

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

10(182)
2011

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Казаков П. В. Генетические алгоритмы многокритериальной оптимизации. Обзор. . . 2
Елтаренко Е. А. Операторы мягкой логики в многокритериальных задачах. 8
Норенков И. П. Редукция графа работ в задаче синтеза расписаний. 18
Стрижов В. В., Крымова Е. А. Выбор моделей в линейном регрессионном анализе. . 21

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

- Бобков С. Г. Архитектурные и конструктивные решения вычислительных систем с производительностью 1—100 Тфлопс. 26
Гливенко Е. В., Прядко С. А., Фомочкина А. С., Левонян К. А. Методы решения систем линейных и нелинейных алгебраических уравнений, обладающие естественным параллелизмом. 31
Андреев Д. В. Аналоговые мультиплексоры для логической обработки континуальных данных. 35

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

- Берников А. Р., Графов Р. П., Лысенко С. Н., Савенко О. С. Поиск вредоносных программ в распределенных тренажерах с использованием технологии нечеткой логики. 42

УПРАВЛЕНИЕ РИСКАМИ

- Борисов В. В., Сеньков А. В. Интеллектуальное управление рисками в сложных организационно-технических системах. 47

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

- Зак Ю. А. Система вероятностных критериев и ограничений в задачах выбора оптимального портфеля инвестиций. 52

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Беневоленский С. Б., Кириллов Ю. И., Колосянко Д. М., Пожар В. Э., Пожар К. В., Пустовойт В. И. Совершенствование программного обеспечения для обработки данных, получаемых с акустооптических спектрометров. 60
Трахтенгерц Э. А. Информационные технологии коррекции целей и стратегических решений в динамике манипулирования общественным мнением. 64
Малафеев С. С., Черняев А. В. Информационная система визуализации работы экскаватора. 70

ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

- Прохоров А. И., Туманов В. Е., Психа Б. Л., Поликарпов К. И. Информационно-аналитическая система учета результатов интеллектуальной деятельности в научной организации. 72
Морылев Р. И. Инструмент определения распараллеливаемых циклов. 76
Contents 78
Приложение. Майданович О. В., Охтилев М. Ю., Соколов Б. В., Юсупов Р. М. Комплексная автоматизация мониторинга состояния космических средств на основе интеллектуальных информационных технологий

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

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

Редакция:

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

Информация о журнале доступна по сети Internet по адресу <http://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.

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

Генетические алгоритмы многокритериальной оптимизации. Обзор

Рассматриваются возможности и перспективы решения задач многокритериальной оптимизации с помощью специального класса генетических алгоритмов. Отмечаются история развития и особенности таких алгоритмов.

Ключевые слова: многокритериальная оптимизация, принципы Парето, граница Парето, многокритериальные генетические алгоритмы

Введение

Многие оптимизационные задачи являются многокритериальными и предполагают поиск решения сразу по множеству критериев, как правило, конфликтующих и несоизмеримых. Поэтому многокритериальная оптимизация (МО) требует принципиально других математических и алгоритмических средств для поиска решений, чем при однокритериальной оптимизации. В основе реализации таких средств лежат так называемые принципы Парето [1, 2], с помощью которых выполняется оценка качества решений одновременно по всем критериям. Результатом решения задачи многокритериальной оптимизации является не одно, а множество (теоретически бесконечное) Парето-оптимальных решений, точное определение которых в настоящее время остается нерешенной NP-проблемой.

Применяемые, как правило, традиционные методы решения задачи многокритериальной оптимизации [1, 3, 4] связаны либо с преобразованием ее в однокритериальную (ради возможности использования хорошо изученных соответствующих численных методов), либо с непосредственным участием исследователя (ЛПР) в оценке решений, если пространство поиска дискретно и ограничено. В первом случае не используются принципы Парето при поиске решений, во втором — существенно снижается уровень автоматизации их определения. В итоге возможностей традиционных методов МО часто оказывается недостаточно при решении сложных

задач многокритериальной оптимизации. Они характеризуются общей постановкой (любое число и вид переменных, критериев оптимизации, ограничений, а также неизвестные топологии пространств решений и критериев) и часто являются основой математических моделей многих комплексных прикладных задач.

На фоне ограниченности традиционных численных методов популярными становятся альтернативные биоэвристические оптимизационные методы научного направления "Искусственный интеллект", прежде всего — многоточечные или так называемые популяционные методы: ройной оптимизации [5], эволюционные алгоритмы [6—8]. Последние благодаря своим принципиальным особенностям [6—8], а также независимости от вида оптимизируемых функций, возможности определения множества оптимальных решений за один запуск считаются естественно предназначенными для решения задач МО [9]. Развитием и исследованием соответствующих эволюционных алгоритмов занимается отдельное постоянно развивающееся направление эволюционных вычислений — MOEA (*Multi-Objective Evolutionary Algorithms*) [9, 10]. К настоящему времени в русскоязычных публикациях отсутствует систематизированная информация о нем. Поэтому в данной статье предпринята попытка обобщить сведения об основных методах этого направления — многокритериальных генетических алгоритмах (МГА).

1. Постановка задачи многокритериальной оптимизации

В общем случае задача многокритериальной оптимизации [1] включает n оптимизируемых переменных, m критериев оптимальности и k ограничений. Математически постановка задачи МО может быть представлена следующим образом:

$$Y = F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \rightarrow \min;$$

$$\varphi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_k(x)); x = (x_1, x_2, \dots, x_n) \in X;$$

$$y = (y_1, y_2, \dots, y_m) \in Y; \{f_1, f_2, \dots, f_m\} \in F.$$

Множество X здесь обозначает пространство решений, имеющее произвольную форму непрерывную, дискретную, смешанную; Y — пространство критериев; F — обобщенный векторный критерий.

При выполнении соответствующих ограничений данные пространства сужаются, образуя множества $X_\varphi = \{x \in X | \varphi(x) \leq 0\}$ и $Y_\varphi = F(X_\varphi) = \bigcup_{x \in X_\varphi} \{f(x)\}$ (рис. 1).

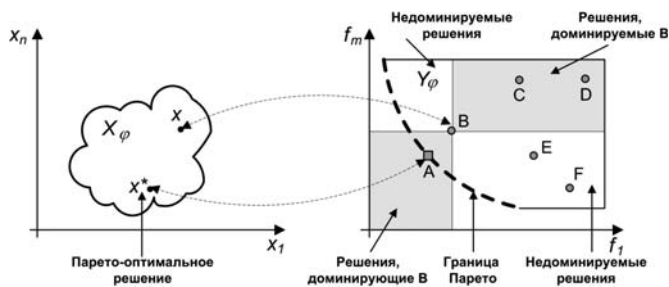


Рис. 1. Отображение пространств решений и критериев

Целью многокритериальной оптимизации является сужение множества X_ϕ до так называемого множества недоминируемых решений. Для определения таких решений используются принципы Парето-доминирования и Парето-оптимальности.

Принцип Парето-доминирования, $\forall x^{(1)}, x^{(2)} \in X_\phi$:

1. $x^{(1)} > x^{(2)}$ ($x^{(1)}$ строго доминирует $x^{(2)}$), если

$$f_i(x^{(1)}) < f_i(x^{(2)}), i = 1, \dots, m.$$

2. $x^{(1)} \geq x^{(2)}$ ($x^{(1)}$ доминирует $x^{(2)}$), если

$$f_i(x^{(1)}) \leq f_i(x^{(2)}) \wedge \exists i: f_i(x^{(1)}) < f_i(x^{(2)}), i = 1, \dots, m.$$

3. $x^{(1)} < > x^{(2)}$ ($x^{(1)}$ не доминирует $x^{(2)}$), если

$$(f_i(x^{(1)}) \not\leq f_i(x^{(2)}) \wedge f_i(x^{(2)}) \not\leq f_i(x^{(1)})), i = 1, \dots, m.$$

На рис. 1 решение В строго доминирует решения С, D, но само доминирует решением А. По отношению к другим изображенным точкам В является недоминируемым, несравнимым.

Принцип Парето-оптимальности состоит в том, что решение задачи МО есть множество недоминируемых по отношению ко всем остальным решений. Точка x^* называется оптимальной по Парето (эффективной, не улучшаемой), если $\nexists x \in X_\phi: x \geq x^*$. На рис. 1 эти точки изображены квадратами. Множество эффективных точек образует множество Парето $X_P = \{x^* \in X_\phi \mid \nexists x \in X_\phi: x \geq x^*\}$, которое в общем случае может включать любое число решений. Соответственно, в пространстве критериев множество решений из X_P образует так называемую компромиссную кривую или границу Парето $Y_P = f(X_P) = \bigcup_{x \in X_P} \{f(x)\}$.

Необходимость одновременного исследования в процессе оптимизации пространства решений и критериев порождает дополнительные сложности для применения любых численных методов:

- практически невозможно оценить взаимосвязи между пространствами решений и критериев;
- отсутствие информации о числе и возможном распределении оптимальных решений в пространстве переменных и критериев;
- трудно контролируемое снижение точности поиска и рост вычислительной сложности при увеличении числа критериев, переменных.

2. Подходы к решению задач многокритериальной оптимизации на основе генетических алгоритмов

Эти подходы связаны прежде всего с различными принципами конструирования fitness-функции и реализации процедуры отбора решений. Можно выделить три таких подхода [10, 11]:

- однокритериальная оптимизация на основе конструирования суперкритерия;
- многокритериальная оптимизация без использования принципов Парето;
- многокритериальная оптимизация с использованием принципов Парето.

Однокритериальная оптимизация на основе конструирования суперкритерия основывается на известных способах [1, 4] свертки критериев, назначении каждому из них весового коэффициента важности $w_i \geq 0, i = 1, \dots, m$. В итоге задача МО сводится к однокритериальной:

$$\sum_{i=1}^m w_i f_i(x) \rightarrow \min, \quad \sum_{i=1}^m w_i = 1.$$

Для получения множества оптимальных решений необходим многократный запуск генетического алгоритма с различными значениями весовых коэффициентов. В результате создается эффект одновременной оптимизации разных функций, лучшее значение которых связано с одной точкой на границе Парето при выполнении условий выпуклости и непрерывности пространства критериев, а также их предварительной нормализации.

В целом, при решении задач МО данный подход является необоснованно популярным и ориентирован на сведение многокритериального поиска решений к однокритериальному. Теоретически это упрощает и ускоряет процесс локализации оптимальных решений, однако на практике можно рассчитывать на определение только части Парето-оптимальных решений. При этом оптимальными они оказываются лишь в очень локальном смысле [11].

Многокритериальная оптимизация без использования принципов Парето отличается от предыдущего учетом при оценке качества решений всех критериев независимо друг от друга, но без явного использования принципов Парето. Для этого при отборе каждая хромосома оценивается только по одному критерию, выбранному случайно или по определенным правилам. В итоге на уровне популяции создается эффект полноценного многокритериального поиска. Данный подход был реализован в генетическом алгоритме VEGA (*Vector Evaluated Genetic Algorithm*) [12], который считается исторически первым МГА и отправной точкой направления МОЕА.

Многокритериальная оптимизация с использованием принципов Парето в настоящее время является основным подходом при решении задач многокритериальной оптимизации с помощью генетических алгоритмов. Относительно других он предполагает

полноценную многокритериальную оптимизацию одновременно по всем критериям в соответствии с принципами Парето.

В основе этого подхода лежит формальный принцип Парето-доминирования и его алгоритмическая реализация, предложенная Д. Голдбергом (D. Goldberg) [6]. Данный алгоритм представляет собой итерационную процедуру многоуровневого ранжирования хромосом и заключается в следующем.

Шаг 1. Присвоить начальное значение ранга $R = 1$.

Шаг 2. Используя принцип Парето-доминирования, найти множество $C_{nondom} \subseteq P$ недоминируемых хромосом популяции P размером n_P и назначить им ранг R : $rank(c) = R, c \in C_{nondom} \subseteq P$.

Шаг 3. Удалить (временно) из популяции хромосомы с рангом R :

$$\forall c \in C_{nondom}: P = P - \{c\}, n_P = n_P - 1.$$

Шаг 4. Увеличить значение ранга $R = R + 1, C_{nondom} = \emptyset$.

Шаг 5. Если всем хромосомам назначен ранг ($P = \emptyset$), то закончить, иначе перейти к шагу 2.

Таким образом, все решения популяции оцениваются по принципу Парето-доминирования и получают так называемый ранг недоминируемости (или ранг Парето). Чем ниже его значение, тем меньше решений доминируют данное. Связанные с хромосомами ранги Парето являются основой для конструирования fitness-функции.

Представленная процедура вычисления рангов Парето имеет некоторые модификации, реализованные в ряде оригинальных МГА (рассматриваются далее). Следует заметить, что непосредственное использование принципа Парето-доминирования является основной причиной увеличения временной сложности МГА. В худшем случае (в популяции ни одно из двух решений не доминируют друг друга) такая добавочная сложность является квадратичной [10]

3. Особенности многокритериального поиска решений генетическими алгоритмами

В общем случае качество решения задачи многокритериальной оптимизации генетическими алгоритмами зависит от:

- возможности исследования поискового пространства в направлении к глобально Парето-оптимальным решениям;
- поддержания разнообразия популяции в пространствах решений, критериев в течение всего поиска;
- сохранения Парето-оптимальных решений от их "разрушения" генетическими операторами в череде поколений.

Разработки в МГА, направленные на практическое достижение этих целей, условно можно разделить на следующие группы [10, 11]:

- методы назначения ранга Парето и вычисления приспособленности хромосом (*Pareto ranking and fitness assignment*);
- методы поддержания разнообразия популяции с использованием нишевых механизмов и разделяющей fitness-функции (*Pareto niching and fitness sharing*);
- использование в МГА вторичной популяции (архива хромосом), отобранных по специальным признакам (*Secondary populations and chromosome archive*);
- методы распараллеливания вычислений (*Parallelization of computation*).

Объединение хромосом в ниши позволяет сохранять разнообразие популяции в процессе поиска. Ниши (*niches*) представляют собой группы хромосом, похожие в некоторой степени. При этом число ниш, занимаемых некоторой хромосомой, обратно пропорционально влияет на ее приспособленность. Тем самым преимущество в дальнейшем отборе получают наиболее "оригинальные" хромосомы. Группировка по нишам хромосом выполняется на этапе вычисления их приспособленностей с помощью так называемой разделяющей функции (*sharing function*) $f_{share}(c_i, c_j)$ [11]:

$$f_{share}(c_i, c_j) = \begin{cases} 1 - \left(\frac{d(c_i, c_j)}{\sigma_{share}} \right), & \text{если } d(c_i, c_j) < \sigma_{share}; \\ 0 & \text{иначе,} \end{cases} \quad (1)$$

где $d(c_i, c_j)$ — метрика вычисления меры близости между парой различных хромосом; σ_{share} — радиус ниши, принимающий значения из интервала (0, 1] (для нормализованного поискового пространства); $i, j = 1, \dots, n_P$.

Тогда число ниш, разделяемых хромосомой c_i , и итоговое значение ее приспособленности, соответственно могут быть определены как

$$nc(c_i) = \sum_{j=1}^{n_P} f_{share}(c_i, c_j); f^s(c_i) = f(c_i)nc(c_i), \quad (2)$$

где $f(c_i)$ — исходное значение приспособленности хромосомы (в частности, ее ранг Парето).

Значение метрики $d(c_i, c_j)$ зависит от выбранного пространства для выделения ниш (поиска, решений, критериев). Чаще всего используется пространство критериев [10].

Вторичная популяция (Парето-архив хромосом) в настоящее время считается ключевой составляющей МГА. Основная цель такого архива заключается в хранении и упорядочении найденных на различных этапах работы МГА Парето-оптимальных решений. Необходимость в этом вызвана следующими причинами:

- вычислением рангов Парето хромосом только относительно текущей популяции;

- возможной потерей Парето-оптимальных решений в процессе поиска;
- возможным превышением мощности множества Парето размера популяции.

Также с помощью Парето-архива в МГА реализуется стратегия элитизма, часто применяемая в одно-критериальных генетических алгоритмах. В различных версиях МГА обработка Парето-архива выполняется по-разному и включает в себя: обновление элементов Парето-архива, взаимодействие хромосом Парето-архива и основной популяции, участие хромосом Парето-архива в генетических операторах. Все это достаточно сильно увеличивает добавочную временную сложность МГА, что, в общем, пропорционально увеличению размера Парето-архива. Для управления этим используется ограничение размера Парето-архива и его принудительное сокращение, например, с помощью кластеризации [11].

Распараллеливание вычислений становится для МГА особенно актуальным из-за существенно большей временной сложности, чем у обычных генетических алгоритмов. Применение соответствующих методов [13—15] позволяет увеличить как скорость поиска, так и точность, число Парето-оптимальных решений. Как правило, это следующие методы: запуск одного МГА на различных процессорах (работающих как синхронно, так и асинхронно); распределение популяции МГА среди процессоров на основе моделей "главный—подчиненный", "островная модель" и др.

4. Поколения генетических алгоритмов многокритериальной оптимизации

В настоящее время принято выделять три основных поколения МГА (см. таблицу).

Поколение "0" объединяет генетические алгоритмы, не использующие при оценке решений принципов МО (первые два подхода, см. п. 2). Основными представителями этого поколения являются WBGA [16] и VEGA [12]. В WBGA fitness-функция конструируется на основе свертки критериев, при этом весовые коэффициенты наряду с переменными оптимизации образуют структуру хромосомы и тем самым постоянно изменяются в процессе поиска. В VEGA пропорциональный отбор применяется

к выделенным из основной популяции размером n_{pt} подпопуляциям размером n_p/m . Приспособленность хромосомы каждой такой подпопуляции оценивается в соответствии с j -м критерием, $j = 1, \dots, n_p/m$. Далее все подпопуляции перемешиваются для получения новой популяции исходного размера. Считается, что получаемые таким образом решения являются оптимальными лишь в локальном смысле, так как их оценка выполняется лишь по отдельному случайно выбранному критерию, игнорируя при этом другие. В итоге к окончанию работы генетического алгоритма имеет место лишь частичное заполнение (в крайних участках) итоговой границы Парето. Главное достоинство VEGA заключается в высоком быстродействии (прежде всего из-за отсутствия взаимного сравнения решений по всем критериям) при большом числе оптимизируемых критериев.

Главным недостатком этого поколения является преобладание однокритериальности при поиске решений,

Поколение I отличается от предыдущего использованием его представителями принципов Парето при вычислении fitness-функции. Основными МГА здесь являются Nondominated Sorting Genetic Algorithm (NSGA) [17], Multi-objective Genetic Algorithm (MOGA) [18], Niched Pareto Genetic Algorithm (NPGA) [19].

Nondominated Sorting Genetic Algorithm (NSGA) основывается на правилах оценивания хромосом посредством их группировки по уровню Парето-доминирования с помощью алгоритма Д. Голдберга (D. Goldberg). Эта процедура выполняется перед оператором отбора, в итоге все хромосомы получают ранг Парето (рис. 2, а). Затем отдельно для каждого ранга с помощью разделяющей функции (1), (2) корректируется приспособленность хромосом для их более равномерного распределения вдоль границы Парето.

Multi-Objective Genetic Algorithm (MOGA). В этом алгоритме, как и в NSGA, для оценивания хромосом вычисляется ранг Парето, но по другим правилам. Ранг хромосомы равен числу хромосом, которые ее доминируют. Из рис. 2, б видно, что по этому правилу решения, лежащие на одной границе Парето, имеют разные ранги (например, J, K, L).

Поколения генетических алгоритмов многокритериальной оптимизации

Поколение МГА	Основные представители	Особенности
Поколение "0"	Vector evaluated GA (VEGA. J. Schaffer); Weight-based GA (WBGA. P. Hajela, C. Lin)	Не используют принципы Парето. Отсутствуют средства поддержания разнообразия популяции
Поколение I	Nondominated Sorting Genetic Algorithm (NSGA. N. Srinivas, K. Deb); Multi-objective Genetic Algorithm (MOGA. C. Fonseca, P. Fleming); Niched Pareto Genetic Algorithm (NPGA. L. Horn, N. Nafpliotis, D. Goldberg)	Используют принципы Парето для оценки решений. Применяются концепции ниш (niching), разделяющая функция (fitness-sharing) для поддержания разнообразия популяции
Поколение II	Strength Pareto Evolutionary Algorithm (SPEA. E. Zitzler, L. Thiele); Improved SPEA (SPEA2. E. Zitzler, M. Laumanns, L. Thiele); Fast Nondominated Sorting Genetic Algorithm (NSGA-II. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan)	Наследует из предыдущего поколения. Развита реализация стратегии элитизма: Парето-архив и управление им. Управление плотностью решений вдоль границы Парето (crowding)

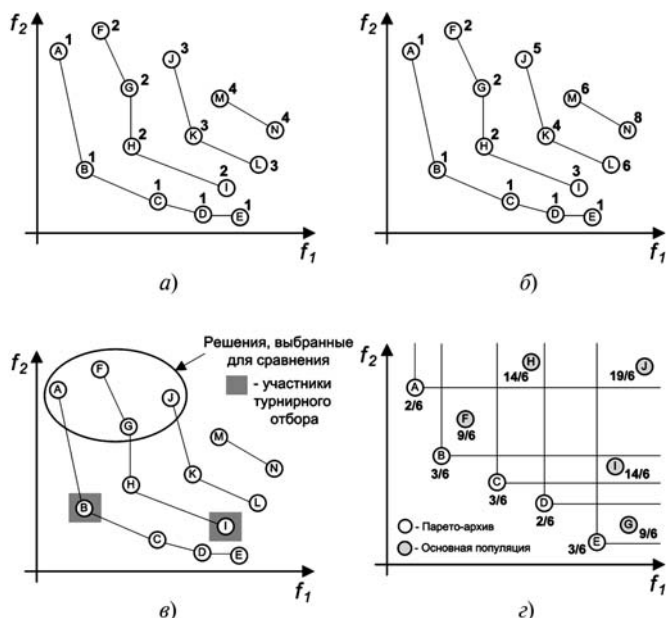


Рис. 2. Основные принципы оценки решений в МГА: а — алгоритм NSGA; б — алгоритм MOGA; в — алгоритм NPGA; г — алгоритм SPEA

В некоторых задачах, с неравномерными границами Парето разных уровней, это может привести к их фрагментациям и, как следствие, — потере ряда Парето-оптимальных решений [10].

Niched-Pareto Genetic Algorithm (NPGA) использует схему турнирного отбора с привлечением принципов Парето-доминирования. Идея алгоритма заключается в следующем. Две случайно выбранные хромосомы c_1 , c_2 сравниваются с подмножеством хромосом (обычно 10% популяции) остальной популяции (рис. 2, в). Если одна из них доминируется хотя бы одной хромосомой из этого множества, а другая нет, то последняя объявляется "победителем" в турнирном отборе. Если же обе хромосомы c_1 , c_2 являются доминируемыми (недоминируемыми), то "победитель" определяется с помощью разделяющей функции (1), (2).

Результаты применения МГА первого поколения показали их высокую эффективность при решении как тестовых задач МО, так и прикладных [9, 11, 20]. Однако отсутствие у этих алгоритмов средств управления качеством Парето-оптимальных решений (получение границы Парето с высокой протяженностью и плотностью решений) определило пути дальнейшего развития МГА.

Поколение II является актуальным в настоящее время и развивает возможности своих предшественников, прежде всего за счет реализации стратегии элитизма посредством следующих способов:

- сохранение "элитных" хромосом в основной популяции;
- хранение и обновление "элитных" хромосом в Парето-архиве, при необходимости интеграция их в основную популяцию.

Под "элитными" в МГА второго поколения следует понимать хромосомы, соответствующие недоминируемым решениям, в общем случае имеющим ранг Парето, равный единице.

Strength Pareto Evolutionary Algorithm (SPEA) [11]. На каждой итерации работы этого алгоритма недоминируемые хромосомы копируются в Парето-архив. Затем для каждого его элемента вычисляется так называемая сила (*strength*) (далее Парето-сила). Ее значение пропорционально числу хромосом из основной популяции, которые доминирует выбранная хромосома из Парето-архива:

$$S(c_i) = \frac{|\{c_j \in P \wedge c_i \succ c_j\}|}{n_P + 1},$$

где $S(c_i) \in [0, 1)$ — Парето-сила хромосомы c_i , $i = 1, \dots, n_A$, из Парето-архива A размером n_A .

Другими особенностями SPEA являются следующие:

- приспособленность хромосом популяции определяется только в зависимости от значений fitness-функции соответствующих хромосом Парето-архива:

$$f(c_j) = 1 + \sum_{c_i \in A, c_i \succ c_j} S(c_i), \forall c_j \in P;$$

- все хромосомы популяции и Парето-архива участвуют в операторе отбора;
- для сокращения размера Парето-архива без потери качества границы Парето используются методы кластеризации.

SPEA в явном виде не использует нишевые механизмы, подобно МГА первого поколения. Для поддержания разнообразия популяции и контроля равномерного распределения решений вдоль границы Парето используется следующий способ. На рис. 2, г показано пространство критериев, содержащее решения из основной популяции (A, B, C, D, E) и Парето-архива (F, G, H, I, J). Над каждым решением указана приспособленность либо Парето-сила соответствующих хромосом. При этом координаты каждого решения разделяют пространство критериев на прямоугольные области. Разные решения могут принадлежать разному числу таких областей — чем их больше, тем больше решений доминируют данное. Например, решение J доминируется всеми остальными решениями. Вместе с этим данные прямоугольные области могут рассматриваться как ниши. Принадлежность им определяется не с позиций некоторой меры близости, а на основе Парето-доминирования. Это исключает необходимость подбора значения управляющего параметра σ_{share} .

Improved SPEA (SPEA2) [21] расширяет возможности своей первой версии, прежде всего за счет непосредственного учета при вычислении приспособленностей хромосом плотности их соседства

относительно друг друга. Для этого используется следующая зависимость:

$$m(c_i) = \frac{1}{(\sigma_{c_i}^{(k)} + 1)},$$

где $m(c_i)$ — плотность соседства хромосом рядом с c_i , $i = 1, \dots, n_P$; $\sigma_{c_i}^{(k)}$ — расстояние в пространстве критериев между хромосомой c_i и ее k -м ближайшим соседом; $k = \sqrt{n_P + n_A}$.

Результирующая приспособленность c_i вычисляется как сумма ее Парето-силы и плотности $m(c_i)$.

В SPEA2 обязательным является полное заполнение до заданного значения N_A Парето-архива в каждом поколении. При этом если размер Парето-архива меньше этого значения, то в него добавляется необходимое число лучших по приспособленности, но доминируемых хромосом из текущей популяции и Парето-архива предыдущего поколения. Иначе — из Парето-архива удаляются хромосомы с постепенно ухудшающимся значением $\sigma_{c_i}^{(k)}$. Соответственно качество и скорость поиска в SPEA2 во многом зависят от параметра N_A .

Fast Nondominated Sorting Genetic Algorithm (NSGA-II) [22] представляет собой усовершенствованную версию NSGA. В NSGA-II для каждой хромосомы c_i определяется как число хромосом, которые доминируют c_i , так и множество хромосом, которых непосредственно сама c_i доминирует. В этом алгоритме в явном виде не используется Парето-архив, поэтому качество поиска поддерживается за счет развитых механизмов управления плотностью размещения решений вдоль границ Парето разных уровней. Для этого применяется специальный количественный показатель, который называется протяженностью "сгущивания" (*crowding distance*) (рис. 3). Например, значение "сгущивания" решения В выше, чем у решения А, потому что $A1 + A2 > B1 + B2$.

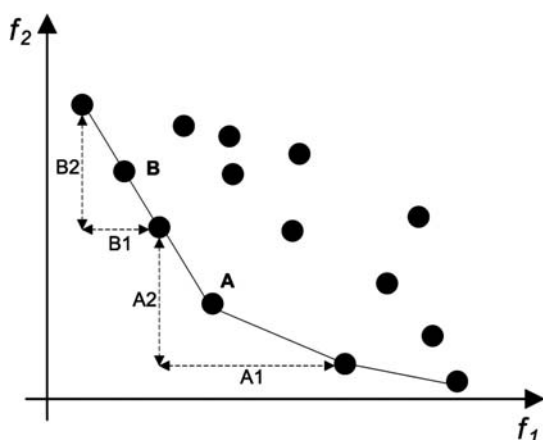


Рис. 3. Определение "сгущенности" хромосом вдоль границы Парето

Значение этого показателя используется в турнирном отборе — при совпадении рангов Парето у сравниваемых хромосом; "победителем" объявляется хромосома с меньшим значением показателя crowding distance.

Важным достоинством этого МГА является отсутствие дополнительных управляющих параметров. В NSGA-II основная популяция является как источником новых решений, так и Парето-архивом. Поэтому ее размер становится определяющим параметром эффективности этого МГА.

Генетические алгоритмы SPEA2, NSGA-II являются наиболее популярными представителями второго поколения как с точки зрения их практического использования при решении различных прикладных задач, так и в качестве обязательных участников тестовых испытаний с другими эволюционными алгоритмами [9]. Например, эволюционными стратегиями, для которых также есть специальные модификации для МО, наиболее известные — PAES (Pareto Archived Evolution Strategy) [23], CMA-ES (Evolution Strategies with Covariance Matrix Adaptation) [24].

Заключение

Многокритериальные генетические алгоритмы представляют собой отдельный класс этих эволюционных методов. Наследуя известные преимущества стандартного генетического алгоритма, МГА благодаря своим особенностям позволяют эффективно, а главное — в естественном виде, решать задачи многокритериальной оптимизации любой сложности. Это подтверждается многочисленными исследованиями МГА как в синтетических тестах, так и при решении многих прикладных инженерных, экономических и научных оптимизационных задач. Актуальное в настоящее время второе поколение МГА определяет как текущие, так и будущие тенденции развития этих биоэвристических оптимизационных методов. В частности, это повышение качества аппроксимации границы Парето для задач с числом критериев, большим двух, интеграция в МГА механизмов самоадаптации значений управляющих параметров, увеличение производительности алгоритмов за счет их реализации на параллельных вычислительных архитектурах и др. Эффективная практическая реализация этого станет определяющим для появления многокритериальных генетических алгоритмов третьего поколения.

Список литературы

1. Соболь И. М., Статников Р. Б. Выбор оптимальных параметров в задачах со многими критериями. 2-е изд., перераб. и доп. М.: Дрофа, 2006. 175 с.
2. Ногин В. Д. Принятие решений в многокритериальной среде: количественный подход. 2-е изд., испр. и доп. М.: Физматлит, 2004. 176 с.
3. Карпенко А. П., Федорук В. Г. Обзор программных систем многокритериальной оптимизации. Отечественные системы // Информационные технологии. 2008. № 1. С. 15—22.

4. **Черноруцкий И. Г.** Методы принятия решений. СПб.: БХВ-Петербург, 2005. 416 с.
5. **Карпенко А. П., Селиверстов Е. Ю.** Глобальная оптимизация методом роя частиц. Обзор // Информационные технологии. 2010. № 2. С. 25—34.
6. **Coldberg D.** Genetic Algorithms in Search, Optimization and Machine Learning. N. Y.: Addison-Wesley Publishing Company, Inc. 199. 412 p.
7. **Аверченков В. И., Казаков П. В.** Эволюционное моделирование и его применение. Брянск: БГТУ, 2009. 200 с.
8. **Гладков Л. А., Курейчик В. В., Курейчик В. М.** Генетические алгоритмы. М.: Физматлит, 2006. 320 с.
9. **Deb K.** Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, 2009. 536 p.
10. **Van Veldhuizen D., Lamont G.** Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art // Evolutionary computation. 2000. № 8(2). P. 125—147.
11. **Zitzler E.** Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph. D thesis, Swiss Federal Institute of Technology Zurich, 1999.
12. **Schaffer J.** Multiple objective optimization with vector evaluated genetic algorithms // Genetic Algorithms and their Applications. Proceedings of the First International Conference on Genetic Algorithms. Lawrence Erlbaum. 1985. P. 93—100.
13. **Карпенко А. П., Уторов А. Н., Федорук В. Г.** MPI-балансер загрузки многопроцессорной вычислительной системы для решения задачи многокритериальной оптимизации // Информационные технологии. 2009. № 6. С. 28—34.
14. **Карпенко А. П., Овчинников В. А., Семенихин А. С.** Программная система PRADIS/FRONT для построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельного генетического алгоритма // Информационные технологии. 2009. № 8. С. 27—33.
15. **Курейчик В. М., Кныш Д. С.** Параллельный генетический алгоритм. Модели и проблемы построения // Интегрированные модели и мягкие вычисления в искусственном интеллекте: Сб. науч. трудов V Международной науч.-практ. конф. М.: Физматлит, 2009. Т. 1. С. 41—52.

16. **Hajela P., Lin C.** Genetic search strategies in multicriterion optimal design // Structural Optimization. 1992. Vol. 4. P. 99—107.
17. **Srinivas N., Deb K.** Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms // Evolutionary Computation. 1994. № 2(3). P. 221—248.
18. **Fonseca C., Fleming P.** Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization // In Proceedings of the Fifth International Conference on Genetic Algorithms. Morgan Kaufman Publishers. 1993. P. 416—423.
19. **Horn J., Nafpliotis N., Goldberg D.** A Niched Pareto Genetic Algorithm for Multiobjective Optimization // In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence. IEEE Service Center. 1994. Vol. 1. P. 82—87.
20. **Казаков П. В.** Возможности генетических алгоритмов для решения задачи многокритериальной оптимизации инвестиционного портфеля // Двенадцатая национальная конференция по искусственному интеллекту КИИ-2010: труды конференции: В 4-х т. М.: Физматлит, 2010. Т. 4. С. 105—111.
21. **Zitzler E., Laumanns M., Thiele L.** SPEA2: Improving the Strength Pareto Evolutionary Algorithm. / K. Giannakoglou et al., editor, EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. 2002. P. 95—100.
22. **Deb K., Pratap A., Agarwal S., Meyarivan T.** A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II // IEEE Transactions on Evolutionary Computation. 2002. 6 (2). P. 182—197.
23. **Knowles J., Corne D.** Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy // Evolutionary Computation. 2000. 8(2). P. 149—172.
24. **Hansen N.** The CMA evolution strategy: a comparing review / Lozano J. A., Larranaga P., Inza I., and Bengoetxea E., ed. // Towards a new evolutionary computation. Advances on estimation of distribution algorithms, Springer, 2006. P. 75—102.

УДК 519.816;004.89

Е. А. ЕлтARENKO, канд. техн. наук, доц.,
Национальный исследовательский
ядерный университет "МИФИ"
e-mail: EAEltarenko@mephi.ru

Операторы мягкой логики в многокритериальных задачах

Выделены классы предпочтений в многокритериальных задачах, названные мягкой логикой. Для них сформулированы аксиомы предпочтений. Разработаны процедуры построения операторов агрегирования для описания таких предпочтений, основанных на генерирующих функциях. Доказаны теоремы, позволяющие определить требования к генерирующим функциям для обеспечения выполнения аксиом мягкой логики. Приведены примеры операторов мягкой логики.

Ключевые слова: многокритериальные задачи, агрегирование критериев, предпочтения, мягкая логика

В работе [1] изложен подход к агрегированию критериев в многокритериальных задачах (МКЗ), основанный на аппроксимации предпочтений ли-

ца, принимающего решения (ЛПР), операторами агрегирования. В этой же работе сформулированы аксиомы предпочтений, при выполнении которых подход может использоваться. Вместе с тем, можно выделить дополнительные свойства предпочтений, которые встречаются на практике. Рассмотрим одно из таких свойств.

Часто ЛПР выбирает наиболее предпочтительный объект (вариант), руководствуясь правилом: объект не должен иметь ни одной плохой характеристики (низкого значения критерия). Неслучайно при выборе товаров по критерию "цена—качество" обычно выбирают среднюю цену и среднее качество, варианты с высокой ценой и средним качеством отвергаются, а с низкой ценой и средним или высоким качеством вызывают недоверие у покупателя. В качестве примера рассмотрим задачу выбора квартиры при покупке. Цена 1 кв. м на рынке квартир отражает предпочтения покупателей. Комплексными критериями оценки квартир являются: местоположение дома, качество дома, расположение квартиры в доме, качество квартиры. Низкое значение одного критерия резко уменьшает цену, причем это уменьшение нельзя компенсировать другими критериями. Так, если дом находится в "плохом"

районе, то цена квартир в нем не может быть высокой (поэтому элитные или бизнес-класса дома не строят в таком районе). Если квартира находится на последнем или первом этаже, то цена падает на 10—15 %. Маленькая площадь кухни не может компенсироваться большой площадью других комнат.

Если критерии переведены в относительные единицы u_j ($j = 1, \dots, m$), то низким значениям критерия u_j^- соответствуют значения, близкие к нулю, а высоким значениям — близкие к единице. Приведенные выше примеры говорят о том, что вариант, для которого значение хотя бы одного из критериев близко к нулю, считается низким по предпочтению. Это можно интерпретировать как операцию конъюнкции с булевыми переменными. Но в работе [1] указано, что для предпочтений P выполняется аксиома единственности объекта с нулевым предпочтением. Приняв $u^- = 0$, выполнение этой аксиомы записываем в виде: $P(u_j^-, \dots, \bar{u}_m^-) = 0$ и для любого $u_j > 0$ выполняется $P(u_j, \bar{u}_j^-) > 0$. Поэтому описанное выше свойство предпочтений назовем *мягкой конъюнкцией* или *квазиконъюнкцией* и сформулируем соответствующую аксиому предпочтений.

Аксиома мягкой конъюнкции. При данных весах (коэффициентах относительной важности)

критериев $V_1, \dots, V_m$ $\left(\sum_{j=1}^m V_j = 1 \right)$ предпочтения

$$P(u_j, \dots, u_m) < \sum_{j=1}^m V_j u_j.$$

Это означает, что уменьшение значения u_j уменьшает предпочтение нелинейно. Аддитивный оператор

$\sum_{j=1}^m V_j u_j$ описывает предпочтения, в которых

уменьшение критерия u_j компенсируется увеличением других критериев. В мягкой конъюнкции такая компенсация невозможна. Значения, близкие к нулю, одних критериев определяют предпочтение объекта, а увеличение других критериев, хотя и увеличивают предпочтения, но незначительно.

Далее будет рассмотрено и свойство мягкой дизъюнкции предпочтений, но оно реже имеет место при выборе вариантов.

Постановка задачи

Изложенный в работе [1] подход к аппроксимации предпочтений ЛПР предусматривает описание предпочтений операторами агрегирования, построение которых основано на генерирующих функциях. Дополнив систему аксиом предпочтений аксиомой мягкой логики, необходимо выполнить следующее:

- определить требования к генерирующим функциям;
- сформулировать условия, при которых оператор является мягкой конъюнкцией/дизъюнкцией.

1. Аксиомы предпочтений

Приведем аксиомы предпочтений, в которых используются веса критериев V_j .

Аксиома 1 (монотонности предпочтений). Предпочтения монотонны по всем критериям: $P(u_j^+, \bar{u}_j) > P(u_j^-, \bar{u}_j)$, где $u_j^+ > u_j^-$, для любого дополнения $\bar{u}_j$.

Аксиома 2. Существует единственный объект с минимальным нулевым предпочтением: $P(u_j^-, \bar{u}_j^-) = 0$, при $u^- = 0$.

Аксиома 3. Существует единственный объект с максимальным единичным предпочтением $P(u_j^+, \bar{u}_j^+) = 1$ при $u^+ = 1$.

Аксиома 4 (важности критериев). Для любых двух критериев с $V_j > V_s$ и любым фиксированным дополнением $\bar{u}'_{j,s}$ условная функция предпочтения

$$P(u_j, u_s^-, \bar{u}'_{j,s}) > P(u_j^-, u_s, \bar{u}'_{j,s}), \text{ где } u_j^- = 0 \text{ и } u_s^- = 0.$$

Из $V_j = V_s$ следует $P(u_j, u_s^-, \bar{u}'_{j,s}) = P(u_j^-, u_s, \bar{u}'_{j,s})$.

Аксиома 5 (логики предпочтений). При заданных весах $V_1, \dots, V_m$ либо предпочтения $P(u_j, \dots, u_m) <$

$\sum_{j=1}^m V_j u_j$ для любых значений критериев, что со-

ответствует мягкой конъюнкции (квазиконъюнкции), либо предпочтения являются мягкой дизъюнкцией (квазидизъюнкцией), т. е. $P(u_j, \dots, u_m) >$

$\sum_{j=1}^m V_j u_j$ для $\forall u_j$.

Случаю $P(u_j, \dots, u_m) = \sum_{j=1}^m V_j u_j$ соответствует аддитивный оператор.

2. Генерация типов операторов мягкой логики

Аксиомы монотонности и важности критериев выполняются в функционале следующего вида:

$$P(u_j, \dots, u_m) = \psi \left(\sum_{j=1}^m f_j(V_j, u_j) \right), \text{ где функции } \psi(x) \text{ и}$$

$f_j(V_j, u)$ (V_j — параметр) монотонно-возрастающие, а для каждой пары функций $f_j(V_j, u)$, $f_s(V_s, u)$ должны выполняться неравенства $f_j(V_j, u) > f_s(V_s, u)$ при $V_j > V_s$ и $f_j(V_j, u) = f_s(V_s, u)$ при $V_j = V_s$ (аксиома важности). Для удобства сравнения функций $\psi(x)$ и $f_j(V_j, u)$, представим функционал как

$$P(u_j, \dots, u_m) = \psi \left(\sum_{j=1}^m \varphi_j^{-1}(V_j, u_j) \right), \quad (1)$$

где $\varphi_j^{-1}(V_j, u)$ — обратная функция от функции $\varphi_j(V_j, x)$.

Для построения $\varphi_j(V_j, x)$ ($j = 1, 2, \dots, m$) будем использовать одну функцию $\varphi(x)$ с параметром V_j , который позволит формировать разные $\varphi_j(V_j, x) = \varphi(V_j, x)$. Данное ограничение на $\varphi_j(V_j, x)$ связано с выполнением аксиомы 5 логики предпочтений. Эти вопросы рассматриваются в п. 4.

В результате получим три типа функционалов, которые в работе [1] определены как операторы агрегирования:

$$h(u_1, \dots, u_m) = \psi \left(\sum_{j=1}^m V_j \varphi^{-1}(u_j) \right); \quad (2)$$

$$g_a(u_1, \dots, u_m) = \psi \left(\sum_{j=1}^m \varphi^{-1}(k V_j u_j) \right); \quad (3)$$

$$g_b(u_1, \dots, u_m) = \psi \left(d \sum_{j=1}^m \varphi^{-1}(V_j u_j) \right), \quad (4)$$

где $\varphi(x)$ и $\psi(x)$ — нормированные генерирующие функции; k, d — масштабные коэффициенты; V_j ($j = 1, 2, \dots, m$) — веса критериев

$$\sum_{j=1}^m V_j = 1. \quad (5)$$

Для построения операторов мягкой логики будем использовать $\varphi(x)$ и $\psi(x)$, монотонно-возрастающие на интервале $x \in [0; 1]$, причем $\varphi(x)$ — выпуклая на этом интервале. Функции $\varphi(x)$ и $\psi(x)$ нормированные, т. е. для выполнения аксиомы 2 принимают значения $\varphi(0) = \psi(0) = 0$, а для аксиомы 3 — значения $\varphi(1) = \psi(1) = 1$.

Операторы типа h и g отличаются способом учета весов. Для выполнения аксиомы 3 проводится масштабирование операторов типа g . В операторе типа g_a (3) масштабный коэффициент k определяется из уравнения

$$\sum_{j=1}^m \varphi^{-1}(k V_j) = 1, \quad (6)$$

а в операторе g_b (4) d определяется из уравнения

$$\sum_{j=1}^m d \cdot \varphi^{-1}(V_j) = 1, \text{ решением которого будет}$$

$$d = \left(\sum_{j=1}^m \varphi^{-1}(V_j) \right)^{-1}. \quad (7)$$

В работе [1] рассмотрены вопросы влияния генерирующих функций на значения оператора. Функция $\varphi(x)$ определяет выпуклость линий безразличия оператора. На основе $\varphi(x)$ определяется функция $F(u_1, \dots, u_m)$: для оператора h функция

$$Fh(u_1, \dots, u_m) = \sum_{j=1}^m V_j \varphi^{-1}(u_j); \text{ для оператора } g_a$$

$$Fg_a(u_1, \dots, u_m) = \sum_{j=1}^m \varphi^{-1}(k V_j u_j); \text{ для } g_b \quad Fg_b(u_1, \dots, u_m) = d \sum_{j=1}^m \varphi^{-1}(V_j u_j).$$

Функция $\psi(x)$ учитывает совместное влияние на значение оператора всех критериев, оператор агрегирования равен $P(u_1, \dots, u_m) = \psi(F(u_1, \dots, u_m))$.

3. Сравнительный анализ типов операторов

Рассматривая g_a как оператор, в котором вес критерия используется в качестве множителя u_j , вместо $\varphi(x)$ введем $\varphi_{ga}(x) = \frac{1}{k} \varphi(x)$, где k — масштабный коэффициент. Тогда оператор (3) будет иметь вид $g_a = \psi \left(\sum_{j=1}^m \varphi_{ga}^{-1}(V_j u_j) \right)$, а оператор (4) вид

$$g_b = \psi \left(\sum_{j=1}^m \varphi_{gb}^{-1}(V_j u_j) \right), \text{ где } \varphi_{gb}(x) = \varphi(x/d).$$

Используя одну генерирующую функцию $\varphi(x)$, для оператора типа g получим две функции — $\varphi_{ga}(x)$ и $\varphi_{gb}(x)$, которые отличаются условиями нормировки $Fg(u_1, \dots, u_m)$.

Функции $\varphi_{ga}(x)$ и $\varphi_{gb}(x)$ принимают значения на интервале $[0; V_{\max}]$. Значит в операторах типа g используется только часть генерирующей функции $\varphi(x)$ (рис. 1).

Если исходная функция выпукла вниз, $\varphi_{gb}(x)$ использует большую часть исходной функции на интервале $[0; V_{\max}]$, а $\varphi_{ga}(x)$ — меньшую часть, только на интервале $[0; k V_{\max}]$ ($k < 1$). В результате получим $\varphi_{gb}(x)$ более выпуклую по сравнению с $\varphi_{ga}(x)$. В случае, когда максимальное значение веса мало (это харак-

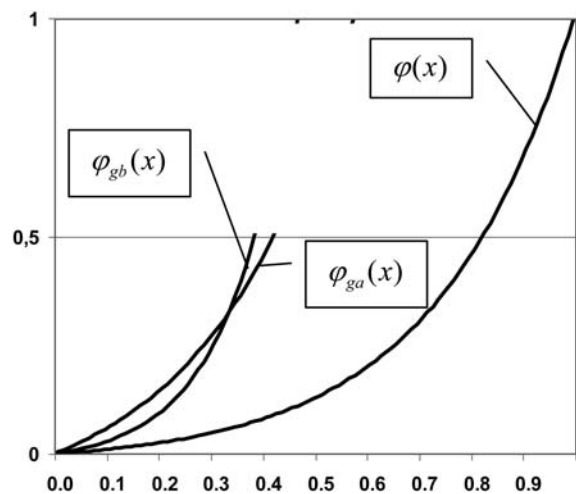


Рис. 1. Генерирующие функции операторов

терно при значительном числе критериев) функция $\varphi_{ga}(x)$ близка к линейной, а сумма $\sum_{j=1}^m \varphi_{ga}^{-1}(V_j u_j)$ близка к $\sum_{j=1}^m V_j u_j$.

Если же исходная функция выпукла вверх, наоборот, $\varphi_{ga}(x)$ использует большую часть исходной функции на интервале $[0; kV_{\max}]$ ($k > 1$), а $\varphi_b(x)$ — меньшую часть на интервале $[0; V_{\max}]$. Получается, что $\varphi_{ga}(x)$ более выпуклая, чем $\varphi_{gb}(x)$. В случае, когда максимальное значение веса мало, уже функция $\varphi_{gb}(x)$ близка к линейной. На рис. 1 приведены исходная показательная генерирующая функция $\varphi(x) = \frac{e^{ax}-1}{e^a-1}$, $\varphi_a(x)$ и $\varphi_b(x)$. При весах $V_1 = 0,2$; $V_2 = 0,3$; $V_3 = 0,5$ масштабные коэффициенты равны $k = 0,18$, $d = 0,47$.

