result;
    memset (MaxResult, 0, sizeof (MaxResult));
    _TreeMax(Root);
    Result = MaxResult [0];
    for (int i = 1; i < nProcs; i++)
        if (MaxResult[i] > Result) Result = MaxResult [i];
    return Result;
}
```

Также представляет интерес параллельная реализация сортировки массива целых чисел методом Шелла. Метод предполагает последовательное выполнение стадий, на каждой из которых решается (в параллельном режиме) серия однотипных подзадач сортировки фрагментов массива. Выделение стадий реализуется группировкой этапов плана, каждому из которых (за исключением первого, на котором формируется план) ставится в соответствие подзадача сортировки. Запишем реализующий код:

```
reenterable Shell (char Plan, int * Arr, int N,
int incr) {
    static int Incr[20] = {1};
    int i, j;
    plan_group_parallelize;
    if (Plan) { /* Генерируем общий план исполнения */
        for (i = 0; Incr[i] < N; Incr[i +
1] = 3*Incr[i] + 1, i++);
```

⁴ Реализация алгоритма с применением многих параллельных расширений C/C++ потребовала бы введения дополнительной переменной (очереди обработки или массива ссылок на узлы дерева) и цикла обработки, итерации которого распределялись бы между процессорами. Естественнее выглядела бы реализация поиска с применением T-параллелизма, но такой подход может быть менее эффективным, учитывая внутренние затраты на организацию контроля готовности значений переменных.

```
while (i--) {
    int NLists = N%Incr[i] ? Incr[i]: Incr[i];
    int * B = Arr;
    plan_group_last; /* Отмечаем начало новой
стадии */
    for (j = 0; j < NLists; j++)
        plan_last (0, B++, N, Incr[i]); /* Соз-
дание подзадачи */
}
else /* Решение подзадачи */
    /* Сортировка фрагмента массива Arr[0, incr,
2*incr, ...] */
    InternalSort (Arr, N, incr);
}
/* Вызов: Shell (1, массив, число_элементов, 0); */
```

Заключение

Предложен новый формализм процедур с планированием повторного входа, который дает более компактную и очевидную реализацию для ряда задач, традиционно решаемых с помощью вспомогательных типовых структур данных: очереди, стека и дека. Также возможна разработка аналогов простых и рекурсивных генераторов.

Предложен простой вариант распараллеливания процесса решения задач на многопроцессорных и многоядерных системах с общей памятью при использовании процедур с повторным входом. Параллелизация решения сводится к адекватному внутреннему распараллеливанию исполнения плана работ с балансировкой, которое реализуется системой программирования. Это позволяет "линеаризовать" задачи параллельной обработки нелинейных структур данных и выделить скрытый параллелизм некоторых рекурсивных алгоритмов, отчасти заменив рекурсию планированием.

В настоящее время разработан прототипный вариант препроцессора⁵, транслирующий декларации и вызовы процедур с повторным входом в код на стандартном языке C++.

Работа была выполнена при финансовой поддержке Минобразования и науки (грант РНП.2.2.1.1.7280).

Список литературы

1. **Рассел С., Норвиг П.** Искусственный интеллект: современный подход. М.: Вильямс, 2007. 1408 с.
2. **Остерн М. Г.** Обобщенное программирование и STL: Использование и наращивание стандартной библиотеки шаблонов C++. СПб.: Невский Диалект, 2004. 544 с.
3. **Бобровский С. И.** Технологии Delphi 2006. Новые возможности. СПб.: Питер, 2006. 288 с.
4. **Кнут Д.** Искусство программирования. Т. 1. Основные алгоритмы. 3-е изд. М.: Вильямс. 2006. 720 с.
5. **Сузи Р. А.** Язык программирования Python: Учеб. пос. М.: ИНТУИТ, БИНОМ, Лаборатория знаний, 2006. 328 с.
6. **Воеводин В. В., Воеводин Вл. В.** Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
7. **Эндрюс Г. Р.** Основы многопоточного, параллельного и распределенного программирования. М.: Вильямс, 2003. 512 с.

⁵ Доступен по ссылке: <http://www.pekunov.chat.ru/Progs.htm>

УДК 004.738.5:001.891.57:002

Г. А. Тарнавский, д-р физ.-мат. наук, вед. науч. сотр., e-mail: gennady.tarnavsky@gmail.com,
А. В. Алиев, аспирант, e-mail: ak2003@ngs.ru,
В. С. Анищик, e-mail: VladimirAnishchik@gmail.com,
Институт вычислительной математики
и математической геофизики СО РАН,
г. Новосибирск,
А. Г. Тарнавский, С. Б. Жибинов,
Научно-исследовательский институт
прикладной информатики
Сибирского государственного университета
телекоммуникаций и информатики,
С. С. Чесноков,
Новосибирский государственный
технический университет

Информационные технологии и проблемы создания Центра компьютерного моделирования в Интернете

Анализируются технологические, экономические и правовые проблемы создания особого Web-ресурса в Интернете — Центра компьютерного моделирования.

Ключевые слова: информационные технологии, Интернет, дистанционное компьютерное моделирование, распространение научных знаний.

Введение

В настоящее время выполняется проект, который поддержан Российским фондом фундаментальных исследований (РФФИ, грант 08-07-12001), предназначенный для реализации современных потребностей в коммерциализации научных продуктов. Практическая реализация этого проекта требует решения фундаментальных проблем, связанных с разнообразными научными, техническими, правовыми, социальными и психологическими аспектами разработки специализированного Web-ресурса, особой точки обмена произведенным научным продуктом, в том числе и на платной основе, в системе Интернет. Одна из главных целей проекта состоит в подготовке и продвижении во Всемирной Сети функционально ориентированного портала — "торговой площадки", на которой обеспечивается коммерческое завершение научных разработок (рис. 1, см. четвертую сторону обложки).

Ранее при поддержке РФФИ был выполнен проект 04-07-90002, результатом которого явилось создание в среде Интернет Информационно-вычислительного центра (далее — Центр) по распространению научных знаний в области компьютерного моделирования задач аэродинамики и физической газовой динамики (<http://www.SciShop.ru>). Также был проведен запуск созданного Web-ресурса в опытную эксплуатацию в целях отработки современных информационных технологий в многостороннем плане, которая в перспективе имела основную цель — коммерциализацию проведенных научных разработок [1–8]. При создании экспериментального Интернет-центра были успешно решены многие сложные проблемы, связанные с реализацией его дизайн-проекта. В деталях разработана, практически реализована и верифицирована архитектура (коммуникационный граф), определено структурирование на разделы и стандартизованы сегменты Центра. В эти сегменты входят библиотеки научной литературы, табличные и графические базы данных, содержащие результаты ранее проведенных расчетов, а также процессорные системы, позволяющие посетителю самому провести компьютерное моделирование ряда классов задач в Интернете в режиме реального времени и получить в свое распоряжение результаты.

Цель проекта

Целью данного реализуемого проекта является совершенствование Центра, функционирующего в настоящее время в экспериментальном режиме, и превращение его в Центр нового поколения — активно посещаемый бизнес-портал, "торговую точку" Интернета по распространению научных знаний.

Проводится существенное дополнение контента действующего Центра, выполняются работы по повышению комфортности пребывания клиента в Центре, улучшению качества обслуживания, расширению визуального статического графического представления научных материалов на базе современных технологий мультимедиа (динамические изображения, возможность конструирования презентаций из полученных клиентом таблично-цифровых результатов работы процессорных систем и т. п.).

Расширяется концепция коммерческой системы функционирования Центра с использованием не только электронной платежной системы WebMoney, но и других актуальных биллинговых систем (Yandex-деньги, E-gold и др.). В целях привлечения новых клиентов используется политика активного маркетинга, в том числе взаимный обмен ссылками с научными порталами Сети высокой посещаемости. Совершенствуются системы защиты сайта от хакерских атак, спама и несанкционированного доступа к сегментам Центра и, при необходимости, используются системы шифрования внутренних потоков информации.

Центр ориентирован на три аспекта использования: как электронная книга (в т. ч. учебник), как электронный справочник (база данных) и как инструмент для научных исследований (процессорная система), и предназначен для хранения, пополнения и систематизации накопленной информации, обеспечения непрерывности научного прогресса и его ускорения в данной области за счет преемственности осуществленных разработок.

Методология разработки

Последние достижения в области информационных технологий и Интернета дают основания считать, что в настоящее время уже складываются новые методы организации процесса обмена научными знаниями, а в ближайшем будущем, по-видимому, произойдет повсеместный переход от традиционных способов распространения научных знаний через бумажные журналы и книги к их электронным аналогам. Следует также ожидать массового появления аналогичных электронных форм передачи вещественных результатов научной деятельности в области математического моделирования (вычислительных методов, алгоритмов, компьютерных программ и их финала — полученных числовых значений, структурированных в виде банков табличных и/или графических данных).

Весьма перспективной представляется новая форма передачи знаний в области компьютерного моделирования от автора-разработчика к пользователю-потребителю не через описание алгоритмов и результатов в журналах, а создание специалисту-прикладнику, аспиранту, студенту возможности непосредственно в любое время ознакомиться с функционированием созданных компьютерных программ и результатами расчета в режиме удаленного доступа через Всемирную сеть, самостоятельно запустив собственную, интересующую его задачу на счет. При этом, как и вообще в науке, так и в Интернет-сообществе в частности, передача результатов научной деятельности может осуществляться и безвозмездно, и на платной, коммерческой основе.

Ранее был реализован проект создания в Интернете локального портала Web-сети — информационно-вычислительного центра "Аэромеханика" с широкими функциями по распространению и обмену результатами научной деятельности (который в настоящее время является разделом "Центр-1" расширенного Web-ресурса). Проект опирался на опыт деятельности авторов в области математического моделирования различных проблем механики сплошных сред. Деятельность Центра на первом этапе реализации проекта была направлена на внедрение современных информационных технологий обеспечения и поддержки фундаментальных научных и прикладных исследований в области высокоскоростной аэродинамики с дополнением в дальнейшем разработками в других областях знания.

Была разработана общая концепция создания в Интернете Информационно-вычислительного центра и создан его коммуникационный граф. Детально исследовались проблемы конструирования этого Web-ресурса

не только как информационного Интернет-сайта, но и как специфической торговой площадки, на которой предлагается особый, интеллектуальный, продукт. Был решен большой ряд вопросов, таких как специфика взаимоотношений "клиент — Центр" для обеспечения комфортности пребывания посетителя в Центре, выработка позиций научного и художественного дизайна, проведение эффективной маркетинговой политики, защита процессорных систем и баз данных от несанкционированного доступа для хищения информации и защиты сайта в целом от хакерских атак. Проведена тщательная разработка общей архитектуры Центра и его главных структур — информационно-вычислительных комплексов (ИВК). Эти комплексы являются независимыми ветвями дерева графа и реализованы как унифицированные последовательности страниц соответствующих линий Web-сайта. Данные линии ориентированы на многоцелевое взаимодействие с клиентами — посетителями Центра, с предоставлением следующих услуг:

- доступ к библиотекам, содержащим научные статьи в соответствующих областях знания;
- доступ к табличным и/или графическим базам данных, содержащим результаты компьютерного моделирования соответствующих научных задач;
- доступ к процессорным системам, позволяющим клиенту самому организовать и провести компьютерное моделирование интересующих его задач;
- доступ к шлюзам для выхода на сайты (рис. 2, см. четвертую сторону обложки) ведущих российских и зарубежных журналов в соответствующих областях знания;
- доступ на сегмент "Форум" для получения консультаций и обсуждения проблем.

В дальнейшем будет продолжено совершенствование функционального наполнения и сервисного обеспечения этих линий. Подчеркнем, что потребительские свойства вычислительного инструментария определяются не только степенью эффективности его применения в своей научной области, но и степенью комфортности пребывания клиента в Центре и уровнем его обслуживания. В новых версиях ИВК "Удар" и "Поток" пользовательские сервисы продолжают всесторонне совершенствоваться. В распоряжение пользователя предоставляются в свободном доступе все научные монографии и статьи авторов комплекса, а также имеющаяся программная документация. Результаты решения классов задач в области высокоскоростной аэродинамики организуются в базы данных, доступных клиенту.

Контент Web-ресурса и проблемы интеллектуальной собственности

Размещаемые в Центре научные продукты разделяются на две группы (следует даже использовать термин "группировки") объектов "торговой площадки". Различия этих группировок лежат в плоскости экономических и юридических проблем — прав собственности на интеллектуальные продукты. Эти проблемы являются весьма существенными, и в настоящем проекте будут

разрабатываться и использоваться различные методики их решения.

Первая группировка научных продуктов — собственные разработки участников проекта. Этот спектр научных разработок был создан непосредственно членами команды исполнителей проекта. Данные разработки размещены в разделе сайта "Центр-1" — это информационно-вычислительные комплексы "Астра" (А. В. Алиев), "Удар" (Г. А. Тарнавский), "Поток" (Г. А. Тарнавский, А. В. Алиев), "Нано" (Г. А. Тарнавский, В. С. Анишик) с соответственно вытекающими отсюда авторскими правами. Эти авторские права закреплены целым рядом публикаций в российских и зарубежных изданиях. Также подтверждены их авторские права и на разработки соответствующих структур Центра. Таким образом, авторские права разработчиков защищены действующим законодательством РФ, а порядок их использования — внутренними соглашениями.

В действующей версии Центра функционируют проблемно-ориентированные информационно-вычислительные комплексы "Удар" и "Поток", и предполагается их дальнейшая модернизация. ИВК "Удар" (рис. 3) обеспечивает расчет ударно-волновых структур на входе в

воздухозаборник гиперзвукового прямоточного воздушно-реактивного двигателя (ГПВРД). ИВК "Поток" (рис. 4, см. четвертую сторону обложки) предназначен для компьютерного моделирования аэродинамики высокоскоростного движения объектов в реальных газовых средах.

В настоящем проекте проводится дополнение Центра, помимо аэромеханики, знаниями из других областей науки. Расширение контента Центра будет обеспечено включением в его состав новых сегментов, в которые войдут эффективно функционирующие вычислительные комплексы математического моделирования ряда научных областей. Первый из них, "Астра", созданный на базе современного вычислительного SPH-метода, ориентирован на компьютерное исследование процессов гравитационной газовой динамики (эволюция межгалактических туманностей, образование звездно-планетарных систем из проточешества и т. п.). Второй комплекс, "Нано", предназначен для компьютерного моделирования физико-химических процессов формирования специальных наноструктур при конструировании новых полупроводниковых материалов для микропроцессорной техники.

Поскольку исполнителям проекта принадлежат (с учетом их вклада) авторские права на создаваемый Web-ресурс, то в настоящий момент не просматриваются предпосылки правовых коллизий между авторами инфраструктуры Web-ресурса и авторами контента, наполняющего эту инфраструктуру, поскольку эти группы авторов являются одними и теми же лицами. Это обстоятельство существенно, и весьма облегчает процесс разработки Web-ресурса.

Вторая группировка научных продуктов — неопределенный ассортимент. Реализуемый проект нацелен на создание Центра по распространению научных знаний, т. е. интеллектуальной продукции, разработанной не только исполнителями проекта. К участию в нем приглашаются как специалисты по профильным тематикам, так и специалисты в других областях знания, обладающие завершенным (в той или иной мере) научным товаром и желающие его распространять, в том числе и на коммерческой основе.