Если рассматривать каждый критерий отдельно, то при изменении u_j в интервале $[0; 1]$ критерий имеет свою функцию $\varphi_j(x)$, которая получается из исходной $\varphi(x)$ и зависит от веса V_j . Ниже приведены $\varphi_j(V_j, x)$ для каждого типа оператора:

$$h: \varphi_{jh}(V_j, x) = \varphi\left(\frac{x}{V_j}\right); \quad (8)$$

$$g_a: \varphi_{jga}(V_j, x) = \frac{\varphi_{ga}(x)}{V_j k} = \frac{1}{V_j k} \varphi(x); \quad (9)$$

$$g_b: \varphi_{jgb}(V_j, x) = \frac{1}{V_j} \varphi\left(\frac{x}{d}\right). \quad (10)$$

Если обе генерирующие функции линейные $\varphi(x) = \psi(x) = x$, то все типы операторов совпадают и равны аддитивному: $P(u_1, \dots, u_m) = \sum_{j=1}^m V_j u_j$.

Определим требования к генерирующим функциям, при которых операторы будут операторами мягкой конъюнкции.

4. Требования к генерирующим функциям для выполнения аксиомы логики предпочтений

Прежде чем переходить к формулировке теорем существования операторов мягкой логики, докажем ряд лемм, которые потребуются в дальнейшем.

Случай 1. Генерирующая функция $\varphi(x)$ выпукла вниз

Лемма 1. Для возрастающей выпуклой вверх функции $f(u)$ и любых коэффициентов $\alpha_j \left(\sum_{j=1}^m \alpha_j = 1 \right)$:

а) выполняется неравенство

$$\sum_{j=1}^m \alpha_j f(u_j) \leq f\left(\sum_{j=1}^m \alpha_j u_j\right); \quad (11)$$

б) максимум $\sum_{j=1}^m \alpha_j f(u_j)$ для любых u_j с $\sum_{j=1}^m \alpha_j u_j = t - \text{const}$ достигается при равенстве всех $u_j = t$.

Доказательство. Неравенство (11) известно как неравенство Йенсена [2]. Оно переходит в равенство при равных $u_j = t$, откуда и следует утверждение (б). Лемма доказана.

Лемма 2. Для возрастающей выпуклой вверх функции $f(u)$ сумма $F(u_1, \dots, u_m) = \sum_{j=1}^m f(\alpha_j u_j)$, в ко-

торой $\sum_{j=1}^m \alpha_j = 1$ и для u_j выполняется условие

$$\sum_{j=1}^m \alpha_j u_j = t - \text{const}, \text{ достигает максимума при } u_j = \frac{t}{m \alpha_j} \quad (j = 1, 2, \dots, m).$$

Доказательство. Представим $F(u_1, \dots, u_m)$ как взвешенную сумму $F(u_1, \dots, u_m) = m \sum_{j=1}^m \frac{1}{m} f(\alpha_j u_j)$. В соответствии с утверждением (а) леммы 1

$$F(u_1, \dots, u_m) = m \sum_{j=1}^m \frac{1}{m} f(\alpha_j u_j) \leq m f\left(\sum_{j=1}^m \frac{1}{m} \alpha_j u_j\right) = m f\left(\frac{1}{m} \sum_{j=1}^m \alpha_j u_j\right) = m f\left(\frac{t}{m}\right) \text{ или } \sum_{j=1}^m f(\alpha_j u_j) \leq m f\left(\frac{t}{m}\right). \quad (12)$$

Неравенство (12) переходит в равенство при $\alpha_j u_j = \frac{t}{m}$, т. е. когда нулевое рассеяние $\alpha_j u_j$. Следовательно, максимум $F(u_1, \dots, u_m)$ достигается, когда $u_j = \frac{t}{m \alpha_j}$. Лемма доказана.

В вышеприведенных леммах на переменные u_j нет ограничений.

$$\text{В операторе } g_a \quad Fg_a = \sum_{j=1}^m \varphi^{-1}(k V_j u_j) = \sum_{j=1}^m \varphi_{ga}^{-1}(V_j u_j),$$

где $\varphi_{ga}^{-1}(u)$ возрастающая выпуклая вверх функция.

В операторе $g_b \quad Fg_b = d \sum_{j=1}^m \varphi^{-1}(V_j u_j)$, где $d > 0$ и $\varphi_{gb}^{-1}(u)$ также возрастающая выпуклая вверх функция.

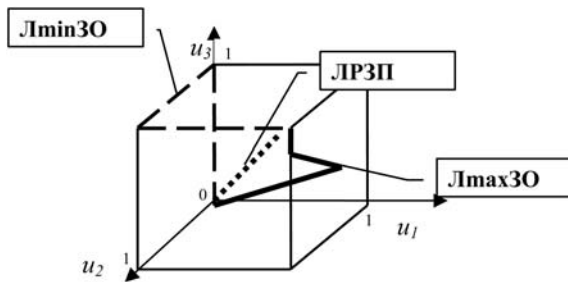


Рис. 2. Линии экстремальных значений оператора

В соответствии с леммой 2 максимальные значения Fg_a и Fg_b достигаются при

$$u_j = \frac{t}{mV_j} \quad (j = 1, \dots, m) \quad t \in [0; 1]. \quad (13)$$

Множество точек (13) есть параметрическое задание прямой в пространстве u_j ($t \in [0; 1]$ — переменный параметр), которую назовем *линией максимальных значений оператора* (ЛмаксЗО). В отличие от лемм на переменные в Fg_a и Fg_b есть ограничения: каждое $u_j \in [0; 1]$. Поэтому рассмотрим процесс построения одномерных функций $Fg_a(t)$ и $Fg_b(t)$ на ЛмаксЗО.

Присвоим индексы критериев в соответствии с порядком возрастания весов $V_1 \geq V_2 \geq \dots \geq V_m$. Если переменная u_1 с минимальным весом V_1 достигает границы $u_1 = \frac{t}{mV_1} = 1$ (рис. 2), то необходимо

двигаться по плоскости с $u_1 = 1$, а остальные переменные принимают одинаковые значения и вычисляются в соответствии с выражением $u_j = \frac{(t - V_1)}{(m - 1)V_j}$

($j = 2, \dots, m$). При достижении следующей переменной границы $u_2 = 1$ остальные переменные определяются по формуле $u_j = \frac{(t - V_1 - V_2)}{(m - 2)V_j}$ ($j = 3, \dots, m$) и т. д.

Функция $Fg_a(t)$ строится следующим образом: на интервале $[0; mV_1]$

$$Fg_a(t) = \sum_{j=1}^m \varphi^{-1}\left(kV_j \frac{t}{mV_j}\right) = m\varphi^{-1}\left(k \frac{t}{m}\right),$$

далее на интервале $(mV_1; (m - 1)V_2 + V_1]$

$$Fg_a(t) = \varphi^{-1}(kV_1) + (m - 1)\varphi^{-1}\left(k \frac{t - V_1}{m - 1}\right),$$

на следующем интервале $((m - 1)V_2 + V_1; (m - 2)V_3 + V_1 + V_2]$

$$Fg_a(t) =$$

$$= \varphi^{-1}(kV_1) + \varphi^{-1}(kV_2) + (m - 2)\varphi^{-1}\left(k \frac{t - V_1 - V_2}{m - 2}\right)$$

и т. д. на последнем интервале $(1 - V_m; 1]$

$$Fg_a(t) = \sum_{j=1}^{m-1} \varphi^{-1}(kV_j) + \varphi^{-1}(k(t - 1 + V_m)).$$

Аналогично строится $Fg_b(t)$.

В операторе h $Fh = \sum_{j=1}^m V_j \varphi^{-1}(u_j)$ функция $\varphi^{-1}(u)$ —

возрастающая и выпукла вверх, поэтому в соответствии с леммой 1 максимального значения Fh достигает при одинаковых значениях переменных:

$$u_j = t \quad (j = 1, \dots, m). \quad (14)$$

Для этого типа оператора ЛмаксЗО равна $Fh(t) = \varphi^{-1}(t)$. Линию (14) в параметрическом виде назовем *линией равных значений переменных* (ЛРЗП) (см. рис. 2).

Приведенные выше $Fh(t)$, $Fg_a(t)$, $Fg_b(t)$ соответствуют максимальным значениям, поэтому введем для них общее обозначение $F_{\max}(t)$. Одномерную функцию $P_{\max}(t) = \psi(F_{\max}(t))$ будем называть *функцией максимальных значений оператора* (ФмаксЗО).

Наряду с максимальными определим минимальные значения $F_{\min}(t)$. Как следует из лемм 1 и 2, при

условии $\sum_{j=1}^m V_j u_j = t$ максимальные значения достигаются при равных значениях переменных $u_j = t/m$

в $Fh = \sum_{j=1}^m V_j \varphi^{-1}(u_j)$ и при равных $V_j u_j = t/m$ в

$Fg = \sum_{j=1}^m \varphi_g^{-1}(V_j u_j)$. То есть максимальное $F(t)$ достигается при нулевом рассеянии слагаемых $V_j u_j$

в $\sum_{j=1}^m V_j u_j = t$, минимальное значение $F(t)$ будет

при максимальном рассеянии $V_j u_j$ в $\sum_{j=1}^m V_j u_j = t$.

Построим в параметрическом виде линию с максимальным рассеянием u_j . Функцию $F(u_1, \dots, u_m)$

запишем в общем виде $F(u_1, \dots, u_m) = \sum_{j=1}^m \varphi_j^{-1}(u_j)$.

При фиксированном t максимальное рассеяние u_j в $\sum_{j=1}^m V_j u_j = t$ будет при одном любом слагаемом $V_j u_j$ равном 1, а остальных слагаемых, равных 0,

т. е. $u_j = t/V_j$. Таким образом, на интервале $t \in [0; V_m]$, где V_m — максимальный вес, $F_{\min}(t) = \varphi_m^{-1}(u_m) = \varphi_m^{-1}(t/V_m)$. При достижении u_m единицы

следующее слагаемое $V_{m-1}u_{m-1}$ в $\sum_{j=1}^m V_j u_j = t$ ($t > V_m$) должно равняться 1, т. е. $u_{m-1} = t - V_m/V_{m-1}$. Значит на интервале $t \in [V_m; V_m + V_{m-1}]$

$$F_{\min}(t) = \varphi_m^{-1}(1) + \varphi_{m-1}^{-1}(t - V_m/V_{m-1}).$$

На последнем интервале $t \in [1 - V_1; 1]$

$$F_{\min}(t) = \sum_{j=2}^m \varphi_j^{-1}(1) + \varphi_1^{-1}((t - 1 + V_1)/V_1).$$

Полученная линия минимальных значений оператора (Лmin3O) приведена на рис. 2.

Для оператора h $\varphi_j^{-1}(u_j) = V_j \varphi^{-1}(u_j)$ и функция $Fh_{\min}(t)$ на интервале j , определяемом

$$t \in \left[\sum_{s=j+1}^m V_s; V_j + \sum_{s=j+1}^m V_s \right], \text{ равна } \sum_{s=j+1}^m V_s + V_j \varphi^{-1} \left(\frac{1}{V_j} \left(t - \sum_{s=j+1}^m V_s \right) \right).$$

$$\text{Для оператора } g_a \quad \varphi_j^{-1}(u_j) = \varphi^{-1}(kV_j u_j), \text{ на том же интервале } Fg_{a\min}(t) = \sum_{s=j+1}^m \varphi^{-1}(kV_s) + \varphi^{-1} \left(k \left(t - \sum_{s=j+1}^m V_s \right) \right).$$

На рис. 3 приведены примеры $Fg_{a\max}(t)$ и $Fg_{a\min}(t)$. При одной генерирующей функции $\varphi(x)$ различие между $Fg_{a\max}(t)$ и $Fg_{a\min}(t)$ зависит от значений весов. При равенстве весов $Fg_{\min}(t)$ в точках перелома касается прямой $y = t$. На рис. 3 приведена также функция $Fg(t)$, определяющая значения $Fg(u_1, \dots, u_m)$ на линии равных значений переменных. Сформулируем ряд утверждений, касающихся выпуклых вниз функций $\varphi(x)$. Они очевидны, поэтому приводятся без доказательств.

Утверждение 1. Для операторов типа g на линии равных значений переменных $Fg(t) > Fg_{\max}(t)$ и $Fg(t) < Fg_{\min}(t)$.

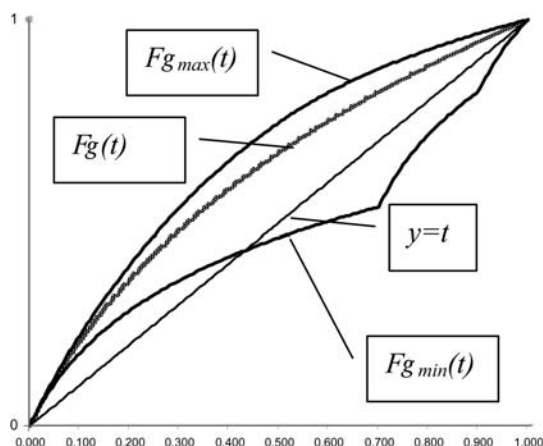


Рис. 3. Экстремальные функции $Fg(t)$ для выпуклой вниз $\varphi(x)$

Утверждение 2. Для операторов типа h на линии равных значений переменных $Fh(t) = Fh_{\max}(t) = \varphi^{-1}(t)$ и $Fh_{\min}(t) \geq t$.

Утверждение 3. При максимальном рассеянии весов, что соответствует $V_1 \rightarrow 1$ и остальные $V_j \rightarrow 0$, для всех типов операторов $F_{\min}(t) \rightarrow F_{\max}(t) \rightarrow \varphi^{-1}(t)$.

Случай 2. Генерирующая функция $\varphi(x)$ выпукла вверх

Для данного случая во всех леммах, рассмотренных ранее, знаки неравенства и min, max меняются на противоположные. Так леммы 1 и 2 формулируются следующим образом.

Лемма 3. Для возрастающей выпуклой вниз функции $f(u)$ и любых коэффициентов $\alpha_j \left(\sum_{j=1}^m \alpha_j = 1 \right)$:

а) выполняется неравенство $\sum_{j=1}^m \alpha_j f(u_j) \geq f \left(\sum_{j=1}^m \alpha_j u_j \right)$;

б) минимум $\sum_{j=1}^m \alpha_j f(u_j)$ для любых u_j с $\sum_{j=1}^m \alpha_j u_j = t - \text{const}$ достигается при равенстве всех $u_j = t$ (нулевое рассеяние u_j), максимум $\sum_{j=1}^m \alpha_j f(u_j)$ соответствует максимальному рассеянию u_j .

Лемма 4. Для возрастающей выпуклой вниз функции $f(u)$ сумма $F(u_1, \dots, u_m) = \sum_{j=1}^m f(\alpha_j u_j)$, в ко-

торой $\sum_{j=1}^m \alpha_j = 1$ и для аргументов u_j выполняется

условие $\sum_{j=1}^m \alpha_j u_j = t - \text{const}$, достигает минимума

при $u_j = \frac{t}{m\alpha_j}$ ($j = 1, 2, \dots, m$).

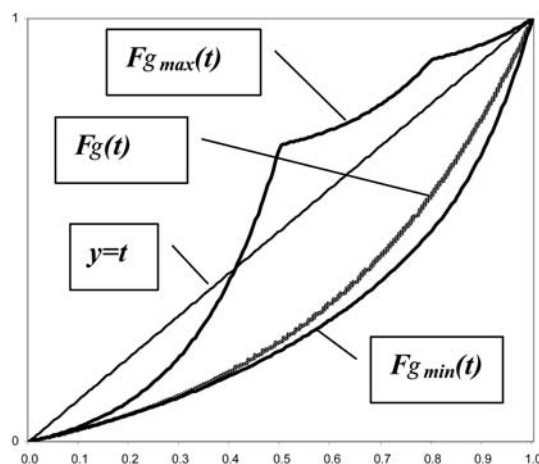


Рис. 4. Экстремальные функции $Fg(t)$ для выпуклой вверх $\varphi(x)$

Максимум $F(u_1, \dots, u_m)$ будет при максимальном рассеянии $\alpha_j u_j$.

Линия максимальных значений оператора (Лmax3O) для выпуклых вверх $\varphi(x)$ определяется по той же процедуре, что Лmin3O для выпуклых вниз $\varphi(x)$, а Лmin3O — по процедуре Лmax3O для выпуклых вниз $\varphi(x)$. На рис. 4 приведены экстремальные функции $F_g(t)$ для выпуклой вверх $\varphi(x)$. Утверждения 1 и 2 заново сформулируем для рассматриваемого случая.

Утверждение 4. Для операторов типа g на линии равных значений переменных $Fg(t) < Fg_{\max}(t)$ и $Fg(t) > Fg_{\min}(t)$.

Утверждение 5. Для операторов типа h на линии равных значений переменных $Fh(t) = Fh_{\min}(t) = \varphi^{-1}(t)$ и $Fh_{\max}(t) \leq t$.

Теоремы существования операторов мягкой логики

Теорема 1. Чтобы оператор был мягкой конъюнкцией, необходимо и достаточно, чтобы на линии максимальных значений операторов $F_{\max}(t)$ была больше $\psi^{-1}(t)$.

Доказательство. Необходимость. Пусть оператор является мягкой конъюнкцией, т. е. выполняется $P(u_j, \dots, u_m) < \sum_{j=1}^m V_j u_j$. Для оператора, построенного на паре генерирующих функций,

$$\psi(F(u_1, \dots, u_m)) < \sum_{j=1}^m V_j u_j. \quad (1)$$

Учитывая, что $\psi^{-1}(u)$ — возрастающая функция, из неравенства (1) получим

$$F(u_j, \dots, u_m) < \psi^{-1}\left(\sum_{j=1}^m V_j u_j\right). \quad (2)$$

Примем сумму $\sum_{j=1}^m V_j u_j$ в выражении (16) за t .

Максимальное $F(u_j, \dots, u_m)$ при условии $\sum_{j=1}^m V_j u_j = t$

есть $F_{\max}(t)$. Из $P(u_j, \dots, u_m) < \sum_{j=1}^m V_j u_j$ следует

$$F_{\max}(t) < \psi^{-1}(t).$$

Достаточность. Возьмем произвольное множество значений переменных $u_1, \dots, u_m$, определим

для него $t = \sum_{j=1}^m V_j u_j \left(\sum_{j=1}^m V_j = 1 \right)$. В соответствии с

леммами 1, 2, 3, 4 максимальное значение $F(u_1, \dots, u_m)$ для данного t находится на линии максимальных значений и равно $F_{\max}(t)$, т. е. $F(u_1, \dots, u_m) \leq F(t)$.

Положительная возрастающая функция $\psi(x)$ не меняет знака неравенства:

$$\psi(F(u_1, \dots, u_m)) \leq \psi(F(t)). \quad (3)$$

По условию теоремы $F(t) < \psi^{-1}(t)$ опять же возрастающая функция не меняет знак: $\psi(F(t)) < t$. С учетом неравенства (17) последнее позволяет записать утверждение теоремы: $P(u_j, \dots, u_m) = \psi(F(u_1, \dots, u_m)) \leq \psi(F(t)) < t = \sum_{j=1}^m V_j u_j$. Теорема

доказана.

Отметим, что условие теоремы 1 можно записать в следующем виде: функция максимальных значений оператора должна быть меньше линейной: $P_{\max}(t) < t$.

Теорема 2. Чтобы оператор был мягкой дизъюнкцией, необходимо и достаточно, чтобы на линии минимальных значений операторов $F_{\min}(t)$ было меньше $\psi^{-1}(t)$.

Доказательство аналогично теореме 1.

От общих теорем перейдем к отдельным типам операторов.

Теорема 3. Для квазиконъюнктивных операторов типа h необходимыми и достаточными условиями являются:

а) генерирующая функция $\varphi(x)$ выпукла вниз и $\varphi(x) > \psi(x)$, или б) функция $\varphi(x)$ выпукла вверх и $\psi(x) < x$.

Доказательство. При выпуклой вниз $\varphi(x)$ для оператора h в соответствии с утверждением 1 $Fh_{\max}(t) = \varphi^{-1}(t)$. По теореме 1 для мягкой конъюнкции необходимо и достаточно $Fh_{\max}(t) < \psi^{-1}(t)$ или $\varphi^{-1}(t) < \psi^{-1}(t)$. Откуда следует $\varphi(x) > \psi(x)$.

Для выпуклой вверх $\varphi(x)$ в соответствии с утверждением 5 $Fh_{\max}(t) \leq t$, примем $Fh_{\max}(t) = t$. По теореме 1 для мягкой конъюнкции необходимо и достаточно $Fh_{\max}(t) < \psi^{-1}(t)$ или $\psi^{-1}(t) > t$. Откуда следует $\psi(x) < x$.

Теорема 4. Для квазидизъюнктивных операторов типа h необходимыми и достаточными условиями являются: а) генерирующая функция выпукла вверх и $\varphi(x) < \psi(x)$, или б) функция $\varphi(x)$ выпукла вниз и $\psi(x) > x$.

Теорема 5. Если генерирующие функции $\varphi(x)$ и $\psi(x)$ одного вида, выпуклы вниз и $\varphi(x) \geq \psi(x)$, то оператор типа g_a является мягкой конъюнкцией при любых значениях весов.

Доказательство. Значение масштабного коэффициента зависит от рассеяния весов V_j . Минимальное значение k принимает при равенстве весов: $k = m\varphi(1/m)$, а максимальное значение $k = 1$ — при максимальном рассеянии. При равенстве весов $Fg_{a\max}(t) = m\varphi^{-1}(t\varphi(1/m))$, что меньше $\varphi^{-1}(t)$, тогда в соответствии с теоремой 1 g_a — оператор мягкой конъюнкции. Максимальное рассеяние соответствует

$k = 1$, в соответствии с утверждением 3 $Fg_{a\max}(t) = \varphi^{-1}(t)$. Следует иметь в виду, что при увеличении k выпуклость $\varphi_{ga}(x)$ увеличивается. Таким образом, при любых значениях весов $Fg_{a\max}(t) \leq \varphi^{-1}(t)$. Из условия теоремы $\varphi(x) \geq \psi(x)$ или $\varphi^{-1}(t) \leq \psi^{-1}(t)$ следует, что $Fg_{a\max}(t) \leq \psi^{-1}(t)$. На основании теоремы 1

делаем вывод, что оператор $g_a(u_1, \dots, u_m) < \sum_{j=1}^m V_j u_j$.

Следует подчеркнуть, что условие $\varphi(x) \geq \psi(x)$ в теореме является достаточным, но не необходимым. Для весов критериев, при которых $k < 1$, оператор g_a будет мягкой конъюнкцией и при неравенстве $\varphi(x) < \psi(x)$, пока в соответствии с теоремой 1 $Fg_{a\max}(t) < \psi^{-1}(t)$. Теорема доказана.

Замечание 1. Для оператора g_b при равенстве весов $d = (m\varphi^{-1}(1/m))^{-1}$, а $Fg_{b\max}(t) = \varphi^{-1}\left(\frac{t}{m}\right) / \varphi^{-1}(1/m) < \varphi^{-1}(t)$, а при максимальном рассеянии $d = 1$, а $Fg_{b\max}(t) = \varphi^{-1}(t)$. Но для оператора g_b $\varphi_{gb}(x) = \frac{1}{V_j} \varphi\left(\frac{x}{d}\right)$ выпуклость увеличивается при уменьшении d и условие $Fg_{b\max}(t) \leq \varphi^{-1}(t) = \psi^{-1}(t)$ может не выполняться. Поэтому для оператора g_b условие $\varphi(x) \geq \psi(x)$ для любых значений весов и любых генерирующих функций не является достаточным.

Теорема 6. Если генерирующие функции $\varphi(x)$ и $\psi(x)$ одного вида, выпуклы вверх и $\varphi(x) \leq \psi(x)$, то оператор типа g_a является мягкой дизъюнкцией при любых значениях весов.

Доказательство аналогично теореме 5.

5. Генерация операторов на основе видов генерирующих функций

В качестве примеров видов генерирующих функций рассмотрим три:

степенную $q_1(x) = x^p$, показательную $q_2(x) = \frac{e^{ax} - 1}{e^a - 1}$ и тангенциальную $q_3(x) = \frac{\operatorname{tg}(ax)}{\operatorname{tg}(a)}$. Для

удобства сравнения разных видов функций в работе [1] введен единый для всех функций параметр λ , характеризующий выпуклость функций. Параметр конкретного вида функции определяется из уравнения $\lambda_q = q(1 - \lambda_q)$. На рис. 5 приведены три указанных вида функций с параметром $\lambda = 0,3$.

Параметр $\lambda < 0,5$ указывает, что функция выпукла вниз. Обратная функция $\hat{q}(x) = q^{-1}(x)$ будет выпуклой вверх, причем для нее параметр выпуклости $\lambda_q = 1 - \lambda_q$.

Из выпуклой вниз $q(x)$ можно получить еще одну выпуклую вниз функцию отображением $q(x)$ относительно оси: $y = 1 - x$. Эта функция имеет вид $\tilde{q}(x) = 1 - q^{-1}(1 - x)$. Взяв обратную функцию от $\tilde{q}(x)$, получим вторую выпуклую вверх функцию $\ddot{q}(x) = \tilde{q}^{-1}(x)$.

Таким образом, из одной функции $q(x)$ получили дополнительно три ее подвида $\tilde{q}(x)$, $\hat{q}(x)$ и $\ddot{q}(x)$. Ниже в таблице приведены эти функции для трех видов.

На рис. 6 приведены четыре подвида показательной функции $q(x)$ с параметром выпуклости $\lambda = 0,3$.

Используя подвиды функции при заданных весах, построим четыре $Fg_{a\max}(t)$. Для построения квазиконъюнктивного оператора возьмем в качестве генерирующей функции $\psi(x)$ тоже показательную выпуклую вниз $q(x)$. Изменяя параметр выпуклости λ_ψ ,

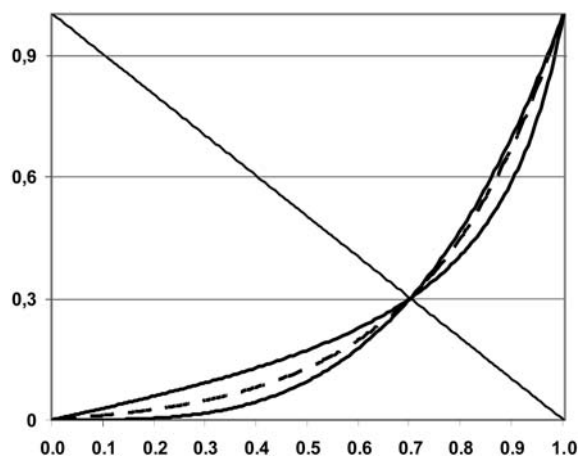


Рис. 5. Примеры видов генерирующих функций

Генерация подвидов выпуклых функций

Наименование функций	Выпуклые вниз		Выпуклые вверх	
	$q_i(x)$	$\tilde{q}(x) = 1 - q^{-1}(1 - x)$	$\hat{q}_i(x) = q_i^{-1}(x)$	$\ddot{q}(x) = \tilde{q}^{-1}(x) = 1 - q(1 - x)$
Степенные	x^p	$1 - (1 - x)^{1/p}$	$x^{1/p}$	$1 - (1 - x)^p$
Показательные	$\frac{e^{ax} - 1}{e^a - 1}$	$1 - \frac{1}{a} \ln((e^a - 1)(1 - x) + 1)$	$\frac{1}{a} \ln((e^a - 1)x + 1)$	$\frac{1 - e^{ax}}{1 - e^{-a}}$
Тангенциальные	$\frac{\operatorname{tg}(ax)}{\operatorname{tg}(a)}$	$1 - \frac{1}{a} \operatorname{arctg}((1 - x)\operatorname{tg}(a))$	$\frac{1}{a} \operatorname{arctg}(x\operatorname{tg}(a))$	$1 - \frac{\operatorname{tg}(a(1 - x))}{\operatorname{tg}(a)}$

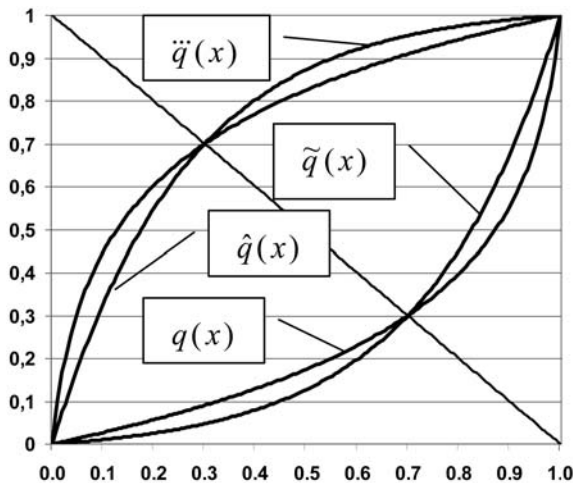


Рис. 6. Подвиды показательной функции

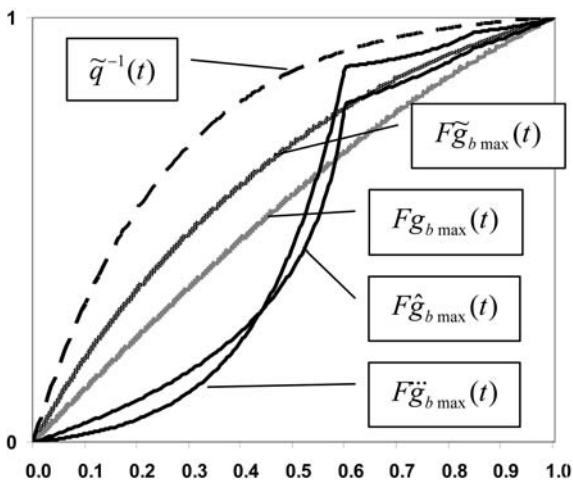


Рис. 7. Генерация операторов на основе $q(x)$ одного вида

получим $\psi^{-1}(x)$ такую, которая будет удовлетворять требованиям теоремы 1: $\psi^{-1}(t) > Fg_{a\max}(t)$ для всех четырех $Fg_{a\max}(t)$, $F\tilde{g}_{a\max}(t)$, $F\hat{g}_{a\max}(t)$, $F\ddot{g}_{a\max}(t)$. На рис. 7 приведены для показательной функции все $Fg_{a\max}(t)$. Получим для одной $\psi(x)$ четыре разных оператора мягкой конъюнкции.

Также получим четыре оператора, используя в качестве $\psi(x)$ выпуклую вниз показательную функцию $\tilde{q}(x)$. Следовательно, используя один вид генерирующей функции, можем получить восемь разных операторов только типа g_a . Аналогично получим восемь показательных операторов типа g_b и восемь типа h .

Если использовать в качестве $\psi(x)$ выпуклые вверх показательные функции $\hat{q}(x)$ и $\ddot{q}(x)$, получим столько же операторов мягкой дизъюнкции.

Используя закон де Моргана (двойственности) [3, стр. 385], из оператора мягкой дизъюнкции

$g_{ad}(u_1, \dots, u_m)$ можем сформировать мягкую конъюнкцию:

$$g_{ac}(u_1, \dots, u_m) = 1 - g_{ad}(1 - u_1, \dots, 1 - u_m).$$

Следовательно, в общем случае на основе четырех подвидов функции можно сформировать восемь разных $g_{ac}(u_1, \dots, u_m)$ и восемь по закону де Моргана из $g_{ad}(u_1, \dots, u_m)$. То же самое для операторов типа $g_b(u_1, \dots, u_m)$,

Для операторов типа h на основе генерирующей функции одного вида формируются восемь разных операторов, а образованные по закону де Моргана совпадают с построенными. Например, возьмем для квазидизъюнкции выпуклую вверх $\psi(x) = \ddot{q}(x)$ с выпуклостью $\lambda_\psi > 0,5$ и $\phi(x) = \ddot{q}(x)$ с $\lambda_\phi > \lambda_\psi > 0,5$ и получим квазиконъюнктивный оператор

$$\begin{aligned} h_c(u_1, \dots, u_m) &= 1 - \ddot{q}_\psi \left(\sum_{j=1}^m V_j \ddot{q}_\phi^{-1}(1 - u_j) \right) = \\ &= q_\psi \left(1 - \sum_{j=1}^m V_j (1 - q_\phi^{-1}(u_j)) \right) = q_\psi \left(\sum_{j=1}^m V_j q_\phi^{-1}(u_j) \right), \end{aligned}$$

что эквивалентно использованию в качестве $\psi(x)$ и $\phi(x)$ функции $q(x)$ с $\lambda_\psi < \lambda_\phi < 0,5$.

Для некоторых функций, как, например, степенной, операторы типа g_a и g_b совпадают, и число разных операторов будет меньше. Существуют функции, для которых $q(x) = \tilde{q}(x)$, они симметричны относительно оси $y = 1 - x$. Для таких функций будет всего два подвида и число разных операторов уменьшается вдвое.

В качестве функции $\psi(x)$ необязательно использовать тот же вид, что и $\phi(x)$, тем более, что $\psi(x)$ необязательно должна быть выпуклой (см. теоремы 1, 2), достаточно $\psi(x)$ быть монотонно-возрастающей, например S-образной с точкой перегиба. Поэтому при выборе вида функции $\psi(x)$ и ее параметров, следует руководствоваться только условиями теоремы о существовании операторов мягкой логики.

В качестве примера построения оператора g_b мягкой конъюнкции возьмем $\phi(x) = \frac{\text{tg}(ax)}{\text{tg}(a)}$ и

$\psi(x) = 1 - (1 - x)^{1/p}$. Сначала запишем выражение для Fg_b :

$$Fg_b = d \sum_{j=1}^m \phi^{-1}(V_j u_j) = \frac{d}{a} \sum_{j=1}^m \arctg(\text{tg}(a) V_j u_j).$$

Итоговый вид оператора:

$$\begin{aligned} g_b(u_1, \dots, u_m) &= \\ &= 1 - \left(1 - \frac{d}{a} \sum_{j=1}^m \arctg(\text{tg}(a) V_j u_j) \right)^{1/p}. \end{aligned}$$

Остается определить параметры a и p , при которых этот оператор будет мягкой конъюнкцией,

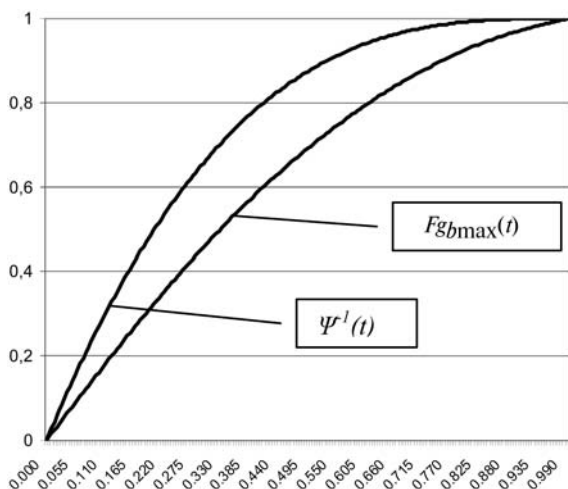


Рис. 8. Проверка условий теоремы о мягкой конъюнкции оператора

т. е. надо выполнить условия теоремы 1. На рис. 8 приведена функция $Fg_{b\max}(t)$ (параметры $a = 1,36$, $d = 0,47$) и $\psi^{-1}(t) = 1 - (1 - t)^p$ (параметр $p = 3$). Так как $\psi^{-1}(t) > Fg_a(t)$, то при данных параметрах a и p оператор является мягкой конъюнкцией.

В работе [1] показано, что показательный оператор типа g_a с $\varphi(x) = \psi(x) = \frac{e^{ax} - 1}{e^a - 1}$ является мультипликативной функцией полезности [4]: $1 + Cg_{ac}(u_1, \dots, u_m) =$

$\prod_{j=1}^m (1 + CkV_j u_j)$, где $C = e^a - 1$. В соответствии с теоремой 5 мультипликативная функция полезности является оператором мягкой конъюнкции при любых весах. Показательный оператор типа g_a

с $\varphi(x) = \psi(x) = \frac{1 - e^{-ax}}{1 - e^{-a}}$ тоже является мультипликативной функцией полезности, но это оператор мягкой дизъюнкции.

Степенная функция с $\varphi(x) = \psi(x) = x^p$, известная как L^p метрика, не при любых весах является мягкой конъюнкцией или дизъюнкцией. Это следует из того, что оператор типа g_a совпадает с g_b , а для операторов g_b см. замечание 1.

6. Идентификация операторов агрегирования

Вопросы идентификации операторов по результатам опроса ЛПР изложены в работе [1].

Свои предпочтения ЛПР задает с помощью весов критериев V_j и в виде оценок псевдообъектов в результате его опроса по определенному плану, который включает:

а) функцию равных значений переменных (ФРЗП); ЛПР оценивает предпочтения объектов $P(u^i) = P(u_1^i, \dots, u_m^i)$ ($i = 1, \dots, n$) с равными значениями критериев $u_j = u^i$;

б) множество значений в вершинах куба (МЗВК); ЛПР оценивает предпочтения объектов по плану полного факторного эксперимента [5, стр. 100—103], каждый критерий меняется на двух уровнях u^- и u^+ , получаем 2^m точек: $P(u_1^-, \dots, u_m^-)$, $P(u_j^+, \bar{u}_j^-, \dots, u_m^-) = W_j$, $P(u_j^+, u_j^+, \bar{u}_{j,s}^-, \dots, u_m^-)$, $P(u_j^-, \bar{u}_j^+, \dots, u_m^-)$, $P(u_1^+, \dots, u_m^+)$. Оценка $P(u_1^-, \dots, u_m^-)$ в МЗВК должна равняться ФРЗП при $u_j = u^-$, а $P(u_1^+, \dots, u_m^+)$ в МЗВК равна ФРЗП при $u_j = u^+$. В принципе необязательно в план опроса включать все 2^m точек. Можно проводить опрос по $m + 2$ точкам $P(u^-)$, $P(u_j^+, \bar{u}_j^-)$, $P(u^+)$, и поменяв значения уровней u^+ и u^- , провести новый опрос по $m + 2$ точкам.

Очевидно, что при мягкой конъюнкции все оценки ЛПР должны быть меньше значений аддитивного оператора $\sum_{j=1}^m V_j u_j$. Это означает, что ФРЗП меньше линейной функции: $P(u) < u$.

МЗВК тоже представляется в виде одномерной дискретной функции, если по оси абсцисс (аргумент МЗВК) принять $x_{\text{МЗВК}} = \sum_{j=1}^m V_j$ по тем крите-

риям, которые равны u^+ в точке куба, а по ординате принять значения МЗВК. Функция МЗВК при нулевом значении аргумента равна $P(u^-)$, а при единичном значении равна $P(u^+)$. Для аддитивного оператора все значения МЗВК находятся на прямой:

$$y = P(u^-) + (P(u^+) - P(u^-))x_{\text{МЗВК}}. \quad (4)$$

Для мягкой конъюнкции все значения МЗВК меньше линейной функции.

Для мягкой дизъюнкции ФРЗП $P(u) > u$, а МЗВК больше линейной функции.

Алгоритм идентификации операторов включает три этапа.

На первом этапе подбирается оператор (тип оператора и генерирующие функции $\varphi(x)$ и $\psi(x)$ из конечного множества), который наиболее точно описывает полученные от ЛПР оценки. Эти вопросы изложены в работе [1]. Для успешной идентификации необходимо иметь большое разнообразие генерирующих функций. Раздел 5 посвящен этой проблеме. Если необходимо описать предпочтения оператором мягкой конъюнкции, то при подборе генерирующих функций и их параметров надо руководствоваться теоремами 1—5.

На втором этапе гладкие функции $\varphi(x)$ и $\psi(x)$ интерполируем кусочно-линейными. В последующем на третьем этапе применяется алгоритм кусочно-линейной аппроксимации, который изменяет кусочно-линейную функцию, уменьшая расхожде-

ние между оценками ЛПР и расчетными значениями оператора.

В условиях теоремы 1 о существовании операторов мягкой конъюнкции требование по выпуклости относится только к $\varphi(x)$, а $\psi(x)$ может быть невыпуклой. Поэтому в алгоритме кусочно-линейной аппроксимации снижение требований к $\psi(x)$ позволяет повысить точность описания оценок ЛПР оператором.

Заключение

Статья развивает подход к описанию предпочтений в МКЗ, изложенный в работе [1].

Рассматривается класс предпочтений, для которого в систему аксиом, приведенных в работе [1] — монотонности предпочтений, важности критериев, добавлена аксиома логики предпочтений. Для описания данного класса предпочтений используются специальный класс операторов агрегирования, названный мягкой логикой, он включает операторы мягкой конъюнкции (квазиконъюнкция) и мягкой дизъюнкции (квазидизъюнкция).

Доказаны теоремы, определяющие требования к генерирующим функциям для построения операторов мягкой логики.

Рассмотрены вопросы генерации операторов на основе типов операторов, видов генерирующих функций и с использованием закона двойственности. На основе одной функции, например показательной, можно построить до 40 различных операторов мягкой конъюнкции.

Приводятся примеры операторов мягкой логики, в том числе и ранее неизвестные. Из известных операторов мультипликативная функция полезности является оператором мягкой логики при любых значениях весов, а L^p -метрика не отвечает аксиоме мягкой логики.

Список литературы

1. Елтаренко Е. А. Аппроксимация предпочтений в многокритериальных задачах // Информационные технологии. 2011. № 6. С. 23—32.
2. Рокафеллар Р. Выпуклый анализ: пер. с англ. М.: Мир, 1973. 470 с.
3. Корн Г., Корн Т. Справочник по математике (для научных работников и инженеров): пер. с англ. М.: Наука, 1973. 832 с.
4. Кини Р. Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения: пер. с англ. М.: Радио и связь, 1981. 560 с.
5. Низаметдинов Ш. У. Анализ данных: учеб. пособие. М.: МИФИ, 2006. 248 с.

УДК 004.89

И. П. Норенков, д-р техн. наук, проф., зав. каф.
МГТУ им. Н. Э. Баумана
e-mail: inorenkov@gmail.com

Редукция графа работ в задаче синтеза расписаний

Генетические алгоритмы являются основным методом решения задач синтеза расписаний. Однако при больших размерах задач эффективность генетических алгоритмов оказывается невысокой. В статье предложен подход к снижению размера модели для задач синтеза расписаний с взаимосвязанными работами. Подход основан на редукции модели, заключающейся в использовании графа кластеров вместо графа работ.

Ключевые слова: синтез расписаний, ациклический орграф, генетические алгоритмы, кластеризация, редукция графа

Введение

Во многих приложениях требуется решать задачи синтеза расписаний. В этих задачах фигурируют множество динамических объектов, называемых работами, и множество ресурсов (исполнителей), называемых машинами, и требуется распределить

работы по машинам и во времени. Как правило, задачи решаются при наличии ограничений и дополнительных условий, к числу которых могут относиться данные, характеризующие типы ресурсов, их производительность, затраты на использование ресурсов, зависимости этих затрат от типа работы, число стадий обслуживания работ и т. п. Разнообразие таких ограничений и условий обуславливает разнообразие типов задач синтеза расписаний. Так, различают задачи JSSP (*job shops scheduling problem*), OSSP (*open shops scheduling problem*) [1], FSP (*flow shops scheduling problem*) [2], FSM (*family scheduling model*) [3] и другие, обычно обозначаемые $n/m/a/b$, где n — число работ; m — число машин; a — тип задачи; b — критерий оптимальности.

Отдельный класс задач синтеза расписаний составляют задачи с работами, взаимосвязанными отношениями предшествования, называемые задачами DAGSP (*DAG scheduling problems*). В этом классе задач принято использовать представление работ и их отношений в виде ациклического орграфа, обозначаемого DAG (*direct acyclic graph*) [4], в котором множество вершин отображает работы, а множество дуг — отношения предшествования работ.

Характерными задачами DAGSP являются задачи распределения вычислительных задач по процессорам мультипроцессорных систем или в GRID-сис-

темах, производственных операций по бригадам исполнителей или единицам технологического оборудования. Наибольшее внимание к настоящему времени уделялось разработке алгоритмов для задач планирования вычислительных процессов в многопроцессорных системах на основе различных эвристических подходов [4–6]. Ставится задача кластеризации работ, где каждый формируемый кластер представляет собой подмножество работ, последовательно без прерываний исполняемых на конкретном ресурсе [5]. В качестве методов решения задач синтеза расписаний используют преимущественно генетические алгоритмы (ГА) [7, 8 и др.], в том числе и для решения DAGSP [6].

Однако применение генетических алгоритмов в задачах большого размера (число работ 10^4 и более) затруднительно как из-за значительных затрат процессорного времени, так и из-за снижения точности результатов. Примерами приложений, в которых имеют место столь большие размеры задач DAGSP, могут служить задачи планирования работ при строительстве крупных объектов или задачи определения оптимальных траекторий обучения и синтеза электронных учебных пособий в образовательной сфере.

Для преодоления трудностей применения ГА для задач синтеза расписаний целесообразно использовать кластеризацию работ. В работе [5] кластеризация являлась целью решения задачи, и ГА применялся к исходной задаче большого размера. В работе [9] авторы используют двухфазный ГА: на первой фазе работы распределяются по кластерам, на второй фазе проводится назначение кластеров на процессоры и упорядочение работ внутри кластеров. Хотя авторы работы [9] отмечают, что кластеризация является средством снижения размера задачи, однако ГА применяется на обеих фазах алгоритма, причем на первой фазе размер задачи определяется общим числом связей между работами, как правило, превышающим число работ.

В данной статье предложен метод синтеза расписаний, в соответствии с которым ГА применяется к редуцированной задаче приемлемого размера.

Постановка задачи

Задача DAGSP может быть представлена в виде графа $G(V, E)$, где $V = (v_1, v_2, \dots, v_n)$ — множество вершин, соответствующих работам; $E = \{e_{ij}\}$ — множество дуг, причем $e_{ij} = \{v_i, v_j\}$ означает, что работа v_j может выполняться только после завершения работы v_i . Заданы также матрицы $D = \{d_{kl}\}$, где d_{kl} — затраты времени на выполнение k -й работы на l -м ресурсе; $S1 = \{s1_{pq}\}$ и $S2 = \{s2_{pk}\}$, где $s1_{pq}$ и $s2_{pk}$ — затраты соответственно времени и средств на передачу данных от p -го к q -му ресурсу, и вектор $C = \{c_l\}$, где c_l — цена единицы времени использования l -го ресурса. Для многих задач характерна однотипность ресурсов, при этом D — вектор, все c_l одинаковы, именно такие задачи рассматриваются далее.

Расписание представляет собой распределение кластеров, сформированных из работ, по исполнителям и во времени. В качестве векторного критерия оптимального распределения используется множество частных критериев, основными из них являются суммарное время T_0 занятости исполнителей, срок выполнения всего множества работ, общее число N_a межкластерных дуг. Показатели T_0 и N_a могут быть переведены в стоимостные при задании **S2** и **C**.

В генетическом алгоритме кластеризации, предложенном в работе [9], расписания кодируются в виде хромосомы, длина которой составляет $|E|$ генов. Существенного снижения длины хромосомы, а, следовательно, и сложности решения задачи, можно добиться, если кодирование расписаний выполнять с помощью хромосомы, число генов в которой определяется числом m потенциальных кластеров.

Метод кластеризации работ. Кластеризация основана на следующих свойствах DAG.

1. Все вершины DAG могут быть проранжированы следующим образом: вершины, не имеющие входящих дуг, и их выходные дуги получают нулевой ранг $r = 0$. Вершина, у которой старший ранг входящих дуг равен r , и ее выходные дуги получают ранг $r + 1$. На рис. 1 показан фрагмент DAG (направления дуг снизу—вверх) с ранжированными вершинами.

2. Все вершины DAG могут быть разделены на поддеревья B_i с корневыми вершинами q_i , $i = [1:m]$, m — число кластеров, так, что q_i имеет минимальный ранг среди вершин поддерева, а каждая некорневая вершина поддерева ранга r инцидентна дуге, исходящей из некоторой вершины ранга $r - 1$ того же поддерева. Далее кластеры обозначаются так же, как их корневые вершины. На рис. 1 поддеревья выделены контурами из штриховых линий.

3. Кластеры образуются из вершин одного из поддеревьев, причем корневая вершина поддерева становится корневой вершиной кластера.