"Неопределенность ассортимента товара" делает процесс создания инфраструктуры Центра, его систем, конструкций и сегментов весьма сложной и вместе с тем интересной задачей. Помимо чисто технических трудностей, здесь весьма высока доля правовых проблем,

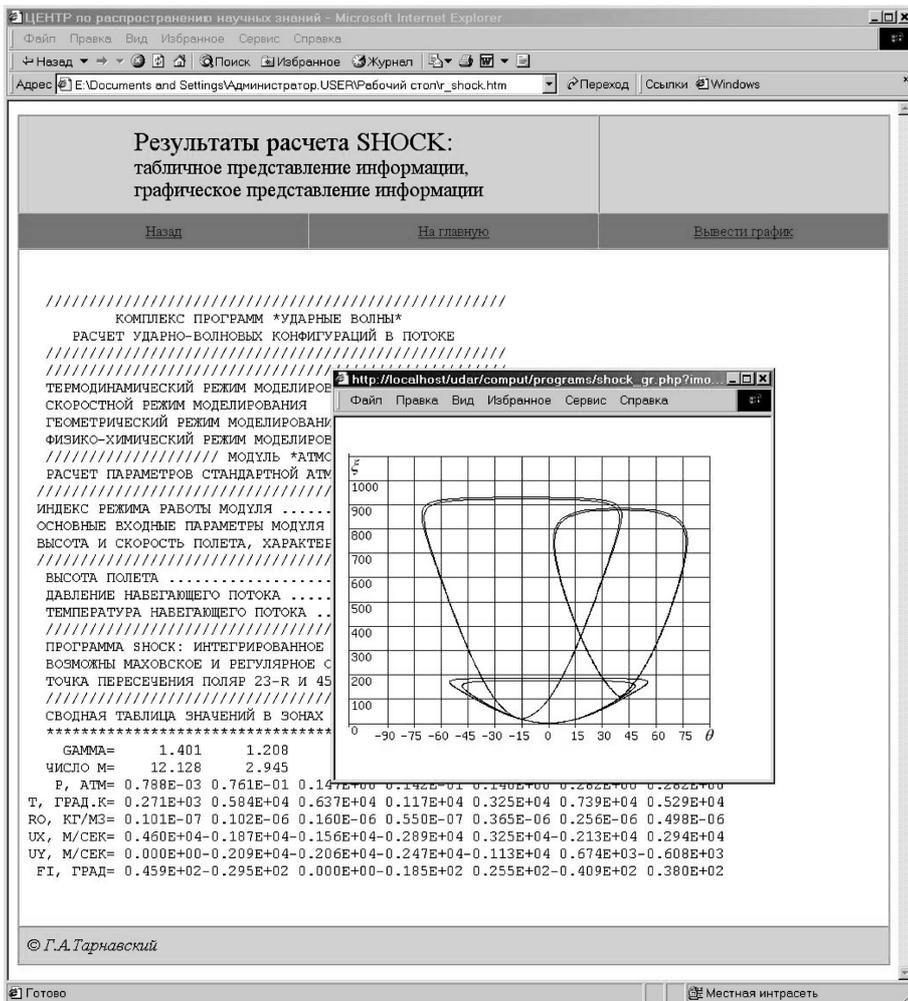


Рис. 3. Информационно-вычислительный комплекс "Удар". Таблично-цифровое и графическое представление результатов компьютерных исследований

взаимоотношений собственников продукта и собственников торговой площадки. В частности, возникает вопрос: если при размещении контента проводится его доработка (улучшение дизайна, повышение эффективности, оптимизация управления, проведение маркетинга, обеспечение электронными платежными системами и т. п.), а эта доработка, как правило, связана с высокими затратами времени и интеллектуальных усилий, то получают ли доработчики авторские права на улучшенный продукт? Какова их доля в авторском праве? Как распределять доход, если он возникает при коммерциализации (продажах) продукта или его части?

Все это ставит вопросы юридического, психологического и даже эстетического плана личностных взаимоотношений субъектов интеллектуального труда.

Подчеркнем, что вторая группировка будет содержать научные разработки любых других представителей научного сообщества, так как предлагаемый проект ориентирован на создание Центра продвижения научного продукта не только участников проекта. Разрабатываемый Центр — это открытая система, которая доступна для всех лиц, от студентов и до ведущих специалистов различных областей науки, причем не только в режиме "чтения" и получения услуг Центра, но и для размещения своего научного продукта для его продвижения в научной среде или для коммерческого использования.

Для обеспечения высокого научного уровня сайта, защиты его от информационно-научного шума, от лженаучных разработок и т. д. принимаются следующие меры:

- анкетирование специалиста, желающего разместить на сайте свои разработки, что позволит сделать выводы об основных научных результатах авторов разработок, сферах их применения, имеющихся внедрениях и т. д.;
- научная экспертиза размещаемых на сайте разработок. Таким образом, открытость Центра не означает его доступности для размещения научной продукции низкого уровня. Кроме ограничений экспертного характера, размещаемый на сайте научный продукт, естественно, должен удовлетворять определенным научным, технологическим и эстетическим требованиям сайта.

Крайне важным является вопрос о защите авторских прав, порядке использовании разработок, выплатах роялти в случаях коммерческой реализации научного продукта клиента.

Качественное решение этого вопроса в значительной степени зависит от состояния правовой базы действующего законодательства РФ, в частности, по таким понятиям, как электронный документ, электронная подпись и т. д.

Для обеспечения защиты прав специалистов, желающих разместить свой научный продукт в Центре для его продвижения, разработан типовой договор о размещении научного продукта, регулирующий отношения пользователя с собственником сайта по следующим основным вопросам:

- защита авторских прав;
- порядок использования научного продукта;
- порядок выплаты вознаграждения;

- условия приведения научного продукта в соответствии с технико-информационными требованиями Web-ресурса.

Кроме этого, в договор включаются и некоторые другие группы вопросов по защите авторских прав. Следует отметить, что структура договора о размещении научного продукта содержит типовые блоки действующих юридических договоров по перечисленным выше проблемам.

Деятельность в области информационных технологий проводится, в частности, в соответствии с законом РФ № 3523-1 "О правовой охране программ для электронных вычислительных машин и баз данных" (в ред. 2002 г.), в котором содержится положение: "Предоставляемая настоящим Законом правовая охрана не распространяется на идеи и принципы, лежащие в основе программы для ЭВМ или базы данных или какого-либо их элемента, в том числе на идеи и принципы организации интерфейса и алгоритма, а также языки программирования". Поэтому в ходе выполнения проекта не предполагается находить решения с их потенциальной патентоспособностью ввиду соответствующих ограничений на законодательном уровне. Также не предполагается использование решений и конструкций, защищенных правами сторонних организаций. Все права на функционирующие программные комплексы и сегменты, выставленные на сайте, принадлежат исключительно их авторам и регулируются соответствующими законами РФ.

В связи с весьма широким спектром интересных и трудных проблем команда исполнителей проекта составлена из специалистов различного профиля (экономика и право, математическое моделирование, аэромеханика, гидродинамика, нанотехнологии, Web-технологии, параллельные системы, компьютерная графика и дизайн, электронные бизнес-системы, психология).

Главные направления развития Центра

Многообразие спектра поставленных задач структурируется на ряд крупных сегментов, в соответствии с чем принят Общий план работ по развитию действующего и активно посещаемого Web-ресурса.

1. Модернизация действующих комплексов Центра "Удар" и "Поток". Создаются и выставляются в Центре новые версии действующих комплексов "Удар-2" и "Поток-5", расширяющие моделирование высокоскоростной аэродинамики в область гиперзвукового полета в атмосфере Земли и движения объектов в атмосферах Марса (версия 5.1), Венеры, Юпитера, Сатурна и других тел Солнечной системы, имеющих газовую оболочку (версии 5.2, 5.3, 5.4).

2. Расширение контента Центра системами "Астра" и "Нано". Расширение контента Центра обеспечивается за счет включения в его состав новых сегментов "Астра" и "Нано", которые, как и предыдущие сегменты "Удар" и "Поток", содержат собственные библиографические разделы, табличные и графические базы данных, а также процессорные системы, позволяющие клиенту-посетителю Центра организовать и провести в режиме реального времени во время пребывания в Центре компьютерного моделирования задач из соответствующих науч-

ных областей макро-, мезо-, и наномеханики сплошных сред (высокоскоростная аэродинамика, физическая газовая динамика, вычислительная астрофизика, нанотехнологии в микроэлектронике).

3. Расширение контента Центра научными продуктами "неопределенного ассортимента". К участию в проекте приглашаются специалисты, имеющие действующие программные комплексы компьютерного моделирования (без ограничений на области научных приложений) и желающие их распространять как безвозмездно (*free soft*), так и на коммерческой основе. Разработан эскизный проект типового договора о размещении научного продукта в Центре с экспериментальной отработкой его положений. Создается экспертный совет Центра, который будет определять: целесообразность размещения того или иного научного продукта и степень его технической и эстетической готовности к размещению на структурах Центра, перспективы распространения и возможный контингент потребителей.

4. Совершенствование дизайна предоставляемой информации. Проводится расширение созданного ранее статического дизайна, создание принципиально новых средств сопровождения пребывания клиента в Центре на базе современных мультимедиа (динамическая графика со звуковым сопровождением, оформление полученных данных в виде презентаций и т. п.).

5. Повышение комфортности пребывания посетителей в Центре. Условия комфортности включают в себя следующее:

- понятность и простота навигации по сайту;
- понятность и простота работы с библиотеками, табличными и графическими базами данных, содержащих теоретические и вычислительные результаты проведенных ранее исследований;
- доступность запуска процессорных систем (понятность смысла физико-математических параметров вычислительных комплексов, поддержка режимов автоматизированного ввода, наличие всплывающих подсказок);
- понятность и надежность работы постпроцессорных систем (интуитивная ясность смысла и полнота таблично-цифровой и графической информации);
- постпроцессорное обслуживание посетителя в случае существенного времени, требующегося на выполнение расчета (паковка полученных результатов в файл и отсылка его по Сети на адрес, указанный клиентом);
- комфортабельность ожидания окончания расчета при времени исполнения задания в течение 2—10 мин (музыкальный антракт, развлекательные видео, игры и т. п.) и удобство организации возвращения пользователя на сайт в целях получения своих результатов после длительных (2—10 ч) расчетов.

6. Совершенствование систем защиты сайта. Усиливаются системы защиты, которые предназначены для отражения хакерских атак, спама и несанкционированного доступа в целях хищения и/или порчи информации.

7. Организация активного маркетинга. Проводятся регистрация сайта в соответствующих Интернет-ката-

логах научной тематики; оптимизация кода портала для поисковых систем Google, Yahoo, Яндекс; энергичная техническая поддержка пользователей на форуме и по электронной почте; гибкая ценовая политика с возможностью бесплатного ознакомления пользователей с базовым функционалом выставляемых программных продуктов.

8. Расширение круга электронных платежных систем. Организуется использование для коммерческого сегмента Центра не только системы WebMoney, но и зарубежных электронных платежных систем типа E-Gold, PayPal, MoneyBookers, EvoCash и др. в целях существенного расширения круга клиентов. Проводится установка универсальной платежной системы "Robokassa", в которой осуществляется конвертация электронных денег различного типа, что повышает эффективность использования контента Центра с увеличением потребительского спроса.

9. Документирование Центра. Продолжается составление полного описания программной реализации Центра: общей архитектуры, сегментов вычислительных систем (методов, алгоритмов и программ), баз данных, систем сопровождения, пре- и постпроцессорных систем обеспечения расчета, средств дизайна, защиты информации, маркетинга и т. п.

10. Коммерческое функционирование в Интернете. Проводится непрерывное исследование востребованности контента Центра, анализ глубины и ширины посещаемости страниц Web-ресурса.

Организация вычислений в Центре

Важнейшим атрибутом данного сетевого ресурса, как и любого сайта, является комфортность пребывания клиента в Интернет-центре и удобная форма пользования его сегментами: библиографическим разделом, базами данных, содержащих уже полученную таблично-графическую информацию, и в особенности, процессорными системами. Именно это обычно вызывает наибольшие затруднения у пользователя любого вычислительного комплекса, особенно многофункционального. В проекте этому вопросу уделяется большое внимание.

Системы препроцессорной подготовки заданий (ввод параметров и запуск процессорных систем) организованы в ясной, удобной и предельно простой форме, не вызывающей двойного толкования и затруднений у специалистов даже с небольшим опытом. Это потребовало разработки специализированных алгоритмов и создания программных структур многоуровневой поддержки пользователя, которые и в дальнейшем будут развиваться и совершенствоваться.

Операции с процессорными системами осуществляются пользователем не на собственном компьютере при установке на нем комплекса, а в режиме дистанционного доступа по Всемирной Сети на его локальном портале — непосредственно в Центре компьютерного моделирования. Это дает возможность посетителю Центра в режиме реального времени провести изучение вычислительного комплекса, организовать решение интересующей его задачи и получить результаты компьютерных исследований. Обычно передача вычислительного комплекса за-

ключается в приобретении лицензии, документации и кодов компьютерной программы. После этого покупатель устанавливает приобретенный продукт на собственном компьютерном оборудовании. Как правило, это происходит с большими затруднениями, которые могут быть вызваны разнообразными причинами — от использования разных версий операционной системы до особенностей установленных у продавца и покупателя поддерживаемых систем.

При размещении процессорных систем в Центре таких проблем нет. Все интерфейсы налажены, хорошо отработаны и "притерты". В указанных диапазонах вариации параметров функционирования комплексов — безотказное (заметим, что на форуме сайта всегда можно задать любой вопрос и получить разъяснение). Еще одним преимуществом этого подхода является то, что пользователь освобождается от необходимости закупки аппаратного обеспечения (часто весьма недешевого), необходимого для осуществления нужных ему расчетов, фактически "арендуя" его у создателей сайта лишь на время решения своей задачи.

Подчеркнем, что такой эффективный метод эксплуатации процессорных комплексов посетителем Центра потребовал разработки и реализации оригинальных решений. Так как ни один из провайдеров Интернета не позволит проводить на своем узле массовые и, возможно, длительные вычисления, забирающие высокие ресурсы и уменьшающие пропускную способность каналов, то потребовалось использовать другую схему проведения расчетов.

В Центре клиент организует вычислительную задачу (выбирает процессорную систему и вводит в нее параметры) и запускает ее на счет. Системы этого сегмента сайта по сопровождению заданий пакуются в файл задание и отправляются по Сети в вычислительный центр, содержащий спектр компьютеров, в том числе многопроцессорные системы с общей памятью на базе *Intel Itanium-2*. Здесь выполняется решение задачи, результаты снова пакуются в файл с атрибутами клиента и отправляются либо в Центр, если клиент ждет, либо на его домашний адрес в Сети. Эта схема отлично зарекомендовала себя при опытной эксплуатации.

Заключение

Способ организации вычислений непосредственно в Центре самими пользователями существенно отличается от обычной цепочки действий исследователя, желающего воспользоваться "чужим" программным комплексом: поиск необходимого вычислительного инструментария (например через их описание в журналах) — контакт с разработчиком — заключение договора — передача программ — их установка на собственном компьютере — модернизация операционной среды — отладка интерфейсов — проведение вычислений и получение научных результатов.

Программные комплексы, ориентированные на массового потребителя, от студентов до специалистов, обеспечивающие комфортабельность условий работы,

более привлекательны и коммерчески успешны. Цель проекта — совершенствование Центра и превращение его в Центр нового поколения — активно посещаемый бизнес-портал, "особую точку" Интернета по распространению научных знаний.