Корневые вершины выбираются среди вершин малых рангов, начиная с $r = 0$, таким образом, что если на уровне ранга r_k имеется структура "разветвление" (рис. 2), то кандидатами на статус корневых

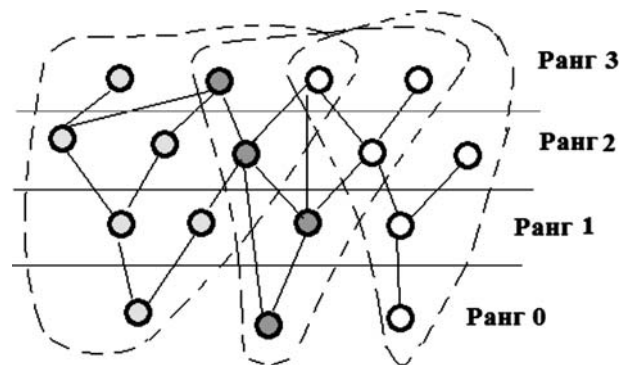


Рис. 1

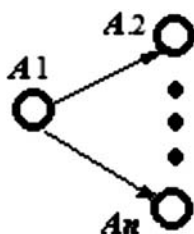


Рис. 2

вершин становятся вершина $A1$ ранга r_k и смежные с ней вершины $A2 \dots An$ на уровне $r_k + 1$. Для ограничения сверху размеров кластеров в качестве корневых выбираются также вершины некоторого порогового ранга $r_{\text{пор}}$ аналогично тому, как это выполнялось для вершин ранга 0. При этом кластеры с корневыми вершинами с $r = 0$ или с близкими к $r = 0$ рангами будем называть кластерами первого яруса, кластеры с корневыми вершинами вблизи $r = r_{\text{пор}}$, $2r_{\text{пор}}$ и т. д. — кластерами соответственно второго, третьего ярусов и т. д.

Варианты расписания различаются выбором параметра $r_{\text{пор}}$ и способом включения некорневых вершин, входящих в несколько поддеревьев, в один из кластеров. Способ распределения вершин по кластерам и составляет сущность алгоритма кластеризации, выполняемого в соответствии с описываемым ниже эвристическим правилом.

Кластеризация начинается с помещения в каждый кластер по одной корневой вершине. Кластеры q_i нумеруются последовательно ярус за ярусом, причем способ нумерации внутри каждого яруса определяет эффективность кластеризации. Далее выполняется циклический процесс по параметру r . На r -м витке распределяются по кластерам вершины ранга r по следующему правилу: очередная вершина помещается в смежный кластер, имеющий наибольший номер, называемый приоритетом кластера.

Соблюдение этого правила обеспечивает бесконечность искомого редуцированного графа кластеров $G_k = \{X, Y\}$, в котором вершины множества X соответствуют кластерам, дуги множества Y — межкластерным связям. В частности, соблюдается однонаправленность дуг в любой паре кластеров, т. е.

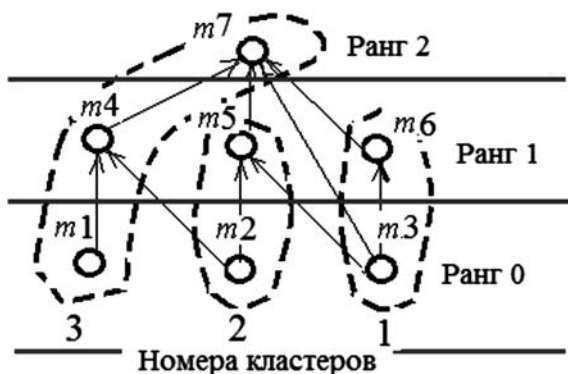


Рис. 3

при наличии дуги из кластера p в кластер q не может быть дуг из q в p .

Обеспечение бесконечности иллюстрируется примером фрагмента графа (рис. 3), в котором имеются три вершины графа G ранга 0 и соответственно создаются три кластера. При распределении вершин ранга 1 для вершины $m4$ смежными являются кластеры 3 и 2 и вершина помещается в кластер с большим номером — в кластер 3. Аналогичным образом вершина $m5$ попадает в кластер 2, $m6$ — в кластер 1, $m7$ — в кластер 3. В результате кластер 1 имеет только исходящие дуги, в кластер 2 входят дуги только из кластера 1, а выходят только в кластер 3, кластер 3 имеет только входящие дуги, т. е. обеспечивается однонаправленность.

Очевидно, что изменяя способ нумерации кластеров, получаем разные расписания с разной эффективностью. Квазиоптимальную нумерацию целесообразно искать с помощью ГА, в котором гены соответствуют кластерам, а аллели равны их номерам (приоритетам).

Заключение

Выполненная редукция графа работ существенно уменьшает размер задачи. Вместо хромосом длиной в n генов, где n — число вершин (или ребер) графа работ G , используются хромосомы с числом генов, равным числу m кластеров, причем обычно $m \ll n$. Это обуславливает возможность применения генетических алгоритмов для синтеза расписаний при большой мощности множества работ.

Работа выполнена при финансовой поддержке РФФИ (код проекта 10-07-00401-а).

Список литературы

1. Liaw C. F. A hybrid genetic algorithm for the open shop scheduling problem // European Journal of Operational Research. 2000. N 124. P. 28—42.
2. Hejazi S. R., Saghafian S. Flowshop-scheduling problems with makespan criterion: a review. // International Journal of Production Research. 2005. Vol. 43, N 14. P. 2895—2929.
3. Potts C. N., Kovalyov M. Y. Scheduling with batching: A review // European Journal of Operational Research. 2000. N 120. P. 228—249.
4. Kwok Y., Ahmad I. Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors // ACM Computing Surveys. 1999. Vol. 31, N 4. P. 406—471.
5. Kianzad V., Bhattacharyya S. S. Multiprocessor Clustering for Embedded Systems. URL: <http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA455028>
6. Correa R. C., Ferreira A., Rebreyend P. Scheduling Multiprocessor Tasks with Genetic Algorithms // IEEE Trans. Parallel and Distributed Systems. 1999. Vol. 10, N 8. P. 825—837.
7. Syswerda G., Palmucci J. The application of genetic algorithms to resource scheduling // Proc. of Fourth int. conf. on GA. 1991. P. 502—508.
8. Norenkov I., Goodman E. Solving Scheduling Problems via Evolutionary Methods for Rule Sequence Optimization // Soft Computing in Engineering Design and Manufacture / P. K. Chawdry et al., eds. Springer, 1998. P. 350—355.
9. Rinehart M., Kianzad V., Bhattacharyya S. S. A Modular Genetic Algorithm for Scheduling Task Graphs. URL: <http://www.cs.imd.edu/Library/TRs/CS-TR-4497/CS-TR-4497.pdf>

В. В. Стрижов, канд. физ.-мат. наук, научный сотр.,
e-mail: strijov@gmail.com,
Е. А. Крымова, аспирант,
Вычислительный центр РАН

Выбор моделей в линейном регрессионном анализе¹

При решении практических задач восстановления регрессии для отыскания адекватной модели предлагается использовать порожденные признаки, полученные с помощью измеряемых исходных признаков. Это влечет существенное повышение размерности признакового пространства и, как следствие, необходимость использования алгоритмов выбора признаков: предполагается, что число признаков сравнимо или превосходит число элементов регрессионной выборки. Рассматриваются и сравниваются классические и новые алгоритмы выбора признаков. Работа алгоритмов проиллюстрирована прикладными задачами.

Ключевые слова: регрессионный анализ, выбор моделей, линейные модели, оценивание параметров, сравнение алгоритмов

Введение

Решение практических задач восстановления регрессионной зависимости требует рассмотрения большого числа порождаемых признаков. Процедура построения регрессионных моделей состоит из двух шагов. На первом шаге на основе свободных переменных — результатов измерений, порождается набор признаков. На втором шаге выбираются признаки. При выборе признаков выполняется настройка параметров модели и оценивается ее качество. Модель с настроенными параметрами, доставляющая минимум заданному функционалу качества, называется моделью оптимальной структуры. Целью данной работы является обзор и сравнение алгоритмов выбора признаков. Мотивацией работы является тот факт, что решение практических задач восстановления регрессионной зависимости требует рассмотрения большого числа порождаемых признаков.

Развитие методов отбора признаков в регрессионном анализе имеет насыщенную историю. Среди методов отбора признаков широкое распространение получил шаговый метод, впервые предложенный в 1960 г. М. А. Эфроимсоном [1]. На каждом шаге признаки проверяются на возможность добавления в модель или удаления из модели. При этом могут использоваться критерии: F-критерий,

критерий Акаике, байесовский критерий [2, 3], критерий Маллоуза [4].

В 1970 г. А. Хоэрл и Р. Кеннард предложили метод гребневой регрессии, в котором использовалась регуляризация [5], позволяющая улучшить устойчивость решения [6].

В книге Н. Дрейпера 1966 г. [7] приведен ступенчатый алгоритм. На каждом шаге выбирается признак с наибольшей проекцией на вектор ответов, делается шаг в направлении признака [8]. Среди полученных моделей находится оптимальная, тем самым проводится отбор признаков.

В методе группового учета аргументов на каждом шаге выполняется отбор моделей и построение на их основе более сложных моделей [9]. Метод позволяет сократить перебор и осуществить отбор признаков.

В работах Шена 1989 г. [10] рассмотрен алгоритм последовательного добавления признаков с ортогонализацией (*Forward Orthogonal search*). Отбор признаков происходит автоматически при выборе оптимальной модели.

Еще один метод регуляризации, Лассо, был предложен Тибширани в 1996 г. [11]. В нем вводится ограничение на L_1 -норму вектора параметров модели, что приводит к обнулению части параметров модели.

В 2004 г. Эфрон, Хастис, Джонстон и Тибширани предложили метод наименьших углов (*Least Angle Regression*) [12]. На каждом шаге признак выбирается так, что вектор регрессионных остатков равноуголен уже добавленным в модель признакам.

Сравнение алгоритмов выполнялось на задаче моделирования волатильности опционов. Использовались исторические данные торгов опционом Brent Crude Oil. В таблицу результатов входят значения функционала качества на обучающей и контрольной выборке, критерий Акаике, число переменных модели. Исходя из этих значений делается вывод об эффективности работы алгоритмов.

Постановка задачи

Задана выборка $D = \{(\mathbf{x}^i, y^i)\}_{i=1}^m$ — множество m пар, состоящих из строк значений n свободных переменных $\mathbf{x}^i = (x_j^i)_{j=1}^n \in \mathbf{R}^n$, и значения зависимой переменной $y^i \in \mathbf{R}^1$, где $i \in I = \{1, \dots, m\}$ и $j \in J = \{1, \dots, n\}$. Задано разбиение выборки $I = L \cup C$ на обучающую и контрольную.

Задан класс регрессионных моделей $F = \{f_s\}$ — параметрических функций, линейных относительно параметров,

$$y^i = f_s(\beta_s, \mathbf{x}^i) = \sum_{j \in J_s} \beta_j x_j^i, \quad (1)$$

¹Работа выполнена при финансовой поддержке РФФИ, проект 10-07-00422.

здесь $s \in \{1, \dots, 2^n\}$ — индекс модели; $\beta_s = (\beta_j)_{j \in J_s}$ — вектор параметров модели f_s .

Под сложностью модели (1) понимается число линейно входящих параметров. Введено ограничение на сложность модели (1): в множество F могут входить только модели сложностью не больше R . Алгоритм выбора модели задает метод оптимизации, использующийся для отыскания оптимальной модели на обучающей выборке $\{(\mathbf{x}^i, y^i) : i \in L\}$.

Пусть случайная аддитивная переменная v регрессионной модели

$$y = f(\beta, \mathbf{x}) + v$$

распределена нормально $v : N(0, \sigma^2)$. При условии гомоскедастичности регрессионных остатков распределение зависимой переменной имеет вид

$$p(y|x, \beta, \sigma^2, f) = \frac{\exp\left(-\frac{1}{2\sigma^2}[y^i - f(\beta, \mathbf{x}^i)]^2\right)}{(2\pi\sigma^2)^{\frac{n}{2}}}.$$

Это распределение задает критерий качества модели

$$S = \sum_{i \in W} (y^i - f(\beta, \mathbf{x}^i))^2, \quad (2)$$

здесь W — некоторое множество индексов. Параметры модели находятся из минимизации критерия (2) на обучающей выборке, т. е. при $W \stackrel{\Delta}{=} L$. Требуется найти такую модель оптимальной структуры $f_s \in F$, которая доставляет минимум функцио-

налу качества (2) на контрольной выборке $W \stackrel{\Delta}{=} C$.

Для удобства описания алгоритмов выбора признаков обозначим вектор-столбец $\mathbf{x}_j = (x_j^1, \dots, x_j^m)$ и $\mathbf{y} = (y^1, \dots, y^m)$. Тогда $\mathbf{y} = \beta_1 \mathbf{x}_1 + \dots + \beta_n \mathbf{x}_n$ или $\mathbf{y} = \mathbf{X}\beta$, где $\mathbf{X}$ — матрица признаков со столбцами $\mathbf{x}_1, \dots, \mathbf{x}_n$. Предполагается, что векторы $\mathbf{x}_j, \mathbf{x}_k$ $j, k \in J$, $j \neq k$, линейно независимы.

Генетический алгоритм

В основе алгоритма лежит порождение моделей с помощью операций скрещивания и мутации из текущего множества моделей (популяции). Используем вектор выбора признака: $\mathbf{c} = (c_1, \dots, c_n)$, $c_i \in \{0, 1\}$. Заданы параметры: число лучших моделей в популяции F , число моделей для скрещивания F_1 , вероятность выбора модели для мутации P_2 . Начальный набор моделей выбирается случайным образом. Итеративно выполняются следующие шаги.

1. Отбор: согласно критерию (2) при $W \stackrel{\Delta}{=} C$ выбирается F лучших моделей.

2. Случайным образом выбираются F_1 моделей.

3. Скрещивание: из двух моделей порождаются новые. Выбранные модели случайным образом разбиваются на пары. В каждой паре переменные выбора $\mathbf{c}^q$ и $\mathbf{c}^p$ разбиваются части точкой, выбираемой случайно из $\{1, \dots, n\}$. Происходит обмен компонент $\mathbf{c}^p$ и $\mathbf{c}^q$:

$$\begin{cases} (c_1^q, \dots, c_k^q, c_{k+1}^q, \dots, c_n^q) \\ (c_1^p, \dots, c_k^p, c_{k+1}^q, \dots, c_n^p) \end{cases} \rightarrow \begin{cases} (c_1^q, \dots, c_k^q, c_{k+1}^p, \dots, c_n^p) \\ (c_1^p, \dots, c_k^p, c_{k+1}^q, \dots, c_n^q). \end{cases}$$

4. Каждая модель из полученного множества с вероятностью P_2 подвергается мутации: случайным образом выбирается индекс j , значение компоненты c_j меняется на $\bar{c}_j$ (если $c_j = 0$, то после операции $c_j = 1$).

После операций 3 и 4 новые модели настраиваются из минимизации критерия (2) при $W \stackrel{\Delta}{=} L$. Процесс повторяется, пока не выполнится условие останова.

Метод группового учета аргументов (МГУА)

Алгоритмы МГУА воспроизводят схему массовой селекции, согласно которой последовательно порождаются и выбираются модели возрастающей сложности. В рамках МГУА применяются следующие критерии. Внутренним называется критерий (2), использующий только обучающую выборку X_L . Внешним называется критерий, использующий тестовую выборку X_C , параметры модели настроены из минимизации внутреннего критерия [9].

На каждом шаге алгоритма происходит порождение моделей. Затем выбираются P лучших моделей согласно внешнему критерию. Обозначим линейную комбинацию признаков как $f(x_1, x_2) = \beta_1 x_1 + \beta_2 x_2$, где β_1, β_2 — коэффициенты модели. Верхним индексом зависимой переменной обозначается номер шага. Порождение моделей происходит следующим образом. На первом шаге строится $Q_1 = C_n^2$ моделей — линейных комбинаций двух признаков. Выбирается P_1 лучших моделей $y_1^1, \dots, y_{P_1}^1$ согласно внешнему критерию. На втором шаге из пар $y_1^1, \dots, y_{P_1}^1$ образуются новые модели. После отыскания параметров выбирается P_2 лучших моделей $y_1^2, \dots, y_{P_2}^2$:

$$y_1^1 = f(x_1, x_2), y_1^2 = f(y_1^1, y_2^1),$$

$$y_2^1 = f(x_1, x_3), y_2^2 = f(y_1^1, y_3^1),$$

.....

$$y_{Q_1}^1 = f(x_{n-1}, x_n), y_{Q_2}^2 = f(y_{P_1-1}^1, y_{P_1}^1).$$

Шаг k селекции:

$$\begin{aligned} y_1^k &= f(y_1^{k-1}, y_2^{k-1}), \\ y_2^k &= f(y_1^{k-1}, y_3^{k-1}), \\ &\dots \\ y_{Q_k}^k &= f(y_{P_{k-1}}^{k-1}, y_{P_{k-1}}^{k-1}). \end{aligned}$$

Процесс продолжается, пока значение внешнего критерия уменьшается.

Шаговая регрессия

Шаговыми методами называются методы, заключающиеся в пошаговом удалении или добавлении признаков согласно определенному критерию. Обычно используется критерий Фишера

$$F = \frac{S_1 - S_2}{S_2} \frac{m - n_2}{n_1 - n_2},$$

где индекс 2 соответствует основной линейной модели; индекс 1 соответствует линейной модели, которая является модификацией второй модели; n_1, n_2 — соответствующие числа параметров в моделях. Если значение критерия больше заданного, то вторая модель считается лучше. Шаговая регрессия включает два шага: Add (добавление) и Del (удаление признаков).

Алгоритм Add. Рассмотрим текущий набор индексов признаков A . Начальным набором является набор $A = \emptyset$. Последовательно к текущему набору A добавляется индекс j^* , определяемый следующим образом:

$$j^* = \arg \max_{j \in J} F_{\text{Add}} \propto \arg \max_{j \in J} \frac{S(A) - S(A \cup \{j\})}{S(A \cup \{j\})}.$$

Алгоритм Del. Начальный набор состоит из всех признаков. На каждом шаге происходит удаление признака j^* :

$$j^* = \arg \min_{j \in J} F_{\text{Del}} \propto \arg \min_{j \in J} \frac{S(A \setminus j) - S(A)}{S(A)}.$$

Алгоритм шаговой регрессии заключается в чередовании шагов Add и Del при проверке значимости уже добавленных ранее признаков. Начальным набором является набор индексов $A = \emptyset$. Останов алгоритма проводится при достижении минимума, заданного критерием Маллоуза C_p :

$$C_p = \frac{S}{\text{MSE}} + 2k - m,$$

где MSE — среднеквадратичная ошибка, вычисленная для модели, построенной по всем признакам; S — значение (2) на модели, построенной по k признакам. Критерий штрафует модели с большим числом признаков.

Недостатком метода является то, что важная переменная может никогда не войти в модель, а второстепенные признаки будут включены. Метод используется, когда число объектов сильно превышает число признаков.

Гребневая регрессия

Алгоритм заключается во введении регуляризующего слагаемого в функционал (2). Это позволяет уменьшить число обусловленности матрицы $X^T X$ и, как следствие, повысить устойчивость решения. Строго говоря, метод не приводит к отбору признаков, так как в решение входят все признаки.

Параметры модели находятся из минимизации нового критерия:

$$\beta^* = \arg \min_{\beta} \left(\sum_{i=1}^m \left(y^i - \sum_{j=1}^m \beta^j x_j^i \right)^2 + \tau \|\beta\|^2 \right).$$

Решением задачи минимизации является вектор $\beta^* = (X^T X + \tau I_n)^{-1} X^T y$, где I_n — единичная матрица размерности $n \times n$. Увеличение коэффициента регуляризации τ приводит к уменьшению нормы вектора β^* и повышению эффективной размерности признакового пространства [13].

На рис. 1 (левый график) показано пространство параметров модели. Критерий S — квадратичная функция относительно компонент β , поэтому кривая $S = \text{const}$ является эллипсоидом. Параметр τ , отличный от нуля, задает сферу. При этом касание эллипсоида сферой в нулевой точке исключено, и обнуления параметров модели не происходит. Метод улучшает устойчивость параметров регрессионной модели, но не приводит к обращению в ноль ни одного из них (рис. 2, см. третью сторону обложки).

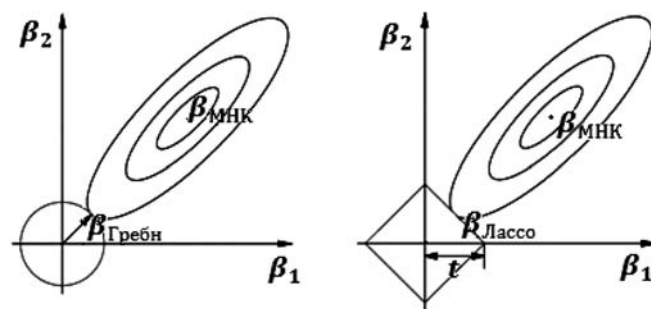


Рис. 1. Пространство параметров и ограничения, которые задача регуляризации накладывает на параметры

Лассо

(Least absolute shrinkage and selection operator)

В методе Лассо вводится ограничение на сумму абсолютных значений параметров модели. В отличие от гребневой регрессии в Лассо некоторые параметры становятся равными нулю, значит, происходит отбор признаков.

Параметры модели находятся из минимизации критерия (2) при $\sum_{j=1}^n |\beta_j| \leq t$, где t — параметр регуляризации. Чем меньше t , тем больше коэффициентов β_j обнуляется. Таким образом, Лассо осуществляет отбор признаков.

Метод решения. Каждый параметр β_j записывается в виде $\beta_j^+ - \beta_j^-$, где β_j^+ и β_j^- неотрицательны. Тогда ограничения-неравенства принимают вид

$$\beta_j^+ \geq 0, \beta_j^- \geq 0, \sum_{j=1}^n (\beta_j^+ + \beta_j^-) \leq t.$$

Таким образом, начальная задача с n переменными и $2n$ ограничениями преобразуется в задачу с $2n$ переменными и $(2n + 1)$ ограничениями.

На рис. 1 (правый график) показано пространство параметров модели. Кривая $S = \text{const}$ является эллипсоидом. Параметр t , отличный от нуля, задает многомерный октаэдр в этом пространстве. Точка касания эллипсоида и октаэдра является решением нормального уравнения при фиксированном t . При касании сферы и ребра октаэдра происходит обнуление коэффициента.

Ступенчатая регрессия

Алгоритм ступенчатой регрессии состоит в последовательном добавлении признаков, наиболее коррелирующих с вектором регрессионных остатков. Начальный набор признаков пуст, вектор остатков $\mathbf{r}_0 = \mathbf{y}$.

Рассмотрим k -й шаг алгоритма. Отыскивается признак с индексом $j = \arg\max_i \mathbf{r}_k^T \mathbf{x}_i$. Обновляется вектор остатков

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \varepsilon \text{sign}(\mathbf{r}_k^T \mathbf{x}_j) \mathbf{x}_j,$$

где ε — достаточно маленькое число. Выбор большого ε , т. е. $\varepsilon = \frac{|\mathbf{r}_k^T \mathbf{x}_j|}{\|\mathbf{x}_j\|^2}$, приводит к алгоритму Add.

Процесс продолжается, пока значение критерия (2) не будет незначительно изменяться за шаг.

В алгоритме ступенчатой регрессии направление для шага находится так же, как в Add, но шаг берется достаточно маленьким. Это приводит к большим вычислительным затратам, но и к более тщательному отбору признаков.

Последовательное добавление признаков с ортогонализацией

Метод последовательного добавления состоит в последовательном добавлении признаков с процедурой ортогонализации признаков-столбцов матрицы $\mathbf{X}$. Ортогонализация делает возможным вычис-

ление индивидуального вклада каждого признака в вектор значений зависимой переменной.

Запишем ортогональное разложение матрицы: $\mathbf{X} = \mathbf{Q}\mathbf{R}$, где $\mathbf{Q}$ — матрица, столбцы которой являются ортогональным базисом матрицы признаков $\mathbf{X}$, а $\mathbf{R}$ — верхняя треугольная матрица. Тогда $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} = \mathbf{Q}\mathbf{g}$, где $\mathbf{g} = \mathbf{R}\boldsymbol{\beta}$. Пусть на k -м шаге получен вектор остатков $\mathbf{r}_k$. Обозначим A_k — текущий набор признаков, $\bar{A}_k$ — остальные признаки. Начальное значение вектора остатков $\mathbf{r}_0 = \mathbf{y}$ (рис. 3, а), текущий набор признаков A_0 пуст.

Рассмотрим k -й шаг алгоритма.

1. Находим признак с индексом j , который составляет наименьший угол с вектором остатков $\mathbf{r}_k$,

$$j = \max_{i \in A_{k-1}} \frac{\mathbf{x}_i^T \mathbf{r}_k}{\|\mathbf{x}_i\| \|\mathbf{r}_k\|}.$$

Признак j добавляется в текущий набор A_k и удаляется из $\bar{A}_k$.

2. Находим проекцию вектора $\mathbf{r}_k$ на $\mathbf{x}_j$ (рис. 3, б):

$$\mathbf{r}_k^{\text{Pr}} = \frac{\mathbf{x}_j^T \mathbf{r}_k}{\|\mathbf{x}_j\|^2} \mathbf{x}_j.$$

3. Находим параметр, соответствующий добавленному признаку,

$$g_j = \|\mathbf{r}_k^{\text{Pr}}\| / \|\mathbf{x}_j\|. \quad (3)$$

4. Обновляется вектор остатков (рис. 3, в): $\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{r}_k^{\text{Pr}}$.

5. Признаки с индексами из $\bar{A}_k$ проецируются на подпространство, ортогональное пространству признаков из A_k (рис. 3, в). Происходит ортогонализация

$$\text{векторов признаков: } \mathbf{X}_{\bar{A}_k} = \mathbf{X}_{\bar{A}_{k-1}} - \mathbf{x}_j \frac{(\mathbf{x}_j^T \mathbf{X}_{\bar{A}_{k-1}})}{\|\mathbf{x}_j\|^2}.$$

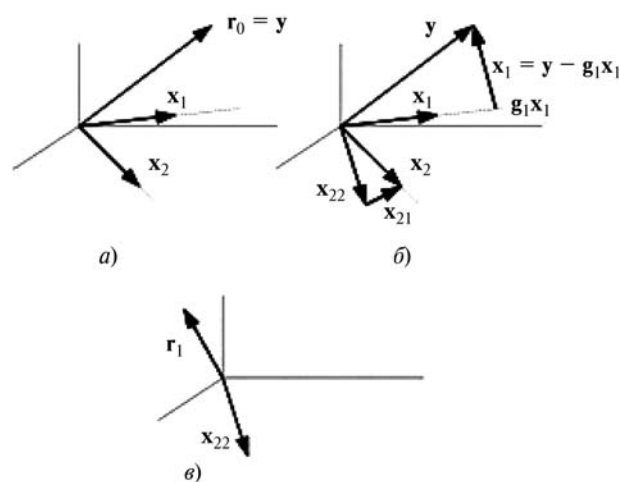


Рис. 3. Шаги алгоритма последовательного добавления признаков с ортогонализацией

Вектор параметров β находится из формулы (3): $\beta = R^{-1}g$. Алгоритм последовательно добавляет признаки за n шагов, выбирая наименее коррелированные признаки.

Метод наименьших углов

На каждом шаге алгоритма происходит изменение вектора параметров модели так, чтобы доставить добавляемому признаку наибольшую корреляцию с вектором регрессионных остатков.

Алгоритм последовательно вычисляет приближение зависимой переменной $\mu = X\beta$. На k -м шаге только k элементов вектора β отличны от нуля. Для приближений используется вектор корреляций столбцов матрицы X с вектором остатков $y - \mu$: $c(\mu) = X^T(y - \mu)$.

На k -м шаге приближение зависимой переменной y вычисляется как

$$\mu_k = \mu_{k-1} + \gamma_k u_k.$$

Здесь u_k — вектор, вычисляемый следующим образом. Пусть A — подмножество индексов признаков. Тогда A задает матрицу

$$X_A = [s_{j_1} x_{j_1}, \dots, s_{j_{|A|}} x_{j_{|A|}}], j \in A,$$

где множитель $s \in \{+1, -1\}$; $|A|$ — мощность множества A . Обозначим ковариационную матрицу

$G = X_A^T X_A$ и $h_A = (1_A^T G^{-1} 1_A)^{-1/2}$, где 1_A — вектор, состоящий из $|A|$ единиц. Определим вектор $u_A = X_A w_A$,

где $w_A = h_A G^{-1} 1_A$. Вектор u образует со столбцами матрицы X_A одинаковые углы, меньшие $\frac{\pi}{2}$. Справедливо, что $X_A^T u_A = h_A 1_A$ и $\|u_A\| = 1$.

Выполнение алгоритма. Начальная оценка вектора значений зависимой переменной $\mu = 0$. Вычислим текущую оценку μ_A и вектор корреляций

$$c = X^T(y - \mu_A).$$

Найдем текущий набор индексов A , соответствующих признакам с наибольшими абсолютными значениями корреляций $C = \max_{j=1, \dots, m} |c_j|$ и $A = \{j : |c_j| = C\}$. Пусть $s_j = \text{sign}(c_j)$ для $j \in A$. Построим матрицу X_A , вычислим h_A . Вычислим вектор u_A и вектор скалярных произведений $a = X^T u_A$. Пересчитывается значение вектора μ_A :

$$\mu_{A+} = \mu_A + \hat{\gamma} u_A,$$

где

$$\hat{\gamma} = \min_{j \in A^C} + \left\{ \frac{C - c_j}{A - a_j}, \frac{C + c_j}{A + a_j} \right\}. \quad (4)$$

Минимум берется по всем положительным значениям аргументов для каждого j из множества

$\bar{A} = \{1, \dots, n\} \setminus A$ индексов признаков, не вошедших в множество A . Добавим в множество A индекс j , где j доставляет минимум соответствующему значению $\hat{\gamma}$. Можно показать, что $\hat{\gamma}$ в выражении (4) — минимальная положительная величина, при которой новый индекс j может быть добавлен в набор A . Шаги повторяются n раз, по числу признаков. На рис. 4 (см. третью сторону обложки) приведена иллюстрация частного случая описанного выше алгоритма для двух признаков.

Вычислительный эксперимент

Сравнительный анализ алгоритмов выполнен на исторических данных торгов опционом Brent Crude Oil. Сетка цен исполнения опциона $K = \{1400, 1425, \dots, 1575, 1600\}$. Регрессионная выборка $\{(x^i, y^i)\}_{i=1}^m = \{(\langle K_p, t_p \rangle, \sigma_p)\}_{i=1}^m$ построена по исходным данным — историческим ценам опциона $C_{K,t}$ и базового инструмента P_p , где $K \in K$, $t \in T$. Для каждого значения K_i и t_p , $i = 1, \dots, m$, вычислялось значение предполагаемой волатильности

$$\sigma_i = \arg \min_{\sigma \in [0, 1,5]} (C_{K_p, t_i} - C(\sigma, P_{t_i}, B, K_p, t_i)),$$

где справедливая цена опциона C вычислена по формуле Блэка—Шоулза [14]. Длина истории составляет 314 отсчетов времени. Число признаков 160.

Задано множество порождающих функций $G = \{1/x, \sqrt{x}, \ln(x), \tanh(x)\}$. Регрессионная выборка была случайным образом разбита на контрольную и обучающую, равные по мощности. Значения ошибок на обучении и контроле были усреднены по 10 запускам алгоритмов на различных разбиениях.

Результаты экспериментов показаны в таблице. Для каждого алгоритма вычислялись ошибки S_L и S_C на обучении и контроле (2), значения информационных критериев Акаике $AIC = m \left(\ln \frac{S}{m} \right) + 2k$,

Байеса $BIC = m \left(\ln \frac{S}{m} \right) + k \ln m$, Маллоуза (C_p), де-

Результаты работы алгоритмов выбора признаков

Алгоритм	S_L	S_C	AIC	BIC	C_p	$\lg \kappa$	k
Генетический алгоритм	0,073	0,107	1152	1072	337	13	26
МГУА	0,146	0,194	1076	1045	745	6	10
Шаговая регрессия	0,128	0,154	1092	1055	644	7	12
Гребневая регрессия	0,111	0,146	819	330	832	33	160
Лассо	0,121	0,147	1089	1034	611	5	18
Ступенчатая регрессия	0,071	0,096	1157	1077	324	9	26
Добавление с ортогонализацией	0,106	0,135	1105	1044	527	7	20
Метод наименьших углов	0,098	0,095	1102	1017	492	7	28

сятичный логарифм числа обусловленности k матрицы значений отобранных признаков и сложность модели k .

Заключение

Рассмотренные алгоритмы можно разделить на группы: алгоритмы последовательного добавления или удаления признаков (FOS, LARS, Add, ступенчатая регрессия), алгоритмы с введением ограничения на функционал качества (гребневая регрессия и Лассо), алгоритмы перебора моделей (МГУА, генетический алгоритм, полный перебор), алгоритмы добавления и удаления признаков (шаговая регрессия, предложенные методы).

Описанные выше алгоритмы выполнены на языке MATLAB и могут быть использованы для решения задач регрессионного анализа. Библиотека алгоритмов находится по адресу <http://strijev.com/examples/mdlselection.zip> и свободно распространяется на условиях лицензии GNU GPL.

Список литературы

1. **Efroymson M. A.** Multiple regression analysis. New York: Ralston, Wiley, 1960.

2. **Akaike H.** A bayesian analysis of the minimum aic procedure // Ann. Inst. Statist. Math. 1978. Vol. 2, N 30. P. 9–15.
3. **Hocking R. R.** A biometrics invited paper the analysis and selection of variables in linear regression // Biometric. 1976. Vol. 32, N 1. P. 1–49.
4. **Mallows C. L.** Some comments on c_p // Technometrics. 1973. Vol. 15. P. 661–675.
5. **Hoerl A. E., Kennard R. W.** Ridge regression: Biased estimation for nonorthogonal problems // Technometrics. 1970. Vol. 3, N 12. P. 55–67.
6. **Bjorkstrom A.** Ridge regression and inverse problems // Tech. rep. Stockholm University, Sweden, 2001.
7. **Draper N. R., Smith H.** Appied Regression Analysis. New York: John Wihley and Sons, 1998.
8. **Hastie T., Taylor J., Tibshirani R., Walther G.** Forward stage-wise regression and the monotone lasso // Electronic Journal of Statistics. 2007. Vol. 1, N 1. P. 1–29.
9. **Malada H. R., Ivakhnenko A. G.** Inductive Learning Algorithms for Complex Systems Modeling. Boca Raton, Florida: CRC Press, 1994. 368 p.
10. **Chen Y. W., Billings C. A., Luo W.** Orthogonal least squares methods and their application to non-linear system identification // International Journal of Control. 1989. Vol. 2, N 50. P. 873–896.
11. **Tibshirani R.** Regression shrinkage and selection via the lasso // Journal of the Royal Statistical Society. 1996. Vol. 32, N 1. P. 267–288.
12. **Efron B., Hastie T., Johnstone I., Tibshirani R.** Least angle regression // The Annals of Statistics. 2004. Vol. 32, N 3. P. 407–499.
13. **Hastie T., Tibshirani R., Friedman J.** The Elements of Statistical Learning. Springer. 2001. 533 p.
14. **Hull J. C.** Options. Futures and Other Derivatives. Prentice Hall, 2000.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

УДК 004.382.2

С. Г. Бобков, д-р техн. наук, зав. отделением,
НИИ системных исследований РАН, г. Москва
e-mail: s_g_bob@mail.ru

Архитектурные и конструктивные решения вычислительных систем с производительностью 1–100 Тфлопс

Рассматриваются базовые принципы создания вычислительных систем производительностью от единиц до сотен терафлопс.

Ключевые слова: суперЭВМ, суперкомпьютерные технологии, архитектура вычислительной системы

К концу XX века потенциал инженерных технологий теорий микроскопического взаимодействия в рамках моделей сплошной среды был практически исчерпан. Дальнейшее развитие ядерной и тер-

моядерной энергетики, электроники, авиастроения, биотехнологий и т. д. стало невозможным без проведения полномасштабных инженерных расчетов сложных технических и биологических систем с учетом атомно-молекулярного взаимодействия, требующих уже в среднесрочной перспективе (до 2020 г.) использования суперЭВМ экзафлопного класса (10^{18} оп/с). По целому ряду причин технологии таких ЭВМ, скорее всего, будут иметь существенные ограничения по поставкам. Поэтому для возможности его реализации необходимо создать собственную элементную базу. Проект по созданию таких ЭВМ невозможно выполнить без опыта работ по проектированию 10-, 100- и 1000-Тфлопсных ЭВМ на собственной элементной базе. В настоящее время в России отсутствуют суперЭВМ на отечественной элементной базе, и задача создания отечественных ЭВМ производительностью 10 Тфлопс и выше является важнейшим этапом создания суперЭВМ экзафлопной производительности [1].

Одной из основных задач применения ЭВМ производительностью до 10 Тфлопс является создание машиностроительных систем предсказательного мо-

делирования и виртуального прототипирования. Оценки показали, что наиболее оптимальным вариантом являются ЭВМ производительностью порядка 3—5 Тфлопс для работы трех конструкторов. При создании таких сложных систем, как самолет или автомобиль в целом, большая часть ЭВМ должна быть компактной и не требовать дорогостоящего обслуживания, и только на конечных стадиях проектирования требуются ЭВМ с производительностью десятки и сотни терафлопс.

В настоящее время в России суперЭВМ создаются только на импортной элементной базе и с использованием импортных модулей. В статье рассматриваются основные конструктивные и архитектурные решения построения отечественных суперЭВМ, а также возможности повышения их производительности под выделенные классы задач. В основу выбора механического конструктива положено требование создания в рамках единого решения как высокопроизводительных компактных компьютеров с производительностью порядка 3 Тфлопс, так и суперкомпьютеров производительностью до сотен и единиц тысяч терафлопс.

Проведенные ранее исследования показали, что платы больших размеров имеют лучшие электрические показатели, прежде всего по распространению высокочастотных сигналов, чем платы небольших размеров (меньше 160×233 мм). Существует ряд стандартов на платы центрального процессора большой площади. Можно выделить три основных стандарта:

- ATX с размером платы 305×244 мм;
- SSI (Blade) с площадью платы $393,7 \times 269,24$ мм;
- Евромеханика 8U—12U.

Наиболее востребованным размером для больших систем из ряда Евромеханики, принятым также в системах Compact PCI 3.0, является размер 8—9U с шириной 280 мм, т. е. плата имеет размер $322,25—366,7 \times 280$ мм. В стандарте VME64-9U размер платы определен как $9U \times 400$ мм. Выбрав размеры плат стандартов ATX или SSI, разумно выбрать также соответствующие корпуса, системы охлаждения и питания. В этом случае существенно уменьшается объем работ, не нужно будет разрабатывать механические конструкции, а для производимых конструкций, как правило, имеется расчет тепловых режимов, максимальных мощностей, способов отвода теплоты. Однако такие системы рассчитаны на микропроцессоры с очень большим тепловыделением (100—200 Вт) и их небольшое

число на плате. Кроме того, с точки зрения полезного объема такие системы существенно проигрывают модульным системам Евромеханики за счет того, что требуют отдельного корпуса для каждой платы. В этом случае не выполняется требование создания компактных высокопроизводительных систем. В формате Евромеханики 9U система охлаждения делается сразу под все модули в крейте или даже в стойке, что также уменьшает объем компьютера. Возможен также кондуктивный или водяной отвод теплоты. То есть с точки зрения развития конструктив Евромеханики 9U представляется лучшим.

Платы центральных процессоров (ЦП) современных суперЭВМ являются многопроцессорными платами, основными элементами которых являются подсистема памяти и коммуникационные каналы. Рассмотрим, какое число основных узлов платы ЦП можно разместить на плате формата Евромеханики 9U. Ожидаемый размер корпуса микропроцессора 42×42 мм. Память можно расположить тремя способами: непосредственно располагать на плате микросхемы памяти, устанавливать модули DIMM и спроектировать отдельную плату памяти. Размер для панельки под DIMM на плате составляет $142 \times 7,75$ мм, этот размер также является одним из параметров, ограничивающих число микропроцессоров на плате. Если использовать дополнительную плату с микросхемами памяти, то ограничивающий размер сильно уменьшится, поскольку разъемы для подключения такой платы можно выбрать очень компактными. Более затратный вариант с точки зрения площади платы — установка микросхем памяти непосредственно на плате. Этот вариант также наименее гибок. Однако он наилучший по возможностям теплового отвода. Возможен комбинированный вариант, когда микросхемы памяти, например для ведущего микропроцессора (производящего инициализацию платы при включении питания), устанавливаются на плате центрального процессора, а для остальных микропроцессоров память устанавливается на дополнительной плате. Наиболее простым и менее трудозатратным является вариант с DIMM. В табл. 1 приведены достоинства и недостатки трех вариантов. Цифра 1 — лучший вариант, 3 — худший.

По суммарным показателям первый и третий варианты примерно одинаковы. С точки зрения эксплуатации изделий лучшим является третий вариант. При наличии в микропроцессоре двух 128-разрядных каналов памяти для первого варианта на плате

Таблица 1

Сравнительный анализ расположения ОЗУ

Варианты расположения ОЗУ	Площадь платы ЦП	Возможности теплоотвода	Гибкость	Трудозатраты	Ремонто-пригодность	Надежность	Стоимость
1. DIMM	2	3	1	1	2	3	2
2. На плате ЦП	3	1	3	2	3	1	1
3. Отдельные платы	1	2	2	3	1	2	3

ЦП Евромеханики 9U можно будет расположить максимально 12 микропроцессоров (панельки памяти можно располагать только вертикально из-за особенностей системы охлаждения), для второго варианта — 8 микропроцессоров и для третьего варианта — 12 микропроцессоров. При наличии в микропроцессоре четырех 64-разрядных подсистем памяти с кодом Хэмминга число микропроцессоров на плате уменьшается соответственно до 8, 6 и 10. В разрабатываемом микропроцессоре предполагается установить четыре 64-разрядные подсистемы памяти с кодом Хэмминга.

Важнейшим параметром, ограничивающим число микропроцессоров, является потребляемая мощность. В современных серверах устанавливаются микропроцессоры мощностью порядка 130 Вт и системные контроллеры (чипсеты) мощностью 25 Вт, соответственно мощность микропроцессора с подсистемой памяти составляет порядка 160 Вт, а для двухпроцессорного варианта — 320 Вт. В планируемом отечественном микропроцессоре при относительно невысокой частоте (порядка 1–1,5 ГГц) можно ожидать потребляемую мощность микропроцессора с подсистемой памяти на уровне 25 Вт. В этом случае для 8-процессорной платы потребляемая мощность составит 300 Вт, для 10-процессорной — 350 Вт и для 12-процессорной — 400 Вт. Число плат в крейте может быть до 21. В случае использования модулей DIMM платы придется ставить через одну позицию (slot) из-за большой высоты DIMM, т. е. возможно установить максимум 10 плат, в остальных случаях плата ЦП может быть установлена в каждой позиции (плата ОЗУ ставится на высоту мезонина, разъемы RMC). При наличии в крейте 12 процессорных плат, в каждой из которых установлено 12 микропроцессоров, мощность крейта составит 5000 Вт (с учетом мощности модуля коммутатора). Отвести такую теплоту сложно, можно считать, что системы с 12 платами, в каждой из которых по 12 микропроцессоров, являются на данный момент предельными.

С учетом сказанного конструктив Евромеханика 9U с размером платы $366,7 \times 280$ мм является наиболее оптимальным для предлагаемых суперЭВМ. С учетом оптимизации по отводу теплоты в крейте целесообразно устанавливать 10 модулей ЦП с 10 микропроцессорами в каждом модуле. Потребляемая мощность крейта составит 3700 Вт (с учетом модуля коммутатора). Пиковая производительность микропроцессора ожидается на уровне 32 Гфлопс (двойная точность), соответственно вычислительная мощность однокрейтных систем составит порядка 3,2 Тфлопс. В стойке возможно установить четыре крейта, таким образом, одностоечный вариант предлагаемого компьютера с пиковой производительностью 12,8 Тфлопс позволяет решить задачу создания современных машиностроительных систем предсказательного моделирования.

Рассмотрим возможные архитектуры наплатной коммуникационной системы. Наиболее производительной архитектурой является архитектура соединений "каждый с каждым". Однако для 10-процессорной системы этот вариант предполагает наличие в каждом микропроцессоре девяти каналов для связи микропроцессоров и один канал для дополнительных внешних связей, что в настоящее время технически неосуществимо. Обычно в современных микропроцессорах такого класса имеются три дуплексных канала шириной 4X–8X с частотой передачи до 10–20 ГГц. Можно считать, что четыре канала в каждую сторону является разумным пределом для создаваемых микропроцессоров. Соответственно на плате ЦП могут реализовываться архитектуры с использованием данных четырех каналов или архитектуры с дополнительными микросхемами коммутаторов. Простейшей архитектурой является архитектура "кольцо". Для построения такой архитектуры требуется только два канала в каждом микропроцессоре. Для микропроцессоров с четырьмя каналами возможно организовать дополнительные связи для уменьшения пути передачи информации. Например, для 6-процессорной системы возможны связи, показанные на рис. 1.

В этой структуре каждый узел связан с соседним узлом и с узлом, расположенным через один от соседнего узла. Таким образом, каждый узел соединяется с четырьмя узлами напрямую и только с одним узлом через дополнительный узел. Условно имеется одна задержка t для соединения с четырьмя узлами и две задержки t для соединения с одним узлом. При большом числе узлов, каждый из которых имеет четыре канала, можно организовывать дополнительные связи с соседними узлами и с узлами, расположенными через два узла, через три узла и т. д. Нетрудно получить число возможных узлов без нарушения симметрии для каждой из такой структуры. Если имеются дополнительные связи через один узел, то число узлов $N = 6 + 2n$, где n — целое число. Если имеются связи через два узла, то число узлов $N = 9 + 3n$, если связи устанавливаются через произвольное число i узлов, то возможное чис-

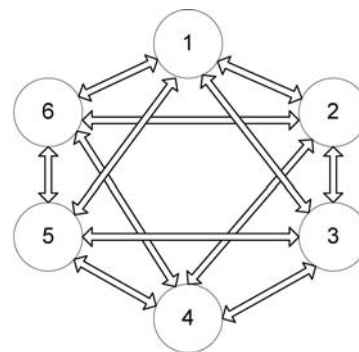


Рис. 1. Архитектура "кольцо" с дополнительными связями "через один узел"

ло узлов без нарушения симметрии: $N = 6 + 3i + (2 + i)n$, где $i = 0, 1, 2, \dots$; $n = 0, 1, 2, \dots$

Наиболее важным параметром для таких структур является время прохождения информации до наиболее удаленных узлов. Это время определяется в основном числом задержек t . Для наглядности приведены табл. 2—5, где для различных архитектур (связи "через один узел", "через два узла", ...)

Таблица 2

Распределение числа узлов с разными задержками (разным удалением от каждого узла) в системе с архитектурой "через один узел"

Число задержек t	6 узлов в системе	8 узлов в системе	10 узлов в системе	12 узлов в системе	14 узлов в системе	16 узлов в системе
1	4	4	4	4	4	4
2	1	3	4	4	4	4
3	—	—	1	3	4	4
4	—	—	—	—	1	3
5	—	—	—	—	—	—

Таблица 3

Распределение числа узлов с разными задержками (разным удалением от каждого узла) в системе с архитектурой "через два узла"

Число задержек t	9 узлов	12 узлов	15 узлов	18 узлов	21 узел	24 узла	27 узлов
1	4	4	4	4	4	4	4
2	4	5	6	6	6	6	6
3	—	2	4	5	6	6	6
4	—	—	—	2	4	5	6
5	—	—	—	—	—	2	4

Таблица 4

Распределение числа узлов с разными задержками (разным удалением от каждого узла) в системе с архитектурой "через три узла"

Число задержек t	12 узлов	16 узлов	20 узлов	24 узла	28 узлов
1	4	4	4	4	4
2	6	7	8	8	8
3	1	2	6	7	8
4	—	—	1	4	6
5	—	—	—	—	1

Таблица 5

Распределение числа узлов с разными задержками (разным удалением от каждого узла) в системе с архитектурой "через четыре узла"

Число задержек t	15 узлов	20 узлов	25 узлов	30 узлов	35 узлов	40 узлов
1	4	4	4	4	4	4
2	6	7	8	8	8	8
3	4	6	8	9	10	10
4	—	2	4	6	8	9
5	—	—	—	2	4	6
6	—	—	—	—	—	2

показано, сколько для каждого узла имеется узлов с различными задержками t .

Четыре канала в узле позволяют также строить архитектуру "решетка". Для девяти узлов на плате максимальная задержка не превышает $4t$. Видно, что она проигрывает по времени задержки архитектуре "кольцо" с дополнительными связями.