Методы достижения этой цели предполагают совершенно другие способы передачи созданного научного продукта от авторов-разработчиков к клиентам-потребителям как на коммерческой, так и некоммерческой основе. Эффективность такой передачи (продажи, аренды) зависит не только от содержания научного продукта, но и определяется многими факторами, аналогичными тем, которые имеют место в различных системах торговли материальными ценностями, и связана с хорошо продуманной маркетинговой политикой.

Очень многое в данной проблематике зависит также от психологических факторов, таких как доступность и/или комфортность пользования продуктом, аспекты межличностного взаимодействия продавца с покупателем, уровень цен и т. п. При этом реализация научного продукта имеет свои специфические проблемы, отсутствующие при распространении других видов материальной продукции. На выявление, анализ и решение таких проблем и ориентирован выполняемый проект.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 08-07-12001-офи).

Список литературы

1. **Тарнавский Г. А., Жибинов С. Б., Алиев А. В., Анищик В. С., Тарнавский А. Г.** Интернет-центр компьютерного моделирования в научных исследованиях и прикладном проектировании // Высокие технологии, фундаментальные и прикладные исследования, образование. СПб.: Изд-во Санкт-Петербургского политехнического университета, 2008. Т. 13. С. 114—119.
2. **Тарнавский Г. А., Тарнавский А. Г., Гилев К. В.** Информационно-вычислительный Интернет-центр "Аэромеханика". Первая линия: программный комплекс "Удар" // Вычислительные методы и программирование. 2005. Т. 6, № 1. С. 27—28.
3. **Жибинов С. Б., Тарнавский Г. А., Тарнавский А. Г., Алиев А. В.** Интернет-центр по распространению научных знаний. Информационно-вычислительный комплекс "Поток" // Труды ИВМиМГ СО РАН. Сер.: Информатика. 2006. С. 181—200.
4. **Тарнавский Г. А.** Ударно-волновые режимы течения на входе в диффузор гиперзвукового прямоточного воздушно-реактивного двигателя: влияние высоты и скорости полета // Теплофизика высоких температур. 2005. Т. 43, № 1. С. 57—70.
5. **Тарнавский Г. А., Алиев А. В., Тарнавский А. Г.** Компьютерное моделирование в аэромеханике: программный комплекс "Поток-5" // Авиакосмическая техника и технология. 2007. № 4. С. 27—38.
6. **Алиев А. В., Тарнавский Г. А.** Иерархический SPH-метод для математического моделирования в гравитационной газовой динамике // Сиб. электронные математические заметки. 2007. Т. 4. С. 420—480.
7. **Тарнавский Г. А., Анищик В. С., Тарнавский А. Г.** Влияние защитных масок при отжиге кремниевой пластины на формирование наноразмерных гетероструктур легирующих примесей фосфора // Нано- и микросистемная техника. 2008. № 3. С. 57—65.
8. **Тарнавский Г. А., Тарнавский А. Г., Алиев А. В., Жибинов С. Б.** Интернет-центр SciShop.ru по распространению научных знаний: статистика посещений и первый опыт функционирования бизнес-портала // Труды ИВМиМГ СО РАН. Сер.: Информатика. 2007. Т. 7. С. 108—115.

К. В. Пшинник, руководитель отдела продаж,
ООО "Мегаплан",
А. С. Козицын, канд. физ.-мат. наук,
ведущий науч. сотр., НИИ механики МГУ,
Е. В. Елкин, аспирант механико-математического факультета МГУ,
e-mail: yolkin.evgeny@gmail.com

Выделение значимой информации из Web-ресурсов

С ростом популярности метасети Интернет растет число Интернет-сайтов и количество информации, которую эти сайты представляют. Однако основной объем текста этих страниц в настоящее время, как правило, составляют информационные блоки, не несущие полезной информации для пользователя. В статье рассмотрен метод автоматического разделения HTML-страниц на незначимую и содержательную части.

Ключевые слова: коллекции документов, автоматическая обработка, значимость, информация, поисковые системы, индексация текстов.

Введение

Современные поисковые системы в сети Интернет все чаще сталкиваются с проблемой выделения значимой информации из общего контента Web-страниц. Среднестатистическая HTML-страница, как правило, содержит малое количество текстовой информации, которая интересует пользователя. Основным объемом кода страниц составляют реклама, навигационное меню, элементы оформления и другие информационные блоки, не несущие полезной информации для пользователя. В связи с этим обстоятельством возникает необходимость проведения исследований, направленных на определение значимости информационных элементов страницы для улучшения результата последующего поиска.

В работе, результаты которой представлены в настоящей публикации, в качестве значимой рассматривается текстовая информация, отличающая данную страницу от других аналогичных страниц того же информационного источника.

Целью настоящей работы является разработка метода автоматического разделения HTML-страниц на незначимую и содержательную части (будем называть ее *значимой* частью).

1. Известные подходы

Большинство существующих методов анализа структуры Web-страниц используют анализ dom-дерева отдельной HTML-страницы [1] или выделение повторяющихся фрагментов страниц с одного сайта [2, 3].

Механизмы выделения повторяющихся фрагментов страниц с одного сайта, как правило, показывают результаты лучшего качества и являются более универсаль-

ными. Недостатком такого подхода является необходимость одновременной обработки всех или значительной части страниц ресурса. Как следствие, реализации таких алгоритмов имеют большую вычислительную сложность. Кроме того, в существующих алгоритмах используются точные критерии оценки значимости информационных блоков, что делает их неустойчивыми к незначительному "шуму".

2. Предлагаемый алгоритм

В отличие от известных методов предлагаемый оригинальный алгоритм основан на вероятностном подходе к определению повторяемости информации и на комплексном анализе значимости текста. Рассматриваемый алгоритм определения значимости информации основан на выделении повторяющихся фрагментов и состоит из следующих четырех этапов:

1) токенизация — разделение файла на минимальные неделимые части HTML-кода — токены. Они будут использоваться для поиска повторяющихся элементов в файлах;

2) кластеризация — разделение множества файлов на группы схожих файлов;

3) определение рейтинга токена внутри кластера;

4) определение степени значимости каждого токена.

Рассмотрим содержание каждого из этих этапов более подробно.

Токенизация

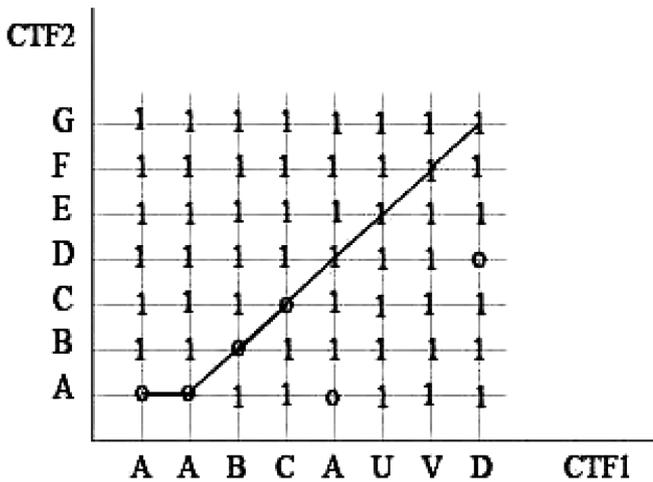
Токенизация — это процесс разбиения HTML-файла на минимальные неделимые смысловые единицы.

Токеном, или минимальной неделимой единицей HTML-файла, будем называть отдельный тег, без символов "<" и ">" или текст между тегами. Скрипты (<script>, <noscript>), комментарии (<!--, -->), стилевые теги (<style>, </style>) не несут значимой информации и исключаются из дальнейшей обработки. После проведения токенизации исходный файл представляется в виде последовательности токенов.