Одной из наиболее распространенных архитектур в современных серверах является группирование четверок соединений "каждый с каждым", т. е. три канала используются для связи "каждый с каждым", а дополнительные каналы используются для обмена с другими четверками. При 16 узлах на плате максимальная задержка для такой архитектуры не превышает $3t$ и остается еще четыре канала для связи с другими модулями. Такая архитектура по быстродействию примерно соответствует архитектуре "кольцо" с дополнительными связями.

Для архитектуры с дополнительной микросхемой коммутатора, имеющей число каналов, больших числа узлов, задержка в передаче данных между любыми двумя узлами составляет $2t$ (рис. 2). Для 10-процессорного модуля достаточно иметь 12-канальный коммутатор (два канала для связи с другими модулями). Данная архитектура позволяет создавать системы с наименьшими максимальными задержками, однако за счет того, что здесь отсутствуют связи с задержкой $1t$, средняя производительность на отдельных задачах может проигрывать предыдущей архитектуре. Следует также учитывать, что выход из строя коммутатора приведет к выходу всего узла, что свойственно любым архитектурам типа "звезда".

Одним из важнейших параметров коммуникационной системы является ее энергопотребление. Высокоскоростные каналы вносят существенный вклад (до 20 % в зависимости от частоты) в энергопотребление коммуникационной системы, и уменьшение числа каналов может привести к существенному снижению потребления питания. В этом отношении архитектура, представленная на рис. 2, является наилучшей из рассмотренных (число каналов в данной архитектуре почти в 2 раза меньше).

Таким образом, сравнительный анализ показывает, что наиболее оптимальной архитектурой высокопроизводительных компактных комплексов

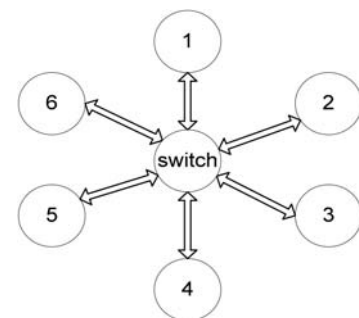


Рис. 2. Архитектура "звезда" с использованием коммутатора

является архитектура коммутируемых каналов с использованием микросхемы коммутатора. Для создания требуемых вычислительных систем целесообразно использовать конструктив Евромеханики 9U. Использование предложенных принципов позволит создать высокопроизводительные комплексы с числом микропроцессоров порядка 100 на один крейт и пиковой производительностью до 3,2 Тфлопс.

Важнейшим параметром рассматриваемых систем является приближение производительности системы на пользовательских задачах к пиковой производительности. Достижение этого возможно при:

- создании высокопроизводительной коммуникационной системы, позволяющей обеспечить требуемый межпроцессорный, межмодульный и межкрейтовый обмен информацией;
- создание быстродействующей подсистемы памяти, обеспечивающей поступление данных в микропроцессор в соответствии со скоростью обработки данных;
- минимизации задержек в микропроцессоре на выполнение требуемых функций.

Первый случай обеспечивается переходом на предельные частоты каналов обмена (в настоящее время это порядка 20 ГГц), снижением начальной задержки на передачу (latency), введением дополнительных режимов коротких (сообщений) и длинных передач, передач данных без преобразования микропроцессором их виртуальных адресов в физические, широковещательных и групповых передач и пр. Важнейшим фактором повышения производительности является минимизация задержек t , рассмотренная выше (оптимизация архитектуры коммутационной системы).

Создание быстродействующей подсистемы памяти может быть обеспечено только комплексом мер. Это введение многофункциональных режимов ускоренного обмена с памятью (DMA), предвыборки данных [2], увеличение частоты и ширины памяти, создание нескольких каналов памяти, увеличение объема и быстродействия кэш-памяти всех уровней и увеличение путей доступа к ним.

Минимизация задержек на выполнение функций реализуется в современных микропроцессорах целым рядом архитектурных решений, повышением частоты микропроцессоров, созданием многоядерных систем. Имеются две основные тенденции развития современных микропроцессоров: повышение сложности ядер микропроцессоров, позволяющее эффективно выполнять несколько операций за такт, и увеличение числа ядер микропроцессора. При достижении современных предельных технологических норм изготовления кристаллов микросхем возникают проблемы с одинаковостью ВАХ транзисторов в разных частях кристаллов с линейными размерами больше 2 мм, что приводит к ограниче-

нию на размер ядра и соответственно на его сложность.

В то же время эффективное программирование многоядерных процессоров современными средствами — весьма трудоемкая задача. Требуется создание всего комплекса системного программного обеспечения под многоядерные системы. Все в целом требует огромных финансовых вложений, недоступных российским разработчикам. Поэтому предлагается осуществлять минимизацию задержек в микропроцессоре на выполнение требуемых функций под выделенные задачи, т. е. создавать суперЭВМ под наиболее востребованные в России задачи. В этом случае возможна оптимизация архитектуры под эти задачи и достижение предельной производительности на данных задачах при сохранении универсальности самого микропроцессора. Такой подход был опробован при создании микропроцессора 1890BM5Ф [3].

Одно из основных отличий микропроцессора 1890BM5Ф от многих выпускаемых современных микропроцессоров заключается в его ориентации на задачи цифровой обработки сигналов при сохранении общей архитектуры универсального микропроцессора. В результате четыре бабочки Фурье выполняются 40 операциями, затрачивается 12 тактов микропроцессора, до шести операций вещественной арифметики выполняются за один такт. Несмотря на то, что микропроцессор наиболее эффективно выполняет задачи обработки изображений и обработки сигналов, это стандартный универсальный 64-разрядный RISC микропроцессор. В результате производительность микропроцессора поднята в 3 раза по сравнению с современными универсальными микропроцессорами. В микропроцессоре 1890BM6Я эта цифра увеличена еще в 2,5 раза.

Предложенные общие принципы создания суперЭВМ на основе отечественной элементной базы позволяют создать как настольные суперЭВМ для создания рабочих мест конструкторов, так и мощные вычислительные комплексы с производительностью сотни терафлопс. Оптимизация под требуемые задачи архитектуры микропроцессоров, коммуникационных систем и конструктивов позволяет до 10 раз поднять производительность по сравнению с используемыми принципами построения отечественных суперЭВМ в настоящее время.

Список литературы

1. Велухов Е. П., Бетелин В. Б., Кушниренко А. Г. Промышленность, инновации, образование и наука в России. М.: Наука, 2009. 141 с.
2. Бобков С. Г., Булгаков В. Ю. Влияние опережающего считывания из оперативного запоминающего устройства на скорость выполнения программ // Вопросы кибернетики НСК "Кибернетика" АН СССР. М., 1992.
3. Микропроцессор гибридный / Бобков С. Г., Аряшев С. И., Барских М. Е., Бычков К. С., Зубковский П. С. // Патент № 2359315. 2009.

Е. В. Гливенко, д-р техн. наук, проф.,

С. А. Прядко, аспирант,

А. С. Фомочкина, аспирант,

К. А. Левонян, аспирант,

РГУ нефти и газа им. И. М. Губкина, г. Москва

e-mail: sergeypryadko@gmail.com

Методы решения систем линейных и нелинейных алгебраических уравнений, обладающие естественным параллелизмом

Предлагаются новые вычислительные методы для решения систем линейных и нелинейных алгебраических уравнений, изначально обладающие естественным параллелизмом. Приводятся примеры их реализации на многопроцессорных структурах.

Ключевые слова: геометрическая интерпретация, сетка, узел

Геометрическая интерпретация задач

В настоящей статье предлагается развитие идей, сформулированных в работе [1] и связанных с организацией параллельных структур. При этом используется понятие геометрической интерпретации задач для построения приближенных вычислительных методов, изначально обладающих естественным параллелизмом.

Решение системы линейных алгебраических уравнений

Пусть задана система уравнений

$$Ax = b. \quad (1)$$

Представим ее в виде

$$x = (A + E)x - b, \quad (2)$$

где E — единичная матрица.

Представление системы в виде (2) обычно используется для применения метода итераций. Необходимым и достаточным условием сходимости итераций в этом случае является тот факт, что норма матрицы $(A + E)$ меньше единицы или, иначе говоря, матрица $(A + E)$ осуществляет сжатое отображение.

Здесь же мы используем представление (2) для решения системы, когда матрица $(A + E)$ произвольна. Введем обозначения:

$$\bar{A} = A + E; \quad \bar{b} = -b, \quad (3)$$

а отображение

$$y = \bar{A}x + \bar{b} \quad (4)$$

назовем $f(x)$.

Очевидно, что у этого отображения будет существовать единственная неподвижная точка, координаты которой составляют решение системы (1). Наша задача заключается в том, чтобы найти неподвижную точку преобразования $f(x)$.

Эту задачу мы будем решать, применяя понятие, известное в современной геометрии под названием *степень отображения* [2].

Постановка задачи

Постановка задачи в нашем случае будет сформулирована следующим образом.

Пусть в n -мерном пространстве задач есть n -мерный симплекс. Наша задача будет состоять в том, чтобы ответить на вопрос, лежит ли решение системы (1), т. е. неподвижная точка преобразования $f(x)$, внутри данного симплекса. Если окажется, что решение лежит внутри симплекса, то это значит, что мы "поймали" решение и уточнять его мы сможем последующим делением симплекса.

Использование степени отображения

Применим преобразование $f(x)$ к $(n + 1)$ точкам — вершинам нашего симплекса. Получим $(n + 1)$ векторов, начальные точки которых будут совпадать с вершинами симплекса, а конечные — с образами этих вершин.

Перенесем эти векторы в начало координат и пронормируем их. Концы построенных таким образом векторов будут располагаться на единичной сфере и определять отображение f всего симплекса на единичную сферу. Случай, соответствующий попаданию решения системы (1) внутрь симплекса, будет соответствовать тому, что степень отображения f по модулю равна единице. Если решение системы (1) не попадает внутрь симплекса, то степень отображения f равна нулю [3].

В нашем случае это будет соответствовать следующему: если все $(n + 1)$ построенные векторы лежат в одном полупространстве, то это означает, что решение системы (1) не содержится внутри симплекса, если же построенные векторы в совокупности не содержатся ни в каком полупространстве, то это значит, что решение лежит внутри симплекса.

Алгоритм обнаружения неподвижной точки преобразования $f(x)$, т. е. решения системы (1)

У n -мерного симплекса имеется $(n + 1)$ вершина: $x_1, x_2, \dots, x_n + 1$. Координаты каждой вершины подставляем в выражение $f(x)$. Получим

$$\begin{aligned} y_1 &= \bar{A}x_1 + \bar{b}; \\ y_2 &= \bar{A}x_2 + \bar{b}; \\ &\dots \\ y_{n+1} &= \bar{A}x_{n+1} + \bar{b}. \end{aligned} \quad (5)$$

В результате получим $(n + 1)$ векторов:

$$\begin{aligned}\bar{\xi}(x_1) &= y_1 - x_1; \\ \bar{\xi}(x_2) &= y_2 - x_2; \\ &\dots \\ \bar{\xi}(x_{n+1}) &= y_{n+1} - x_{n+1}.\end{aligned}\quad (6)$$

Пронормировав их, получим векторы

$$\begin{aligned}\xi(x_1) &= \frac{y_1 - x_1}{|y_1 - x_1|}; \\ \xi(x_2) &= \frac{y_2 - x_2}{|y_2 - x_2|}; \\ &\dots \\ \xi(x_{n+1}) &= \frac{y_{n+1} - x_{n+1}}{|y_{n+1} - x_{n+1}|},\end{aligned}\quad (7)$$

длины которых равны единице.

Теперь, если окажется, что векторы $\xi(x_1)$, $\xi(x_2)$, ..., $\xi(x_{n+1})$ не лежат ни в каком n -мерном полупространстве, то это значит, что решение нашей системы (точка с его координатами) лежит внутри симплекса.

Таким образом, теперь задача сводится к тому, как определить, лежат ли $(n + 1)$ векторов в n -мерном пространстве (иногда удобно проводить нормировку векторов, а иногда и нет).

Проиллюстрируем идею геометрического метода решения двух уравнений с двумя неизвестными, когда $n = 2$. Отображение $f(x)$ в этом случае выглядит так:

$$\begin{aligned}x_{1\text{нов}} &= \bar{a}_{11}x_1 + \bar{a}_{12}x_2 + \bar{b}_1; \\ x_{2\text{нов}} &= \bar{a}_{21}x_1 + \bar{a}_{22}x_2 + \bar{b}_2.\end{aligned}\quad (8)$$

Рассмотрим произвольный треугольник Q на плоскости (x_1, x_2) (рис. 1).

Нам предстоит выяснить, лежит ли точка, координаты которой являются решением системы уравнений (1), внутри треугольника. Подставляем последовательно в правую часть уравнений (8) координаты точек (x_1^1, x_2^1) , (x_1^2, x_2^2) , (x_1^3, x_2^3) . В левой части

при этом получатся координаты точек $(x_{1\text{нов}}^1, x_{2\text{нов}}^1)$,

$(x_{1\text{нов}}^2, x_{2\text{нов}}^2)$, $(x_{1\text{нов}}^3, x_{2\text{нов}}^3)$. Рассмотрим три

вектора ξ_1, ξ_2, ξ_3 , начала которых расположены в вершинах треугольника, а концы — в точках $(x_{1\text{нов}}^1, x_{2\text{нов}}^1)$, $(x_{1\text{нов}}^2, x_{2\text{нов}}^2)$, $(x_{1\text{нов}}^3, x_{2\text{нов}}^3)$

(рис. 1). Перенесем эти векторы в начало координат, а концы — на единичные окружности (рис. 2 и рис. 3). Если начальные точки векторов ξ_1, ξ_2, ξ_3 лежат в одной плоскости, то точка, координаты которой являются решением системы (8), лежит вне

треугольника Q . Если же не найдется никакой полуплоскости, в которой бы лежали все векторы ξ_1, ξ_2, ξ_3 , то это значит, что точка, координаты которой удовлетворяют системе уравнений (1), лежит внутри треугольника Q , т. е. мы "поймали" решение.

На рис. 2 векторы ξ_1, ξ_2, ξ_3 не лежат все ни в одной полуплоскости. На рис. 2, а — это явная ситуация, на рис. 2, б — сомнительная. На рис. 3 векторы ξ_1, ξ_2, ξ_3 лежат в одной полуплоскости, на рис. 3, а — это явная ситуация, на рис. 3, б — сомнительная.

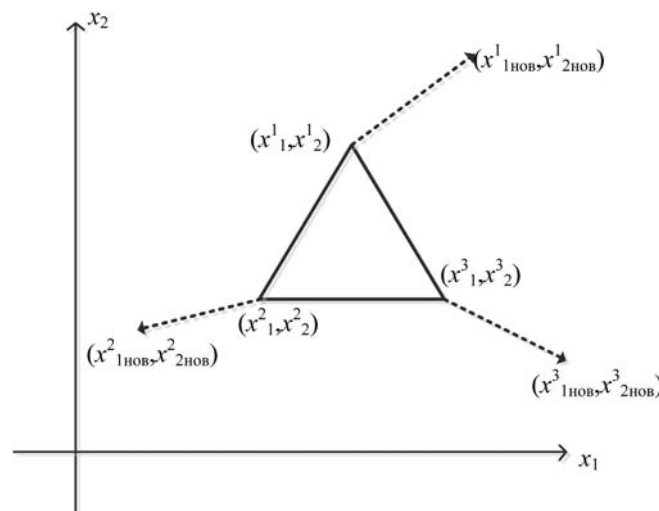


Рис. 1. Двухмерный симплекс на плоскости

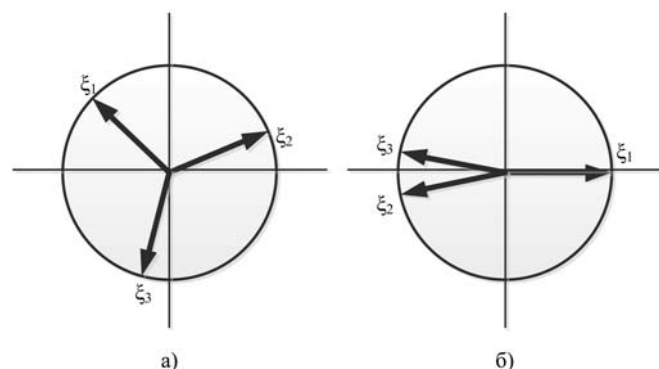


Рис. 2. Расположение векторов в разных полуплоскостях: а — явное; б — сомнительное

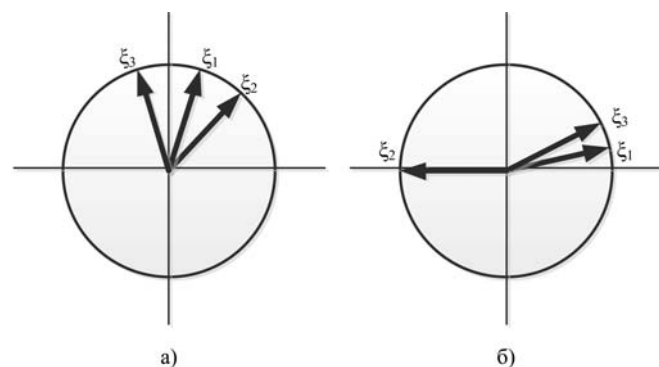


Рис. 3. Расположение векторов в одной полуплоскости: а — явное; б — сомнительное

Метод размножения

Предлагается методика, которая позволяет выяснить, лежат ли векторы в одном полупространстве или нет.

Рассмотрим набор ортогональных координатных линий, в рамках которого рассматривается система линейных алгебраических уравнений. На каждой из этих линий выделим по два единичных вектора с противоположными направлениями. Всего таких векторов будет $2n$. Обозначим их z_j ($j = 1, 2, \dots, 2n$).

Рассмотрим скалярные произведения (z_j, ξ_i) для всевозможных пар (j, i) . Построим функцию $\Phi(j, i)$ следующим образом:

$$\Phi(j, i) = \begin{cases} (z_j, \xi_i), (z_j, \xi_i) \geq 0; \\ 0, (z_j, \xi_i) < 0. \end{cases} \quad (9)$$

Первый этап метода. Вычисляем последовательно $\Phi(1, i)$ для всех $i = 1, 2, n+1$. Находим $\max_i \Phi(1, i)$.

Затем вычисляем $\Phi(2, i)$ для всех $i = 1, 2, \dots, n+1$ и находим $\max_i \Phi(2, i)$.

Далее вычисляем $\Phi(j, i)$ для всех $i = 1, 2, \dots, n+1$ и находим $\max_i \Phi(j, i)$ для всех $j = 1, 2, \dots, 2n$.

Затем находим минимум этих $2n$ максимумов и получаем

$$V = \min_j \max_i \Phi(j, i). \quad (10)$$

Второй этап метода. Выполним процесс "размножения", заключающийся в следующем. Построим попарно различные суммы векторов

$$\begin{aligned} \bar{W}_k &= \xi_l + \xi_m, \\ l &\neq m, \\ l &= 1, 2, \dots, n+1, \\ m &= 1, 2, \dots, n+1, \\ k &= 1, 2, \dots, \frac{n(n+1)}{2}, \end{aligned} \quad (11)$$

которые после нормировки обозначим

$$W_k = \frac{\bar{W}_k}{|\bar{W}_k|}. \quad (12)$$

Для этих векторов W_k построим функции

$$\bar{\Phi}(j, i) = \begin{cases} (z_j, W_i), (z_j, W_i) \geq 0; \\ 0, (z_j, W_i) < 0 \end{cases} \quad (13)$$

аналогично тому, как мы поступили с векторами ξ_i , строя функции $\Phi(j, i)$.

Аналогично предыдущей процедуре построим минимакс $\bar{V}$ следующим образом:

$$\begin{aligned} \bar{V} &= \min_j \max_k \bar{\Phi}(j, k), \\ j &= 1, 2, \dots, n, \\ k &= 1, 2, \dots, \frac{n(n+1)}{2}. \end{aligned} \quad (14)$$

Признаком того, что векторы лежат в одном полупространстве, является тот факт, что

$$\bar{V} \leq V. \quad (15)$$

В противном случае, т. е. в том случае, когда решение системы находится внутри симплекса (мы "поймали" решение),

$$\bar{V} > V. \quad (16)$$

Для численного эксперимента была использована моделирующая программа [4], позволяющая для различных систем линейных алгебраических уравнений и различных симплексов, как содержащих точку — решение системы, так и не содержащих ее, вычислять значения V и $\bar{V}$ для векторов

$$\begin{aligned} \xi_1 &= \frac{y_1 - x_1}{|y_1 - x_1|}; \\ \xi_2 &= \frac{y_2 - x_2}{|y_2 - x_2|}; \\ &\dots \\ \xi_{n+1} &= \frac{y_{n+1} - x_{n+1}}{|y_{n+1} - x_{n+1}|}. \end{aligned} \quad (17)$$

Результаты численного эксперимента показывают, что выделенный признак обладает достаточной эффективностью при $n > 7$. Эффективность признака возрастает с возрастанием n — размерности пространства или числа уравнений системы. Численный эксперимент был проведен до $n = 90$.

Решение системы нелинейных алгебраических уравнений

Метод геометрической интерпретации для решения систем нелинейных уравнений. Рассмотрим пример. Пусть задана система, состоящая из двух квадратных уравнений:

$$\begin{cases} A_1x^2 + B_1y^2 + C_1xy + D_1x + E_1y + H_1 = 0; \\ A_2x^2 + B_2y^2 + C_2xy + D_2x + E_2y + H_2 = 0. \end{cases} \quad (18)$$

Рассмотрим две функции от двух переменных

$$\begin{aligned} F_1(x, y) &= A_1x^2 + B_1y^2 + C_1xy + D_1x + E_1y + H_1; \\ F_2(x, y) &= A_2x^2 + B_2y^2 + C_2xy + D_2x + E_2y + H_2. \end{aligned} \quad (19)$$

Уравнения системы (18) определяют линии уровня для функций $F_1(x, y)$ и $F_2(x, y)$, где соответствующие функции равны нулю. Решения этой системы будут координатами точек, являющихся пересечением линий уровня (рис. 4).

Рассмотрим поведение функций $F_1(x, y)$, $F_2(x, y)$ в окрестности их линий уровня (рис. 4). Пусть функция $F_1(x, y)$ больше нуля вне эллипса, а меньше нуля — внутри него. Это значит, что в окрестности эллипса, где $F_1(x, y) = 0$, найдутся как точки, где $F_1(x, y) > 0$, так и точки, где $F_1(x, y) < 0$. То же самое справедливо для функции $F_2(x, y)$. Пусть вне

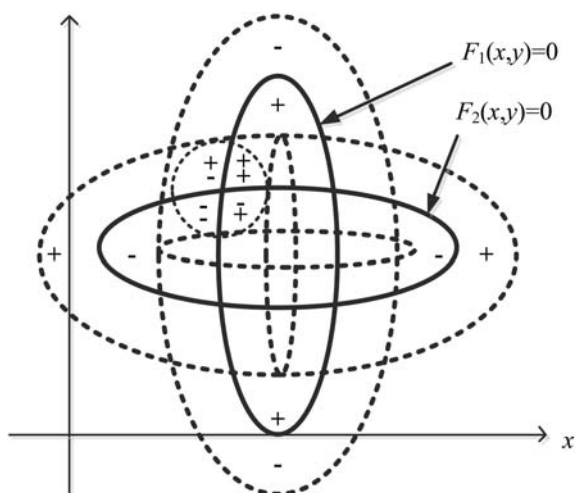


Рис. 4. Геометрическая интерпретация решения систем нелинейных уравнений

второго эллипса она меньше нуля, а внутри него — больше нуля. Это значит, что в окрестности эллипса, где $F_2(x, y) = 0$, найдутся как точки, где $F_2(x, y) > 0$, так и точки, где $F_2(x, y) < 0$.

Рассмотрим точки пересечения эллипсов, т. е. те точки, координаты которых являются решениями системы (18). В окрестности этих точек найдутся точки четырех типов:

$$\text{1-й тип точки, где } \begin{cases} F_1(x, y) > 0; \\ F_2(x, y) > 0; \end{cases} \quad (20)$$

$$\text{2-й тип точки, где } \begin{cases} F_1(x, y) > 0; \\ F_2(x, y) < 0; \end{cases} \quad (21)$$

$$\text{3-й тип точки, где } \begin{cases} F_1(x, y) < 0; \\ F_2(x, y) > 0; \end{cases} \quad (22)$$

$$\text{4-й тип точки, где } \begin{cases} F_1(x, y) < 0; \\ F_2(x, y) < 0; \end{cases} \quad (23)$$

а именно окрестность точки-решения разделится на четыре области, каждая из которых состоит из точек одного типа.

Метод геометрической интерпретации состоит в отыскании точек-решений по этому признаку.

Решение системы на многопроцессорном компьютере. Использование многопроцессорного компьютера для нахождения точек-решений состоит в следующем.

Объясним это на примере, который был разобран выше. На плоскости (x, y) построим набор точек с координатами (x_i, y_j) ($i = 1, 2, \dots, N; j = 1, 2, \dots, N$). Это будут узлы некоторой сетки. В каждом узле вычисляются значения функций $F_1(x_i, y_j)$, $F_2(x_i, y_j)$.

Эти вычисления на многопроцессорной машине выполняются параллельно разными процессорами для разных точек. Далее нам предстоит выделить области, в которых найдутся узлы всех четырех типов (20)—(23).

По нашему критерию, если область содержит хотя бы по одному узлу из каждого типа, значит эта область содержит точку-решение. Если при этом наборе узлов данное условие не выполняется, строим набор узлов в другом месте. Если мы нашли область, содержащую решение, то уточнение решения будем проводить делением этой области на более мелкие подобласти и повторять процедуру в каждой области.

Алгоритм решения

Алгоритм решения поставленной задачи, т. е. решения системы нелинейных уравнений от n неизвестных, основан на методике геометрической интерпретации задачи, которая при использовании многопроцессорных компьютеров обеспечивает хорошее распараллеливание, а именно позволяет эффективно использовать процессоры в каждом такте.

Рассмотрим предлагаемый алгоритм при решении системы из двух нелинейных уравнений

$$\begin{cases} F_1(x, y) = 0; \\ F_2(x, y) = 0. \end{cases} \quad (24)$$

Представим рассматриваемую область плоскости (x, y) в виде набора узлов, отстоящих друг от друга на некоторое расстояние Δx и Δy (например, шага). В каждом из узлов вычисляем вначале $F_1(x, y)$ и обращаем внимание на знаки в каждом узле.

В том случае, если во всех узлах знаки одинаковые и параметр шага достаточно мал, то мы дальше эту область можем не рассматривать как потенциально содержащую точку-решение. Если же одинаковые знаки у нас окажутся при достаточно большом параметре шага, то уменьшим параметр шага, т. е. перейдем к более мелкой сетке узлов.

Пусть в ряде узлов у нас появится знак "плюс", а в других — знаки "минус". Тогда поделим рассматриваемую область на более мелкие подобласти, среди которых нас особенно будут интересовать те, в которых обнаруживаются как "плюсы", так и "минусы". Эти подобласти будут кандидатами на содержание внутри себя кривой $F_1(x, y) = 0$. Дальнейшая работа будет идти именно с этими областями.

Теперь ту же процедуру повторим, вычисляя в узлах сетки значения $F_2(x, y)$. Для $F_2(x, y)$ также выделим области, содержащие как узлы, где $F_2(x, y) > 0$, так и узлы, где $F_2(x, y) < 0$.

Рассматривая пересечения потенциальных областей для $F_1(x, y) = 0$ и для $F_2(x, y) = 0$, получим области, содержащие решения системы, т. е. точки, где как $F_1(x, y) = 0$, так и $F_2(x, y) = 0$.

Если у нас имеется n уравнений с n неизвестными, то процедура будет аналогичной описанной. Области в этом случае будут областями n -мерного пространства, а слово "кривые" мы заменим словами "гиперповерхностями".

При использовании многопроцессорных компьютеров, следуя идеям геометрической интерпре-

тации задач, поручаем отдельному процессору работу с одним из узлов. Это позволит анализировать области большого размера и эффективно использовать многопроцессорность компьютера.

Решение системы из трех уравнений на персональном компьютере. Создана компьютерная система с интерфейсом, реализованная на персональном компьютере для решения систем трех квадратных уравнений с тремя неизвестными. Идея этой программы та же, что и описанная выше, но обзор рассматриваемой области выполняется на персональном компьютере. Таким образом, последовательно перебираются узлы двумерной (или уже в этом случае трехмерной) сетки. И в этих узлах последовательно вычисляются значения $F_1(x_i, y_j, z_k)$, $F_2(x_i, y_j, z_k)$ и $F_3(x_i, y_j, z_k)$ ($i, j, k = 1, \dots, N$). Интерфейс позволяет вводить исходную систему уравнений с экрана.

Система, реализованная на персональном компьютере, использовалась для определения предельного дебита фонтанирующей скважины при течении двухфазной жидкости [5].

Список литературы

1. Гливенко Е. В. Параллельный процессор первичной обработки информации. М.: Радио и связь, 1992. 104 с.
2. Дубровин Б. А., Новиков С. П., Фоменко А. Т. Современная геометрия. М.: Наука, 1979. 760 с.
3. Александров П. С. Комбинаторная топология. М.-Л.: ОГИЗ, 1947. 660 с.
4. Бухарова Н. И., Гливенко Е. В., Демьянов В. Л. Параллельный метод геометрической интерпретации в интерактивном режиме // Вопросы радиоэлектроники. Сер. ЭВМ. 1999. № 1.
5. Гливенко Е. В., Мухтарулин В. С., Петрова Г. Н. Параллельные вычисления и геометрическая интерпретация задач. М.: Нефть и Газ, 2006. 111 с.

УДК 004.3.047

Д. В. Андреев, д-р техн. наук, проф.,
Ульяновский государственный
технический университет,
e-mail: andreev@ulstu.ru

Аналоговые мультиплексоры для логической обработки континуальных данных

Представлены два метода структурной организации аналоговых мультиплексоров, основанные на предложенных процедурах разложения функций бесконечнозначной логики и дающие более простые по сравнению с известными схемные решения. Рассмотрены особенности настройки этих мультиплексоров на воспроизведение бесконечнозначных логических функций соответствующего класса.

Ключевые слова: бесконечнозначная логика, функциональная декомпозиция, универсальные логические модули

Введение

Развитие электронных средств автоматизации, вычислительной техники и систем управления связано с решением проблемы обеспечения универсальности указанных средств. Наличие универсальности дает пользователю возможность самостоятельно осуществлять настройку используемого стандартного средства сообразно своему проекту.

В результате решения данной проблемы применительно к цифровой технике появились микропро-

цессорные средства и средства на ПЛИС, развитию которых уделялось и продолжает уделяться большое внимание. При этом все чаще отмечается быстрый и значительный рост числа внутрикристалльных межсоединений, который неблагоприятно влияет на распределение ресурса внутрикристалльного пространства, и числа внешних выводов, который затрудняет практическое использование упомянутых средств. Устранить такие недостатки можно за счет повышения информационного содержания сигналов, действующих на внешних выводах и внутри кристалла. Однако логическая обработка таких сигналов потребует применения соответствующих методов и средств, основанных на математическом аппарате расширенных логик и схемотехнике многозначных и бесконечнозначных автоматов.

Разработкой таких методов и средств занимались многие исследователи, но проблеме обеспечения универсальности многозначных и бесконечнозначных автоматов должного внимания уделено не было. Здесь наиболее значимым результатом [1] является разработка математических моделей и схемных решений аналоговых мультиплексоров, универсальных в полных классах логических функций от заданного числа аргументов. Отметим, что упомянутые модели сформированы за счет суперпозиции базовых операций предикатной алгебры выбора.

Аппаратурная сложность указанных мультиплексоров вызывает необходимость разработки новых принципов их более простой структурной организации. В статье задача такой разработки решается на основе декомпозиционных построений, опирающихся на аппарат бесконечнозначной логики [2].

Разложения произвольной функции бесконечнозначной логики и их схемная реализация

Пусть $X = [x_{\min}, x_{\max}]$ — замкнутый интервал с серединой $x_0 = 0,5(x_{\min} + x_{\max})$. Основные операции бесконечнозначной логики (БЛ) определяются на X в виде $x_1 \vee x_2 = \max(x_1, x_2)$ — БЛ-дизъюнкция, $x_1 \wedge x_2 = \min(x_1, x_2)$ — БЛ-конъюнкция, $\bar{x} = 2x_0 - x$ — БЛ-инверсия. При $x_1, x_2, x \in \{0, 1\}$, $x_{\min} = 0$, $x_{\max} = 1$ либо $x_1, x_2, x \in \{0, \dots, k-1\}$, $x_{\min} = 0$, $x_{\max} = k-1$ указанные БЛ-операции вырождаются в одно-

Таблица 1

Табличное представление БЛ-функции $f(x_1, x_2) = x_1 \bar{x}_2 \vee \bar{x}_1 x_2$

Вариант упорядочения	$f(x_1, x_2)$	Вариант упорядочения	$f(x_1, x_2)$
$x_1 \leq x_2 \leq \bar{x}_2 \leq \bar{x}_1$	x_2	$\bar{x}_1 \leq x_2 \leq \bar{x}_2 \leq x_1$	$\bar{x}_2$
$\bar{x}_1 \leq \bar{x}_2 \leq x_2 \leq x_1$	$\bar{x}_2$	$x_1 \leq \bar{x}_2 \leq x_2 \leq \bar{x}_1$	x_2
$\bar{x}_2 \leq x_1 \leq \bar{x}_1 \leq x_2$	$\bar{x}_1$	$\bar{x}_2 \leq \bar{x}_1 \leq x_1 \leq x_2$	$\bar{x}_1$
$x_2 \leq \bar{x}_1 \leq x_1 \leq \bar{x}_2$	x_1	$x_2 \leq x_1 \leq \bar{x}_1 \leq \bar{x}_2$	x_1

Таблица 2

Табличное представление произвольной БЛ $_{\vee, \wedge}$ -функции $f(x_1, x_2, x_3, x_4)$

i	Вариант упорядочения	$f(x_1, x_2, x_3, x_4)$
1	$x_1 \leq x_2 \leq x_3 \leq x_4$	$f((x^{(1)})_4, (x^{(1)})_3, (x^{(1)})_2, x_4)$
2	$x_1 \leq x_2 \leq x_4 \leq x_3$	$f((x^{(1)})_4, (x^{(1)})_3, (x^{(2)})_2, x_4)$
3	$x_1 \leq x_3 \leq x_2 \leq x_4$	$f((x^{(1)})_4, (x^{(2)})_3, (x^{(1)})_2, x_4)$
4	$x_1 \leq x_4 \leq x_2 \leq x_3$	$f((x^{(1)})_4, (x^{(2)})_3, (x^{(2)})_2, x_4)$
5	$x_1 \leq x_3 \leq x_4 \leq x_2$	$f((x^{(1)})_4, (x^{(3)})_3, (x^{(1)})_2, x_4)$
6	$x_1 \leq x_4 \leq x_3 \leq x_2$	$f((x^{(1)})_4, (x^{(3)})_3, (x^{(2)})_2, x_4)$
7	$x_3 \leq x_1 \leq x_2 \leq x_4$	$f((x^{(2)})_4, (x^{(2)})_3, (x^{(1)})_2, x_4)$
8	$x_4 \leq x_1 \leq x_2 \leq x_3$	$f((x^{(2)})_4, (x^{(2)})_3, (x^{(2)})_2, x_4)$
9	$x_3 \leq x_1 \leq x_4 \leq x_2$	$f((x^{(2)})_4, (x^{(3)})_3, (x^{(1)})_2, x_4)$
10	$x_4 \leq x_1 \leq x_3 \leq x_2$	$f((x^{(2)})_4, (x^{(3)})_3, (x^{(2)})_2, x_4)$
11	$x_2 \leq x_1 \leq x_3 \leq x_4$	$f((x^{(2)})_4, (x^{(1)})_3, (x^{(1)})_2, x_4)$
12	$x_2 \leq x_1 \leq x_4 \leq x_3$	$f((x^{(2)})_4, (x^{(1)})_3, (x^{(2)})_2, x_4)$
13	$x_2 \leq x_3 \leq x_1 \leq x_4$	$f((x^{(3)})_4, (x^{(1)})_3, (x^{(1)})_2, x_4)$
14	$x_2 \leq x_4 \leq x_1 \leq x_3$	$f((x^{(3)})_4, (x^{(1)})_3, (x^{(2)})_2, x_4)$
15	$x_3 \leq x_2 \leq x_1 \leq x_4$	$f((x^{(3)})_4, (x^{(2)})_3, (x^{(1)})_2, x_4)$
16	$x_4 \leq x_2 \leq x_1 \leq x_3$	$f((x^{(3)})_4, (x^{(2)})_3, (x^{(2)})_2, x_4)$
17	$x_3 \leq x_4 \leq x_1 \leq x_2$	$f((x^{(3)})_4, (x^{(3)})_3, (x^{(1)})_2, x_4)$
18	$x_4 \leq x_3 \leq x_1 \leq x_2$	$f((x^{(3)})_4, (x^{(3)})_3, (x^{(2)})_2, x_4)$
19	$x_2 \leq x_3 \leq x_4 \leq x_1$	$f((x^{(4)})_4, (x^{(1)})_3, (x^{(1)})_2, x_4)$
20	$x_2 \leq x_4 \leq x_3 \leq x_1$	$f((x^{(4)})_4, (x^{(1)})_3, (x^{(2)})_2, x_4)$
21	$x_3 \leq x_2 \leq x_4 \leq x_1$	$f((x^{(4)})_4, (x^{(2)})_3, (x^{(1)})_2, x_4)$
22	$x_4 \leq x_2 \leq x_3 \leq x_1$	$f((x^{(4)})_4, (x^{(2)})_3, (x^{(2)})_2, x_4)$
23	$x_3 \leq x_4 \leq x_2 \leq x_1$	$f((x^{(4)})_4, (x^{(3)})_3, (x^{(1)})_2, x_4)$
24	$x_4 \leq x_3 \leq x_2 \leq x_1$	$f((x^{(4)})_4, (x^{(3)})_3, (x^{(2)})_2, x_4)$

именные операции соответственно двузначной либо k -значной логики.

Любая дающая отображение $X^n \rightarrow X$ функция в форме суперпозиции конечного числа основных БЛ-операций, примененных к аргументам $x_1, \dots, x_n \in X$ (континуальным переменным) является БЛ-функцией. Таким образом, всякая БЛ-функция принимает значение одного из своих аргументов или его отрицания и может быть задана таблицей (см., например, табл. 1), в которой перечисляют все $2^n n!$ вариантов упорядочения $\hat{x}_{i1} \leq \dots \leq \hat{x}_{in} \leq x_0 \leq \hat{x}_{in} \leq \dots \leq \hat{x}_{i1}$ ($i1, \dots, in$ — перестановка чисел $1, \dots, n$) аргументов $x_1, \dots, x_n$ и их отрицаний $\bar{x}_1, \dots, \bar{x}_n$ с указанием для каждого варианта того аргумента (или его отрицания), которому равна задаваемая БЛ-функция [3]. Здесь $\hat{x}_{ij} = x_{ij}$ либо $\hat{x}_{ij} = \bar{x}_{ij}$, причем в первом случае $\hat{x}_{ij} = \bar{x}_{ij}$, а во втором — $\hat{x}_{ij} = x_{ij}$.

Нетрудно усмотреть, что система $\{\vee, \wedge, -\}$ является полной в классе всех БЛ-функций. При этом система $\{\vee, \wedge\}$ обладает полнотой для класса БЛ-функций, сохраняющих значение одного из своих аргументов $x_1, \dots, x_n \in X$. Любую функцию последнего класса (БЛ $_{\vee, \wedge}$ -функцию) можно задать таблицей (см., например, табл. 2), которая сопоставляет каждому из $n!$ вариантов упорядочения $x_{i1} \leq \dots \leq x_{in}$ ($i1, \dots, in$ — перестановка чисел $1, \dots, n$) аргументов одну из континуальных переменных $x_1, \dots, x_n$.

В табл. 2 $f((x^{(r_{i1})})_4, (x^{(r_{i2})})_3, (x^{(r_{i3})})_2, x_4) \in \{x_1, x_2, x_3, x_4\}$ — значение функции $f(x_1, x_2, x_3, x_4)$ при i -м ($i = \overline{1, 4!}$) варианте упорядочения ее аргументов, который определяется набором $x_1 = (x^{(r_{i1})})_4$, $x_2 = (x^{(r_{i2})})_3$, $x_3 = (x^{(r_{i3})})_2$, характеризующим переменные x_g ($g = \overline{1, 3}$) и x_4 как соответственно r_{ig} -ю по величине среди $5 - g$ переменных $x_g, \dots, x_4$ и r_{i4} -ю по величине среди двух переменных x_3, x_4 ($r_{i1} \in \{1, 2, 3, 4\}$, $r_{i2} \in \{1, 2, 3\}$, $r_{i3}, r_{i4} \in \{1, 2\}$, $r_{i3} \neq r_{i4}$).

Введем аналогичное разложению Шеннона двузначной логической функции по одной переменной разложение БЛ $_{\vee, \wedge}$ -функции $f(x_1, \dots, x_n)$:

$$f(x_1, \dots, x_n) = \gamma_1(x_j)_n f(x_1, \dots, x_{j-1}, (x^{(1)})_n, x_{j+1}, \dots, x_n) + \dots + \gamma_n(x_j)_n f(x_1, \dots, x_{j-1}, (x^{(n)})_n, x_{j+1}, \dots, x_n), \quad (1)$$

где

$$\gamma_r(x_j)_n = \begin{cases} 1 & \text{при } x_j = (x^{(r)})_n; \\ 0 & \text{при } x_j \neq (x^{(r)})_n \end{cases} \quad (2)$$

есть характеристическая функция, равная единице либо нулю, когда переменная x_j — соответственно

r -я либо не r -я по величине среди переменных, от которых зависит разлагаемая функция ($r = \overline{1, n}$, $(x^{(1)})_n \leq \dots \leq (x^{(n)})_n$; $\gamma_1(x_j)_n + \dots + \gamma_n(x_j)_n = 1$).

Справедливость равенства (1) обосновывается следующим образом. Когда $x_j = (x^{(r)})_n$ ($r \in \{1, \dots, n\}$), в правой части этого равенства получим $f(x_1, \dots, x_{j-1}, (x^{(r)})_n, x_{j+1}, \dots, x_n)$, но точно такое же значение имеет и его левая часть.

Характеристическая функция $\gamma_r(x_j)_n$ представима в Σ — Π форме вида [4]

$$\gamma_r(x_j)_n = \sum_{k=1}^N \langle \xi_{x_1}(x_j) \dots \xi_{x_{j-1}}(x_j) \xi_{x_{j+1}}(x_j) \dots \xi_{x_n}(x_j) \rangle_k, \quad (3)$$

где $x_1, \dots, x_n$ — переменные, от которых зависит разлагаемая функция $f(x_1, \dots, x_n)$. Формула (3) содержит $N = C_{n-1}^{r-1}$ (число сочетаний из $n-1$ по $r-1$)

неповторяющихся произведений $\xi_{x_1}(x_j) \dots \xi_{x_{j-1}}(x_j) \times \xi_{x_{j+1}}(x_j) \dots \xi_{x_n}(x_j)$, каждое из которых состоит из $r-1$ инверсных ($\xi(x_j) = \bar{\xi}(x_j)$) и $n-r$ неинверсных ($\xi(x_j) = \xi(x_j)$) характеристических функций, причем

$$\xi_{x_m}(x_j) = \begin{cases} 1 & \text{при } x_j \leq x_m; \\ 0 & \text{при } x_j > x_m. \end{cases} \quad (4)$$

Пример 1. Пусть $n = j = 4$, тогда имеем

$$\gamma_1(x_4)_4 = \xi_{x_1}(x_4) \xi_{x_2}(x_4) \xi_{x_3}(x_4);$$

$$\gamma_2(x_4)_4 = \bar{\xi}_{x_1}(x_4) \xi_{x_2}(x_4) \xi_{x_3}(x_4) + \xi_{x_1}(x_4) \bar{\xi}_{x_2}(x_4) \xi_{x_3}(x_4) + \xi_{x_1}(x_4) \xi_{x_2}(x_4) \bar{\xi}_{x_3}(x_4);$$

$$\gamma_3(x_4)_4 = \bar{\xi}_{x_1}(x_4) \bar{\xi}_{x_2}(x_4) \xi_{x_3}(x_4) + \bar{\xi}_{x_1}(x_4) \xi_{x_2}(x_4) \bar{\xi}_{x_3}(x_4) + \xi_{x_1}(x_4) \bar{\xi}_{x_2}(x_4) \bar{\xi}_{x_3}(x_4);$$

$$\gamma_4(x_4)_4 = \bar{\xi}_{x_1}(x_4) \bar{\xi}_{x_2}(x_4) \bar{\xi}_{x_3}(x_4).$$

Выражение (1) назовем формулой разложения БЛ $_{\vee, \wedge}$ -функции $f(x_1, \dots, x_n)$ по характеристическим функциям γ переменной x_j . Поскольку выбор переменной x_j может быть произвольным, будем считать, что разложение выполняется, начиная с переменной x_1 и далее по порядку.

Последовательно применяя разложение вида (1) к остаточным от указанного разложения БЛ $_{\vee, \wedge}$ -функ-

циям, зависящим от меньшего числа переменных, получим

$$f(x_1, \dots, x_n) = \sum_{i=1}^{n!} f((x^{(r_{i1})})_n, \dots, (x^{(r_{i(n-1)})})_2, x_n) \gamma_{r_{i1}}(x_1)_n \dots \gamma_{r_{i(n-1)}}(x_{n-1})_2, \quad (5)$$

где $r_{iv} \in \{1, \dots, n+1-v\}$ ($v = \overline{1, n-1}$); $f((x^{(r_{i1})})_n, \dots, (x^{(r_{i(n-1)})})_2, x_n) \in \{x_1, \dots, x_n\}$ — значение БЛ $_{\vee, \wedge}$ -функции $f(x_1, \dots, x_n)$ при i -м варианте упорядочения ее аргументов $x_1, \dots, x_n$, который определяется набором $x_1 = (x^{(r_{i1})})_n, \dots, x_{n-1} = (x^{(r_{i(n-1)})})_2$, характеризующим переменные x_v и x_n как соответственно r_{iv} -ю по величине среди $n+1-v$ переменных $x_v, \dots, x_n$ и r_{in} -ю по величине среди двух переменных x_n-1, x_n ($r_{in} \in \{1, 2\}$, $r_{in} \neq r_{i(n-1)}$).

Нетрудно усмотреть, что любая n -арная БЛ $_{\vee, \wedge}$ -функция представима выражением (5), полученным в результате введенного разложения по характеристическим функциям γ переменных $x_1, \dots, x_{n-1}$. Следовательно, формула (1), определяющая процедуру такого разложения, может быть использована в качестве математической модели логического

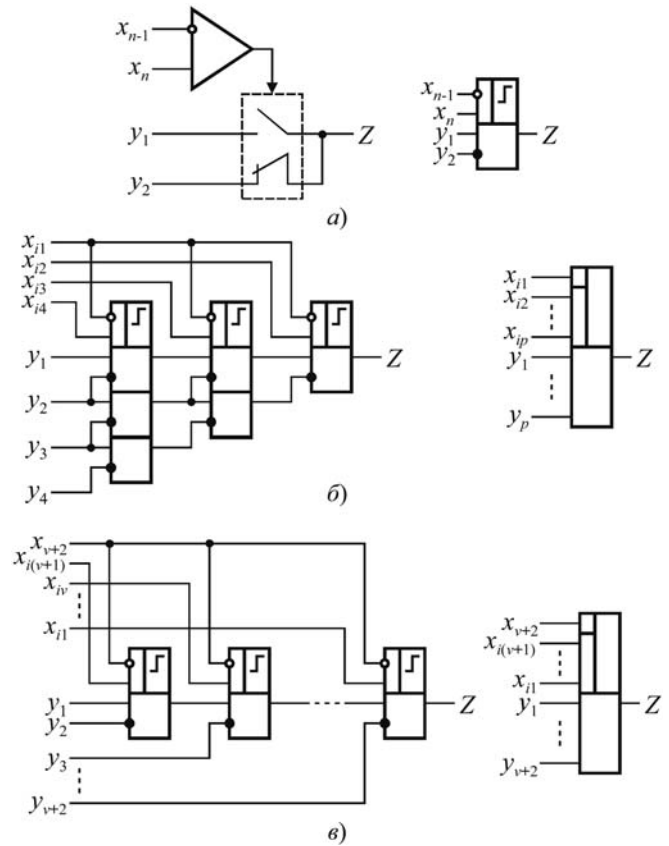


Рис. 1. Базисные элементы аналоговых мультиплексоров

модуля (мультиплексора), универсального в классе всех $БЛ_{\vee, \wedge}$ -функций от n аргументов.

Для построения указанного модуля потребуется $n - 1$ типов логических элементов, причем v -й тип должен реализовывать разложение вида (1) по характеристическим функциям γ переменной x_v , выбранной из $p = n + 1 - v$ континуальных переменных $x_v, \dots, x_n$.

При $v = n - 1$ поставленному условию удовлетворяет предложенный в работе [4] релятор (рис. 1, а), для которого

$$Z = y_1 \xi_{x_n}(x_{n-1}) + y_2 \bar{\xi}_{x_n}(x_{n-1}),$$

где y_1, y_2, x_{n-1}, x_n — аналоговые сигналы (напряжения); $\xi_{x_n}(x_{n-1})$ — характеристическая функция вида (4).

При $v = \overline{1, n-2}$ поставленному условию удовлетворяет построенный в базе реляторов ранговый идентификатор [4]. На рис. 1, б изображена его схема при $p = 4$. Воспроизводимая идентификатором операция имеет вид

$$Z = \gamma_1(x_{i1})_p y_1 + \dots + \gamma_p(x_{i1})_p y_p,$$

где $\gamma_r(x_{i1})_p$ — характеристическая функция (2), определяемая по формуле (3), в которой $r = \overline{1, p}$, $\{x_1\} \cup \dots \cup \{x_{j-1}\} \cup \{x_{j+1}\} \cup \dots \cup \{x_n\} = \{x_{i2}, \dots, x_{ip}\}$; $x_{i1}, \dots, x_{ip}, y_1, \dots, y_p$ — аналоговые сигналы (напряжения). Здесь $i1 = v, i2 = v + 1, \dots, ip = n$.

На рис. 2 представлена схемная реализация в базе реляторов и ранговых идентификаторов разложения (1) по характеристическим функциям γ переменных $x_1, \dots, x_{n-1}$ (здесь $n = 4$). Схема (рис. 2) является схемой логического модуля, универсального в классе всех $БЛ_{\vee, \wedge}$ -функций от n аргументов. Настройка мультиплексора $F_1(n)$ (рис. 2) на воспроизведение любой заданной $БЛ_{\vee, \wedge}$ -функции $f(x_1, \dots, x_n)$ осуществляется фиксацией значений этой функции на соответствующих переключательных входах реляторов.

Сложность $L(F_1(n))$, глубина $H(F_1(n))$ и число $G(F_1(n))$ настроечных входов мультиплексора $F_1(n)$ со-

ставляют $L(F_1(n)) = n - 1 + \sum_{q=1}^{n-1} 2q + \sum_{q=2}^{n-1} \frac{n!}{q!} \left(q - 1 + \sum_{w=1}^{q-1} 2w \right)$ компараторов, ключей, $H(F_1(n)) = 2$

и $G(F_1(n)) = n!$. Приведенное значение глубины обусловлено тем, что максимальное время задержки распространения сигнала от входа к выходу в схеме (рис. 2) определяется выражением $\tau = \tau_K + \tau_{KL}$, где τ_K и τ_{KL} — время задержки компаратора и ключа.

На одноименных компараторных входах элементов, образующих в мультиплексоре $F_1(n)$ вертикаль-

ную группу, действуют одинаковые сигналы. Это означает, что принцип управления переключательными входами в элементах указанной группы одинаков, и последние могут быть объединены с удалением управляющей части во всех, кроме одного, элементах объединяемой группы. На рис. 3 изображена схема, полученная в результате такого объединения из схемы по рис. 2.

Сложность мультиплексора $F_1(n)$, когда он построен на объединенных реляторах и объединенных ранговых идентификаторах, определяется выражением

$$L(\tilde{F}_1(n)) = n - 1 + \sum_{q=1}^{n-1} 2q + \sum_{q=2}^{n-1} q - 1 + \frac{n!}{q!} \sum_{w=1}^{n-1} 2w.$$

Введем следующее разложение $БЛ_{\vee, \wedge}$ -функции $f(x_1, \dots, x_n)$:

$$f(x_1, \dots, x_n) = \gamma_1(x_2)_2 f(x_1, (x^{(1)})_2, x_3, \dots, x_n) + \gamma_2(x_2)_2 f(x_1, (x^{(2)})_2, x_3, \dots, x_n), \quad (6)$$

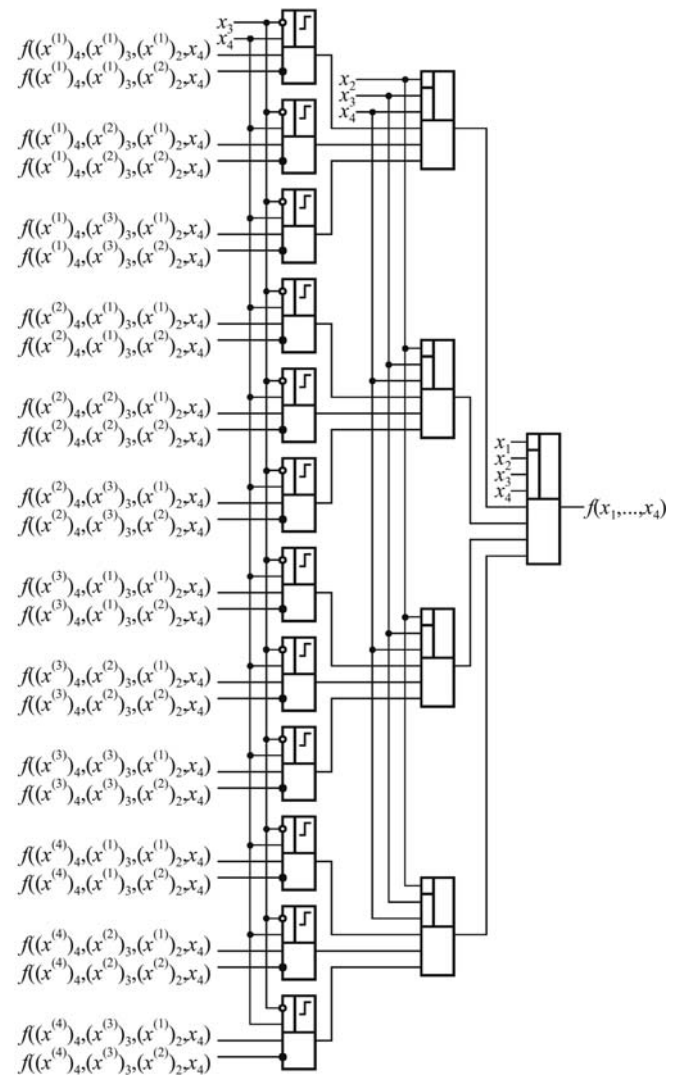


Рис. 2. Схема мультиплексора $F_1(n)$

где $\gamma_1(x_2)_2, \gamma_2(x_2)_2$ — характеристические функции, представляемые согласно выражению (3).

Здесь коэффициенты разложения по характеристическим функциям переменной x_{v+1} ($v \in \{1, \dots, n-2\}$) определяются выражением

$$\begin{aligned} f(x_1, (x^{(r_{i1})})_2, \dots, (x^{(r_{iv})})_{v+1}, x_{v+2}, \dots, x_n) = \\ = \lambda_1(x_{v+2})_{v+2} f(x_1, (x^{(r_{i1})})_2, \dots, (x^{(r_{iv})})_{v+1}, (x^{(1)})_{v+2}, \\ x_{v+3}, \dots, x_n) + \dots + \lambda_{v+2}(x_{v+2})_{v+2} f(x_1, (x^{(r_{i1})})_2, \\ \dots, (x^{(r_{iv})})_{v+1}, (x^{(v+2)})_{v+2}, x_{v+3}, \dots, x_n), \quad (7) \end{aligned}$$

где $r_{iv} \in \{1, \dots, v+1\}$; набор $x_2 = (x^{(r_{i1})})_2, \dots, x_{v+1} = (x^{(r_{iv})})_{v+1}$ задает i -й ($i = \overline{1, (v+1)!}$) вариант упорядочения континуальных переменных $x_1, \dots, x_{v+1}$; $(x^{(r)})_{v+2}$ ($r = \overline{1, v+2!}$) представляет переменную x_{v+2} как r -ю по величине среди упорядоченных переменных $x_1, \dots, x_{v+1}$ ($(x^{(1)})_{v+2} \leq \dots \leq (x^{(v+2)})_{v+2}$); $\lambda_1(x_{v+2})_{v+2} + \dots + \lambda_{v+2}(x_{v+2})_{v+2} = 1$.

Характеристическая функция $\lambda_r(x_{v+2})_{v+2}$ реализуется выражением

$$\begin{aligned} \lambda_r(x_{v+2})_{v+2} = C(\xi_{x_{i1}}(x_{v+2}) \dots \\ \dots \xi_{x_{i(v-r+2)}}(x_{v+2}) \bar{\xi}_{x_{i(v-r+3)}}(x_{v+2}) \dots \bar{\xi}_{x_{i(v+1)}}(x_{v+2})), \quad (8) \end{aligned}$$

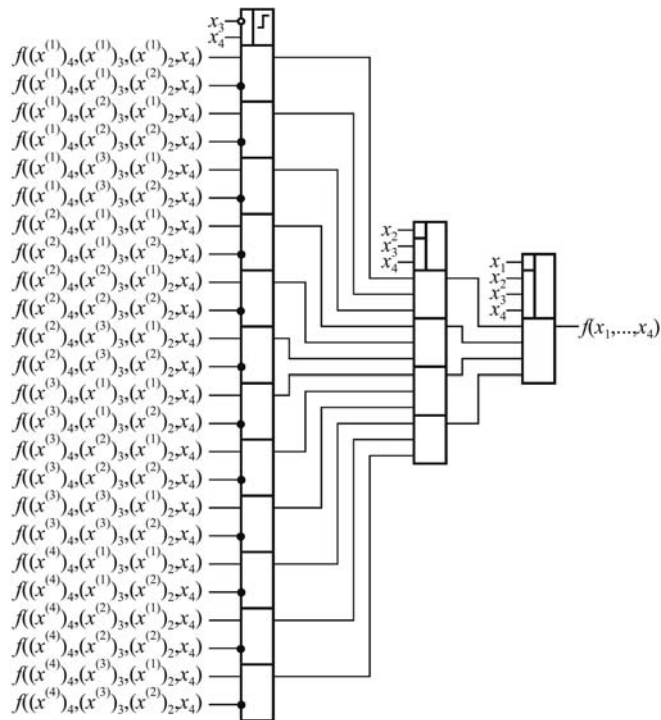


Рис. 3. Схема мультиплексора $\tilde{F}_1(n)$

полученным на основе результатов работы [5]. В (8) содержится $v - r + 2$ неинверсных и $r - 1$ инверсных характеристических функций (4), число $v + 1$ которых определяется числом упорядоченных континуальных переменных $x_{i1} \geq \dots \geq x_{i(v+1)}$ ($i1, \dots, i(v+1)$ — перестановка чисел $1, \dots, v+1$).

Оператор $C()$ преобразует произведение по правилу $\bar{\xi}_{x_{ik}}(x_{v+2}) \dots \bar{\xi}_{x_{i(v+1)}}(x_{v+2}) = \bar{\xi}_{x_{ik}}(x_{v+2})$ ($k \in \{1, \dots, v+1\}$).

Пример 2. При $v = 3, x_4 \geq x_3 \geq x_2 \geq x_1$ и $r = \overline{1, 5}$ согласно (8) можно записать

$$\begin{aligned} \lambda_1(x_5)_5 &= C(\xi_{x_{i1}}(x_5) \xi_{x_{i2}}(x_5) \xi_{x_{i3}}(x_5) \xi_{x_{i4}}(x_5)) = \\ &= \xi_{x_{i1}}(x_5) \xi_{x_{i2}}(x_5) \xi_{x_{i3}}(x_5) \xi_{x_{i4}}(x_5) = \\ &= \xi_{x_4}(x_5) \xi_{x_3}(x_5) \xi_{x_2}(x_5) \xi_{x_1}(x_5); \\ \lambda_2(x_5)_5 &= C(\xi_{x_{i1}}(x_5) \xi_{x_{i2}}(x_5) \xi_{x_{i3}}(x_5) \bar{\xi}_{x_{i4}}(x_5)) = \\ &= \xi_{x_{i1}}(x_5) \xi_{x_{i2}}(x_5) \xi_{x_{i3}}(x_5) \bar{\xi}_{x_{i4}}(x_5) = \\ &= \xi_{x_4}(x_5) \xi_{x_3}(x_5) \xi_{x_2}(x_5) \bar{\xi}_{x_1}(x_5); \\ \lambda_3(x_5)_5 &= C(\xi_{x_{i1}}(x_5) \xi_{x_{i2}}(x_5) \bar{\xi}_{x_{i3}}(x_5) \bar{\xi}_{x_{i4}}(x_5)) = \\ &= \xi_{x_{i1}}(x_5) \xi_{x_{i2}}(x_5) \bar{\xi}_{x_{i3}}(x_5) = \xi_{x_4}(x_5) \xi_{x_3}(x_5) \bar{\xi}_{x_2}(x_5); \\ \lambda_4(x_5)_5 &= C(\xi_{x_{i1}}(x_5) \bar{\xi}_{x_{i2}}(x_5) \bar{\xi}_{x_{i3}}(x_5) \bar{\xi}_{x_{i4}}(x_5)) = \\ &= \xi_{x_{i1}}(x_5) \bar{\xi}_{x_{i2}}(x_5) = \xi_{x_4}(x_5) \bar{\xi}_{x_3}(x_5); \\ \lambda_5(x_5)_5 &= C(\bar{\xi}_{x_{i1}}(x_5) \bar{\xi}_{x_{i2}}(x_5) \bar{\xi}_{x_{i3}}(x_5) \bar{\xi}_{x_{i4}}(x_5)) = \\ &= \bar{\xi}_{x_{i1}}(x_5) = \bar{\xi}_{x_{i4}}(x_5). \end{aligned}$$

Равенства (6) и (7) доказываются следующим образом. Если $x_2 = (x^{(r)})_2$ ($r \in \{1, 2\}$), то в правой части равенства (6) имеем $f(x_1, (x^{(r)})_2, x_3, \dots, x_n)$, но точно такое же значение имеет и его левая часть. Когда $x_{v+2} = (x^{(r)})_{v+2}$ ($r \in \{1, \dots, v+2\}$), в правой и левой частях равенства (7) получим $f(x_1, (x^{(r_{i1})})_2, \dots, (x^{(r_{iv})})_{v+1}, (x^{(r)})_{v+2}, x_{v+3}, \dots, x_n)$.

Для схемной реализации выражения (7) удобно использовать позиционный идентификатор [4], схема которого содержит реляторы (см. рис. 1, а) и изображена на рис. 1, в. Указанный идентификатор воспроизводит операцию

$$Z = \lambda_1(x_{v+2})_{v+2} y_1 + \dots + \lambda_{v+2}(x_{v+2})_{v+2} y_{v+2},$$

где $\lambda_r(x_{v+2})_{v+2}$ ($r = \overline{1, v+2}$) — характеристическая функция, определяемая по формуле (8); $x_{i1} \geq \dots \geq x_{i(v+1)}$ и $x_{v+2}, y_1, \dots, y_{v+2}$ — аналоговые сигналы (напряжения).

На рис. 4 представлена сформированная из позиционных идентификаторов и релятора схема аналогового мультиплексора $F_2(n)$, реализующего

разложения (6), (7) по характеристическим функциям переменных $x_2, \dots, x_n$ (при $n = 4$). Этот мультиплексор является универсальным логическим модулем (УЛМ) в классе всех БЛ $_{\vee, \wedge}$ -функций $f(x_1, \dots, x_n)$. Настройка $F_2(n)$ на воспроизведение заданной БЛ $_{\vee, \wedge}$ -функции осуществляется фиксацией ее значений на соответствующих переключательных входах позиционных идентификаторов крайней левой вертикальной группы.

Отметим, что позиционные идентификаторы в любой вертикальной группе могут быть объединены. При этом удаляются компараторы релейторов во всех, кроме одного, идентификаторах объединяемой группы, а каждый освободившийся вход управления ключей подсоединяется к выходу того оставшегося компаратора, на входах которого действуют сигналы, одноименные входным сигналам удаленного компаратора, выход которого был соединен с данным входом управления ключей.

Сложность, глубина и число настроечных входов мультиплексора $F_2(n)$, сформированного из объединенных позиционных идентификаторов, определяются соответственно выражениями

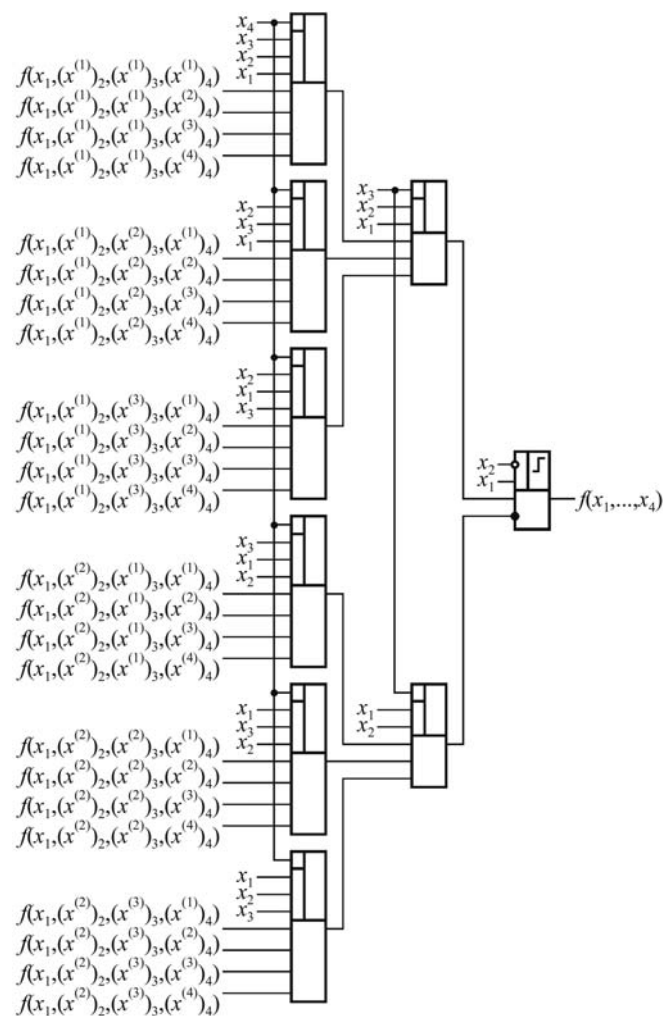


Рис. 4. Схема мультиплексора $F_2(n)$

$$L(\tilde{F}_2(n)) = \sum_{q=1}^{n-1} q(1 + 2(q!)) \text{ компараторов, ключей,}$$

$$H(\tilde{F}_2(n)) = 2 \text{ и } G(\tilde{F}_2(n)) = n!.$$

Нетрудно видеть, что $L(\tilde{F}_2(n)) < L(\tilde{F}_1(n))$ при $n > 2$. Еще одно преимущество модуля $F_2(n)$ обусловлено возможностью увеличения его порядка без изменения исходной схемы. Однако объединенные элементы УЛМ $F_2(n)$ имеют, в отличие от объединенных элементов модуля $F_1(n)$, пересечения соединений выходов компараторов с входами управления ключей, т. е. обладают в большей степени непланарной структурой, что затрудняет их однокристальное аппаратное воплощение.

Мультиплексоры $F_1(n)$, $F_2(n)$ кроме воспроизведения БЛ $_{\vee, \wedge}$ -функций $f(x_1, \dots, x_n)$ могут использоваться для реализации предикатных, предикатно-аргументных и аргументных функций [1], которые возможно однозначно задать на множестве ситуаций $x_{i1} \leq \dots \leq x_{in}$ ($i1, \dots, in$ — перестановка чисел $1, \dots, n$).

При четном n мультиплексоры $F_1(n)$, $F_2(n)$ являются универсальными в классе всех БЛ-функций от $0,5n$ аргументов $x_1, \dots, x_{0,5n}$. В этом случае необходимо переименовать остальные переменные, например, следующим образом: $x_{0,5n+1} \rightarrow \bar{x}_1, \dots, x_n \rightarrow \bar{x}_{0,5n}$. Отметим аппаратную избыточность модулей $F_1(n)$, $F_2(n)$, когда они реализуют произвольную БЛ-функцию $f(x_1, \dots, x_{0,5n})$ ($n > 2$). Указанная избыточность обусловлена тем, что некоторые из их $n!$ переключательных входов оказываются неиспользуемыми, поскольку значение $2^{0,5n}(0,5n)!$, определяющее число вариантов упорядочения $\hat{x}_{i1} \leq \dots \leq \hat{x}_{i(0,5n)} \leq x_0 \leq \dots \leq \hat{x}_{i(0,5n)} \leq \dots \leq \hat{x}_{i1}$ переменных $x_1, \dots, x_{0,5n}$, $\bar{x}_1, \dots, \bar{x}_{0,5n}$, меньше $n!$.

С учетом табл. 1 построим табл. 3. Здесь $x_3 = \bar{x}_1$, $x_4 = \bar{x}_2$; $f((x^{(r_{i1})})_4, (x^{(r_{i2})})_3, (x^{(r_{i3})})_2, x_4) \in \{x_1, x_2, \bar{x}_1, \bar{x}_2\}$ — значение функции $f(x_1, x_2)$ при i -м ($i = \overline{1, 8}$) варианте упорядочения ее аргументов и их отрицаний, который определяется набором $x_1 = (x^{(r_{i1})})_4$, $x_2 = (x^{(r_{i2})})_3$, $x_3 = (x^{(r_{i3})})_2$, характеризующим переменные x_g ($g = \overline{1, 3}$) и x_4 как, соответственно, r_{ig} -ю по величине среди $5 - g$ переменных $x_g, \dots, x_4$ и r_{i4} -ю по величине среди двух переменных x_3, x_4 ($r_{i1} \in \{1, 2, 3, 4\}$, $r_{i2} \in \{1, 2, 3\}$, $r_{i3}, r_{i4} \in \{1, 2\}$, $r_{i3} \neq r_{i4}$); $f(x_1, (x^{(r_{i1})})_2, (x^{(r_{i2})})_3, (x^{(r_{i3})})_4) \in \{x_1, x_2, \bar{x}_1, \bar{x}_2\}$ — значение функ-

Табличное представление произвольной БЛ-функции $f(x_1, x_2)$

i	Вариант упорядочения	$f(x_1, x_2)$	$f(x_1, x_2)$
1	$x_1 \leq x_2 \leq x_4 \leq x_3$	$f((x^{(1)})_4, (x^{(1)})_3, (x^{(2)})_2, x_4)$	$f(x_1, (x^{(2)})_2, (x^{(3)})_3, (x^{(3)})_4)$
2	$x_1 \leq x_4 \leq x_2 \leq x_3$	$f((x^{(1)})_4, (x^{(2)})_3, (x^{(2)})_2, x_4)$	$f(x_1, (x^{(2)})_2, (x^{(3)})_3, (x^{(2)})_4)$
3	$x_4 \leq x_1 \leq x_3 \leq x_2$	$f((x^{(2)})_4, (x^{(3)})_3, (x^{(2)})_2, x_4)$	$f(x_1, (x^{(2)})_2, (x^{(2)})_3, (x^{(1)})_4)$
4	$x_2 \leq x_1 \leq x_3 \leq x_4$	$f((x^{(2)})_4, (x^{(1)})_3, (x^{(1)})_2, x_4)$	$f(x_1, (x^{(1)})_2, (x^{(3)})_3, (x^{(4)})_4)$
5	$x_4 \leq x_3 \leq x_1 \leq x_2$	$f((x^{(3)})_4, (x^{(3)})_3, (x^{(2)})_2, x_4)$	$f(x_1, (x^{(2)})_2, (x^{(1)})_3, (x^{(1)})_4)$
6	$x_2 \leq x_3 \leq x_1 \leq x_4$	$f((x^{(3)})_4, (x^{(1)})_3, (x^{(1)})_2, x_4)$	$f(x_1, (x^{(1)})_2, (x^{(2)})_3, (x^{(4)})_4)$
7	$x_3 \leq x_4 \leq x_2 \leq x_1$	$f((x^{(4)})_4, (x^{(3)})_3, (x^{(1)})_2, x_4)$	$f(x_1, (x^{(1)})_2, (x^{(1)})_3, (x^{(2)})_4)$
8	$x_3 \leq x_2 \leq x_4 \leq x_1$	$f((x^{(4)})_4, (x^{(2)})_3, (x^{(1)})_2, x_4)$	$f(x_1, (x^{(1)})_2, (x^{(1)})_3, (x^{(3)})_4)$

ции $f(x_1, x_2)$ при i -м варианте упорядочения ее аргументов и их отрицаний, определяемом в данном случае набором $x_2 = (x^{(r_{i1})})_2$, $x_3 = (x^{(r_{i2})})_3$, $x_4 = (x^{(r_{i3})})_4$, в котором $(x^{(r_{iq})})_{g+1}$ представляет переменную x_{g+1} как r_{ig} -ю ($r_{ig} \in \{1, \dots, g+1\}$) по величине среди $g+1$ континуальных переменных $x_1, \dots, x_{g+1}$, причем переменные $x_1, \dots, x_g$ упорядочены.

На рис. 5, а (рис. 5, б) приведена схема безызбыточного логического модуля, универсального в классе БЛ-функций от двух аргументов, построенная на основе схемы по рис. 3 (рис. 4) и табл. 3. Указанный модуль имеет те же достоинства и недостатки, что и порождающий его УЛМ.

Пусть $F_3(4)$ есть универсальный в классе всех БЛ $_{\vee, \wedge}$ -функций $f(x_1, x_2, x_3, x_4)$ аналоговый мультиплексор, построенный согласно [1]. Тогда его сложность, глубина и число настроечных входов составляют $L(F_3(4)) = 72$ компаратора, ключа, $H(F_3(4)) = 2$ и $G(F_3(4)) = 4!$. Таким образом, $L(F_3(4)) > L(\tilde{F}_1(4))$ и $L(F_3(4)) > L(\tilde{F}_2(4))$. При этом

$$H(F_3(4)) = H(\tilde{F}_1(4)) = H(\tilde{F}_2(4)), \quad G(F_3(4)) = G(\tilde{F}_1(4)) = G(\tilde{F}_2(4)).$$

Заключение

Предложенные автором декомпозиционные процедуры обеспечивают построение аналоговых мультиплексоров с меньшими аппаратными затратами, что подтверждается приведенными здесь схемными решениями.

Рассмотренные аналоговые мультиплексоры являются универсальными средствами логической обработки континуальных данных, поскольку могут быть настроены на воспроизведение произвольных функций бесконечнозначной логики. Перспективность указанных средств обуславливается прежде всего тем, что они образуют элементный базис СБИС нового класса — аналоговых ПЛИС.

Необходимо отметить возможность реализации бесконечнозначных логических функций в стандартной системе "аналого-цифровой преобразователь → цифровой процессор → цифроаналоговый преобразователь". Однако выполняемое в последней промежуточное преобразование континуальных данных в цифровую форму, отсутствующее в аналоговых мультиплексорах, снижает производительность обработки этих данных.

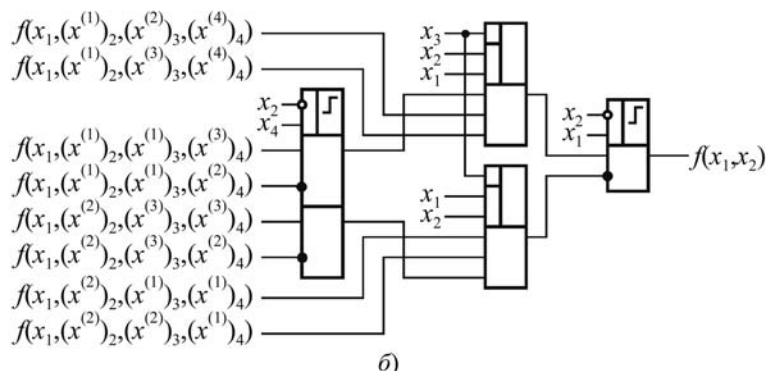
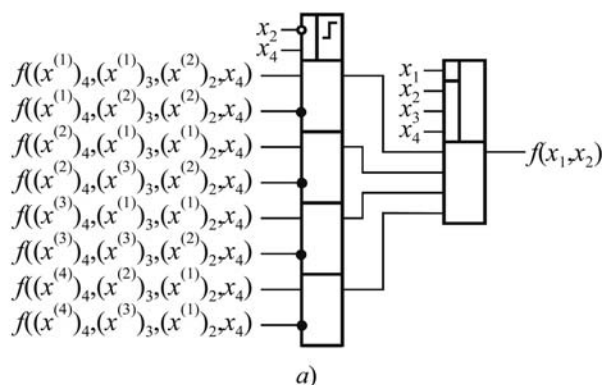


Рис. 5. Результат устранения избыточности схем (рис. 3, 4), возникающей при $x_3 = \bar{x}_1$, $x_4 = \bar{x}_2$

Список литературы

1. Волгин Л. И. Реляторные генераторы предикатных, аргументных и непрерывно-логических функций. Таллинн: АН Эстонии, 1992. 64 с.
2. Левин В. И. Бесконечнозначная логика в задачах кибернетики. М.: Радио и связь, 1982. 176 с.
3. Шимбирев П. Н. Гибридные непрерывно-логические устройства. М.: Энергоатомиздат, 1990. 174 с.
4. Волгин Л. И. Синтез устройств для обработки и преобразования информации в элементном базисе реляторов. Таллинн: Валгус, 1989. 179 с.
5. Андреев Д. В. Алгоритм минимизации ядра ранговых функций предикатной алгебры выбора / Информационные технологии в электротехнике и электроэнергетике: Материалы 4-й Всероссийской науч.-техн. конф. Чебоксары, 2002. С. 132—133.

УДК 004.492.3

А. Р. Берников,

канд. техн. наук, зам. ген. директора,

Научно-технический учебный
тренажерный центр, г. Калининград
e-mail: bernikov@ntutc.kocnig.su

Р. П. Графов, канд. техн. наук, доц.,

С. Н. Лысенко, ст. преподаватель,

О. С. Савенко, канд. техн. наук, декан,
Хмельницкий национальный университет

Поиск вредоносных программ в распределенных тренажерах с использованием технологии нечеткой логики

Рассматривается методология решения проблемы поиска вредоносных программ в распределенных информационных системах, характеризующихся нечеткой информацией о протекающих процессах. Использован механизм нечеткого логического вывода, основанного на продукционных правилах принятия решения. Полученные результаты могут быть использованы для антивирусного диагностирования программного обеспечения различных компьютерных систем.

Ключевые слова: нечеткий логический вывод, поиск вредоносных программ, нечеткие модели принятия решений, нечеткие продукционные правила и стратегии, семантические переменные, функции принадлежности

Известно, что методы дистанционного доступа для построения различных информационных систем (обучающих, управляющих и др.) получили в настоящее время достаточно широкое применение. Основной задачей при этом является, прежде всего, обеспечение защиты пользовательских программ и операционных систем от влияния различных вредоносных программ (ВП), так как потенциальная угроза проникновения их в систему со стороны удаленных станций достаточно велика.

Особое место среди ВП занимают различные вирусные, а также троянские программы, которые в отличие от вирусных программ способны проникать в систему в целях хищения информации, что представляет собой особую опасность.

Имеющиеся многочисленные факты хищения информации в конкретных системах можно объяснить тем, что существующие антивирусные программы во многом ориентированы на выявление известных

ВП, которые достаточно исследованы и классифицированы, но недостаточно адаптированы на распознавание новых подозрительных объектов, проникающих в систему.

В технологиях поиска ВП, как известно, различаются два независимых аспекта [1]: технический и аналитический. Технический компонент — это система сбора информации (о файловой системе, файлах и их цифровом содержимом и др.), на основании которой аналитическая часть в соответствии с принятой политикой безопасности оповещает пользователя о потенциальной угрозе и запрашивает у него дальнейшие указания относительно ВП. К основным методам сбора информации относятся:

- считывание и работа с файлами как с массивом байтов;
- эмуляция кода программы и запуск команд в виртуальной копии компьютера;
- запуск программы в "песочнице" (sandbox²), а также использование технологий виртуализации;
- мониторинг и регистрация всех событий, происходящих в операционной системе;
- поиск системных аномалий относительно некоего эталонного состояния.

Рассмотренные средства обработки содержат в себе элементы искусственного интеллекта, требующие достаточно сложных алгоритмов построения аналитической системы, включая экспертные системы или нейронные сети.

С учетом приведенного анализа предложена технология поиска и идентификации неизвестных ВП, которая сочетает элементы технологии поиска аномалий и экспертных систем. Для ее реализации использован математический аппарат нечеткой логики (НЛ), который в последнее время эффективно используется для решения ряда задач со слабоструктурированными данными.

Механизмы нечеткой логики позволяют строить модели предметной области, адекватные реальным, на основе семантического описания объекта исследования и знаний экспертов, что намного проще разработки сложных математических моделей. При этом с помощью специализированных методов обработки нечисловой информации обеспечиваются достаточно точные решения [2].

Рассмотрим некоторые понятия теории нечетких множеств, необходимые для дальнейшего изложения. Нечеткое множество A определяется как множество упорядоченных пар (кортежей) вида $(\mu_A(u), u)$, где $\mu_A(u)$ — степень принадлежности элемента множества $u \in U$ нечеткому множеству A ,

u — элемент универсального множества U . Следовательно,

$$A = \{\mu_A(u_i), u_i\}; i = 1, \dots, k. \quad (1)$$

Степень принадлежности определяется некоторым действительным числом из интервала $[0, 1]$. Функция принадлежности позволяет для произвольного элемента универсального множества вычислить степень принадлежности его нечеткому множеству. Нечеткое множество связано с лингвистической переменной и их нечеткими переменными, которые позволяют определить ее значение с помощью функции принадлежности.

Например, возьмем лингвистическую переменную "Категория программы". Ее значениями могут быть нечеткие переменные, образуемые с помощью модификаторов (или, не, очень и др.) в виде: неопасная, немного опасная, очень опасная и т. д. Функция принадлежности, если она задана, позволяет оценить далее лингвистическую переменную. Применительно к высказываниям в нечеткой логике применимы функции алгебры логики, которые имеют не числовые, а лингвистические значения истинности, которые адекватны степени принадлежности. Основой семантического описания исследуемой модели является выбор и анализ входных и выходных лингвистических переменных, а также их значений, которые получают с помощью мониторинга и экспертных оценок. Лингвистическая переменная считается заданной, если для нее определены: наименование N , универсальное множество A (область рассуждений), базовое терм-множество T (нечеткие переменные — значения лингвистической переменной), функция принадлежности μ и семантические процедуры преобразования (модификации) переменных термов лингвистической переменной.

Для исследования нечетких моделей используется механизм логического вывода (рис. 1), позволяющий получить четкий результат функционирования объекта, описываемого нечетким множеством переменных. Данный механизм содержит следующие этапы: фаззификация, нечеткий логический вывод, дефаззификация (приведение к четкости) [3, 4].

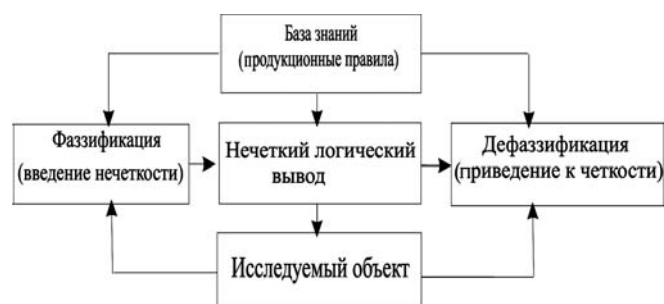


Рис. 1. Общая структура нечеткого вывода

Фаззификация — процесс перевода входных четких величин, полученных на основе экспертного анализа исследуемого объекта, в нечеткие переменные.

На этапе логического вывода используются нечеткие продукционные (if—then) правила базы знаний для преобразования нечетких входных данных после фаззификации в нечеткие выходные.

Дефаззификация — процесс композиции нечетких множеств логического вывода и приведение к четкости, получению результата для принятия решения.

Действия ВП рассматриваются на этапах проникновения в систему, активизации и деструктивных действий, которые образуют один трехуровневый жизненный цикл (ЖЦ). Логический вывод о наличии ВП реализуется с помощью нечеткой модели (НМ), которая основана на семантическом описании и анализе множества возможных состояний ВП в течение жизненного цикла:

$$НМ \rightarrow \{(X, Y, V, Z)\}, \quad (2)$$

где X — универсальное множество входных состояний $x \in X$, принимающих ВП в системе пользователя (функции и действия, используемые ВП для проникновения в систему);

Y — универсальное множество выходных состояний $y \in Y$ (компоненты, которые потенциально могут быть подвержены действию ВП);

V — множество нечетких отношений (x_i, y_j) , характеризующих связи между элементами множеств X и Y ;

Z — характеристические параметры отношений.

На множествах X, Y определены нечеткие множества $(X^п, X^a, X^д)$ и $(Y^п, Y^a, Y^д)$ входных и выходных состояний ВП, где индексы "п", "а", "д" соответствуют этапам ЖЦ (проникновение в систему, активизация и деструктивные действия). В соответствии с формулой (1) множества $X^п, X^a, X^д$ определяются как множества упорядоченных пар:

$$X^п = \{\mu_{X^п}(x), x\}; X^a = \{\mu_{X^a}(x), x\}; X^д = \{\mu_{X^д}(x), x\},$$

где $\mu_{X^п}(x)$, $\mu_{X^a}(x)$ и $\mu_{X^д}(x)$ — функции принадлежности, указывающие на степень принадлежности элемента x множеству X множествам $X^п, X^a, X^д$ соответственно.

Множества $(Y^п, Y^a, Y^д)$ представляются в виде множеств отношений (x_i, y_j) :

$$Y^п = \{(x_i, y_j, \mu_{Y^п}(x_i, y_j)); Y^a = \{(x_i, y_j, \mu_{Y^a}(x_i, y_j));$$

$$Y^д = \{(x_i, y_j, \mu_{Y^д}(x_i, y_j));$$

где выражения $\mu_{Y^п}(x_i, y_j)$, $\mu_{Y^a}(x_i, y_j)$ и $\mu_{Y^д}(x_i, y_j)$ являются функциями принадлежности нечетких отношений (x_i, y_j) множествам $Y^п, Y^a, Y^д$ соответственно, которые определяют степень уверенности в существовании причинно-следственной связи

(x_i, y_j) . Нечеткие отношения множеств $Y^п$, Y^a , $Y^д$ представляются соответствующими матрицами:

$$V_п = |x_i, y_j|, V_a = |x_i, y_j|, V_d = |x_i, y_j|,$$

в которых на пересечении x_i, y_j находятся значения функции принадлежности $\mu_{Y^п}(x_i, y_j)$, $\mu_{Y^a}(x_i, y_j)$ и $\mu_{Y^д}(x_i, y_j)$ соответственно.

В качестве входной лингвистической переменной на множестве X принята "Степень подозрительности". Ее характеристикой является терм-множество T , которое содержит три нечетких переменных (значения лингвистической переменной): "Малая подозрительность $M(x)$ ", "Средняя подозрительность $C(x)$ " и "Высокая подозрительность $B(x)$ ", с областью определения $[0, 1]$. На практике число переменных термина не превышает семи.

Для физической реализации лингвистической переменной необходимо определить значения переменных термина T при заданных входных переменных x . Для этого должны быть найдены функции принадлежности: $\mu_{X^п}(M)$, $\mu_{X^п}(C)$, $\mu_{X^п}(B)$; $\mu_{X^a}(M)$,

$\mu_{X^a}(C)$, $\mu_{X^a}(B)$, а также $\mu_{X^д}(M)$, $\mu_{X^д}(C)$, $\mu_{X^д}(B)$.

На практике с этой целью часто используется ряд типовых форм кривых: треугольная, трапециевидная и гауссова функция принадлежности [4]. Так, на рис. 2, А для нечетких переменных $M(x)$, $C(x)$, $B(x)$ на множествах $X^п$, X^a , $X^д$ их функции принадлежности определены при входных значениях

$x_1 = 2, x_2 = 5, x_3 = 9$: (0,62, 0,78, 0,37), (0,9, 0,5, 0,25) и (0,42, 0,28, 0,5) соответственно. Входные значения определяются и приводятся к определенному формату экспертом по результату мониторинга. Выходную лингвистическую переменную зададим на множестве Y . В качестве такой переменной принята "Опасность поражения", значение которой будет определяться на этапах ЖЦ нечеткой переменной "Степень поражения" с функциями принадлежности: $\mu_{Y^п}(x_i, y_j)$, $\mu_{Y^a}(x_i, y_j)$ и $\mu_{Y^д}(x_i, y_j)$. В данной постановке задачи для оценки лингвистической переменной также учитывается вид поражения: зависание системы, потеря и искажение файлов, нарушение реестра и т. д., что учитывается вектором $Z = \{z_k\}$, в котором компонентам $z_k, k = 1, \dots, r$, (виды поражения) присваиваются нормированные приоритетные веса $P = \{p_k\}$, ($\sum p_k = 1$), учитывающие уровень их опасности для системы пользователя.

Рассмотрим содержательную часть матриц отношений, построенных с учетом [5, 6]. Так, в матрице отношений $V_п = |x_i, y_j|$ элементы x_i — функции (механизмы) и способы проникновения ВП в систему пользователя, $i = 1, \dots, k$; элементы y_j — возможные входы проникновения в систему: порты сетевых протоколов прикладного уровня, $j = 1, \dots, h$ (индексы не совпадают с истинными номерами портов). Например, x_1 — набор функций работы с электронной почтой; x_2 — набор функций работы с системами передачи файлов; x_3 — набор функций удаленного доступа (через консоль и командные

интерпретаторы); x_4 — набор функций работы с web-браузером и т. д. Порт p_{20} — FTP (File Transfer Protocol) — сетевой протокол, предназначенный для передачи файлов в компьютерных сетях; p_{22} — SSH (Secure Shell) — сетевой протокол, который позволяет осуществлять удаленное управление компьютером и передачу файлов и т. д.

На этапе деструктивных действий используется матрица отношений исследуемого объекта и структурных единиц операционной системы (ОС): $V_d = |x_i, y_j|$, в которой x_i — действия ВП, направленные на структурные компоненты ОС пользователя; $i = 1, \dots, \sigma$; y_j — структурные компоненты ОС пользователя, $j = 1, \dots, \tau$. На-

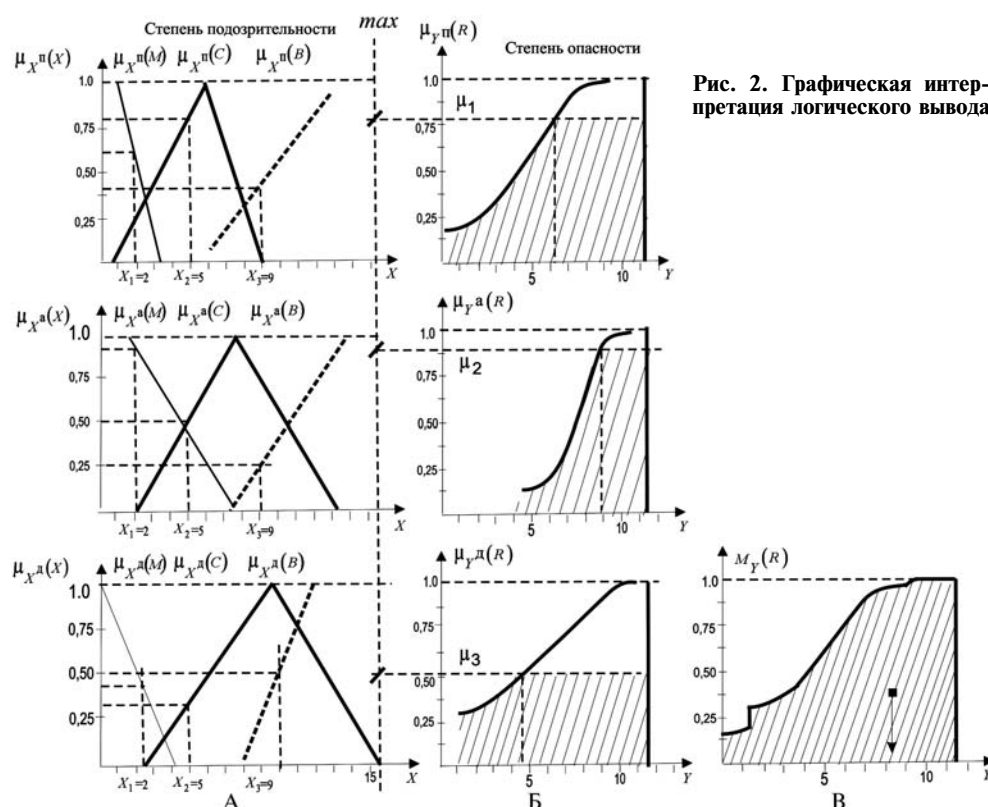


Рис. 2. Графическая интерпретация логического вывода

пример, x_1 — прием и отправление файлов; x_2 — создание и уничтожение файлов; x_3 — создание, запуск и уничтожение процессов и т. д. К структурным компонентам ОС относятся: y_1 — файловая система; y_2 — планировщик процессов (как составляющая ядра операционной системы); y_3 — системные службы (ключи реестра — Windows XP) и т. д.

Из приведенных примеров видно, что поведение ВП на этапах ЖЦ многовариантно, имеет нечеткий характер и в принципе не может быть просто прогнозировано. Кроме того, их функции принадлежности не определены, что делает невозможным оценки выходной лингвистической переменной. В работе данные функции определяются, в отличие от типовых подходов, косвенным путем на основе анализа и оптимизации матриц нечетких отношений V_{Π} , V_a , V_d с привлечением оценок экспертов.

Покажем на примере принцип формирования функции принадлежности $\mu_{Y^{\Pi}}(x_i, y_j)$ на этапе проникновения.

Рассматриваемая задача представляется как нахождение для каждого x_i матрицы $V_{\Pi} = |x_i, y_j|$ наиболее вероятного порта проникновения y_j при заданных признаках опасности z_k . Данная задача сводится к задаче ранжирования [7], в которой в отличие от [4] попарное сравнение экспертов проводится с учетом оценочных признаков, позволяющих учесть особенности сравниваемых объектов.

Исходным для решения задачи является построение матрицы превосходства $S = |s_{ij}|$, элементы которой s_{ij} выражаются любым положительным числом ($s_{ij} = s_i/s_j$; $0 < s_{ij} < \infty$; $s_{ji} = 1/s_{ij}$; $s_{ii} = 1$; $i, j = 1, \dots, m$, где m — число возможных исходов). Элементы s_{ij} матрицы S определяются вычислением значений парных предпочтений по каждому признаку отдельно с учетом их весов $P = \{p_k\}$; $k = 1, \dots, r$, с использованием выражения

$$s_{ij} = \frac{\sum_{k=1}^r s_{ij}^k p_k}{\sum_{k=1}^r s_{jk}^k p_k}; \quad s_{ji} = \frac{1}{s_{ij}}; \quad s_{ii} = 1; \quad i, j = \overline{1, m}.$$

С помощью матрицы S определяется собственный вектор $\Pi = (\pi_1, \dots, \pi_m)$, который соответствует максимальному положительному корню λ характеристического полинома $|S - \lambda \cdot E| = 0$; $S \cdot \Pi = \lambda \cdot \Pi$, где E — единичная матрица. Компоненты вектора Π ($\sum \pi_i = 1$) отождествляются с оценкой $\mu_{Y^{\Pi}}(x_i, y_j)$,

учитывающей принятые признаки опасности. Подобная процедура выполняется для всей матрицы $V_{\Pi} = |x_i, y_j|$, в результате чего получаем оптимизированную матрицу отношений $V_{\Pi} = |x_i, y_j|$, в которой используются лишь отношения x_i, y_j с наиболее выраженным уровнем опасности, определяемым значением, равным $\pi_{\max}$, ($0 \leq \pi_{\max} \leq 1$). На основе

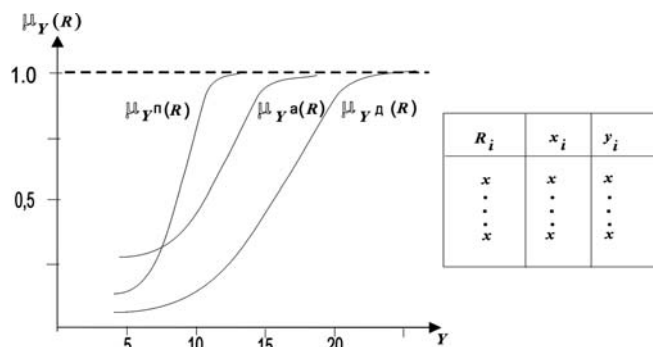


Рис. 3. Примерный вид функций принадлежности выходной переменной

полученной матрицы строится нормированная кривая функции принадлежности $\mu_{Y^{\Pi}}(x_i, y_j)$ выходной переменной y , а также таблица, в которой каждому значению $\mu_{Y^{\Pi}}(x_i, y_j)$ ставится в соответствие пара (x_i, y_j) , позволяющая идентифицировать, в том числе, ресурсы, которые были использованы вредоносной программой на рассматриваемом этапе (рис. 3). Для компактности отношения (x_i, y_j) заменены символом R .