Кластеризация и определение рейтинга токена внутри кластера

Сравнение пары файлов. Для проведения кластеризации и для присваивания рейтинга токенам внутри кластера используется алгоритм сравнения пары файлов, являющийся вариацией алгоритмов динамического программирования, используемых для нахождения максимальной повторяющейся последовательности. Преимуществом выбранного алгоритма является устойчивость к небольшим изменениям файлов.

Рассмотрим рисунок. По горизонтали будем располагать токены одного файла, а по вертикали — токены другого файла. В клетках запишем единицы, если токены строки и столбца не совпадают, и нули, если токены совпадают. Рассмотрим направленный граф, вершинами которого будут соответствовать клетки этого рисунка, обозначаемые индексами соответствующей клетки (n, m) и имеющие вес согласно записям в клетках, а именно — либо нули, либо единицы. Из каждой вершины проведем ребра вправо, вверх и вправо-вверх по диагонали. Задачей алгоритма является нахождение пути, минимального по сумме весов вершин в построенном графе, от точки, соответствующей пересечению первых



символов файлов, до точки, соответствующей пересечению последних символов файлов. Простой вид графа позволяет выполнить поиск за один проход по графу.

Каждой вершине графа соответствует вес $d(n, m)$, зависящий только от токенов, располагающихся на соответствующих местах:

$$d(n, m) = \begin{cases} 0, & \text{если } t_n^1 = t_m^2; \\ 1, & \text{иначе,} \end{cases}$$

где t_n^1 и t_m^2 — токены первого и второго файла соответственно.

Глобальная длина, т. е. длина пути от начальной вершины графа до рассматриваемой вершины, считается итеративно, в предположении, что снизу и справа все длины уже подсчитаны. Формула для подсчета длины:

$$D(n, m) = \min[D(n-1, m), D(n-1, m-1), D(n, m-1)] + d(n, m).$$

Таким образом, $D(n, m)$ — число несовпадений в минимальном по числу несовпадений пути из $(0, 0)$ в (n, m) .

Применяя формулу подсчета длины для каждой вершины графа, получим значение D в конечной точке, соответствующее длине пути по всему графу. Полученное значение назовем *степенью несходства* для двух файлов, и будем использовать его при последующей кластеризации.

Как уже было отмечено, алгоритм сравнения пары файлов необходим на двух разных, перечисленных ранее этапах 2 и 3 работы программы. В зависимости от этапа различаются цели его применения. На этапе 2 (кластеризация) применение этого алгоритма необходимо для определения меры несходства файлов. При этом определяемым параметром является только длина минимального пути.

На этапе 3 (определение рейтинга) алгоритм используется для подсчета числа повторений токенов файлов. В этом случае при проходе клеток рисунка запоминаются значения $p(n, m) \in \{L, U, V\}$, по которым впоследствии будет реконструирован минимальный путь с помощью обратного прохода от конечной точки к начальной:

$$p(n, m) = \begin{cases} L, & \text{если } D(n, m-1) \text{ — минимальная длина;} \\ U, & \text{если } D(n-1, m) \text{ — минимальная длина;} \\ V, & \text{если } D(n-1, m-1) \text{ — минимальная длина.} \end{cases}$$

На первом шаге прохода клеток рисунка путь состоит из одной конечной точки. На каждом последующем ша-

ге в путь включается одна из трех точек, предшествующая текущей крайней точке пути и соответствующая минимуму пути $D(n, m)$. Алгоритм повторяется до тех пор, пока в путь не попадет начальная точка $(0, 0)$. На последнем шаге алгоритм проходит по минимальному пути от точки (n_{\max}, m_{\max}) до точки $(0, 0)$, проверяя для каждой из точек пути, являются ли токены в соответствующих позициях равными в сравниваемых файлах. Если токены равны, то для них увеличивается число повторений, используемое при подсчете рейтинга.

Кластеризация. Задачей кластеризации является разделение исходного множества файлов на кластеры по степени схожести. Такое объединение позволяет выделить файлы, имеющие общую навигационную часть и основу, которую предстоит отфильтровать на следующем шаге.

Перед проведением кластеризации запускается процесс сравнения всех возможных пар файлов из исходного множества. Результатом работы является матрица степеней несходства между файлами. На основе этой матрицы вычисляется матрица расстояний между файлами. Для этого степень несходства каждой пары файлов делится на максимум из длин этих файлов (под длиной файла понимается число его токенов).

Определение. Расстоянием между файлами F_1, F_2 назовем величину, равную

$$m(F_1, F_2) = \frac{c}{\max(|F_1|, |F_2|)},$$

где m — нормированная длина; c — число несовпадений; F_1 и F_2 — списки токенов. Число несовпадений подсчитывается для цепочек токенов по схеме, описанной выше.

Полностью одинаковые файлы будут иметь нормированное расстояние 0, полностью различные файлы — расстояние 1.

На основе построенной матрицы расстояний выполним теоретико-графовую кластеризацию. Для этого представим множество файлов в виде графа $G = G(V, E)$. Вершинам графа (V) будут соответствовать файлы. Сначала составим полный неориентированный граф, проведя ребра от каждой вершины до всех остальных. Назначим ребрам веса w , соответствующие расстоянию между файлами:

$$w((u, v)) = m(F_u, F_v).$$

Минимальным остовным деревом назовем связанный ациклический граф, сумма ребер которого минимальна. Для графа G построим минимальное остовное дерево с помощью алгоритма Крускала [4].

Далее будем проводить кластеризацию итеративно. На каждом шаге удаляем ребро максимальной длины из тех компонент связности минимального остовного дерева, которые не соответствуют критерию кластера, получая все более мелкие компоненты связности. Компоненту связности считаем кластером, если максимальное попарное расстояние между всеми ее вершинами не превышает заданной величины.

Определение степени значимости токенов

Значимость токенов определяется отдельно для каждого кластера, полученного на предыдущем шаге. Для всех файлов кластера выполняется попарное сравнение и определяются повторяющиеся токены.

Рейтингом токена t из файла F назовем число повторений, зафиксированных при попарном сравнении файла F с другими файлами кластера.

Предположение 1. Чем выше рейтинг у токена, тем менее он значим.

Рассмотрим кластер из n файлов. Файлы состоят из навигационной и информативной части. Навигационная часть непостоянна в каждом файле, однако ее изменения незначительны. В таком случае у большинства токенов навигационного меню рейтинг будет близок к $(n - 1)/n$, что при $n > 10$ будет достаточно близко к единице.

Предположение 2. Если тип токена — текстовый, то чем больше длина текста, тем токен более значимый.

Рассмотрим файл, содержащий навигационное меню и значимую информацию. Пункты навигационного меню обычно (при нормальном дизайне сайта) не превышают некоторой длины. В то же время текстовая информация, которая для нас является значимой, имеет значительно больший размер.

Предположение 3. Если тип токена — текстовый и текст разбивается на предложения, то токен более значимый.

В работе [5] рассмотрены методы анализа предложений со знаками препинаний, применимые к извлечению значимой информации. Общая информационная значимость токена определяется по формуле, объединяющей нормированную числовую информацию, подсчитанную на основе трех отмеченных выше предположений.

Формула имеет следующий вид:

$$p(t) = \lambda_1 T_r(t) + \lambda_2 T_L(t) + \lambda_3 T_p(t),$$

где $p(t)$ — вероятность значимости; $T_r(t)$ — слагаемое рейтинга; $T_L(t)$ — слагаемое длины; $T_p(t)$ — слагаемое пунктуации. При этом $\lambda_1 + \lambda_2 + \lambda_3 = 1$, где λ_i — коэффициенты значимости.

Если заранее известен информационный ресурс, на котором предполагается применять алгоритм, то можно подобрать коэффициенты, добившись максимальной его точности. Для неизвестных информационных ресурсов возможно применение универсальных коэффициентов.