Основой для нечеткого вывода служит база нечетких продукционных правил (НПП), определяющих стратегию решения задачи. Типичное продукционное правило базы правил состоит из посылки (антецедента): нечеткие высказывания в форме "если..." и заключения (консеквента) — в форме "то...". Антецедент может содержать несколько посылок, которые объединяются в зависимости от стратегии посредством логических связей "и" или "или".

Каждое из правил нечетких продукций может иметь некоторый вес $F_i = [0, 1]$, который определяет значимость правила или уверенность в степени истинности заключения, получаемого по отдельному нечеткому правилу. В общем виде база, содержащая m правил, имеет вид:

Π_1 : если x_1 это A_{11} ... и(или) ... x_n это A_{1n} ,
то y это R_1 ;

.....

Π_i : если x_1 это A_{i1} ... и(или) ... x_n это A_{in} ,
то y это R_i ;

.....

Π_m : если x_1 это A_{m1} ... и(или) ... x_n это A_{mn} ,
то y это R_m ,

где x_k — входные переменные, $k = 1, \dots, n$; y — выходная переменная; A_{ik} — заданные нечеткие множества с функциями принадлежности ($i = 1, \dots, m$; $k = 1, \dots, n$).

На основе нечетких высказываний, истинность которых установлена в результате фаззификации, оценивается степень истинности нечетких выска-

званий, являющихся заключением соответствующих НПП. Далее выполняется процедура (агрегирование) определения степени истинности левых частей (уровней отсечения — a_i) по каждому из правил системы нечеткого вывода. Так, дизъюнкцией нечетких высказываний является логическая операция, результатом которой является нечеткое высказывание, определяемое как

$$a_i = \max_i (A_{ik}(X_k)).$$

Приведем для примера фрагмент базы правил (три переменные и три правила), в которой для каждого правила использованы связки "или" и значения F_i равны единице:

П1: если $x_1 = M$, или $x_2 = C$, или $x_3 = B$, то $y = R^п$;

П2: если $x_1 = M$, или $x_2 = C$, или $x_3 = B$, то $y = R^а$;

П3: если $x_1 = M$, или $x_2 = C$, или $x_3 = B$, то $y = R^д$.

Возможны и другие стратегии получения базы правил.

В работе используется стратегия на основе правила $\max$ — $\min$ композиции и нечеткой операции $\max$ — дизъюнкции для оценки одинаковых заключений. В рассматриваемом фрагменте при заданных x_1, x_2, x_3 приведенные правила позволяют получить нечеткое заключение о степени истинности подозрительности на исследуемых этапах:

$$\mu_1 = \max\{\mu_{X^п}(M); \mu_{X^п}(C); \mu_{X^п}(B)\} = \max\{0,62; 0,78; 0,37\} = 0,78;$$

$$\mu_2 = \max\{\mu_{X^а}(M); \mu_{X^а}(C); \mu_{X^а}(B)\} = \max\{0,9; 0,5; 0,25\} = 0,90;$$

$$\mu_3 = \max\{\mu_{X^д}(M); \mu_{X^д}(C); \mu_{X^д}(B)\} = \max\{0,42; 0,28; 0,5\} = 0,5.$$

Следующий этап — активизация: процесс нахождения степени истинности каждого из заключений нечетких продукционных правил, который осуществ-

ляется усечением функций принадлежности выходной переменной на уровнях μ_1, μ_2, μ_3 (см. рис. 2, Б):

$$\mu_{Y^п}(R) = 0,78; \mu_{Y^а}(R) = 0,90; \mu_{Y^д}(R) = 0,5.$$

Для композиции (объединения) полученных усеченных функций используется максимальная композиция нечетких множеств

$$M_y(R) = \max_i (\mu_y(R)),$$

где $M_y(R)$ — функция принадлежности итогового нечеткого множества.

На рис. 2 показана графическая интерпретация нечеткого вывода для трех входных переменных и трех нечетких правил.

На рис. 2 этапы интерпретации логического вывода А, Б, В выполняют следующие функции:

А — представление функции принадлежности для нечетких переменных $M(x), C(x), B(x)$ на множествах $X^п, X^а, X^д$ при входных значениях $x_1 = 2, x_2 = 5, x_3 = 9$;

Б — нахождение степени истинности заключений нечетких продукционных правил на уровнях отсечения μ_1, μ_2, μ_3 ;

В — приведение к четкости логического вывода (получение числового значения).

Заключительным является этап дефаззификации — приведение к четкости (получение числового значения для принятия решения) (рис. 2, В). Для дефаззификации в работе использован метод центра тяжести [4], при котором значение выходной переменной равно абсциссе центра тяжести площади, ограниченной графиком функции принадлежности итогового нечеткого множества $M_y(R)$:

$$y = \frac{\sum_{i=1}^n y_i M_y(R) i}{\sum_{i=1}^n M_y(R) i} = \frac{11,3 \cdot 0,78 + 9 \cdot 0,9 + 5 \cdot 2,5}{0,78 + 0,9 + 0,5} = 8,9.$$

Полученный результат интерпретируется как коэффициент опасности поражения системы со стороны ВП, который сравнивается с некоторым нормированным коэффициентом, определяемым принятой стратегией безопасности. Обобщенная схема реализации предложенной методологии приведена на рис. 4, которая содержит подсистемы мониторинга и принятия решения.

Выводы

В статье предложена новая методология и ее использование для поиска и идентификации ВП методами нечеткой логики, позволяющей более просто получать достаточно точные

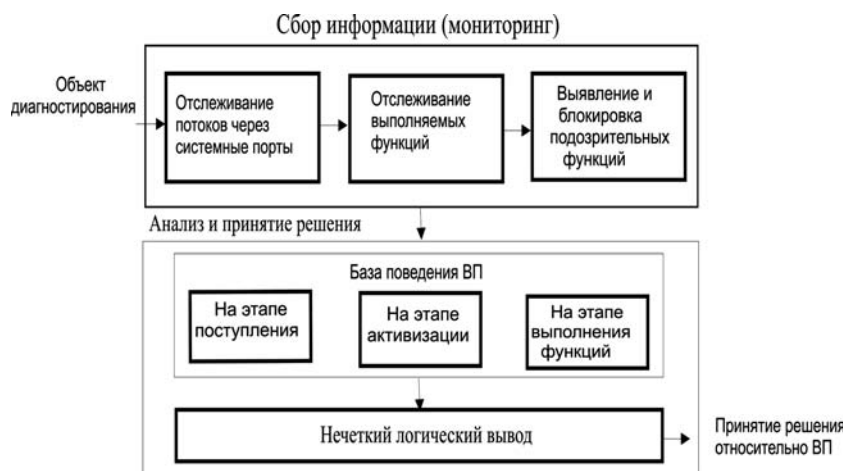


Рис. 4. Обобщенная схема поиска вредоносных программ

решения на основе семантического описания задачи и нечетких продукционных правил.

Существующим методам присущи следующие недостатки: невысокое быстродействие, низкая вероятность идентификации новых ВП, высокие требования отдельных методов к аппаратному обеспечению, сложность модели и ее реализации.

Установлено, что применение нечеткой логики открывает большие возможности для адаптации к любым моделям ВР, а также принятия оптимального решения путем проектирования соответствующей базы продукционных правил и стратегии их обработки.

При этом обеспечивается повышение быстродействия за счет исключения из алгоритмов поиска достаточно сложных компонент и процедур (баз сигнатур, эвристических анализаторов, использование контрольных сумм и т. д.), свойственных существующим методам, а также достаточно высокая реакция и незначительная вероятность пропуска неизвестных ВП.

Результаты работы могут быть использованы разработчиками антивирусного диагностирования. Рабочей средой для разработки и исследования предложенной методологии является программная оболочка FUZZY EXPERT.

Список литературы

1. **Технологии** обнаружения вредоносного кода. Эволюция Алиса Шевченко. URL: www.securelist.com
2. **Заде Л. А.** Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 165 с.
3. **Леоненков А. В.** Нечеткое моделирование в среде MATLAB в fuzzy TECH. СПб.: БХВ—Петербург, 2005. 736 с.
4. **Штовба С. Д.** Проектирование нечетких систем средствами MATLAB. М.: Горячая линия—телеком 2007. 290 с.
5. **Савенко О. С., Лысенко С. М.** Модель процесса поиска троянских программ в персональном компьютере // Радиотехнические и компьютерные системы. Харьков: НАУ "ХАИ", 2008. № 7. С. 87—92.
6. **Х-релиз:** исходники трояна // Информационный портал Хакер.Ру. URL: <http://www.hacker.ru/post/17223/>
7. **Берников А. Р., Графов Р. П.** Согласование экспертных оценок для формирования модели деятельности оператора в тренажерах // Информационные технологии. 2003. № 6. С. 44—47.

УПРАВЛЕНИЕ РИСКАМИ

УДК 004.89

В. В. Борисов, д-р техн. наук, проф.,
А. В. Сеньков, аспирант,
Филиал ГОУ ВПО "Московский энергетический
институт (технический университет)",
г. Смоленск
e-mail: VBorisov@etna-it.ru

Интеллектуальное управление рисками в сложных организационно-технических системах

Предложен метод интеллектуального управления рисками в сложных организационно-технических системах, обеспечивающий: комплексное управление профессиональными и производственными рисками одновременно на нескольких уровнях иерархии сложной системы; гибридизацию интеллектуальных технологий на каждом из этапов управления рисками; адаптивность в процессе управления рисками.

Ключевые слова: интеллектуальное управление рисками, профессиональные и производственные риски, нечеткое моделирование

При возрастании сложности организационно-технических систем и происходящих в них процессов все большее значение приобретает управление рисками, которое осуществляется в условиях комплексного воздействия рискообразующих факторов, неопределенности системных и внешних параметров.

Одними из основных рисков, присущих сложным организационно-техническим системам и имеющим комплексный характер, являются профессиональные [1, 2] и производственные [3—5] риски, трактуемые как сочетание возможности нанесения ущерба системе (технической подсистеме, подсистеме организации труда, подсистеме управления) и его негативных последствий (тяжести этого ущерба).

Комплексное воздействие профессиональных и производственных рисков на организационно-техническую систему прежде всего проявляется в том, что и профессиональные, и производственные рискообразующие факторы могут негативно воздействовать как на сотрудников, так и на оборудование, а также на всю систему в целом [6, 7].

При этом негативные последствия от воздействия этих факторов могут быть двух типов:

- мгновенные последствия, например, получение травм сотрудниками либо отказ оборудования;

- накапливающиеся последствия, например, профессиональные заболевания сотрудников в результате продолжительного воздействия вредных и опасных производственных факторов или отказ оборудования в результате истощения его ресурса.

И в итоге негативные последствия от воздействия профессиональных и производственных рискообразующих факторов могут заключаться в остановке функционирования системы в целом.

В таких условиях управление рисками должно носить не только комплексный, но и интеллектуальный характер. При этом "интеллектуализация" управления рисками должна быть системной. Обязательным условием управления рисками в таком случае является минимизация вмешательств человека в процесс управления рисками, поскольку он является основным источником неточности и неопределенности данных, что в значительной степени влияет на качество процесса управления рисками.

Под интеллектуальной системой понимается адаптивная система, позволяющая строить программы целесообразной деятельности по решению поставленных перед ними задач на основании конкретной ситуации, складывающейся на данный момент в окружающей их среде.

При этом адаптивной системой является система, устойчивая к изменениям свойств управляемого объекта, целей управления или окружающей среды за счет смены алгоритма функционирования, программы проведения или поиска оптимальных решений и состояний.

На основании сказанного выше можно говорить о том, что интеллектуализация должна заключаться:

- в применении нечетких, нейронечетких и других интеллектуальных моделей на каждом из этапов управления рисками [8];
- в гибридизации интеллектуальных технологий на основе одного из следующих подходов:
 - в модели один из компонентов замещается другой моделью;
 - независимые модели, обмениваясь информацией, выполняют различные задачи в целях получения общего решения;
 - одна модель применяется для имитации (и реализации) функционирования другой модели;
- в комплексности подхода к управлению рисками, заключающемся в:
 - управлении как профессиональными, так и производственными рисками;
 - управлении рисками одновременно на нескольких уровнях иерархии сложной организационно-технической системы;
 - в комплексировании различных моделей для реализации отдельных этапов управления рисками;
- в адаптивности (самоадаптации) управления рисками:

- самоадаптации структуры и набора моделей способа управления рисками;
- самоадаптации отдельных моделей, участвующих в управлении рисками;
- самоадаптации знаний моделей, используемых при управлении рисками.

Основными этапами управления рисками являются следующие [9]:

- определение контекста;
- идентификация рисков;
- анализ рисков;
- оценивание рисков;
- обработка рисков и планирование мероприятий для противодействия рискам на каждом из этапов управления рисками.

На каждом из этапов возможно осуществление обмена информацией, анализа и мониторинга.

В зависимости от этапа управления рисками планируемые мероприятия могут быть направлены на:

- уход от риска путем принятия решения о прекращении деятельности (или уход от начала действий), приводящую к появлению риска (предотвращение риска);
- устранение источников риска (опасностей);
- разделение риска с другой стороной или сторонами (снижение последствий риска);
- сохранение риска;
- принятие риска.

Следует отметить, что наиболее эффективным механизмом планирования мероприятий в процессе управления рисками является принятие решений о мерах по устранению источников риска, а также по предотвращению/снижению риска, т. е. до превращения риска в проблему, так как это, в большинстве случаев, требует гораздо меньших ресурсов, чем на ликвидацию последствий риска. Однако в ряде случаев сделать этого не удастся.

Все это обуславливает необходимость комплексного решения проблем управления рисками за счет обоснованного выбора и планирования мероприятий по предотвращению/снижению рисков на всех этапах управления ими.

Вместе с тем, в современных условиях основные задачи интеллектуального управления рисками не могут решаться с помощью традиционных математических методов, основанных на точном и адекватном описании проблемных ситуаций, анализе и выработке управляющих решений. Такие задачи характеризуются большим числом влияющих разнокачественных факторов, неопределенностью, нечеткостью и неточностью данных. Это, в свою очередь, позволяет обосновать необходимость разработки совокупности моделей для реализации основных этапов управления рисками, основанных на методологии нечеткого моделирования.

В таблице представлен обоснованный состав информации, используемой в рамках процессов интеллектуального управления профессиональными и производственными рисками, а также перечень

модельного обеспечения, необходимого для реализации основных этапов интеллектуального управления рисками.

В соответствии с рассмотренными выше требованиями предлагается метод интеллектуального управления рисками, основанный на методологии нечеткого моделирования.

На рисунке представлены основные этапы предлагаемого метода интеллектуального управления рисками.

В рамках процесса определения контекста требуется четко сформулировать или актуализировать цели системы, определить внутренние и внешние параметры, которые должны быть учтены при управлении комплексными рисками, а также область и критерии риска для оставшейся части процесса.

Требования к моделям, применяемым на этапе "Определение контекста":

- модель должна иметь возможность, оперируя большим числом исходных данных, изменять характеристики моделей, применяемых на последующих этапах управления рисками;

- модель должна допускать изменение числа и состава входных параметров;
- модель должна допускать изменение числа и состава выходных параметров (при изменении на этапе мониторинга структуры моделей);
- модель должна учитывать значимость каждого из входных параметров и, как следствие, степень его влияния на выходные параметры (каждый в отдельности).

Оценка риска обобщает процессы идентификации рисков, анализа рисков и оценивания рисков.

Идентификация рисков подразумевает идентификацию источников рисков и зон их влияния, события, характерные для этих рисков, и их причины, а также потенциальные последствия этих событий. Цель этого шага — создание исчерпывающего перечня рисков на базе событий, которые могут повлиять на достижение целей организацией. Критичным является объем идентификации. Получение исчерпывающего перечня рисков обеспечит повышение качества процесса управления рисками в целом. Идентификация рисков проводится с применением модели идентификации рисков.

Состав информационного и модельного обеспечения для интеллектуального управления профессиональными и производственными рисками

Этап управления рисками	Исходная информация	Результирующая информация	Используемые модели и методы
Подготовка к управлению рисками. Определение контекста	Неформализованное описание предметной области Цели системы Информация о предпочтениях заинтересованных сторон Ограничения на управление рисками (внешние и внутренние, в том числе на ресурсы) Произошедшие события Статистическая информация о событиях Сведения о результатах мониторинга применения обработки рисков	Формализованное описание предметной области Обоснованный набор исследуемых факторов системы/процесса (системных факторов) Непосредственно принимаемые решения (могут касаться любого из этапов управления рисками) Сведения о необходимых изменениях в применяемых в процессе управления рисками методах и моделях Информация о предпочтениях заинтересованных сторон	Способ формализации предметной области Модель оценки значимости системных факторов
Идентификация рисков	Формализованное описание предметной области Обоснованный набор исследуемых факторов системы/процесса (системных факторов) Ограничения на управление рисками Произошедшие события Статистическая информация о событиях	Перечень рисков	Модель идентификации рисков
Анализ рисков	Ограничения на управление рисками Произошедшие события Статистическая информация о событиях Перечень рисков	Выявленные последствия реализации рисков и их характеристики Возможность наступления Воздействие на цели организации	Модель анализа комплексных рисков
Оценивание рисков	Критерий риска Выявленные последствия реализации рисков и их характеристики	Ранжированный перечень подлежащих обработки рисков	Модель оценивания комплексных рисков
Обработка рисков	Ранжированный перечень подлежащих обработки рисков Ограничения на управление рисками	План обработки рисков, включающий меры по обработке рисков Сведения об остаточном риске	Модель обработки комплексных рисков
Анализ и мониторинг	План обработки рисков, включающий меры по обработке рисков Сведения об остаточном риске	Сведения о результатах мониторинга применения обработки рисков	

Анализ рисков обеспечивает получение входных данных для расчета характеристик риска и решения о необходимости обработки рисков. Этот шаг включает рассмотрение причин и источников риска, их позитивных и негативных последствий, возможности того, что эти последствия могут произойти, а также определение прочих необходимых атрибутов риска. Анализ рисков может быть проведен с различной степенью детализации в зависимости от риска, цели анализа, доступной информации и ресурсов. Анализ может быть качественным, полуквантитативным или количественным, а также комбинацией перечисленных видов. Анализ рисков проводится с применением модели анализа комплексных рисков.

Этапы "Идентификация рисков" и "Анализ риска" могут выполняться последовательно либо параллельно в зависимости от того, какие модели были выбраны для выполнения этих этапов.

В случае, если выбранные модели одинаковы и представляют одинаковую структуру (например, деревья отказов, деревья решений), то этап идентификации рисков может быть опущен, и, например, для каждого события мы сразу будем получать его характеристики как риска.

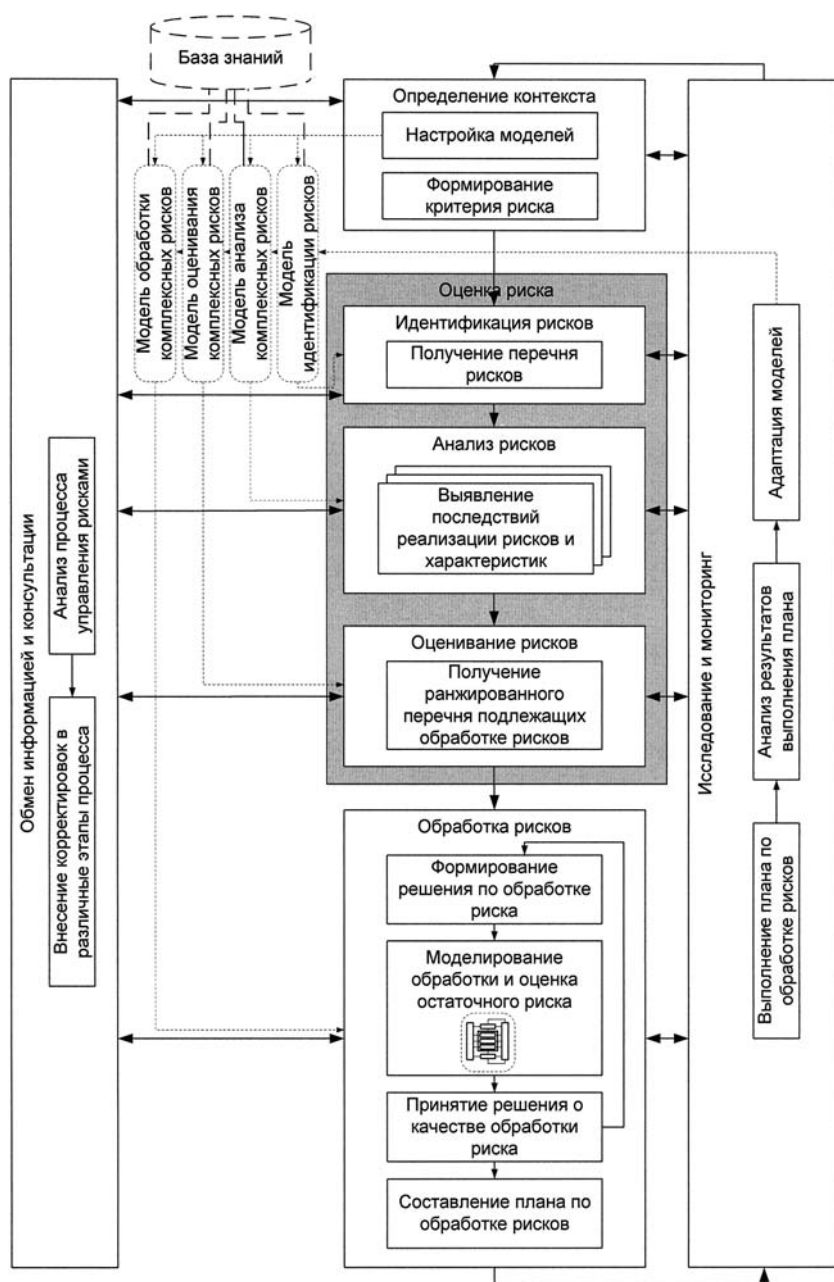
В случае выбора различных моделей этапы должны следовать друг за другом.

Обобщенно требования к моделям, применяемым на этапе "Анализ риска", заключаются в следующем:

- модель должна иметь возможность адаптации на этапе мониторинга;
 - модель должна обеспечивать оценку каждого из идентифицированных на предыдущем этапе рисков;
 - модель должна учитывать весомость влияния каждого входного параметра на каждый выходной;
 - модель должна позволять получать на выходе характеристики рисков вне зависимости от их природы (описанные числовыми значениями, лингвистическими или любыми другими);
 - в дополнение, модель может обеспечивать оценку достоверности полученных характеристик рисков.
- Цель оценивания риска — поддержка принятия решений, а также выставление приоритета обработки риска. Оценивание риска включает сравнение уровней риска, определенных в результате анализа рисков. Необходимость обработки риска определяется на основе этого сравнения. В некоторых случаях оценивание риска может привести к решению продолжить анализ или же не обрабатывать риск. Оценивание рисков проводится с применением модели оценивания комплексных рисков.

Требования к моделям оценивания рисков:

- модель оценивания рисков должна обеспечивать получение итоговой оценки каждого из рисков на основе его характеристик;
- модель должна учитывать критерий риска (в зависимости от обстоятельств это может быть некоторое критериальное значение либо зависимость);
- модель должна быть адаптивной к значимости отдельных характеристик рисков.



Этапы метода интеллектуального управления рисками

Обработка риска включает выбор одного или нескольких способов работы с риском (см. выше).

Результатом обработки рисков является составление и осуществление плана обработки рисков. Целью таких планов является документирование мер, принятых при обработке рисков. Планы должны быть интегрированы с процессами управления системой и обсуждены с заинтересованными сторонами. Обработка рисков проводится с применением модели обработки комплексных рисков.

Требования к модели обработки рисков:

- модель должна обеспечивать возможности гибкой адаптации к особенностям конкретной системы;
- модель должна учитывать временные особенности функционирования системы;
- модель должна обладать некоторой базой знаний об уже обработанных рисках и принятых для их обработки мерах;
- модель должна детализировать характеристики мероприятий, направленных на обработку рисков;
- модель должна учитывать взаимодействия одних и тех же мероприятий на различные риски, и наоборот, — воздействие отдельных рисков на мероприятия.

Анализ и мониторинг включают регулярную проверку состояния системы и ее контроль. Результаты анализа и мониторинга являются исходными данными непосредственно для управления рисками.

Детальный обзор моделей, которые могут применяться при управлении рисками, представлен в [10].

В зависимости от вида сложной организационно-технической системы, а также внутренних и внешних условий управления рисками выбранные модели должны быть адаптивными (для выполнения условия интеллектуальности управления рисками). Такую адаптивность может обеспечить база знаний, включающая наборы правил, применяемых при управлении рисками, а также наборы знаний (данных).

Все модели, применяемые при управлении рисками, могут быть связаны с упомянутой выше базой знаний. Связь может быть нескольких видов:

- выгрузка и применение правил, хранимых в базе знаний;

- обращение к данным, хранимым в базе знаний;
- выгрузка и применение правил с обращением к данным из базы знаний.

Таким образом, можно обеспечить адаптивность моделей управления рисками путем внесения изменений в базу знаний.

Предлагаемый метод позволяет:

- осуществлять интеллектуальное управление профессиональными и производственными рисками в сложных организационно-технических системах;
- формализовать управление рисками, что обеспечивает его автоматизацию и ускорит принятие решений по обработке рисков;
- проводить моделирование поведения сложной организационно-технической системы с точки зрения существующих в ней рисков.

Список литературы

1. **OHSAS 18001:2007.** Occupational health and safety management systems—Requirements.
2. **Ветошкин А. Г., Таранцева К. Р.** Техногенный риск и безопасность. Пенза: Изд-во Пенз. гос. ун-та, 2001.
3. **Надежность** технических систем и техногенный риск: Электронное учебное пособие. URL: <http://www.obzh.ru/nad>
4. **Сеньков А. В.** Анализ рисков в опасных производственных объектах на основе нечеткого дерева отказов // Тез. докл. конф. конкурса научных работ студентов "Росэнергоатом: знания молодых ядерщиков — атомным станциям". Обнинск: ИАТЭ, 2008. 48 с.
5. **Сеньков А. В.** Способ оценки производственных рисков на основе нечетких деревьев отказов // Радиоэлектроника, электротехника и энергетика: Тез. докл. XV Международной науч.-техн. конф. студентов и аспирантов. В 3 т. Т. 1. 2009. 292 с.
6. **Борисов В. В.** Комплексное управление профессиональными и производственными рисками на основе проведения сертификации работ по охране труда // Тез. докл. V Международного конгресса "Управление профессиональными рисками — приоритетное направление в сфере охраны труда Российской Федерации". М.: ФГУ "ВНИИ охраны и экологии труда". 2008. С. 35—36.
7. **Борисов В. В., Сеньков А. В.** Комплексное управление рисками в сложных организационно-технических системах // Математическая морфология. Электронный математический и медико-биологический журнал. Т. 9. Вып. 4. 2010. URL: <http://www.smolensk.ru/user/sgma/MMORPG/N-28-html/TITL-28.htm>
8. **Борисов В. В., Круглов В. В., Федулов А. С.** Нечеткие модели и сети. М.: Горячая линия — Телеком, 2007. 284 с.
9. **ISO 31000:2009.** Risk management — Principles and guidelines.
10. **ISO 31001:2009.** Risk management — Framework of managing risks.

Ю. А. Зак,

д-р техн. наук, независимый исследователь,
Аахен, Германия
e-mail: yuriy_zack@hotmail.com

Система вероятностных критериев и ограничений в задачах выбора оптимального портфеля инвестиций

Задача формирования эффективного портфеля активов ценных бумаг формулируется в виде многокритериальной задачи математического программирования с вероятностными ограничениями. На основе установленных свойств левых частей ограничений и локальных критериев эффективности задачи предложены операторы и итеративные процессы исключения подмножеств переменных, не содержащих допустимые решения, и эффективные оценки перспективности развития различных вариантов.

Разработаны алгоритмы решения задачи на основе модификаций "метода ветвей и границ", позволяющие решать сформулированные проблемы для достаточно широкого класса практических приложений.

Ключевые слова: эффективный портфель активов ценных бумаг, вероятностные ограничения, многокритериальная оптимизация, сепарабельно монотонные функции, модификации метода "ветвей и границ"

Введение

Классическая модель формирования портфеля инвестиций (модель Марковица—Тобина) сводится, как известно (см., например, [1—7]), к решению двухкритериальной оптимизационной задачи. Объем ожидаемой прибыли портфеля (*Portfolie*) рассматривается как средневзвешенная доходность входящих в него активов, а риск портфеля оценивается по известным формулам теории вероятности через вариации и ковариации этих активов. Как правило, не существует *Portfolie*, которые при высоком математическом ожидании прибыли обеспечивают минимальные проценты риска.

Марковиц решал задачу минимизации риска портфеля при обеспечении заданного уровня его доходности. При этом важным предварительным результатом Марковица было доказательство выпуклости эффективного фронта, что обеспечивает

единственность решения сформулированной им оптимизационной задачи.

Доминирующее определение риска как дисперсии доходности объясняется простотой этого измерителя и в какой-то степени традицией. Недостатки дисперсии как модели риска обсуждаются, например, в работах [9—12]. Основные среди этих недостатков — следующие:

- дисперсия определяет **все** отклонения доходности от своего математического ожидания, в то время как с термином "риск" для инвестора связаны только неблагоприятные для него отклонения;
- дисперсия не раскрывает распределения (структуры) отклонений, т. е. одна ценная бумага с преобладанием положительных отклонений может иметь такую же дисперсию, как другая ценная бумага с преобладанием отрицательных отклонений от математического ожидания прибыли. При этом от инвестора будет скрыт больший риск потерь при покупке второй из них;
- значения математического ожидания и дисперсии прибыли от сформированного портфеля инвестиций не дают инвестору полной информации о распределении вероятности ожидаемых объемов прибыли и возможных потерь, т. е. о степени риска при принятии решений.

В работах [7—13] рассматриваются следующие альтернативные измерители риска: математическое ожидание отрицательных отклонений, вероятность получения дохода меньше ожидаемого значения, а также значения моментов отклонений более высоких порядков. Вероятностные ограничения при формировании эффективного портфеля инвестиций рассматривались в монографии [12].

Предлагаемая в работе задача формирования эффективного портфеля активов ценных бумаг рассматривается в многокритериальной постановке. Представлены системы ограничений, в которых некоторые частные критерии эффективности и ограничения определены в виде вероятностей получения объемов прибыли не ниже заданного установленного уровня, а вероятности потерь более заданного установленного значения не должны быть выше некоторого установленного инвестором значения. Такая система вероятностных ограничений и критериев оптимальности дает возможность инвестору в понятных для него показателях более объективно оценить степень риска принимаемого им решения. Исследуются математические свойства сформулированных многоэкстремальных задач. Показано, что функции цели и ограничений задачи относятся

к классу сепарабельно квазимоноотонных функций, определение экстремальных значений которых осуществляется на основе простых вычислений. На основе установленных свойств предлагаются операторы и итеративные процессы выделения областей, не содержащих допустимых решений, методы вычисления грубых и точных оценок оптимальных значений критериев оптимальности для анализируемых подмножеств вариантов. На основе установленных свойств рассматриваемых задач предлагаются методы получения их точных и приближенных решений модифицированными методами "ветвей и границ".

1. Постановка и математическая модель задачи

Рассмотрим задачу в более формальной постановке. Весь объем инвестиций A может быть распределен по K направлениям (источникам финансирования). Пусть α_k — доля общего объема, направленная в k -й источник вложения активов, где

$$\sum_{k=1}^K \alpha_k = 1, 0 \leq \alpha_k \leq 1, k = 1, \dots, K. \quad (1)$$

Тогда объем инвестиций, направленный в k -й источник финансирования, равен $A_k = \alpha_k A$.

Для каждого из направлений инвестиций известны m_k — математическое ожидание объема возможной прибыли на единицу инвестируемого капитала, $V_k = \sigma_k^2$ — дисперсия доли ожидаемой прибыли.

Пусть также известны коэффициенты корреляции случайных величин $0 \leq \rho_{kl} \leq 1, k, l = 1, \dots, K$.

При данном распределении инвестиций математическое ожидание и дисперсия возможной прибыли сформированного портфеля инвестиций определяются выражениями

$$M(E) = \sum_{k=1}^K m_k A_k; \quad (2)$$

$$\sigma^2(E) = V(E) = \sum_{k=1}^K \sum_{l=1}^K A_k A_l \rho_{kl} \sigma_k \sigma_l. \quad (3)$$

Аналогом множества Парето показателей эффективности различных вариантов *Portfolie* в системе координат $m(E)$ и $\sigma^2(E)$ является кривая, для любой точки которой увеличение математического ожидания объема прибыли может быть достигнуто за счет роста доли риска получения потерь. Поэтому при формировании оптимальных *Portfolie* должны использоваться определенные схемы компромисса.

В строгой многокритериальной постановке задача выбора оптимального *Portfolie* представляется в виде

$$F_1 = \max_{\alpha_1, \dots, \alpha_k, \dots, \alpha_K} A \sum_{k=1}^K \alpha_k m_k; \quad (4)$$

$$F_2 = \min_{\alpha_1, \dots, \alpha_k, \dots, \alpha_K} A^2 \sum_{k=1}^K \sum_{l=1}^K \alpha_k \sigma_k \alpha_l \sigma_l \rho_{kl}. \quad (5)$$

В большинстве рассматриваемых в литературе публикаций (см., например, [1, 10–12]), а также в практических приложениях решение задачи осуществляется с помощью линейной свертки критериев:

$$F_3 = \max_{\alpha_1, \dots, \alpha_k, \dots, \alpha_K} \left[A \sum_{k=1}^K \alpha_k m_k - \eta A^2 \sum_{k=1}^K \sum_{l=1}^K \alpha_k \sigma_k \alpha_l \sigma_l \rho_{kl} \right], \quad (6)$$

где $0,2 \leq \eta \leq 0,3$ — некоторый весовой коэффициент.

Для лица, принимающего решение, в ряде случаев значение $\sigma^2(E)$ недостаточно ясно раскрывает картину степени его возможного риска. Было бы существенно понятнее, если бы риск вложения инвестиций был проиллюстрирован некоторой системой вероятностных показателей типа

$$p\{E \leq B_\xi\} \leq q_\xi, \xi = 1, \dots, n, \quad (7)$$

$$p\{E \geq D_\omega\} \geq g_\omega, \omega = 1, \dots, v, \quad (8)$$

т. е. вероятность того, что потери будут не выше некоторой заранее установленной величины B_ξ , должны быть больше значения q_ξ , а вероятность того, что прибыль будет больше или равна некоторой установленной величине D_ω , должна быть не меньше значения g_ω .

Пусть существует несколько относительно независимых друг от друга источников инвестирования, случайные величины показателей которых слабо коррелированы. В соответствии с предельной теоремой теории вероятностей закон распределения суммы прибылей от нескольких источников инвестиций с достаточной степенью точности может быть аппроксимирован нормальным законом распределения с математическим ожиданием и дисперсией, определяемыми выражениями (2) и (3). Значения вероятностей $p\{E \leq B_\xi\}$ и $p\{E \geq D_\omega\}$ в этом случае могут быть легко вычислены для любых значений α_k , определяемых выражением (1).

Рассмотрим постановку задачи выбора оптимального портфеля инвестиций в виде следующей модели многокритериальной оптимизации:

$$\Phi_1 = \max_{\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K} \left\{ \sum_{k=1}^K m_k [\alpha_k A] \right\}; \quad (9)$$

$$\Phi_2 = \min_{\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K} \left\{ \sum_{k=1}^K \sum_{l=1}^K [\alpha_k A] [\alpha_l A] \rho_{kl} \sigma_k \sigma_l \right\}; \quad (10)$$

$$\Phi_{3\xi} = \min_{\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K} p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \leq B_\xi\}, \quad \xi = 1, \dots, n; \quad (11)$$

$$\Phi_{4\omega} = \max_{\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K} [p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \geq D_\omega\}],$$

$$\omega = 1, \dots, v; \quad (12)$$

$$\Phi_{5t} = \max_{\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K} [\bar{B}_t : p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \leq \bar{B}_t\} \leq \beta_t], \quad t = 1, \dots, T; \quad (13)$$

$$\Phi_{6w} = \max_{\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K} [\bar{D}_w : p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \geq \bar{D}_w\} \geq \pi_w], \quad w = 1, \dots, W. \quad (14)$$

Критерии Φ_{5t} , $t = 1, \dots, T$, минимизируют значение потерь или невысоких показателей достигаемой прибыли, которые при формируемом *Portfolie* могут наступить с вероятностью, не выше заданной β_t , а критерии Φ_{6w} , $w = 1, \dots, W$, максимизируют пороговое значение прибыли, получение которой гарантируется с вероятностью, не ниже заданной π_w .

Задача выбора оптимального портфеля инвестиций может быть также сформулирована в виде многокритериальной задачи в условиях системы вероятностных ограничений, устанавливающих более жесткие требования на кривую распределения объема ожидаемой прибыли. Рассмотрим математическую модель с двумя критериями оптимальности (2) и (3), наложив на остальные показатели некоторые пороговые ограничения вида

$$p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \leq D_\omega\} \leq P_\omega^1, \quad \omega = 1, \dots, v; \quad (15)$$

$$p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \geq B_\xi\} \geq P_\xi^2, \quad \xi = 1, \dots, n; \quad (16)$$

$$\{\bar{D}_w \leq S_w^1 : p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \leq \bar{D}_w\} \leq P_w^3\}, \quad w = 1, \dots, W; \quad (17)$$

$$\{\bar{B}_t \geq S_t^2 : p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \geq B_t\} \geq P_t^4\}, \quad t = 1, \dots, T. \quad (18)$$

Применяя линейную свертку критериев (2) и (3) в виде (6), приходим к решению многоэкстремальной задачи математического программирования относительно переменных $0 \leq \alpha_k \leq 1$, $k = 1, \dots, K$, в условиях ограничений (1), (16)–(18).

Выбор эффективного решения сформулированных задач может осуществляться из конечного множества альтернатив [15]. Ниже рассматриваются свойства и методы решения многоэкстремальной задачи математического программирования, в которой вектор переменных задачи может принимать как конечное, так и бесконечное множество значений, принадлежащих n -мерному параллелепипеду $0 \leq \alpha_k \leq 1$, $k = 1, \dots, K$.

Существенной особенностью этой задачи является то, что критерий оптимальности и левые части ограничений задачи не являются аналитическими функциями от переменных задачи, и для вычисления этих значений для каждого конкретного значения вектора переменных требуется значительный

объем вычислений или моделирование системы. Однако из содержательной постановки задачи можно заключить, что эти функции цели задачи и левых частей ограничений задачи обладают некоторыми специфическими свойствами, учет которых позволит предложить эффективные алгоритмы решения сформулированной задачи.

2. Математические свойства частных критериев эффективности и левых частей ограничений задачи

2.1. Сепарабельно квазимоноотонные функции

Функцию $f(x_1, \dots, x_j, \dots, x_n)$, заданную на n -мерном параллелепипеде

$$H = H_1 \times H_2 \times \dots \times H_j \times \dots \times H_n;$$

$$H_j = \{x_j : h_j^1 \leq x_j \leq h_j^2\}, \quad j = 1, \dots, n, \quad (19)$$

будем называть сепарабельно квазимоноотонной в сильном смысле по переменной x_j (в дальнейшем будем обозначать ее $SQMF(j)$), если для любых фиксированных значений $\bar{y}_k \in H_k$, $k = 1, \dots, n$, $k \neq j$, остальных переменных и любых двух значений переменных x_j , равных $x_{1j} = g_j$, $x_{2j} = l_j$, где $h_j^1 \leq g_j < l_j \leq h_j^2$, выполняется одно из пары условий:

- для невозрастающих $SQMF(j)$ (в дальнейшем они обозначаются $S^-QMF(j)$, т. е. верхний индекс "-" означает убывающая)

$$f(\bar{y}_1, \dots, \bar{y}_{j-1}, g_j, \bar{y}_{j+1}, \dots, \bar{y}_n) \leq f(\bar{y}_1, \dots, \bar{y}_{j-1}, l_j, \bar{y}_{j+1}, \dots, \bar{y}_n); \quad (20)$$

- для неубывающих $SQMF(j)$ (в дальнейшем они обозначаются $S^+QMF(j)$, т. е. верхний индекс "+" означает возрастающая)

$$f(\bar{y}_1, \dots, \bar{y}_{j-1}, g_j, \bar{y}_{j+1}, \dots, \bar{y}_n) \geq f(\bar{y}_1, \dots, \bar{y}_{j-1}, l_j, \bar{y}_{j+1}, \dots, \bar{y}_n). \quad (21)$$

Если хотя бы для одной из переменных k или целой группы переменных $\bar{J} \subseteq J$ для некоторого подмножества значений других переменных $j = 1, \dots, n$, $j \neq k$, выполняется условие (20), а для другого, дополняющего его подмножества значений, — условие (21), то $f(x)$ назовем квазимоноотонной в слабом смысле и будем обозначать ее $SQMF$. Класс таких функций рассматривался в работе автора [14, 16].

Рассмотрим поведение и свойства таких функций в системе n -мерных параллелепипедов

$$H_s = H_{1s} \times H_{2s} \times \dots \times H_{js} \times \dots \times H_{ns}, \quad (22)$$

где $H_{js} = \{x_j : h_{js}^1 \leq x_j \leq h_{js}^2\}$, $H_{js} \subseteq H_{j,s-1}$, $s = 1, \dots, S$.

При этом допускается, что некоторые из переменных (множество этих переменных обозначим J_{Dskr}) могут принимать только целочисленные значения. Остальные переменные $J_{Np} = J/J_{Dskr}$ являются непрерывными. Систему параллелепипедов (22), в которых $J_{Dskr} \neq \emptyset$, назовем системой частично це-

лочисленных параллелепипедов (*ChZP*). Здесь и в дальнейшем будем предполагать, что если $j \in J_{Dskr}$ то h_{js}^1 и h_{js}^2 — допустимые целочисленные значения.

Рассмотрим множество функций $f^i(x_1, \dots, x_j, \dots, x_n)$, $i = 0, 1, \dots, m$, каждая из которых является $S^-QMF(j)$ либо $S^+QMF(j)$ по каждой из переменных.

Обозначим:

J_i^- и J_i^+ — соответственно подмножества переменных, по которым $f^i(x_1, \dots, x_j, \dots, x_n)$ является функцией $S^-QMF(j)$ и $S^+QMF(j)$;

$Y_s^{(-j)}$ — некоторый вектор значения всех переменных задачи, за исключением переменной x_j , каждая из которых принадлежит подмножеству H_{ks} , т. е. $x_k \in H_{ks}$, $k = 1, \dots, n$, $k \neq j$.

Для любых значений векторов $Y_s^{(-j)} \in H_s/H_{1s}$ справедливо

$$x_{js}^{i-} = \arg \min_{x_j \in H_{js}} f(x_j | Y_s^{(-j)}) = \begin{cases} h_{js}^1, & \text{если } j \in J_i^+, \\ h_{js}^2, & \text{если } j \in J_i^-, \end{cases} \quad j = 1, \dots, n, i = 0, 1, \dots, m, \quad (23)$$

$$x_{js}^{i+} = \arg \max_{x_j \in H_{js}} f(x_j | Y_s^{(-j)}) = \begin{cases} h_{js}^2, & \text{если } j \in J_i^+, \\ h_{js}^1, & \text{если } j \in J_i^-, \end{cases} \quad j = 1, \dots, n, i = 0, 1, \dots, m. \quad (24)$$

То есть минимальное и максимальное значения таких функций по каждой из переменных x_j при фиксации остальных переменных в пределах параллелепипеда H_s достигается на одном из концов отрезка ее допустимых значений H_{js} .

Обозначим

$$f^{i-}(x) = \min_{x \in H_s} f^i(x), f^{i+}(x) = \max_{x \in H_s} f^i(x), \quad (25)$$

а x_{js}^{i-} , x_{js}^{i+} , $j = 1, \dots, n$ — соответственно значения переменных, при которых достигается $f^{i-}(x)$ и $f^{i+}(x)$, если $x \in H_s$, т. е. в одной из 2^n вершин n -мерного параллелепипеда (22).

2.2. Оценка возможности выполнения системы ограничений задачи

Следует заметить, что левые части ограничений задачи (15)—(18) при фиксированных значениях математических ожиданий, дисперсий и коэффициентов корреляции случайных величин являются сепарабельно невозрастающими слабомонотонными функциями каждой из переменных задачи $(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K)$, свойства которых были рассмотрены выше. Следовательно, если определены граничные значения всех переменных задачи для s -го подмножества значений переменных

$$\alpha \in H^s = \{\alpha_1, \alpha_2, \dots, \alpha_K | h_k^{1s} \leq \alpha_k \leq h_k^{2s}\}, \quad \text{где } 0 \leq h_k^{1s} \leq h_k^{2s} \leq 1, k = 1, \dots, K, \quad (26)$$

то граничная оценка минимального значения функций $p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \leq D_\omega\}$, $\omega = 1, \dots, v$;

и значения $\bar{D}_W$ в выражении $\{\bar{D}_W \geq S_w^2 :$

$p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \leq \bar{D}_W\} \leq P_w^4\}$, $w = 1, \dots, W$,

для этого подмножества переменных значений достигается при значениях $\alpha_k = h_k^1$, $k = 1, \dots, K$. А граничная оценка максимального значения функций

$p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \geq B_\xi\}$, $\xi = 1, \dots, n$, и значения

$\bar{B}_t$ в $\{\bar{B}_t \geq S_t^1 : p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \geq \bar{B}_t\} \geq$

$\geq P_t^3\}$, $t = 1, \dots, T$, — при значениях $\alpha_k = h_k^2$, $k = 1, \dots, K$.

То есть

$$\Psi_{1\omega} [p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \leq D_\omega\}] = p\{E(h_1^{1s}, h_2^{1s}, \dots, h_k^{1s}, \dots, h_K^{1s}) \leq D_\omega\}, \omega = 1, \dots, v; \quad (27)$$

$$\{\Psi_{4w} [\bar{D}_w] \leq S_w^1 : p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \leq \bar{D}_w\} \leq P_w^3\} = \{\bar{D}_w \leq S_w^1 : p\{E(h_1^{1s}, h_2^{1s}, \dots, h_k^{1s}, \dots, h_K^{1s}) \leq \bar{D}_w\} \leq P_w^1\}, w = 1, \dots, W; \quad (28)$$

$$\Psi_{2\xi} [p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \geq B_\xi\}] = p\{E(h_1^{2s}, h_2^{2s}, \dots, h_k^{2s}, \dots, h_K^{2s}) \geq B_\xi\}, \xi = 1, \dots, n; \quad (29)$$

$$\{\Psi_{4t} [\bar{B}_t] \leq S_t^2 : p\{E(\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_K) \geq \bar{B}_t\} \leq P_t^4\} = \{\bar{B}_t \leq S_t^2 : p\{E(h_1^{2s}, h_2^{2s}, \dots, h_k^{2s}, \dots, h_K^{2s}) \geq \bar{B}_t\} \leq P_t^4\}, t = 1, \dots, T. \quad (30)$$

Если не выполняется хотя бы одно из ограничений задачи (27)—(30), то подмножество переменных задачи $\alpha \in H^s$ не содержит значений переменных, удовлетворяющих всей системе ограничений, и оно может быть отброшено как неперспективное.