Слагаемое рейтинга отражает информацию о частоте повторений токена в файлах кластера. Для приведения этой информации к нормированному виду разделим рейтинг на число файлов в кластере:

$$T_r(t) = \frac{|K| - r(t)}{|K|},$$

где $r(t)$ — рейтинг токена; $|K|$ — число документов в кластере.

Определим *слагаемое длины*. Длина текста $L(t)$ равна числу токенов, из которых состоит текст. Область значений функции длины текста $L(t)$ лежит в интервале $[0, +\infty]$. Для нормировки длины рассмотрим функцию вида

$$T_L(t) = 1 - \frac{1}{\eta L(t) + 1}, \quad \eta \in [0, 1],$$

где η — параметр, характеризующий вероятную значимость токена от длины текста, подбирается экспериментально по пороговым значениям L и T . Если требуется, чтобы слагаемое для текста длины L превышало некоторое эмпирическое пороговое значение T , то параметр η вычисляется по формуле

$$\eta = \frac{T}{L(1 - T)}.$$

Информация *слагаемого пунктуации* имеет бинарный вид. Если текстовый токен содержит предложение, то это слагаемое будет равно 1 и внесет в общую сумму вклад, равный своему весовому коэффициенту. Предложением считается текст, содержащий символ ".". Однако, если символ "." следует после одиночно стоящей буквы, то такая последовательность не считается концом предложения.

Значимой информацией признаются текстовые токены, вероятность значимости $P(t)$ которых превышает пороговое значение.

3. Результаты тестирования

Тестирование работы программы проводилось на трех информационных ресурсах: www.rbc.ru, www.lenta.ru и www.kvartirazamkad.ru со следующими универсальными параметрами:

- диагональ кластеризации 0,4;
- пороговое значение формулы $P(t)$ 0,42;
- коэффициент значимости рейтинга 0,4;
- коэффициент значимости длины текста 0,3;
- коэффициент значимости пунктуации 0,3;
- значимость для текста длиной 100 символов . . . 0,4.

На всех выбранных для теста порталах страницы генерируются с помощью средств динамической генерации HTML-страницы. На страницах присутствуют меню навигации, ссылки на ТОП-новости, статьи, множество рекламных баннеров и ссылок. Так как сложность алгоритма очень большая, проводить испытания на всех страницах портала не представлялось возможным и были сформированы представительные выборки. Тестирование выполнялось на персональном компьютере с процессором Intel Pentium 2,4 ГГц и 1024 Мбайт RAM.

В табл. 1 приведены результаты работы на каждом из ресурсов и дана оценка эффективности алгоритма. Рас-

Таблица 1

Результаты работы алгоритма

Площадка	www.rbc.ru	www.lenta.ru	www.kvartirazamkad.ru
Число файлов	103	50	270
Исходный размер файлов	9,666 Мбайт	1,616 Мбайт	14,463 Мбайт
Время работы алгоритма	18 мин	1 мин 30 с	2 ч 46 мин
Число кластеров	2	6	12
Суммарный размер файлов после токенизации	6,366 Мбайт	1,433 Мбайт	12,976 Мбайт
Суммарный размер текстовых токенов	683,62 Кбайт	308,06 Кбайт	1,485 Мбайт
Суммарный размер файлов после прохождения алгоритма	253,78 Кбайт	265,35 Кбайт	434 Кбайт

смотренные информационные порталы имеют схожую структуру организации данных, а именно:

- индексные страницы;
- страницы с описанием разделов портала;
- информационные страницы.

На информационных страницах и страницах с описанием разделов портала алгоритм показал хорошие результаты.

На порталах www.rbc.ru и www.lenta.ru алгоритм в качестве значимой информации определил содержимое тега <title>, заголовков новости, полный текст новости и уникальные ссылки ("ссылки по теме"), находящиеся после новости.

На портале www.kvartirazamakd.ru алгоритм посчитал значимой информацией значение тега <title> и полное описание объекта.

При этом всю рекламную и навигационную информацию алгоритм квалифицировал как незначимую.

На индексных страницах значимой информацией алгоритм определил заголовки и краткое описание новостей.

Устойчивость алгоритма к модельным возмущениям

Одним из основных критериев оценки алгоритмов обработки больших массивов данных в сети Интернет является устойчивость к небольшим изменениям данных. В целях оценки устойчивости предлагаемого алгоритма в исходные файлы перед обработкой вносились небольшие случайные изменения. Во время токенизации, после добавления каждого нового токена, с вероятностью P процентов добавлялся тег, произвольно выбираемый из списка: "<td>", "<tr>", "<p>", "<a>", "<h1>", "<h2>", "<h3>", "<h4>", "
". Эти теги являются наиболее часто встречающимися в HTML-языке. Добавление проводилось в случайных местах файлов. Изменения подвергаются все исходные файлы. Рассмотрим относительное изменение результатов по сравнению с результатами для неизменных файлов.

Тестирование проводилось на 38 случайно выбранных файлах, взятых с проекта www.kvartirazamakd.ru.

В исходные файлы добавлялось 5 %, 20 % и 99 % "шума" (модельных возмущений). После внесения "шума" размер файлов увеличился соответственно на 5 %, 20 % и 99 % относительно исходного размера (табл. 2).

При 5 % "шума" расстояние между файлами изменилось и, как следствие, произошло некоторое перераспределение файлов по кластерам. Один из файлов был добавлен в большой кластер, другой, наоборот, был оттуда исключен. Появилась разница в размерах получившихся данных, так как при наличии одного файла в кластере

Таблица 2

Отличие результатов работы программы от результатов работы с неизменными файлами

Добавляемый шум, %	0	5	20	99
Размер файла, число кластеров	4	4	8	38
Результат работы программы, байт	74 029	74 048	79 640	179 510
Отличие результатов, %	0	0,03	7,58	142,49

все текстовые токены получают нормированный рейтинг 1, что при заданных параметрах гарантирует значимость этого токена.

При 20 % "шума" разница также проявляется, как правило, в файлах, которые образуют единичные кластеры.

Анализ результатов тестирования на устойчивость позволяет сделать выводы о том, что добавление 5 % и даже 20 % случайных тегов практически не отражается на результате. В случае добавления случайных текстовых элементов алгоритм распознает их как значимый текст.

Заключение

В настоящее время все поисковые системы пытаются улучшить релевантность результатов поиска запросам пользователя. Алгоритм, рассмотренный в данной работе, позволяет при поиске учитывать значимость информации на странице, улучшая тем самым релевантность получаемых результатов. После обработки алгоритма страницы портала содержат более значимую информацию, чем после простого удаления тегов. Следует отметить, что алгоритм показал также хорошие результаты на обнаружении и удалении оптимизационных "спам-текстов", находящихся на страницах портала.

Основным отличием предлагаемого алгоритма от известных и традиционно используемых является вероятностный подход к анализу повторяющихся фрагментов, а также комплексный подход к подсчету значимости текста.

Анализ устойчивости алгоритма к модельным возмущениям показал, что при добавлении 5 % шумов результат изменился всего на 0,03 %.

Вместе с тем, существенным минусом описанного алгоритма является большая сложность $O(n^2)$, где n — число обрабатываемых файлов. По этой причине запускать алгоритм на больших порталах целиком, где число страниц более 100 тыс., нецелесообразно. В связи с этим обстоятельством перспективным представляется создание языка описания шаблонов значимости для информационных кластеров, позволяющего на основе небольшой выборки сформировать описания значимости информации в файлах кластера для быстрой последующей обработки больших массивов данных.