2.3. Некоторые свойства систем неравенств, левая часть которых — сепарабельно квазимоноотонные функции

Рассмотрим систему ограничений вида

$$f^i(x_1, \dots, x_j, \dots, x_n) \leq b^i, i = 1, 2, \dots, m. \quad (31)$$

Отметим, что все ограничения вида $f^i(x_1, \dots, x_j, \dots, x_n) \geq b^i$ и $f^i(x_1, \dots, x_j, \dots, x_n) = b^i$ легко могут быть сведены к ограничениям вида (31).

Определим соответственно $Q_{s, \min}^{i, (-j)}$, $Q_{s, \max}^{i, (-j)}$ — векторы значения всех переменных $k = 1, \dots, n$, $k \neq j$ (все, кроме переменной j), компоненты которых соответственно равны x_{ks}^{i-} и x_{ks}^{i+} и вычисляются в соответствии с выражениями (23), (24).

Вычислим

$$t_{js}^{i,+} = \begin{cases} \arg \max_{x_j \in H_{js}} \{x_j | f^i(x_j | Q_{s, \min}^{i, (-j)}) \leq b^i\}, \\ \text{если } j \in J^{i,+}, \\ h_{js}^2, \text{ если } j \in J^{i,-}, \\ i = 1, \dots, m; \end{cases} \quad (32)$$

$$t_{js}^{i,-} = \begin{cases} \arg \min_{x_j \in H_{js}} \{x_j | f^i(x_j | Q_{s, \min}^{i, (-j)}) \leq b^i\}, \\ \text{если } j \in J^{i,+}, \\ h_{js}^1, \text{ если } j \in J^{i,-}, \\ i = 1, \dots, m. \end{cases} \quad (33)$$

Подмножества $\bar{H}_{js}^-$ и $\bar{H}_{js}^+$ определим следующим образом:

$$\begin{aligned} \bar{H}_{js}^- &= \{x_j | h_{js}^1 \leq x_j < \bar{t}_{js}^-\}, \quad \bar{H}_{js}^+ = \{x_j | \bar{t}_{js}^+ < x_j \leq h_{js}^2\}, \\ \bar{H}_{js} &= \bar{H}_{js}^- \cup \bar{H}_{js}^+, \end{aligned} \quad (34)$$

где

$$\bar{t}_{js}^- = \max_{1 \leq i \leq m} t_{js}^{i,-}, \quad \bar{t}_{js}^+ = \min_{1 \leq i \leq m} t_{js}^{i,+}. \quad (35)$$

Определим

$$\tilde{H}_{js} = \{x_j | \bar{t}_{js}^- \leq x_j \leq \bar{t}_{js}^+\}, j = 1, \dots, n. \quad (36)$$

Приведем без доказательства несколько очевидных утверждений.

1. Если для некоторого индекса $j \in J$ справедливо неравенство $\bar{t}_{js}^- > \bar{t}_{js}^+$, то система неравенств (31) в условиях $x \in H_s$ не имеет допустимых решений.

2. Подмножество $\bar{H}_{js} = \bar{H}_{js}^- \cup \bar{H}_{js}^+$, определенное в соответствии с формулами (32)–(35), не содержит допустимых планов, обеспечивающих решение системы неравенств (31) в условиях $x \in H_s$.

3. Подмножество

$$\begin{aligned} \bar{H}_s(j) &= H_{1s} \times \dots \times H_{j-1,s} \times \bar{H}_{js} \times \\ &\times H_{j+1,s} \times \dots \times H_{ns}, \end{aligned} \quad (37)$$

где $\bar{H}_{js}$ определяется согласно выражениям (32)–(35), не содержит допустимых планов.

4. Допустимое решение системы неравенств (31) может искаться только в подмножестве переменных, которое определяется по формуле

$$\begin{aligned} H_{s+1} &= H_{1s} \times \dots \times H_{j-1,s} \times \\ &\times H_{j,s+1} \times \dots \times H_{j+1,s} H_{ns}, \quad H_{j,s+1} = H_{js} / \bar{H}_{js}^i, \end{aligned} \quad (38)$$

где подмножества $H_{j,s+1}$ определяются согласно выражению

$$\begin{aligned} H_{j,s+1} &= \{x_j | h_{j,s+1}^1 \leq x_j \leq h_{j,s+1}^2\}, \\ \text{где } h_{j,s+1}^1 &= t_{js}^1, \quad h_{j,s+1}^2 = t_{js}^2. \end{aligned} \quad (39)$$

2.4. Итеративный процесс исключения областей, не содержащих допустимых планов

На основе отмеченных выше свойств сепарабельно квазимонотонных функций определим следующий итеративный процесс исключения подмножеств, не содержащих допустимых планов, который в дальнейшем будем называть ItP_1.

Введем счетчик числа шагов M , в течение которых $H_{s+1} = H_s$, т. е. не происходит изменения n -мерного параллелепипеда ни по одной из переменных. В начале процесса положим $s = 0$, $M = 0$, $H_{j,0} = H_j$, $j = 1, \dots, n$. На каждом s -м шаге итеративного процесса в цикле для каждой переменной $j = 1, \dots, n$ выполняем вычисления в соответствии с выражениями (32)–(34), (38), (39).

Если на некотором s -м шаге для некоторого индекса переменной $j \in J$ выполняются условия $H_{j,s+1} = \emptyset$, то система ограничений задачи не содержит допустимых планов, и итеративный процесс останавливается.

Если на некотором s_1 -м шаге справедливо $H_{s_1+1} = H_{s_1}$ (т. е. $\bar{H}_{js_1} = \emptyset$), то значение M увеличиваем на 1. Если $M = n$, то полученное подмножество H_{s_1+1} не может быть подвергнуто дальнейшему изменению (сужению), и итеративный процесс останавливается, т. е. дальнейшее исключение областей, не содержащих допустимых планов, на основе описанного алгоритма невозможно. Если $H_{s_1+1} \subset H_{s_1}$ и при этом $M > 0$, то полагаем $M = 0$ и переходим к следующему шагу процесса.

Ясно, что через конечное число шагов процесса мы либо придем к факту о несовместности исходной системы ограничений, либо получим n -мерный параллелепипед $\tilde{H} = H_{s+1} \neq \emptyset$, который не может быть в дальнейшем подвергнут изменениям.

Отметим, что успешное завершение этого итеративного процесса и получение в результате этого n -мерного параллелепипеда $\tilde{H} = H_{s+1} \neq \emptyset$ не гарантирует, что система ограничений исходной задачи совместна, а является только необходимым условием существования допустимых решений. Обозначим H_s — полученное после завершения описанного итеративного процесса множество переменных задачи.

3. Алгоритм решения задачи методом "ветвей и границ"

3.1. Вычисление верхней границы компромиссного критерия эффективности

Грубая оценка оптимального значения критерия эффективности. Принимая во внимание, что математическое ожидание процента ожидаемой прибыли (потерь) и среднее квадратическое отклонение этого показателя также являются неубывающими функциями от каждой из переменных задачи $0 \leq \alpha_k \leq 1$, $k = 1, \dots, K$, верхняя

граница значения компромиссного критерия оптимальности для множества переменных задачи $\alpha \in H^s$ может быть вычислена согласно выражению

$$\Omega_1[F_3(H^s)] = \left[A \sum_{k=1}^K m_k h_k^{2s} - \eta A^2 \sum_{k=1}^K \sigma_k \alpha_k \sigma_l \alpha_l \rho_{kl} h_k^{1s} \right], \quad s = 1, \dots, S. \quad (40)$$

Точная оценка оптимального значения критерия эффективности. Более точная оценка верхней границы значения компромиссного критерия учитывает ограниченный объем инвестиций, требует большего объема вычислений и может быть определена на основе следующего алгоритма.

Пусть задано $\alpha \in H^s = \{\alpha_1, \alpha_2, \dots, \alpha_K | h_k^{1s} \leq \alpha_k \leq h_k^{2s}\}$, где $0 \leq h_k^{1s} \leq h_k^{2s} \leq 1, k = 1, \dots, K$, — некоторое подмножество допустимых значений переменных задачи.

Среди всех переменных выделим подмножество переменных

$$\tilde{K}_1^s = \{\alpha^s \in \bar{H}^s | h_r^{1s} = h_r^{2s}, r = 1, \dots, K_1^s, K_1^s < K\}. \quad (41)$$

Для данного подмножества определено значение переменных α_r^s .

Все подмножество индексов переменных обозначим $\tilde{K}, \tilde{K}_1^s \subseteq \tilde{K}, \tilde{K}_2^s \subseteq \tilde{K} / \tilde{K}_1^s$.

$$\text{Вычислим } \Lambda^s = \sum_{r=1}^{K_1^s} \alpha_r^s. \text{ Если } \Lambda^s \leq 1, \text{ то } \hat{\Lambda}^s = \sum_{k \in \tilde{K}_2^s} \alpha_k^s = 1 - \Lambda^s.$$

Упорядочим математические ожидания $m_k, k \in \tilde{K}_2^s$, по убыванию этих значений в последовательности

$$U^s = \{k_\delta \in \tilde{K}_2^s | m_{k_1} \geq m_{k_2} \geq \dots \geq m_{k_{P^s}}\}. \quad (42)$$

Здесь P^s — число членов в последовательности U и в подмножестве $\tilde{K}_2^s$.

Выберем первые индексы переменных последовательности (43) (если для каждого из них выполняются условия $h_{k_\delta}^{1s} \leq h_{k_\delta}^{2s} \leq (1 - \Lambda^s)$) так, чтобы выполнялись условия

$$\sum_{\delta=1}^{\Delta} h_{k_\delta}^{2s} + x_{k_{\Delta+1}}^s = 1 - \Lambda^s, \quad (43)$$

$$\text{где } h_{k_{\Delta+1}}^{1s} \leq x_{k_{\Delta+1}}^s \leq h_{k_{\Delta+1}}^{2s}.$$

Выбранные из подмножества $r \in \tilde{K}_2^s$ значения Δ переменных для оценки верхней границы математического ожидания процента (или объема) ожидаемой прибыли обозначим $x_{k_\delta}^s, \delta = 1, \dots, \Delta$, где $\Delta \leq P^s$.

Для каждой из переменных $r \in \tilde{K}_2^s$ вычислим значения

$$f_1(r) = \sum_{k \in \tilde{K}_1^s} \sum_{r \in \tilde{K}_2^s} \rho_{kr} \sigma_k \sigma_r + \sum_{r \in \tilde{K}_2^s} \sigma_r^2, r \in \tilde{K}_2^s. \quad (44)$$

Упорядочим индексы переменных задачи $r \in \tilde{K}_2^s$ в порядке возрастания значений $f_1(r)$:

$$W^s = \{r_1 \in \tilde{K}_2^s | f_1(r_1) \leq f_1(r_2) \leq \dots \leq f_1(r_p) \leq \dots \leq f_1(r_{P^s})\}. \quad (45)$$

Выбираем первый член ряда (45), т. е. переменную r_1 и значение этой переменной $y_{r_1}^s = h_{2r_1}^s$. Если

$y_{r_1}^s = h_{2r_1}^s = \tilde{\Lambda}^s$, то переходим к расчетным формулам вычисления точной оценки. В противном случае находим $\hat{\Lambda}_1^s = \hat{\Lambda}^s - h_{r_1}^s$. Выбираем следующий ($p = 2, 3, \dots$) член ряда (45). Определяем

$$y_{r_p}^s = \min(\hat{\Lambda}_{r_p}^s; h_{2r_{p+1}}^s), \hat{\Lambda}_{r_p}^s = (\hat{\Lambda}_{r_{p-1}}^s - y_{r_p}^s), \quad p = 1, \dots, P^s. \quad (46)$$

Эту процедуру продолжаем до тех пор, пока не будет выбран некоторый θ -й индекс, для которого $\Lambda_{r_\theta}^s = 0$.

Более точная оценка максимального значения компромиссного критерия эффективности для подмножества вариантов $\alpha \in H^s$ рассчитывается по формуле

$$\Omega_3[F_3(H^s)] = A \sum_{k \in \tilde{K}_1^s} m_k \alpha_k^s + \sum_{\substack{\delta=1 \\ r_\delta \in \tilde{K}_2^s}}^{\Delta} m_{r_\delta} x_{r_\delta}^s - \eta A^2 \left\{ \sum_{k \in \tilde{K}_1^s} \sum_{l \in \tilde{K}_1^s} \sigma_k \alpha_k^s \sigma_l \alpha_l^s \rho_{kl} + \sum_{k \in \tilde{K}_1^s} \sum_{\substack{p=1 \\ r_p \in \tilde{K}_2^s}}^{\Delta} \sigma_k \alpha_k^s \sigma_{r_p} y_{r_p}^s \rho_{kr_p} + \sum_{\substack{p=1 \\ r_p \in \tilde{K}_2^s}}^{\Delta} y_{r_p}^s \sigma_{r_p}^2 \right\}. \quad (47)$$

3.2. Вычислительная схема алгоритма решения

Рассмотрим схему метода "ветвей и границ" применительно к решению данной задачи.

На начальном этапе решения определим

$$H^0 = \{\alpha_k | 0 \leq \alpha_k \leq 1, k \in \bar{K} = \{1, \dots, k, \dots, K\}\}, \quad (48)$$

$$\Omega_3[F_3(H^0)] = A m_{k_1} - \eta A^2 \sigma_{r_1}^2, \quad (49)$$

$$\text{где } m_{k_1} = \max_{1 \leq l \leq K} (m_l), \sigma_{r_1}^2 = \min_{1 \leq p \leq K} (\sigma_{r_p}^2).$$

Итеративный процесс решения задачи состоит из следующих шагов.

1. Выбираем некоторое подмножество переменных $H^s, s = 0, 1, \dots, S$, задачи с наилучшим значе-

нием границы функции эффективности — $H^{\bar{s}} = \max_{1, \dots, s, \dots, S} \Omega_3[F_3(H^s)]$. Если допустимых подмножеств переменных задачи не существует, то процесс решения задачи останавливается. Это означает, что невозможно обеспечить установленные вероятностные граничные значения и соотношения значений ожидаемых потерь и прибыли. Эксперту или лицу, принимающему решение, в этом случае необходимо ослабить установленные им требования.

2. Проверяем возможность выполнения всех ограничений задачи в соответствии с выражениями (29)—(32). Выполняем итеративные процессы ItP_1 исключения подобластей переменных, не содержащих допустимых планов. Если хотя бы одно из неравенств не выполняется, то подмножество значений переменных H^s не содержит допустимых решений и отбрасывается как неперспективное. Корректируем индексацию оставшихся подмножеств значений переменных и переходим к п. 1. В противном случае переходим к п. 3.

3. Если у выбранного подмножества переменных задачи выполняются условия $h_{1k}^s = \alpha_k^s = h_{2k}^s$, $k = 1, \dots, K$, то получено оптимальное решение задачи, и алгоритм завершает работу. В противном случае переходим к п. 4.

4. Осуществляем разбиение анализируемого подмножества решений задачи на подмножества (ветвление). Выделяем подмножество индексов переменных задачи $\bar{J}_1^{\bar{s}} = \{j_1 \in \bar{K}^{\bar{s}} | h_{1j_1}^{\bar{s}} < h_{2j_1}^{\bar{s}}\}$.

Выбираем какой-либо индекс переменных задачи, удовлетворяющий одному из следующих требований:

$$\bar{k} = \{j_1 \in \bar{J}_1^{\bar{s}} | m_{\bar{k}} = \max_{j_1 \in \bar{J}_1^{\bar{s}}} m_{j_1}\},$$

$$\bar{k} = \{j_1 \in \bar{J}_1^{\bar{s}} | \sigma_{\bar{k}} = \max_{j_1 \in \bar{J}_1^{\bar{s}}} \sigma_{j_1}\}; \quad (50)$$

$$\bar{k} = \{j_1 \in \bar{J}_1^{\bar{s}} | (h_{2\bar{k}}^{\bar{s}} - h_{2j_1}^{\bar{s}}) = \max_{j_1 \in \bar{J}_1^{\bar{s}}} (h_{2j_1}^{\bar{s}} - h_{1j_1}^{\bar{s}})\}. \quad (51)$$

Разбиваем подмножество $H^{\bar{s}}$ на два подмножества

$$\bar{H}^{1s} = \{h_{1\bar{k}}^{\bar{s}} \leq \alpha_{\bar{k}} \leq 0,5(h_{2\bar{k}}^{\bar{s}} + h_{1\bar{k}}^{\bar{s}}); h_{1j}^{\bar{s}} \leq \alpha_j \leq h_{2j}^{\bar{s}}, \\ j = 1, \dots, K; j \neq \bar{k}\}, \quad (52)$$

$$\bar{H}^{2s} = \{0,5(h_{2\bar{k}}^{\bar{s}} + h_{1\bar{k}}^{\bar{s}}) < \alpha_{\bar{k}} \leq h_{1\bar{k}}^{\bar{s}}; h_{1j}^{\bar{s}} \leq \alpha_j \leq h_{2j}^{\bar{s}}, \\ j = 1, \dots, K; j \neq \bar{k}\}. \quad (53)$$

Подмножество $H^{\bar{s}}$ исключаем из дальнейшего рассмотрения.

Проводим переиндексацию всех оставшихся подмножеств и переходим к п. 1.

Через некоторое число шагов работы алгоритма либо будет получено решение задачи с заданной точностью, либо установлен факт о несовместности исходной системы ограничений.

Заключение

При формировании эффективного портфеля активов ценных бумаг для лица, принимающего решение, в ряде случаев значение дисперсии функции распределения объема ожидаемой прибыли недостаточно ясно раскрывает картину степени его возможного риска. Предлагаемая в работе система ограничений и локальных критериев эффективности в виде вероятностей получения объемов прибыли (не ниже заданного установленного уровня) и вероятности потерь (не более заданной, установленной инвестором величины) позволяет более объективно оценить эффективность и степень риска принимаемого им решения.

В работе установлены математические свойства сформулированных многоэкстремальных задач. Показано, что функции левых частей ограничений и локальных критериев эффективности являются сепарабельно монотонными функциями по каждой из переменных задачи, значение глобального экстремума которых на n -мерном параллелепипеде находится в одной из его вершин и может быть определено по результатам простых вычислений.

На основании установленных свойств предложены операторы и итеративные процессы выделения подмножеств переменных, не содержащих допустимых решений, а также эффективные оценки перспективности рассмотрения различных вариантов. Предложенные в работе модификации алгоритмов решения задачи методом "ветвей и границ" делают возможным решение данных важных проблем для практических приложений.

Список литературы

1. Markowitz H. M. Portfolio Selection // The Journal of Finance. 1952. 7(1). P. 77—91.
2. Markowitz H. M. Portfolio Selection // Journal of Finance. 1952. 5. Chapter 5.
3. Markowitz H. M. The Utility of Wealth // The Journal of Political Economy (Cowles Foundation, Paper 57). 1952. LX(2). P. 151—158.
4. Markowitz H. M. The Elimination Form of the Inverse and Its Application to Linear Programming // Management Science. 1957. 3(3). P. 255—269.
5. Markowitz H. M. Portfolio Selection: Efficient Diversification of Investments. // John Wiley & Sons, New York, 1959. (reprinted by Yale University Press, 1970, 2nd ed. Basil Blackwell, 1991).
6. Markowitz H. M., Van Dijk E. Single-Period Mean-Variance Analysis in a Changing World // Financial Analysis Journal. 2003. 59(2). P. 30—44.
7. Markowitz H. M. Market Efficiency: A Theoretical Distinction and So What? // Financial Analysis Journal. 2005. 61(5). P. 17—30.
8. Шарп У. Ф., Гордон Дж. А., Бейли Д. В. Инвестиции. М.: Инфра-М, 1997.
9. Huang Chi-Fu. Foundations for Financial Economics // Elsevier Science Publishing Co., 1988.
10. Шведов А. С. Теория эффективных портфелей ценных бумаг. М.: Изд-во ГУ ВШЭ, 1999.
11. Первозванский А. А., Первозванская Т. Н. Финансовый рынок: расчет и риск. М.: Инфра-М, 1994.
12. Кан Ю. С., Кибзун А. И. Задачи стохастического программирования с вероятностными критериями. М.: Физматлит, 2009. 372 с.
13. Ковалев В. В. Введение в финансовый менеджмент. М.: Финансы и статистика. М., 2003. 768 с.

14. Michalowski W., Ogryczak W. Extending MAD Portfolio Optimization Model to Incorporate Downside Risk Aversion // IASA, IR-98-041, June 1998.

15. Зак Ю. А. Вычислительные схемы последовательных алгоритмов оптимизации. // Автоматика и вычислительная техника. 1980. № 1. С. 46—55.

16. Зак Ю. А. Методы оценки риска в задачах выбора эффективного портфеля ценных бумаг. Принятие решений из конечного множества альтернатив // Управляющие системы и машины. 2010. № 4. С. 84—92.

17. Зак Ю. А. Методы многоэкстремальной оптимизации в условиях ограничений для сепарабельно квазимоноотонных функций // Известия РАН. Теория и системы управления. 2011. № 2. 22 с. (в печати).

ПРИЛОЖЕНИЯ

Доказательства утверждений, приведенных в разделе 2.3.

Утверждение 1

Если существует такое число t_{js}^i , вычисленное согласно выражениям (32), (33), то подмножество $\bar{H}_{js} = \bar{H}_{js}^- \cup \bar{H}_{js}^+$, определенное в соответствии с формулой (34), не содержит допустимых решений, удовлетворяющих неравенству (31), в условиях, что $x \in H_s$.

Если задача (32) или (33) не имеет решения (t_{js}^i не существует), то подмножество H_s не содержит допустимых решений неравенства (31).

Доказательство

Из свойств $SQMF(j)$ (21), (22) как следствие следует справедливость (31) в случае

$$\bar{y}_k = x_{js}^{i,-}, k = 1, \dots, n, k \neq j,$$

т. е. при подстановке в эти выражения компонент вектора $Q_{s, \min}^{i, (-j)}$.

Предположим, что найдется такое значение $p_{js} \in \bar{H}_{js}$, что $p_{js}^i > t_{js}^i$, если $-j \in J_s^i$, либо $p_{js}^i < t_{js}^i$, если $j \in J_s^{i,+}$, и $f^i(p_{js}^i | Q_{s, \min}^{i, (-j)}) \leq b^i$, т. е. $\bar{y}_k \in Q_{s, \min}^{i, (-j)} \forall k \in J_s^{i,-}$.

Здесь $J_s^{i,-}$ и $J_s^{i,+}$ — соответственно подмножества переменных, по которым функция (31) является $S^-QMF(j)$ и $S^+QMF(j)$.

Следовательно, должно выполняться неравенство

$$f^i(p_{js}^i | Q_{s, \min}^{i, (-j)}) \leq f^i(t_{js}^i | Q_{s, \min}^{i, (-j)}).$$

Но это противоречит паре условий (21), (22). Возникшие противоречия являются доказательством утверждения 1.

Утверждение 2

Если справедливо утверждение 1, то не существует никаких других значений переменных $t_{ks}^i \in H_{ks}$, $k = 1, \dots, n$, $k \neq j$, таких что

$$t_{ks}^i > h_{ks}^1, \text{ если } k \in J_s^{i,-}, \\ \text{и } t_{ks}^i < h_{ks}^2, \text{ если } +k \in J_s^i, \quad (\text{П.1})$$

что найдется некоторое другое значение $p_{js}^i \in H_{js}$, которое обеспечит выполнение неравенства

$$f^i(p_{js}^i | Y_s^{(-j)}) \leq b^i,$$

в котором значения одной, группы или всех переменных $k \in J/\{j\}$ выбраны согласно условиям (П.1).

Доказательство

Из утверждения 1 следует справедливость неравенства

$$f^i(p_{js}^i | Q_{s, \min}^{i, (-j)}) > b^i. \quad (\text{П.2})$$

Замена в неравенстве (П.2) одного, группы или всех значений $x_{ks}^{i,-}$ на значение t_{ks}^i , $k = 1, \dots, n$, $k \neq j$, взятых из условий (П.1), может только усилить неравенство (33), что доказывает справедливость утверждения 2.

Утверждение 3

1. Если для некоторого индекса $j \in J$ справедливо неравенство $\bar{t}_{js}^- > \bar{t}_{js}^+$, то система неравенств (31) в условиях $x \in H_s$ не имеет допустимых решений.

2. Подмножество $\bar{H}_{js} = \bar{H}_{js}^- \cup \bar{H}_{js}^+$, определенное в соответствии с формулами (38), (39), не содержит допустимых планов, обеспечивающих решение системы неравенств (31) в условиях $x \in H_s$.

Доказательство

1. Если $\bar{t}_{js}^- > \bar{t}_{js}^+$, то $\tilde{H}_{js} = \emptyset$ и, следовательно, не существует значений x_j , удовлетворяющих системе ограничений (31), и первое положение утверждения 3 доказано.

2. Предположим, что $\tilde{H}_{js}$ содержит допустимые значения x_j и рассмотрим сопутствующие этой посылке следующие два условия.

- Пусть найдется такое значение $p_{js} \in H_{js}$, $p_{js} < \bar{t}_{js}^-$, удовлетворяющее всем ограничениям

$$f^i(p_{js} | Q_{s, \min}^{i, (-j)}) \leq b^i, i = 1, \dots, m.$$

Тогда найдется такое ограничение i_1 , для которого $j \in J^{i_1,+}$, а $t_{js}^{i_1} = \bar{t}_{js}^- > p_{js}$. В соответствии с утверждениями 1 и 2 справедливо

$$f^{i_1}(p_{js} | Y_s^{(-j)}) > b^{i_1},$$

т. е. i_1 -е ограничение при значении $x_j = p_{js}$ не выполняется ни при каких значениях других переменных $x_{ks} \in H_{ks}$, $k = 1, \dots, n$, $k \neq j$.

- Пусть найдется такое значение $l_{js} \in H_{js}$, $l_{js} > \bar{l}_{js}^+$, удовлетворяющее всем ограничениям

$$f^i(l_{js} | Q_{s, \min}^{i, (-j)}) \leq b^i, i = 1, \dots, m.$$

В этом случае найдется такое ограничение i_2 , для которого $j \in J^{i_2, -}$ и $t_{js}^{i_2} = \bar{l}_{js}^+ < l_{js}$. В соответствии с утверждениями 1 и 2 справедливо

$$f^{i_2}(l_{js} | Y_s^{(-j)}) > b^{i_2},$$

т. е. i_2 -е ограничение при значении $x_j = l_{js}$ не выполняется ни при каких значениях других переменных $x_{ks} \in H_{ks}$, $k = 1, \dots, n$, $k \neq j$.

Но эти два условия противоречат исходной посылке, что подмножество $\bar{H}_{js}$ содержит допустимые планы. Два возникших противоречия доказывают второе положение утверждения 3.

Следствия утверждения 3

1. Подмножество

$$\bar{H}_s(j) = H_{1s} \times \dots \times H_{j-1,s} \times \bar{H}_{js} \times H_{j+1,s} \times \dots \times H_{ns},$$

где $\bar{H}_{js}$ определяется согласно выражениям (32)–(35), не содержат допустимых планов.

2. Допустимое решение системы неравенств (31) может искаться только в подмножестве переменных H_{s+1} , которое определяется в соответствии с формулами (38), (39).

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 004.942

С. Б. Беневоленский*, д-р техн. наук, зав. каф.,

Ю. И. Кириллов*, ассистент,

Д. М. Колосьянко*, студент,

В. Э. Пожар**, д-р физ.-мат. наук, зав. лаб.,

К. В. Пожар*, студент,

e-mail: electron-inform@mail.ru;

В. И. Пустовойт**,

д-р физ.-мат. наук, акад. РАН, директор

***"МАТИ" — Российский государственный технологический университет им. К. Э. Циолковского

**Научно-технологический центр

уникального приборостроения РАН

Совершенствование программного обеспечения для обработки данных, получаемых с акустооптических спектрометров

Рассматриваются проблемы, возникающие при обработке информации, получаемой с акустооптического спектрометра, а также возможные пути их решения. Показано, что необходимо совершенствование подходов к обработке спектральной информации и описаны особенности разрабатываемого программного обеспечения, которое позволит повысить удобство и эффективность обработки данных, полученных с акустооптического спектрометра.

Ключевые слова: программное обеспечение для акустооптического спектрометра, анализ спектров, обработка спектров, хранение информации, графическое отображение информации, информационное ядро

Результатом исследований последних лет явилось создание акустооптических спектральных приборов для измерения спектрального распределения оптического излучения в видимом, ультрафиолетовом и инфракрасном диапазонах [1]. В отличие от традиционных спектральных приборов, использующих в качестве диспергирующего элемента дифракционную решетку, акустооптические приборы основаны на оригинальной физической идее об использовании дифракции света на управляемой акустической волне, реализованной в виде перестраиваемых оптических фильтров [2, 3]. На этой основе оказалось возможным создание акустооптических спектрометров (АОС), которые имеют ряд существенных преимуществ, значительно расширяющих область их использования.

На сегодняшний день обзор результатов разработок и исследований показывает, что направление, связанное с созданием и использованием АОС, представляет собой самостоятельную область науки и техники, которую можно назвать акустооптической спектрометрией [4]. Она характеризуется не только расширением области применения стандартных спектрометрических методов, но и созданием принципиально новых методов анализа, основанных на применении аппаратуры с уникальными функциональными возможностями. Новые возможности приводят к постановке новых задач, среди которых можно выделить:

- определение оптимального алгоритма спектрального анализа для определенного типа объектов и построение на этой основе адаптивных методов измерений;
- разработка и обоснование оптимальных методов спектрально-пространственного анализа изображений, в частности с помощью визуализации характеристических свойств объектов;
- изучение модуляционных методов анализа с помощью АОС.

Эти задачи требуют специализированного программного обеспечения (ПО), однако существующее в данный момент ПО функционирует на платформе Excel и Origin, что сужает возможности по обработке информации и, тем самым, снижает ее эффективность. Это обусловлено тем, что данные, получаемые с акустооптических спектрометров, представляют собой массив вещественных чисел. Для импортирования этого массива в Origin или Excel и последующей обработки необходимо проделывать ряд однотипных операций по "разметке" файла. При этом форматы файлов Origin и Excel различны, что также доставляет неудобства при переносе данных из одной программной оболочки в другую. Таким образом, при обработке акустооптической информации налицо недостатки в автоматизации, эргономичности и удобстве использования указанных программных средств, что обуславливает необходимость разработки новой программной среды, которая предлагала бы пользователю удобный дружественный интерфейс, позволяла обрабатывать получаемые акустооптические данные с помощью специализированного математического аппарата и имела встроенную базу данных (БД) для хранения спектральной информации. БД должна позволять анализировать спектральные данные нестационарных процессов, изменяющиеся с течением времени. Кроме того, нужно отметить, что множество спектральных точек представляет собой неравномерно распределенное дискретное множество, поэтому подходы, разработанные для сплошных спектров как непрерывных функций, неприменимы. Также и графическое отображение таких спектров может быть разным, что порождает многовариантность представлений.

Отличительной особенностью разрабатываемого ПО является сценарный подход к обработке и анализу спектральной информации. Сценарий в данном случае — это упорядоченный набор операций обработки спектральных данных, представленных в виде спектральных линий. Такой подход позволяет создать интуитивно понятный интерфейс для графического программирования с использованием принципа потока данных, что очень ценно, так как графический язык программирования позволяет представить сценарий обработки данных в виде

блок-схемы, понятной даже неподготовленному пользователю.

В основе разрабатываемого ПО лежит ядро, объединяющее все функциональные модули разрабатываемого программного обеспечения и включающее в себя две компоненты (рис. 1, см. третью сторону обложки):

- информационное ядро (ИЯ), представляющее собой базу данных, интегрированную с набором программных средств, организующих передачу рабочих данных на всех этапах решения исследовательских задач, включая работу с архивом сценариев;
- блок взаимодействия с пользователем, который представляет собой среду разработки/выполнения сценариев — набор программных средств, предназначенных для создания, выполнения, хранения (в отдельной библиотеке) и загрузки сценариев обработки спектральных данных.

Сами операции обработки/анализа спектральных данных реализуются в виде интерактивных операторов графического языка сценариев. Были созданы следующие библиотеки, группирующие операции по типам решаемых задач:

- библиотека операций предварительной обработки данных (операции алгебраических действий над спектрами: сложение спектров, усреднение множества спектров, вычитание, деление, масштабирование, смещение, логарифмирование; а также сглаживание, фильтрация и т. д.);
- библиотека операций анализа спектральных данных (операции выделения исследуемых спектральных линий, их сортировки, нахождения интенсивности линий, вычисления отношений интенсивности и т. д.);
- библиотека операций интерпретаций информации (инструменты для количественной и качественной оценки излучающего вещества);
- библиотека системных операций (инструменты для работы с данными и их множественной обработки, а также инструменты для организации сопряжений между операциями и средства визуализации данных).

Помимо этого, в структуру ПО будет введен механизм разработки/добавления подобных библиотек операций для возможности дальнейшего расширения системы и адаптации ее под новые задачи. Все библиотеки операций взаимодействуют как с информационным ядром, так и со средой разработки/выполнения сценариев.

При работе с программой пользователь размещает на рабочей области элементы (контейнеры и операции) и соединяет их между собой с помощью соединений. Когда происходит соединение операции с контейнером, она помещается в "Список операций". Если же происходит соединение операции с

другой операцией, тогда "Интерпретатор" просматривает соединения, которые подключены к данной операции. Если у соединения идентификатор не равен -1 , значит оно подключено к другой операции. Таким образом, интерпретатор вычисляет место в списке, в которое нужно вставить новую операцию, т. е. после всех операций, id которых равно id соединений. Если же id соединения равно -1 , то это означает, что соединение подключено к контейнеру, а значит операцию можно добавлять в начало "Списка операций".

Визуализация спектральной информации осуществляется путем передачи данных через контейнер

```
void Plotter::drawGrid(QPainter *painter)
{
    QRect rect(Margin, Margin,
               width() - 2 * Margin, height() - 2 * Margin);
    if (!rect.isValid())
        return;

    PlotSettings settings = zoomStack[curZoom];
    QPen quiteDark = palette().dark().color().light();
    QPen light = palette().light().color();

    for (int i = 0; i <= settings.numXTicks; ++i) {
        int x = rect.left() + (i * (rect.width() - 1) / settings.numXTicks);
        double label = settings.minX + (i * settings.spanX() / settings.numXTicks);

        painter->setPen(quiteDark);
        painter->drawLine(x, rect.top(), x, rect.bottom());
        painter->setPen(light);
        painter->drawLine(x, rect.bottom(), x, rect.bottom() + 5);
        painter->drawText(x - 50, rect.bottom() + 5, 100, 20,
                          Qt::AlignHCenter | Qt::AlignTop,
                          QString::number(label));
    }

    for (int j = 0; j <= settings.numYTicks; ++j) {
        int y = rect.bottom() - (j * (rect.height() - 1) / settings.numYTicks);
        double label = settings.minY + (j * settings.spanY() / settings.numYTicks);

        painter->setPen(quiteDark);
        painter->drawLine(rect.left(), y, rect.right(), y);
        painter->setPen(light);
        painter->drawLine(rect.left() - 5, y, rect.left(), y);
        painter->drawText(rect.left() - Margin, y - 10, Margin - 5, 20,
                          Qt::AlignRight | Qt::AlignVCenter,
                          QString::number(label));
    }

    painter->drawRect(rect.adjusted(0, 0, -1, -1));
}
```

Рис. 2. Программная реализация графической сетки "Средства отображения результатов"

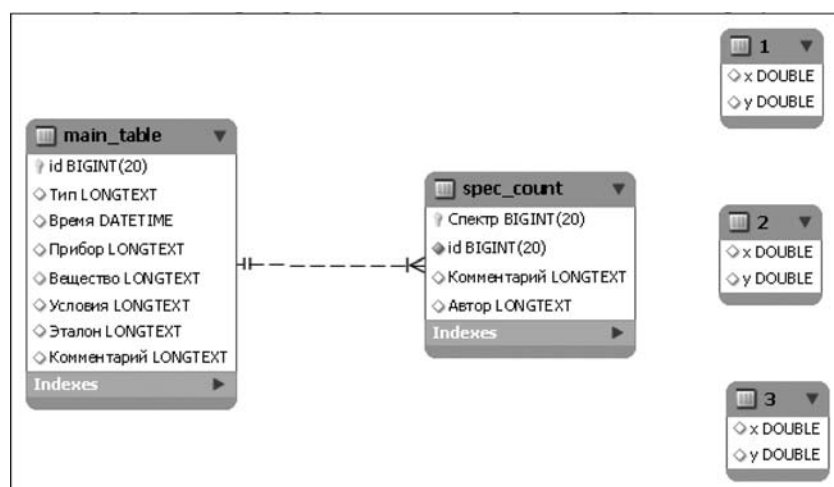


Рис. 3. Структурная схема разработанной базы данных

(двумерный массив) в средство отображения результатов — блок визуализации спектральной информации (БВСИ), который, в свою очередь, должен быть настроен определенным образом для правильного отображения данных и т. д. Режим выбирается путем вариации соответствующих параметров инструмента визуализации.

При выполнении функций визуализации БВСИ имеет возможность отображения нескольких графиков, при этом каждый график может отображаться в своем слое и со своим масштабом. Также предусмотрены функции масштабирования по различным осям, приведения масштаба всех отображаемых графиков к одному (выравнивание масштаба) и увеличения масштаба в заданной области графика.

В ходе анализа возможности использования тех или иных средств для разработки настоящего программного комплекса выбор был сделан в пользу языка C++, позволяющего реализовать кроссплатформенное ПО с модульной архитектурой, и библиотеки Qt, распространяемой по бесплатной лицензии GNU GPL. Пример программной реализации одной из функций БВСИ с использованием этой библиотеки показан на рис. 2. Руководствуясь этими же требованиями, для реализации БД для хранения спектров, полученных с акустооптических спектрометров, было решено использовать язык MySQL.

Связь информационного ядра с БД обеспечивается объектом QSqlDatabase, который предоставляет средства, позволяющие непосредственно выполнять произвольные команды SQL и обрабатывать их результаты.

Разработанная БД предназначена для хранения следующей информации:

- сценариев обработки спектральных данных;
- эталонных спектров;
- исходных данных, получаемых с акустооптического спектрометра;
- промежуточных данных, получаемых в результате выполнения операций над спектрами;
- результатов выполнения сценариев обработки спектральных данных.

В разработанной базе можно выделить три логических уровня (рис. 3):

- уровень главной таблицы ("main table"), в которой содержатся такие данные, как тип, время измерения, прибор, с помощью которого про-

водилось измерение, вещество, состояние, эталон, число измерений, а также вспомогательное поле индекса таблицы второго уровня;

- уровень вспомогательной таблицы (1, 2, ..., n), на котором хранятся указатели на массивы спектров (`spec_count`). Число n определяется числом измерений, т. е. значением поля "processed" таблицы "main_table". На данном уровне также предусмотрена возможность присоединения дополнительных таблиц с измерениями, для чего создано динамическое поле "spec_table_FK";
- уровень таблиц спектров (1, 2, 3, ..., m), на котором хранятся непосредственно массивы спектральной информации в виде таблиц с тремя столбцами: x (значение по оси абсцисс), y (значение по оси ординат), dy (смещение/резервное поле).

Необходимо также отметить, что в структуре БД используются следующие типы данных:

- LONGTEXT — поле строкового типа, используемое для хранения технической информации о приборе, типе и природе исследуемого материала. Может хранить не более 4 294 967 295 символов;
- Date — поле для хранения даты проведения эксперимента. Дата хранится в формате ГГГГ-ММ-ДД;
- UNSIGNED INT — поле целочисленного типа для хранения ключей;
- DOUBLE — поле для хранения данных, снимаемых с прибора. Данные хранятся в виде числа двойной точности.

Для удобства использования разработанного программного комплекса в него была внедрена интерактивная справочная подсистема, которая включает в себя всплывающие контекстные подсказки и отдельный файл подсказки, вызываемый из главного меню. Всплывающая подсказка (tooltip) представляет собой небольшое текстовое сообщение, которое появляется при нахождении курсора мыши на виджете в течение определенного времени.

Всплывающие подсказки отображаются на желтом фоне черными буквами. В основном они предназначены для пояснения назначения кнопок на панели инструментов.

Разработка нового и совершенствование существующего программного обеспечения для акустооптических приборов необходимо для повышения эффективности их функционирования. Учитывая растущие возможности современных ЭВМ, становится возможным совершенствовать расчетный модуль программной оболочки, что зачастую влечет за собой необходимость его создания с нуля. Однако это необходимо для повышения точности расчетов на этапе модельного анализа. Эти работы неотрывно связаны с созданием новых модулей, таких как, например, модуля визуализации или модуля хранения. Рассмотренное в данной статье программное обеспечение даст возможность увеличения степени автоматизации установки АОС и повысит оперативность при обработке, анализе и интерпретации получаемых с АОС спектральных линий. Оно также повысит степень гибкости и адаптивности всей системы в целом благодаря возможности расширения вычислительной базы и базы эталонов.

Список литературы

1. Пустовойт В. И., Пожар В. Э. Управление характеристиками коллинеарного акустооптического фильтра путем модуляции ультразвука // Радиотехника и электроника. 1998. Т. 43. Вып. 1. С. 121—127.
2. Harris S. E., Wallace R. W. Acousto-optic tunable filter // J. Opt. Soc. Amer. 1969. Vol. 59. P. 744—747.
3. Визен Ф. Л., Захаров В. М., Калинин Ю. К., Магомедов З. А., Масленников В. Н., Пустовойт В. И. Акустооптический фильтр. Авт. Свид. 667940, 1979.
4. Мазур М. М., Шорин В. Н., Жогун В. Н., Пустовойт В. И., Пожар В. Э., Калинин Ю. К., Костин Н. С., Мазур Л. И., Торопов В. Н., Шорин Д. В. Акустооптический рамановский спектрометр // Акустооптические, акустические и рентгеноспектральные методы и средства измерений в науке и технике. М.: ВНИИФТРИ, 2005. С. 16—26.

Э. А. Трахтенгерц, д-р техн. наук, проф.,
e-mail: tracht@ipu.rssi.ru
Учреждение Российской академии наук
Институт проблем управления
им. В. А. Трапезникова РАН

Информационные технологии коррекции целей и стратегических решений в динамике манипулирования общественным мнением

Рассматриваются компьютерные методы и алгоритмы коррекции целей и стратегических решений в динамике манипулирования общественным мнением.

Ключевые слова: информационные технологии, алгоритмы коррекции, цель, стратегические решения

Введение

Манипулирование общественным мнением — пропаганда определенной идеологии или точки зрения на какие-либо события — получило очень широкое распространение на самых различных уровнях: межгосударственных, транснациональных корпораций, политических партий, коммерческих организаций и т. д. На их реализацию затрачиваются очень большие средства, при их проведении в большинстве случаев обрабатываются и генерируются огромные объемы информации, зачастую в сжатые сроки. В связи с этим возникла необходимость создания и использования информационных технологий для управления кампаниями манипулирования общественным мнением. Этому вопросу посвящена работа [1]. Настоящая статья является ее продолжением.

Цель работы — показать методы использования информационных технологий и возможности противостояния действиям манипуляторов.

1. Взаимосвязь целей и стратегических решений в динамике манипулирования общественным мнением

Любая кампания манипулирования общественным мнением определяется целями ее проведения и стратегическими методами их реализаций, поэтому формирование целей и стратегий является одной из важнейших составляющих процедуры принятия управленческих решений. Задача формирования цели и реализующих ее стратегий в значительной степени — семантическая. Сегодня эффективных методов формализации семантик нет. Тем не менее компьютерная система манипулирования общественным мнением может помочь руководителю за счет использования методов генерации целей и стратегий, их оценки и ранжирования расширить множество вариантов, которые окажутся в его поле зрения. Варьируя методы оценок, она может влиять на субъективные предпочтения руководителя, способствуя принятию лучших решений.

Мир стал чрезвычайно динамичным. Выбрать "правильную" цель и стратегию ее реализации с учетом быстро меняющейся обстановки и большого числа факторов, требующих учета, без использования информационных технологий становится все сложнее. Поэтому они находят все более широкое применение в компьютерной поддержке принятия решений и их важной составляющей — формировании целей и стратегии их реализации (они рассмотрены в работе [1]) и, тем более, их коррекции в динамике проведения кампании манипулирования общественным мнением.

В статье предполагается, что цели кампании манипулирования и реализующие их стратегии выбраны в процессе планирования кампании манипулирования и выполняются, поэтому рассматри-

Таблица 1

Цели	Стратегии				
	Непрямая атака	Атака двойного стандарта	Неоправданно оптимистические оценки	Фальсификация	Приверженность консерватизму
	1	2	3	4	5
1. Нанесение экономического ущерба конкурирующей коммерческой организации	×	×	—	×	×
2. Компрометация или разоблачение руководителей конкурирующей организации	×	×	×	×	×
3. Дезинформация о действиях чиновников разного уровня, проводящих невыгодную для фирмы политику	×	×	×	×	×
4. Создание обстановки, при которой конкурирующая организация принимает невыгодные для себя условия	×	×	×	×	—
5. Дискредитация продукции конкурирующей организации	×	—	—	×	—

ваются не вопросы формирования целей и стратегий, а их модификация.

Табл. 1 демонстрирует связь целей со стратегиями. Знак "х" на пересечении строки и столбца показывает, что цель, обозначенная в строке таблицы, может быть реализована стратегией, указанной в столбце. Манипуляционные стратегии табл. 1 взяты из работы [2], а типичные цели — из работы [3]. Однако, если цель может быть реализована данной стратегией, это еще не значит, что она реализует цель эффективно.

В процессе анализа, оценки цели и реализующей ее стратегии всегда встает вопрос, следует ли менять стратегию или цель, менять ли и то и другое, или не менять ничего.

2. Компьютерная технология оценки определения необходимости модификации цели манипулирования общественным мнением

Оценка эффективности реализуемой кампании посредством манипуляции цели может быть только субъективной. Даже анализ опросов общественного мнения фокусных групп показывает только результативность пропаганды, но ее результаты не всегда определяют степень достижения поставленной экономической, социальной или организационной цели. Варьируя методы оценок, компьютерная система манипулирования общественным мнением может влиять на субъективные предпочтения руководителей, способствуя принятию лучших решений.

Большую роль в модификации целей и стратегий играют интересы и отношение руководства. Бывают случаи, когда высшее руководство не хочет пересматривать ранее принятые им решения, даже если и открываются новые перспективы. Руководство может любить рисковать, а может, наоборот, — стремиться любыми способами избежать риска. Финансовые ресурсы фирмы также оказывают существенное влияние на коррекцию целей и стратегий манипулирования. Манипуляционная атака на конкурентов требует больших финансовых затрат. Поэтому при выборе стратегий манипулирования организации, имеющие большие финансовые ресурсы либо легкий доступ к ним, имеют для выбора гораздо большее число вариантов стратегий, нежели организации с сильно ограниченными финансовыми возможностями.

Оценка эффективности реализуемой цели в динамике проведения кампании манипулирования, забегая вперед скажем и стратегии, имеет чрезвычайно важное значение. В конечном счете, она нужна для принятия решения: нужно ли что-то менять или нет. Если нет, то в большинстве случаев считается, что кампания манипуляции проводится эффективно. А когда у руководителей появляется ощущение, что "что-то не так", им нужна информация, которая отнюдь не лежит на поверхности. В качестве примера рассмотрим успешность реали-

зации цели "Дезинформация о действиях министра". Критерии успешности проводимой кампании разобьем на три группы. Первую назовем социологическими критериями. Они отражают различные аспекты влияния кампании манипуляции на общество. Вторая и третья группы критериев характеризуют риски и реализуемость. Затем дадим общую оценку успешности. Конечно, разбиение критериев на группы может быть другим.

1. Социологические критерии формируются так же, как и в работе [1]. Будем считать, что каждый эксперт с помощью системы сформировал таблицу набора критериев и их важности. Пример такой таблицы, сформированной одним экспертом, показан в табл. 2.

Для ранжирования критериев компьютерная система составляет таблицу типа табл. 3. На дисплее руководителей и экспертов высвечиваются данные последнего столбца табл. 3, показывающие "веса" критериев, равные нормированным суммам рангов в соответствии с их коллективной оценкой. Они определяются по следующим формулам:

$$r_i = \sum_{j=1}^J r_{ij} n_j, i = \overline{1, I}, \quad (1)$$

где r_{ij} — ранг i -го критерия в табл. 1; n_j — число руководителей, давших данную оценку критерию.