Список литературы

1. **Chakrabarti S.** Integrating the Document Object Model with Hyperlinks for Enhanced Topic Distillation and Information Extraction // In Proceedings of WWW10, May 1–5. Hong Kong. 2001. <http://www10.org/cdrom/papers/489/>
2. **Агеев М. С., Вершинников И. В., Добров Б. В.** Извлечение значимой информации из web-страниц для задач информационного поиска. http://company.yandex.ru/grant/2005/07_Ageev_102942.pdf
3. **Некрестьянов И., Павлова Е.** Обнаружение структурного подобию HTML-документов // Труды Четвертой всероссийской конференции RCDL'2002, 38–54, Дубна, Россия, 2002. <http://meta.math.spbu.ru/~igor/papers/rcdl02-structure/rcdl02-structure.html>
4. http://en.wikipedia.org/wiki/Text_mining
5. **Кузнецов Р. Ф.** Извлечение значимой информации из web-страниц с использованием предложений. http://www.rcdl.ru/papers/2006/paper_61_vl.pdf

CONTENTS

Avdoshin S. M., Savelieva A. A. <i>Efficiency Evaluation of Cryptographic Protection of Information in Enterprise Applications</i>	2
---	---

We introduce a complex approach to evaluating cryptographic protection efficiency. Classically, the research has mostly focused on information system security as a whole, whereas cryptographic tools evaluation techniques have not received as much attention. The main thread of our work is the development of mathematical models of threats to analyze the security of cryptographic systems based on various types of attacks that the cryptographic system is exposed to. The approach allows to build an economic rationale for investments to cryptographic systems and to provide sound arguments for implementing an information security strategy.

Keywords: threat modeling, cryptographic system, risk management, discounted cash flow.

Pereguda A. I., Timashov D. A. <i>A Reliability Model for Information Security Systems</i>	10
---	----

In this paper a new information security systems reliability model has been proposed, assuming the security system has a complex structure. The model can be used to calculate such reliability indices as mean time to system failure and CDF of the time to system failure.

Keywords: reliability model, information security, system structure.

Akhremchik O. L. <i>Information Base for Computer Aided Design of Electric Diagrams of Technological Object Control Systems</i>	17
--	----

The creation model of electric diagrams of technological object control system is considered. The control system hierarchical decomposition are presented as teoretico-plural descriptions of subsystems, contours, chains, elements. The signals of element input and output are allocated as properties of all system levels for automatic synthesis of the principal electric diagrams.

Keywords: control system, properties, structure, element.

Shumarin S. V., Talicky E. N. <i>Optimization Vibroprotection Electronic Equipment by Method Frequency Offset</i>	21
--	----

Possible ways vibroprotection electronic equipment by a method frequency offset are analyzed. The algorithm of optimization mass and overall equipment parameters are offered at frequency offset.

Keywords: electronic equipment, vibroprotection, frequency offset, algorithm.

Karpenko A. P., Ovchinnikov V. A., Semenikhin A. S. <i>Program System Pradis//Front for Construction of Pareto-Set in the Multiple Criteria Optimization Problem of Dynamic Systems Based on a Parallel Genetic Algorithm</i>	27
--	----

The paper discusses distributed program system PRADIS//FRONT for approximate construction of Pareto-set in the multiple criteria optimization problem of dynamic systems based on a parallel genetic algorithm. We use program package PRADIS to model of investigated dynamic system.

Keywords: multiple criteria optimization problem, Pareto-set, Pareto-front, dynamic system, parallel genetic algorithm, program package PRADIS.

Kazakov P. V. <i>The Cluster Based Extension of Genetic Algorithm for Multimodal Optimization Problems Solving</i>	33
---	----

The cluster based extension of standard genetic algorithm is presented in the paper. This version of genetic algorithm increases of a standard genetic algorithm functionality when the multimodal optimization problems are solved. The results of the experiments with this modification of a standard genetic algorithm are presented.

Keywords: genetic algorithm, cluster, multimodal optimization problems, clusterization of population, population diversity.

Antonov V. A. *Information Technology of Optimization by the New Numerical Method and Construction on its Basis Polynoms of Sedate Trends with Selfdefined Exponents* 39

The essence of a new numerical method approach a top pasrabolicheskoy (МППБ), intended for a finding extremism is stated the set function. The methodology and information technology of construction on casually distributed znachenijam dependent size polynoms sedate trends with samoopredelajushchimisja exponents is considered developed on the basis of МППБ. Classification of trends of take the type is offered. The system methodical and processing method of their reception is presented. Practical examples of construction of the specified trends are resulted.

Keywords: an extremum, optimization, a numerical method, approximation function, information technology, regress, a trend, coefficient determinations.

Potapov A. P., Petrov I. B. *Modeling of High Speed Impacts with Smoothed Particle Hydrodynamic Method* 46

Concerned issues in the given article are related to the modeling of high speed impact with different variants of the smoothed particle hydrodynamics method and applying this method to the solution of problems of mechanics of the deformed solid body. Results of solution of the problem of disintegration of discontinuity were obtained and comparative analysis was performed.

Keywords: SPH, elastic-plastic, discontinuity, 3D algorithm.

Volovikov V. V., Uvaysov S. U. *Heat Mass Transfer Model in Radio Electronic Devises* 50

There are results of heat mass transfer model represented as an equivalent circuit in the article. It determines volume-averaged and output temperatures of heat transfer fluid. It uses as a topological heat model component.

Keywords: heat characteristic, mathematic simulation, appliance, heat and mass transfer, topological models.

Shvetsov A. N., Sergushicheva M. A. *Multi-Agent Application Systems' Design Via the DISIT Package*. 54

The paper considers intellectual multi-agent systems' design issues. It defines the design principles and architecture of the DISIT program package tool. The application of the package is shown by example of the maintenance and repair organization system at the geographically distributed power supply enterprise.

Keywords: multi-agent systems, subject domain models, maintenance and repair organization, agent structure.

Arzamastsev A. A., Zenkova N. A., Neudahin A. V. *Technology of Medical Expert System Building on the Basis of the Device of Artificial Neural Networks* 60

The article is about automatical technology of expert systems building developed on the basis device of artificial neural networks. Realization of this technology is considered on an example of construction of medical expert system.

Keywords: the automatical technology, medical expert system, artificial neural networks, intellectual core, client-server technology of access.

Pekunov V. V. *The Procedures with Plan of Reentering in High-Level Languages Use in Traditional and Parallel Programming* 63

The new approach to programming of the algorithms reduced to generation of a series of similar subtasks with planning of sequence of their decision according to strategy of queue, stack or deck is offered. The approach proposes a new formalism to include in high-level programming languages: procedures with plan of reentering. Syntax and semantics of such procedures are described. Parallelization strategy of task decision on shared memory multiprocessor/multicore systems is offered.

Keywords: algorithms, queue, stack, deck, procedure with plan of reentering, parallel programming, generator.

Tarnavsky G. A., Aliev A. V., Anishchik V. S., Tarnavsky A. G., Zhibinov S. B., Chesnokov S. S.
Informational Technologies and Problems of Creation of Computer Simulation Center in Internet 68

The technological, economical, and legal problems of creation of special Web-resource in Internet, Center of computer simulation, are analyzed.

Keywords: informational technologies, Internet, distance computer simulation, scientific knowledge distribution.

Pshinik K. V., Kozitsyn A. S., Elkin E. V. *Allocation of the Significant Information from Web-Resources* 74

With growth of popularity of the Internet, the quantity of sites and the quantity of the information on these sites are growing. However, the basic part of text is the information blocks are not the helpful information for the user. This work describes one method of automatic allocation significant information.

Keywords: collections of documents, automatic processing, signification, information, search engine, text indexation.

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *О. А. Ефремова*.
Корректор *Е. В. Комиссарова*

Сдано в набор 04.06.2009. Подписано в печать 14.07.2009. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 9,8. Уч.-изд. л. 10,81. Заказ 602. 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