Таблица 2

Наименование критерия	Значения критерия	Важность	Ранг критерия
1. Полнота решения исходной проблемы	Хорошо	Очень важно	1
2. Экономический эффект	Удовлетворительно	Важно	2
3. Соответствие менталитету фокусной группы рассматриваемой цели	Отлично	Не очень важно	4
4. Влияние на "болевы" точки фокусной группы	Хорошо	Важно	2

Таблица 3

Наименование критерия	Число руководителей, проставивших оценку					Сумма рангов критериев r_i	"Вес" (значение) критерия K_j
	5	4	3	2	1		
1. Полнота решения исходной проблемы	6	5	3	—	—	59	0,28
2. Экономический эффект	3	7	3	1	—	54	0,25
3. Соответствие менталитету фокусной группы рассматриваемой цели	2	4	5	3	—	47	0,22
4. Влияние на "болевы" точки фокусной группы	2	8	3	1	—	53	0,25

Нормированная сумма рангов K_j , являющаяся его "весом", определяется по формуле

$$K_j = \frac{r_i}{\sum_{i=1}^I r_i} \quad (2)$$

Теперь успешность реализации цели манипулирования может быть определена системой методами, рассмотренными, например, в работах [1, 4, 5]. Это можно сделать по формуле

$$V = \sum a_i k_i$$

где a_i — значение критериев в табл. 2. Тогда, используя значения табл. 2 и 3, получим

$$V = 4 \cdot 0,28 + 3 \cdot 0,25 + 5 \cdot 0,22 + 4 \cdot 0,25 = 3,97.$$

То есть оценка, полученная компьютерной системой по этим критериям, — "хорошо".

2. Понятие риска часто используется в быту, и поэтому приобрело некоторую размытость. Под риском обычно понимается возможность наступления какого-либо события, наносящего ущерб, мешающего достижению поставленной цели или делающего его невозможным. Под оценкой риска можно понимать размер возможного ущерба, степень отступления от намеченной цели, недополучение ожидаемых преимуществ (доходов, политической власти, положения на рынке) и т. д. В литературе приведено несколько классификаций рисков и подходов к их оценкам [6, 7], так или иначе они связаны с характером выбираемых целей.

Для оценки рисков обычно используют теоретико-вероятностные, вероятностно-статистические, статистические и экспертные модели [6]. Поскольку любой вариант в кампании манипулирования общественным мнением проходит, как правило, в уникальной ситуации, то приходится ограничиваться применением только экспертных методов. Для определения опасности рисков проводимой кампании манипулирования система представляет на дисплее руководителей и экспертов список возможных рисков, находящийся в базе данных, просит отметить те риски, которые они хотят внести в окончательный список, и добавить новые, если они считают это нужным. Затем система по одному из алгоритмов, рассмотренных, например, в работах [1, 5], согласовывает список критериев.

В нашем иллюстративном примере будем считать, что согласованы и утверждены следующие критерии рисков достижения целей кампании манипулирования общественным мнением:

- отношение властей;
- реакция средств массовой информации;
- рыночные риски (возможные убытки, вызванные расходами на проведение кампании, не компенсируемые достижением цели).

Согласование оценок риска успешности достижения цели манипулирования общественным мнением может быть проведено по среднему баллу.

Величина риска (угрозы) должна быть оценена всеми руководителями и экспертами. Каждый из них вводит в систему свои оценки риска, и система определяет среднюю оценку цели по формуле

$$X_i = \frac{1}{k} \sum_{k=1}^K x_{ij}^k, \quad i = \overline{1, I}, \quad (3)$$

где x_{ij}^k — оценка риска i -й цели k -м руководителем по j -му риску.

В результате компьютерная система представляет каждому специалисту таблицу типа табл. 4, в верхней строке которой указаны значения критериев риска, данные самим специалистом, а в нижней строке — подсчитанные по формуле (3). Каждый специалист на дисплее будет видеть только свою таблицу, верхние строки которой не известны другим специалистам.

После того как каждый специалист увидел у себя на экране табл. 4, он может скорректировать свои оценки с учетом средних оценок, данных другими специалистами. Они показаны во вторых строках табл. 4. Компьютерная система с учетом проведенных изменений проводит перерасчет второй строки. Последний столбец табл. 4 показывает среднюю балльную оценку риска проведения кампании манипулирования с указанной целью.

3. Перейдем к оценке реализуемости цели кампании манипулирования общественным мнением. Методика оценки реализуемости в каком-то смысле близка к методике оценки рисков. Близость объясняется тем, что эти оценки также в значительной степени субъективные, основанные на опыте, знаниях и интуиции, а не на точном расчете. Задачу оценки реализуемости целей можно оценить по критериям: техническая, экономическая, финансовая и политическая реализуемость, как это сделано, например, в работе [8].

Существуют, конечно, и другие критерии реализуемости целей манипуляции. Теперь перейдем к их оценке. Пусть компьютерная система сгенерировала и согласовала с руководителями перечисленные аспекты реализуемости цели манипуляции общественным мнением. Для оценки реализуемо-

Таблица 4

Наименование цели	Значения критериев риска			Средняя балльная оценка
	Отношение властей	Реакция средств массовой информации	Рыночные риски	
Дезинформация о действиях министра	Риск есть (3)	Риск небольшой (4)	Риск есть (3)	3
	Риск минимальный (5)	Риск минимальный (5)	Риск существенный (2)	4
Примечание. В скобках показана балльная оценка.				

Таблица 5

№	Наименование цели	Значения критериев реализуемости цели			
		Техническая	Экономическая	Финансовая	Политическая
		1	2	3	4
3	Дезинформация о действиях министра	Вполне реализуема (5)	Реализуема с трудом (3)	Реализуема (4)	Реализуема (4)
Примечание. В скобках показана балльная оценка.					

сти компьютерная система предложила руководителям пятибалльную шкалу. По этой шкале руководители дают оценку реализуемости каждой цели по всем рассматриваемым критериям.

В качестве примера оценки реализуемости цели манипуляции рассмотрим ту же цель "Дезинформация о действиях министра". Для этого компьютерная система высвечивает на дисплеях руководителей таблицу типа табл. 5, которую руководители заполняют.

Компьютерная система согласовывает значения критериев, представленные руководителями в табл. 5 одним из способов, рассмотренных, например, в работах [1, 5, 9] или других. Будем считать, что в табл. 5 проставлены уже согласованные значения, и компьютерная система рассчитывает "веса" критериев реализации кампании манипуляции общественным мнением по приведенным ниже формулам.

"Вес" технического критерия:

$$\beta_1 = \frac{y_1}{\sum_i y_i} = \frac{5}{16} = 0,31.$$

"Вес" экономического критерия:

$$\beta_2 = \frac{y_2}{\sum_i y_i} = \frac{3}{16} = 0,19.$$

"Веса" финансового и маркетингового критериев:

$$\beta_3 \text{ и } \beta_4 = \frac{y_3}{\sum_i y_i} = \frac{y_4}{\sum_i y_i} = \frac{4}{16} = 0,25,$$

где y_i — оценка в i -м столбце табл. 5.

Оценку эффективности реализации j -й цели R_j можно выполнить по формуле

$$R_j = \begin{cases} 1, & \text{если } \exists d_{ij} = 1; \\ \sum_i \beta_i d_{ij}, & \text{в противном случае,} \end{cases} \quad (4)$$

где d_{ij} — оценка i -го критерия реализации j -й цели.

В нашем случае $R_1 = 0,31 \cdot 5 + 0,19 \cdot 4 + 0,25 \cdot 3 + 0,25 \cdot 4 = 4,06$, т. е. "Цель реализуется хорошо".

4. Теперь перейдем к общей оценке эффективности реализации цели. Оценка проведения кам-

пании манипуляции по социологическим критериям равна 4,0. Оценка риска по табл. 4 равна 4 и оценка реализуемости по формуле (4) — 4,06. Общая оценка — 4, цель модифицировать не нужно. В случае необходимости ее модификации вариант модификации может быть сгенерирован так, как это показано, например, в работах [1, 5].

3. Компьютерная оценка эффективности реализации стратегии манипулирования общественным мнением

Оценка эффективности реализации стратегии определяет необходимость ее модификации. В табл. 6 приведен пример согласованных критериальных оценок, данных экспертами. Но в таблице нет ответа на вопрос, нужно ли менять стратегию. Для его решения оценим эффективность реализации стратегии методом распознавания образов, несколько модифицировав его с учетом специфики решаемой задачи следующим образом [4].

1. Система разбивает значения результатов стратегии манипулирования на N областей эффективности и пронумеровывает области в нисходящем порядке в зависимости от эффективности результатов. Каждой степени эффективности стратегий ставится в соответствие опорный вектор критериальных оценок, согласованный с руководителями или экспертами. Максимально возможная степень эффективности отнесена к области 1, а минимальная — к области N . Чем больше номер области, тем настоятельней необходима замена стратегии.

2. Для распознавания системой степени эффективности стратегии манипуляции вводится мера близости между r -м опорным вектором и вектором $u_i(\bar{x})$ оценки эффективности реализации i -й стратегии:

$$\rho(u_r(\bar{x}), u_i(\bar{x})) = \min_{\substack{u_r(\bar{x}) \in B_T \\ u_i(\bar{x}) \notin B_T}} \rho(u_r(\bar{x}), u_i(\bar{x})), \quad (5)$$

где $\rho(u_r(\bar{x}), u_i(\bar{x})) = \sqrt{\sum_{j=1}^J K_j (x_i^j - x_r^j)^2}$, K_j — "вес" j -го

критерия, а x_i^j — оценка эффективности реализации i -й стратегии по j -му критерию; x_r^j — оценка

Таблица 6

Значения критериальных оценок			
Полнота решения исходной проблемы	Экономический эффект	Соответствие менталитету фокусной группы рассматриваемой стратегии	Влияние на "болевые" точки фокусной группы
Удовлетворительно	Удовлетворительно	Хорошо	Хорошо

r -го опорного вектора по j -му критерию; B_T — множество опорных векторов, характеризующих степень эффективности реализации стратегии.

Для нахождения степени эффективности i -й стратегии компьютерная система сравнивает вектор $u_i(\bar{x})$ со всеми опорными векторами $u_r(\bar{x})$. Опорный вектор $u_r(\bar{x})$, для которого функция (5) — $\rho(u_i(\bar{x})u_r(\bar{x}))$ достигает минимума, и определяет класс эффективности $u_i(\bar{x})$. Номер класса определяет возможность продолжения реализации стратегии, т. е. возможность ее выполнения для достижения поставленной цели или необходимость ее замены.

Рассмотрим иллюстративный пример. В соответствии с пунктом 1 алгоритма эксперты с помощью компьютерной системы генерируют и согласовывают значения опорных векторов. Результаты согласования показаны в табл. 7. Значения "весов" критериев K_j показаны в шапке табл. 7 в соответствии с табл. 2.

В табл. 7 введено только три области эффективности. Увеличение числа областей позволяет делать более тонкие оценки, но принципиально идею метода не меняет.

Выполнение пункта 2 начнем сравнением по формуле (5) с опорным вектором области эффективности:

$$\begin{aligned}\rho(u_i(\bar{x})u_1(\bar{x})) &= \sqrt{\sum_{j=1}^5 K_j(x_i^j - x_1^j)^2} = \\ &= \sqrt{0,28(3-5)^2 + 0,25(3-4)^2 + 0,22(4-5)^2 + 0,25(4-4)^2} = \\ &= \sqrt{1,59} = 1,26.\end{aligned}$$

Затем проведем такое же сравнение с опорным вектором области 2:

$$\begin{aligned}\rho(u_i(\bar{x})u_2(\bar{x})) &= \sqrt{\sum_{j=1}^5 K_j(x_i^j - x_2^j)^2} = \\ &= \sqrt{0,28(3-4)^2 + 0,25(3-3)^2 + 0,22(4-4)^2 + 0,25(4-4)^2} = \\ &= \sqrt{0,28} = 0,52,\end{aligned}$$

и, наконец, с опорным вектором области 3:

$$\begin{aligned}\rho(u_i(\bar{x})u_3(\bar{x})) &= \sqrt{\sum_{j=1}^5 K_j(x_i^j - x_3^j)^2} = \\ &= \sqrt{0,28(3-3)^2 + 0,25(3-2)^2 + 0,22(4-3)^2 + 0,25(4-2)^2} = \\ &= \sqrt{1,47} = 1,21.\end{aligned}$$

Таким образом, вектор оценки эффективности реализуемой стратегии ближе всего к опорному вектору области 2 и дальше всего от опорного вектора 1. Тем самым, компьютерная система предупреждает руководство, что хотя ситуация еще не требует смены стратегии, но следует тщательно

№ области эффективности	Критериальные оценки			
	Полнота решения исходной проблемы $K_1 = 0,28$	Экономический эффект $K_2 = 0,25$	Соответствие менталитету фокусной группы рассматриваемой стратегии $K_3 = 0,22$	Влияние на "болевые" точки фокусной группы $K_4 = 0,25$
1	Отлично	Хорошо	Отлично	Хорошо
2	Хорошо	Удовлетворительно	Хорошо	Хорошо
3	Удовлетворительно	Плохо	Удовлетворительно	Плохо

проанализировать и проранжировать возможные модификации стратегий.

4. Компьютерное ранжирование модифицированных стратегий

В табл. 1 уже была показана взаимосвязь стратегии и цели. Если считать, что стратегии и цели оцениваются по одним и тем же критериям, и значения этих критериев достаточно близки, то оценки эффективности реализации стратегии и достижения цели будут одинаковы. Можно стратегии и цели оценивать разными критериями, тогда их оценки в большинстве случаев не будут совпадать.

Нахождение лучшего варианта модифицированной стратегии и/или цели является одной из важнейших задач менеджмента. Выбор модификации в значительной степени определяется как причинами ее проведения, так и интересами руководителя (или руководителей), реализующих эту модификацию.

В табл. 8 приведены примеры стратегических задач, которые могут встать перед руководителями различных типов организаций. Хотя таблица не отражает процессы манипулирования, она показывает различие интересов руководителей, которые придется согласовывать в случае выполнения совместных действий. Естественно, этот список не является исчерпывающим. Для решения каждой вновь возникшей задачи обычно приходится модифицировать реализуемые стратегии. Требуется определить, какая комбинация модификаций стратегий с учетом интересов руководителей окажется лучшей.

Методы генерации, оценки и согласования модификаций стратегий и целей не отличаются от аналогичных задач, рассмотренных в [1, 5] и других работах на эту тему, поэтому не будем на них останавливаться, считая их известными. Будем считать, что по выбранной цели манипуляции общественным мнением компьютерной системой предложены согласованные с руководителями вариан-

Таблица 8

Руководители различного уровня и типа организаций	Стратегические задачи
Руководитель фирмы	Обеспечение экономической и информационной безопасности Разработка планов развития предприятия Выпуск акций Оценка эффективности кампаний манипулирования
Собственник организации	Выбор варианта распоряжения собственностью Составление объединительных и раздельных балансов при реструктуризации Обоснование цены купли-продажи предприятия или его доли Увеличение размера выручки при упорядоченной ликвидации предприятия
Руководители инвестирующих финансовых групп	Проверка целесообразности инвестиционных вложений Определение допустимой цены покупки предприятия в целях включения его в инвестпроект
Руководители государственных органов	Экономическая и информационная подготовка предприятия к приватизации Определение облагаемой базы для различных видов налогов Установление выручки от принудительной ликвидации через процедуру банкротства Оценка для судебных целей

ты модификации стратегий, их критерии и оценки. Они показаны в табл. 9.

Для ранжирования стратегий выберем метод Парето. Он позволяет сравнивать все варианты модификаций стратегий между собой по одним и тем же критериям, показывая превосходство одних вариантов над другими.

Для ранжирования вариантов методом Парето как обычно введем булеву переменную:

$$a_{ij} = \begin{cases} 1, & \text{если вариант модификации } i \text{ строго предпочтительнее варианта модификации } j; \\ 0, & \text{в противном случае,} \end{cases}$$

и построим квадратную матрицу, показанную в табл. 10, элементами которой являются значения переменных a_{ij} . Элементы в i -й строке матрицы определяют элементы j , по отношению к которым элемент i строго предпочтителен. Поэтому, если в j -м столбце все нули, значит нет элемента строго предпочтительнее элемента j , и он принадлежит Парето-оптимальному множеству. Исключаем его из дальнейшего рассмотрения, вычеркивая такие столбцы и одноименные строки. В табл. 10 это модификация 2 — "Атака двойного стандарта". Она принадлежит Парето-оптимальному множеству 1-го ранга.

В результате вычеркивания модификации 2 из табл. 10 получаем табл. 11 с нулевым столбцом, оп-

Таблица 9

№ и наименование модификации стратегий	Критериальные оценки			
	Полнота решения исходной проблемы	Экономический эффект	Соответствие менталитету фокусной группы рассматриваемой цели	Влияние на "болевые" точки фокусной группы
1	2	3	4	5
1. Непрямая атака	4	3	5	4
2. Атака двойного стандарта	5	3	5	4
3. Неоправданно оптимистическая оценка	3	3	4	3
4. Фальсификация	3	2	3	2
5. Приверженность консерватизму	4	3	3	2

Таблица 10

№ модификации	1	2	3	4	5
1	0	0	1	1	1
2	1	0	1	1	1
3	0	0	0	1	1
4	0	0	0	0	0
5	0	0	0	1	0

Модификация 2
принадлежит множеству
Парето 1-го ранга

Таблица 11

№ модификации	1	3	4	5
1	0	1	1	1
3	0	0	1	1
4	0	0	0	0
5	0	0	1	0

Модификация 1
принадлежит множеству
Парето 2-го ранга

Таблица 12

№ модификации	3	4	5
3	0	1	1
4	0	0	0
5	0	1	0

Модификация 3
принадлежит множеству
Парето 3-го ранга

Таблица 13

№ модификации	4	5
4	0	0
5	1	0

Модификация 5 принадлежит
множеству Парето 4-го ранга,
а модификация 4 — 5-го ранга

ределяющим модификацию 1 — "Непрямая атака". Эта модификация принадлежит Парето-оптимальному множеству 2-го ранга. Если это необходимо, формируются модификации низших рангов — табл. 12 и 13.

Лучшей модификацией оказалась модификация "Атака двойного стандарта", за ней следует модификация — "Непрямая атака".

Таким образом, компьютерная система управления манипулированием общественным мнением предлагает подготовиться к реализации стратегии "Атака двойного стандарта", а возможно и стратегии "Непрямая атака".

Заключение

Компьютерная система управления манипулированием общественным мнением определяет необходимость модификации целей и стратегий проведения кампании манипулирования и в случае необходимости рекомендует новые модифицированные цели и стратегии.

Список литературы

1. **Трахтенгерц Э. А.** Информационные технологии формирования управленческих решений в процессе манипуляции общественным мнением // Информационные технологии. Приложение. 2011. № 5. 32 с.
2. **Ambrosini V., Bowman C.** Managerial consensus and corporate strategy // European Management Journal. 2003. V. 21, N 2. P. 213—221.
3. **At-Twajjri M. I., Al-Chamdi S. V., Luqmani M.** Prioritization of corporate goals in Saudi Arabia: an exploratory investigation // International Journal of Value-Based Management. 1996. 9. P. 259—270.
4. **Горелик А. Д., Скрипкин В. А.** Методы распознавания. М.: Высшая школа. 2004.
5. **Трахтенгерц Э. А.** Компьютерные методы реализации экономических и информационных управленческих решений. Т. 1, 2. М.: СИНТЕГ, 2009.
6. **Акимов В. А., Лесных В. В., Раднаев Н. Н.** Риски в природе, техносфере, обществе и экономике. М.: Деловой экспресс. 2004. 352 с.
7. **Вишняков Я. Д., Радаев Н. Н.** Общая теория рисков. М.: ACADEMIA, 2007.
8. **Web Allen.** Project planning and feasibility analysis//<http://members.aol.com/AllenWeb/planning.htm>
9. **Трахтенгерц Э. А.** Компьютерные технологии информационного управления в конфликтных ситуациях // Информационные технологии. Приложение. 2009. № 1. 32 с.

УДК 62-932.4

С. С. Малафеев, аспирант,
e-mail: serpg87@gmail.com,

А. В. Черняев, д-р техн. наук, проф.,
"МАТИ"—РГТУ имени К. Э. Циолковского

Информационная система визуализации работы экскаватора

Рассматривается использование системы визуализации работы экскаватора, основанной на технологии трехмерного графического моделирования и использующей данные цифровой системы управления приводами главного движения для определения положения, направления и скорости движения механизмов экскаватора. Рассматриваемая система использует существующие аппаратные ресурсы и позволяет повысить эффективность мониторинга работы экскаваторов горно-добывающего комплекса.

Ключевые слова: экскаватор, мониторинг, визуализация, компьютерная графика

Современное состояние развития горно-добывающих предприятий требует организации централизованного контроля за работой всех задействованных в добыче машин и систем, учета энергопотребления и объема добычи, мониторинга аварийных ситуаций. Для реализации такого централизованного контрольного центра используется вычислительная сеть предприятия, собирающая данные с различных датчиков, навигационных систем транспортных средств, камер наблюдения и др.

Любой компьютер, подключенный к сети предприятия, может использоваться как пульт оператора. Так как в работах на горно-добывающем предприятии задействовано множество различных машин, объем передаваемых и обрабатываемых в сети предприятия данных очень велик, поэтому оптимизация данных о состоянии задействованных в работе машин является важной задачей.

Визуализация работы экскаватора применяется для контроля объема добычи, обнаружения аварийных ситуаций, других отклонений от штатного режима работы и позволяет в реальном времени отслеживать положение механизмов экскаватора в пространстве, скорость и направления их движения. В настоящее время широко используются системы визуализации, основанные на видеонаблюдении. Такие системы включают в себя комплекс специализированных видеокамер, систему передачи данных, видеосервер, служащий для сбора, архивирования и передачи видеоданных на компьютеры наблюдателей. Такая система позволяет в реальном времени проводить наблюдение за ходом работы механизмов экскаватора, отслеживать отклонения от нормального режима работы и аварийные ситуации [1].

Недостатками системы видеонаблюдения являются: значительные объемы передаваемых данных; необходимость установки на экскаваторе дополни-

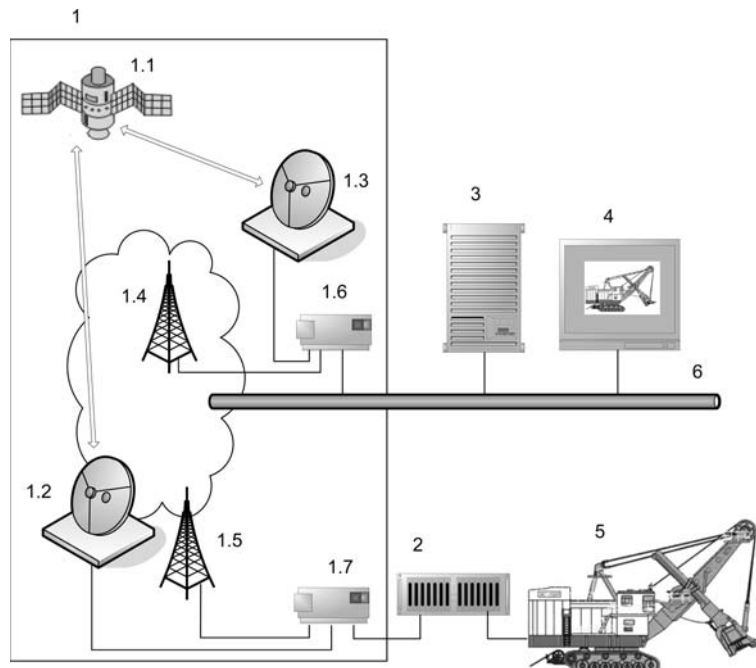
тельного оборудования для ведения видеонаблюдения; влияние погодных условий на возможность наблюдения. При этом основной информацией, передаваемой системой видеонаблюдения, является положение механизмов экскаватора в пространстве, направление и скорость их движения. Поэтому возможна передача на диспетчерский пункт только изменяющихся во времени данных о положении, скорости и направлении движения приводов главного движения и последующее восстановление картины работы экскаватора средствами компьютерной графики. Современные средства трехмерной графики в сочетании с пакетами физического моделирования позволяют с использованием известных и неизменных физических характеристик экскаватора и получаемых данных о координатах, направлении и скорости движения механизмов экскаватора создавать трехмерные модели экскаваторов, с высокой степенью точности отражающие работу реальной машины.

В качестве источников информации о положении механизмов экскаватора возможно использование сигналов управления приводами главного движения экскаватора, а также токи и напряжения двигателей. Эти данные могут быть получены из цифровой системы управления экскаватором без необходимости установки на экскаватор дополнительных датчиков и позволяют точно определять текущее положение, направление и скорость движения механизмов экскаватора.

На рисунке показана функциональная схема системы визуализации работы экскаватора. Движение рабочих органов экскаватора однозначно определяется сигналами управления, поступающими от командоаппаратов [2]. По напряжению на якорной обмотке двигателя и току якорной обмотки вычисляется скорость соответствующего привода. Моделирование работы экскаватора по сигналам управления и выходным переменным привода позволяет визуализировать работу машины, контролировать процесс экскавации, учитывать количество отгруженной горной массы, фиксировать простой экскаватора.

Таким образом, предлагаемая система визуализации работы экскаватора позволяет повысить надежность наблюдения в сложных условиях работы экскаваторов, особенно в темное время суток, при наличии атмосферных осадков, пыли, грязи и др. и уменьшить объем передаваемой информации за счет использования для визуализации работы экскаватора сигналов задания для приводов главного движения, напряжений и токов двигателей указанных приводов.

Другим важным достоинством предлагаемой системы визуализации работы экскаватора является



Функциональная схема системы визуализации работы экскаватора:

1 — средства передачи данных; 1.1 — геостационарный спутник; 1.2 и 1.3 — спутниковые антенны; 1.4 и 1.5 — антенны радиосвязи; 1.6 и 1.7 — модемы; 2 — контроллер передачи данных; 3 — средство обработки информации (сервер); 4 — монитор; 5 — экскаватор; 6 — шина

отсутствие необходимости в установке на экскаваторе дополнительных аппаратных средств, так как указанная информация в цифровой форме имеется в системе управления экскаватором, следовательно, отпадает необходимость использования дополнительных датчиков в системе. Для синтеза трехмерной модели экскаватора не требуются какие-либо специальные средства, за исключением компьютера. Таким образом, все необходимое для реализации данной системы оборудование уже имеется в составе машинного комплекса экскаватора, поэтому для запуска системы визуализации работы экскаватора необходимо установить только соответствующее программное обеспечение. Предлагаемая система визуализации работы экскаватора рекомендована для использования на отечественных экскаваторах ЭКГ-10 и ЭКГ-12.

Статья подготовлена в процессе проведения поисковой научно-исследовательской работы в рамках реализации ФЦП "Научные и научно-педагогические кадры инновационной России" на 2009—2013 годы по проблеме НК-544П(5), проект П-236.

Список литературы

1. Трубецкой К. Н., Кулешов А. А., Клебанов А. Ф., Владимиров Д. Я. Современные системы управления горно-транспортными комплексами / Под ред. К. Н. Трубецкого. СПб.: Наука, 2007. С. 91—93.
2. Малафеев С. И., Серебренников Н. А. Создание электрооборудования и систем управления для экскаваторов на основе мехатронной технологии // Горное оборудование и электромеханика. 2007. № 12. С. 23—28.

УДК 004.418+004.031.4

А. И. Прохоров, науч. сотр.,
e-mail: aipro@icpr.ac.ru,

В. Е. Туманов, канд. химич. наук, зав. сектором,
Б. Л. Психа, д-р химич. наук, ученый секретарь,
Институт проблем химической физики РАН

К. И. Поликарпов, студент,
Московский энергетический институт
(технический университет)

Информационно-аналитическая система учета результатов интеллектуальной деятельности в научной организации*

Рассмотрена информационно-аналитическая система мониторинга и учета результатов интеллектуальной деятельности академического института, в том числе учета результатов научно-технической деятельности. База данных системы аккумулирует в себе сведения обо всех результатах интеллектуальной деятельности, полученных научными подразделениями института.

Ключевые слова: автоматизированная информационная система, он-лайн доступ, управленческий учет, объекты интеллектуальной собственности

Введение

Учет и анализ результатов интеллектуальной деятельности в исследовательских организациях в целях выявления на ранних стадиях выполнения научно-исследовательских работ (НИР) потенциальных объектов интеллектуальной собственности (ОИС) для поддержки принятия решений в области управления инновационной деятельностью являются актуальной задачей в условиях развития новой экономики России и совершенствования взаимодействия науки и бизнеса.

Вопросам разработки автоматизированных систем учета и инвентаризации результатов научно-технической деятельности (РНТД) и результатов интеллектуальной деятельности (РИД) на общероссийском уровне в последнее время уделяется много внимания. Такие системы разработаны и внедрены на государственном уровне [1, 2]. Также разрабо-

таны типовые решения по учету результатов интеллектуальной деятельности для предприятий [3].

В настоящее время для академических институтов таких типовых решений не предложено. Поэтому одним из важных аспектов решения поставленной задачи является разработка и создание электронных баз данных внутреннего учета и мониторинга результатов интеллектуальной деятельности академического института [4, 5].

Целью настоящей работы является описание информационно-аналитической системы внутреннего учета и мониторинга результатов интеллектуальной деятельности академического института, в том числе учета результатов научно-технических работ для поддержки принятия решений в области управления учетом интеллектуальной деятельности в академическом институте. В ИПХФ РАН такая система проходит опытную эксплуатацию с 2006 г.

Постановка задачи

Информационно-аналитическая система внутреннего учета результатов интеллектуальной деятельности (ИАС "Учет РИД") академического института предназначена для:

- сбора сведений о фундаментальных, фундаментально-прикладных и прикладных результатах деятельности подразделений института, полученных в ходе выполнения всех видов работ;
- хранения в электронном виде данных о полученных результатах;
- классификации полученных результатов в соответствии с направлениями деятельности РАН, Отделений РАН и конкретных учреждений РАН;
- извлечения данных о полученных результатах в соответствии с критериями запросов;
- формирования справок и аналитических отчетов для анализа и принятия решений.

Организационным обеспечением ИАС "Учет РИД" является комплект нормативных документов РФ, Министерства образования и науки РФ, Российской академии наук в области управления РИД, а также комплект внутренних нормативных документов института, которые определяют порядок учета и предоставления сведений о РИД в систему.

Информационные потоки поступают в ИАС "Учет РИД" из внешних по отношению к ней автоматизированных информационных систем планового отдела, патентного отдела, отдела кадров и научных подразделений института. Мониторинг за

*Работа выполнена при поддержке гранта РФФИ 09-00481-а.

Учет результатов интеллектуальной деятельности ИПХФ РАН

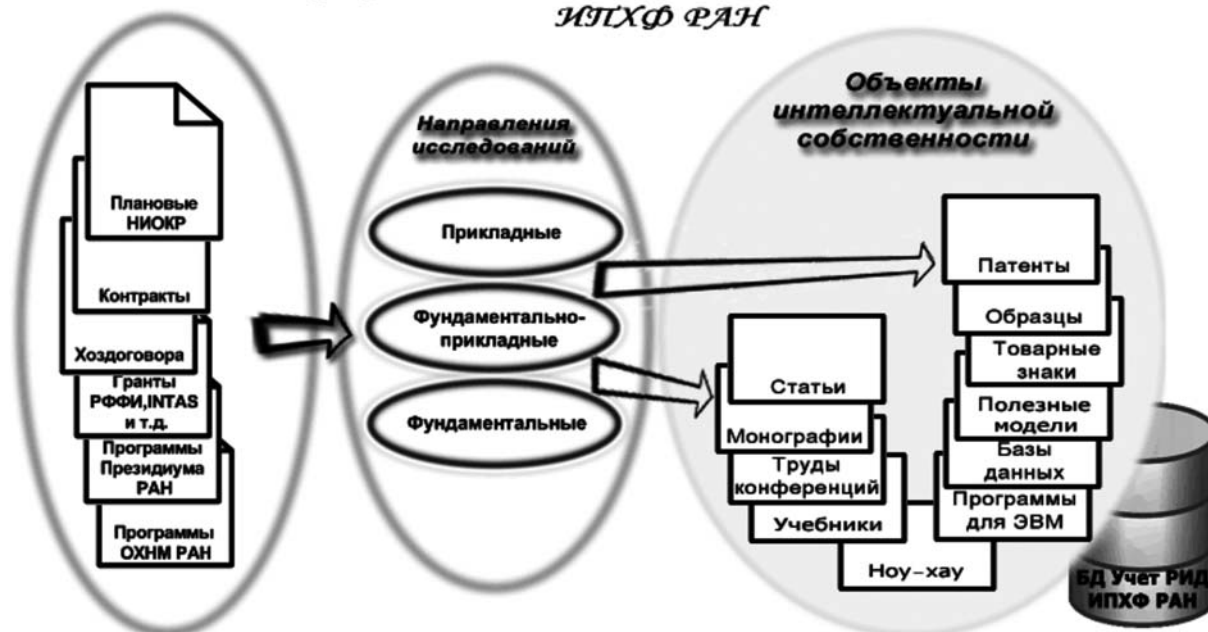


Рис. 1. Процесс производства, мониторинга и учета результатов интеллектуальной деятельности в академическом институте

работой системы осуществляется Ученым секретариатом.

Процесс производства, мониторинга и учета РИД в академическом институте показан на рис. 1.

Автоматизированная система обеспечивает информационную поддержку управленческого учета проводимых в академическом институте работ (плановые работы, государственные контракты, исследовательские гранты, хозяйственные договоры, контракты), позволяет руководству координировать планы деятельности учреждения, оценивать эффективность финансовых вложений в науку в конкретном учреждении. В системе ежегодно аккумулируются результаты работы научных коллективов по реализации исследований по всем видам работ. Анализ сохраняемых в системе результатов представляет руководству развернутую картину научной и инновационной деятельности института.

Информационная система выполняет следующие основные функции:

- осуществляет сбор, хранение, верификацию данных обо всех результатах интеллектуальной деятельности, полученных коллективами института;
- обеспечивает гибкий поиск данных по типовым запросам;
- генерирует типовые отчеты и справки.

Структура и модель данных базы данных информационно-аналитической системы "Учет РИД"

Основными объектами базы данных системы являются следующие.

Объект "Наименование темы и ее классификация".

Классификация выполняемой работы по теме включает в себя следующие атрибуты: индекс приоритетного направления РАН на 2008—2012 гг., индекс научного направления РАН, индекс научного направления института, номер государственной регистрации, шифр работы (внутренний номер темы), вид работы (НИР, ОКР, ТР, НИОКР, ОКТР, НИОКТР).

Объект "Наименование программы, в рамках которой выполнена работа". Работы классифицируются по следующим группам: плановые работы, программы Президиума РАН, программы Отделения РАН, гранты РФФИ, совместные гранты РФФИ, государственные контракты, хозяйственные договоры, зарубежные контракты, гранты МНТЦ, гранты CRDF, гранты INTAS, региональные гранты, персональные гранты поддержки ученых, гранты фондов поддержки науки, другие гранты.

Объект "Заказчик" хранит информацию об организации, которая является заказчиком работы (РАН, РФФИ, Роснаука, др.).

Объект "Головной исполнитель и соисполнители работ" включает в себя информацию об исполнителях и соисполнителях для совместно проводимых работ.

Объект "Дата начала и конца работы" определяет временной интервал выполнения работы.

Объект "Руководитель темы" аккумулирует данные о руководителе работы, о списке исполнителей, о подразделениях, участвующих в работе (отделы, лаборатории).



Рис. 2. Логическая модель данных базы данных ИАС "Учет РИД"

Рис. 3. Регистрация результата интеллектуальной деятельности

Рис. 4. Регистрация объекта интеллектуальной деятельности

Объект "Отчетный период" определяет год предоставления отчета по РИД.

Объект "Формулировка результата" включает в себя сведения о полученном результате: краткое описание, перечень документации (отчет, технический регламент, программы, методики, технологические инструкции), характеристика полученного результата (фундаментальный, фундаментально-ориентированный, прикладной), форма представления сведений о результате (промежуточный отчет, итоговый отчет), ключевые слова.

Объект "Сведения об ОИС" содержит сведения об ОИС, полученном в результате выполнения работ по данной теме.

Логическая модель данных базы данных ИАС "Учет РИД" приведена на рис. 2. Физическая модель данных разработана как витрина данных с неаддитивными фактами (в качестве факта выступает результат) [6].

Информационно-аналитическая система "Учет РИД" и ее интерфейс

Информационно-аналитическая система состоит из следующих приложений:

- "Регистрация результата интеллектуальной деятельности", с помощью которого подразделениями института (научными отделами) фиксируются сведения о РИД в базе данных (рис. 3).
- "Регистрация объекта интеллектуальной собственности", которое предназначено для размещения подразделениями института (научными отделами) сведений об ОИС в базе данных (рис. 4).
- "Поиск результатов и справки". Данное приложение используется для поиска и визуализации сведений о РИД и об ОИС руководством института.
- "Отчеты". В приложении делается вывод и выполняется

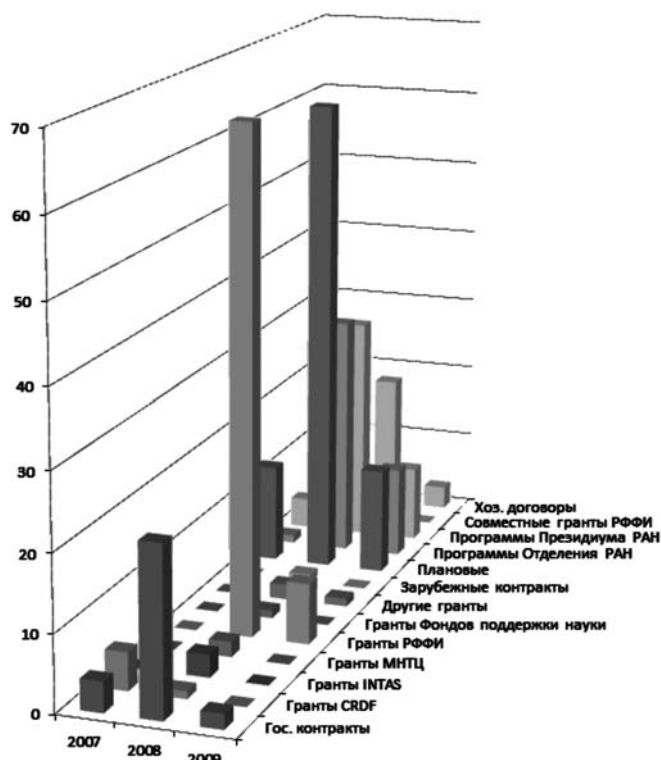


Рис. 5. Пример аналитического отчета

подготовка отчетов о РИД и об ОИС сотрудниками Ученого секретариата для предоставления их руководству института и в вышестоящие организации.

- Модуль "Анализ", который позволяет исследовать сведения о РИД в базе данных на предмет выявления смысловых дубликатов в целях последующего принятия решений об укрупнении тем исследований.

Пример аналитического отчета приведен на рис. 5.

Заключение

Информационно-аналитическая система внутреннего учета РИД академического позволяет:

- учреждениям РАН осуществлять внутренний управленческий учет и мониторинг результатов научной, научно-практической и технологиче-

ской деятельности, проводить оперативный анализ этих результатов и корректировать планы научных исследований;

- поставлять отделениям РАН отчеты о результатах инновационной деятельности института для анализа хода реализации программ фундаментальных исследований отделений РАН.

Автоматизация процесса управленческого учета научной и научно-практической деятельности институтов РАН за счет внедрения внутриинститутских систем учета результатов, полученных в процессе научных исследований и выполнения хозяйственных работ, обеспечивает оперативную поддержку принятия решений руководством института и позволяет оптимизировать планы научных исследований. Кроме того, в рамках функционирования системы происходит интеграция информационных электронных потоков данных об инновационной деятельности подразделений института в информационные электронные ресурсы.

Список литературы

1. База данных РИД Министерства образования и науки Российской Федерации "Государственный учет результатов научно-исследовательских, опытно-конструкторских и технологических работ гражданского назначения". URL: <http://intelpro.extech.ru/>
2. О Федеральной автоматизированной информационной системе (АИС) государственного учета результатов интеллектуальной деятельности. Электронный ресурс. ФГУ "ФАПРИД". URL: <http://www.faprid.ru/pages.php?id=8>
3. АСУ ИнтеллектУМ. URL: <http://www.intellectum.ru/index.php?aid=32>
4. Алдошин С. М., Амосова Е. С., Волохов В. М., Туманов В. Е. Информационная система для поддержки фундаментальных и прикладных исследований в институтах РАН // Сб. тр. Пятой междунар. научно-практ. конф. "Исследование, разработка и применение высоких технологий в промышленности" 28—30 апреля 2008. Санкт-Петербург: Изд-во Политехнического университета. 2008. Т. 12. "Высокие технологии, фундаментальные и прикладные исследования, образование". С. 5—6.
5. Алдошин С. М., Волохов В. М., Рыков А. С., Прохоров А. И., Туманов В. Е. Информационно-аналитическая система внутреннего учета результатов научно-технической деятельности в институте РАН // Приложение к журналу "Открытое образование". Матер. XXXVI Международной конференции и дискуссионного научного клуба "Информационные технологии в науке, социологии, экономике и бизнесе". IT + SE'09. Майская сессия. Украина. Крым. Ялта—Гурзуф. 20—30 мая 2009. С. 133—134.
6. Туманов В. Е. Проектирование хранилищ данных для систем бизнес-аналитики: Учеб. пособие. М.: Интернет-университет информационных технологий: БИНОМ. Лаборатория знаний. 2010. 615 с.

Р. И. Морылев, науч. сотр.,
Южный федеральный университет,
e-mail: rmorylev@gmail.com

Инструмент определения распараллеливаемых циклов

Рассматриваются условия возможности параллельного выполнения циклов на машинах с архитектурой SIMD и MIMD как с общей, так и с распределенной памятью. Представлено описание автоматического инструмента, реализующего описанный анализ в Диалоговом высокоуровневом оптимизирующем распараллеливателе.

Ключевые слова: распараллеливание циклов, информационная зависимость, SIMD, MIMD, общая память, распределенная память

Введение

В последние годы многопроцессорные вычислительные системы вытеснили однопроцессорные. Но несмотря на прогресс в области аппаратного обеспечения, создание программ для таких систем остается трудной задачей, так как требует особых знаний и навыков от программистов. В то время, когда преобладали однопроцессорные системы, было написано огромное число последовательного кода, отказаться от которого по причине сложности и дороговизны переписывания невозможно. Поэтому возникает задача нахождения параллелизма в существующих последовательных программах. Известно, что большую часть времени выполнения программы занимают циклы, поэтому в первую очередь необходимо уметь находить параллелизм в циклах.

В рамках проекта Диалогового высокоуровневого оптимизирующего распараллеливателя (ДВОР) реализован инструмент, позволяющий автоматически определять возможность параллельного выполнения для любого цикла в программе. Он может быть использован как при разработке автоматически распараллеливающих компиляторов, так и в консультационных целях при ручном распараллеливании. Отметим, что данный инструмент позволяет проверить лишь корректность распараллеливания, а решение вопросов его целесообразности и эффективности возлагается на пользователя.

Граф информационных связей

Граф информационных связей — ориентированный граф, вершины которого соответствуют вхождениям переменных, а дуга соединяет пару вершин (u, v) , если выполняются следующие условия [1]:

- эти вхождения обращаются к одной и той же ячейке памяти (т. е. порождают информационную зависимость);
- на графе потока управления программы существует путь от u к v .

Пример 1.

```
for (i = 1; i < N; ++i)
```

```
    A[i] = A[i - 1];
```

Вхождения переменной A в теле цикла порождают информационную зависимость, поскольку оба обращаются к ячейке памяти $A[1]$: вхождение $A[i]$ — на первой итерации цикла, а $A[i - 1]$ — на второй.

Дуги информационных зависимостей бывают четырех видов, в зависимости от типов начала и конца:

- от генератора к использованию — истинная информационная зависимость;
- от использования к генератору — антизависимость;
- от генератора к генератору — выходная зависимость;
- от использования к использованию — входная зависимость.

Информационная зависимость между вхождениями называется циклически независимой (*loop independent dependence*), если эти вхождения обращаются к одной и той же ячейке памяти на одной и той же итерации цикла [1]. Иначе дуга называется циклически порожденной.

Условия распараллеливания циклов

Цикл называется распараллеливаемым, если его итерации можно выполнять одновременно. Для определения возможности параллельного (независимого) выполнения итераций цикла на архитектурах SIMD и MIMD используется граф информационных связей. Доказано, что итерации цикла можно выполнять параллельно, если на графе информационных связей не существует циклически порожденных рассматриваемым циклом дуг зависимости [1, с. 82]. Но это условие является лишь достаточным, более точный набор условий задается архитектурой и типом памяти целевой системы.

На машинах с общей памятью допускается наличие циклически порожденных дуг входной зависимости (от использования к использованию).

На машинах с распределенной памятью возможно параллельное выполнение итераций цикла, даже если на графе информационных связей существуют циклически порожденные дуги антизависимости или входной зависимости. Но при этом необходимо предварительно разослать данные по вычислительным узлам.

Для того чтобы итерации цикла можно было выполнять параллельно на машине с архитектурой SIMD, необходимо выполнение следующих условий [3]:

- границы цикла должны быть известны на этапе компиляции;
- в теле цикла могут быть только операторы присваивания и циклы DO с единственным входом и единственным выходом;
- на графе информационных связей рассматриваемого цикла не должно быть дуг зависимости, ведущих от вхождений, выполняющихся раньше, к вхождениям, выполняющимся позже.

На рис. 1 (см. четвертую сторону обложки) представлена копия экранной формы ДВОР с примером визуализации графа информационных связей.

Сложность представляют циклы, в теле которых присутствуют вызовы функций. В общем случае параллельно выполнять итерации таких циклов нельзя, так как вызываемая функция может иметь побочные эффекты (изменение глобальных переменных, вывод в файл и т. д.). В таких случаях необходимо дополнительно проводить межпроцедурный анализ. Если в результате такого анализа обнаружится, что функция не имеет побочных эффектов, то рассматриваемый цикл можно выполнять параллельно.

Реализация инструмента в ДВОР

Внутреннее представление ДВОР [4] является деревом. Каждому циклу исходной программы соответствует узел внутреннего представления. Любой узел дерева внутреннего представления ДВОР можно пометить. Этот механизм используется для пометки параллельных циклов. После применения алгоритма определения параллельности каждый цикл может быть помечен набором меток в зависимости от архитектуры машины, на которой итерации этого цикла можно выполнять параллельно:

- SISM — архитектура SIMD с общей памятью;
- SIDM — архитектура SIMD с распределенной памятью;
- MISM — архитектура MIMD с общей памятью;
- MIDM — архитектура MIMD с распределенной памятью.

После прохода алгоритма все циклы разделяются на две группы:

- не помеченные — это циклы, параллельное выполнение которых невозможно ни на одной из архитектур;
- помеченные — циклы, итерации которых можно параллельно выполнять на одной или нескольких архитектурах.

Данные метки могут в дальнейшем использоваться другими компонентами ДВОР: генераторами кода, визуализацией и т. д. Например, генератор MPI-кода может использовать эту информацию, чтобы определять, для каких циклов в про-

грамме можно генерировать параллельный код и в каких случаях необходимо генерировать код для предварительной рассылки данных. Далее приводится пример визуализации меток в форме отчета.

В ДВОР граф информационных связей строится автоматически. На рис. 2 (см. четвертую сторону обложки) изображена копия экранной формы ДВОР с визуализацией графа информационных связей для одного из фрагментов программы-решателя `composeV4_100_corr_phi` из пакета ACELAN [2]. Автоматическое построение графа информационных связей позволяет быстро и без участия пользователя определить возможность параллельного выполнения для каждого из циклов в программе.

Визуализация результатов

Программа, определяющая параллельность циклов, в результате своей работы формирует отчет. В заголовке отчета приводится суммарная информация об общем числе циклов в программе, числе распараллеливаемых и нераспараллеливаемых циклов. После заголовка следует подробный список циклов программы. Циклы нумеруются и группируются по функциям, в которых они встречаются. Для краткости выводятся только заголовки циклов в порядке их следования в программе.

Каждому заголовку цикла предшествует одна из следующих меток: SEQUENTIAL или PARALLEL (<метки архитектур>). Метка SEQUENTIAL говорит о том, что помеченный ею цикл нельзя выполнять параллельно. Метка PARALLEL — что итерации цикла можно выполнять параллельно только на машинах с архитектурой, указанной в скобках (SISM, SIDM, MISM или MIDM). Если цикл нельзя выполнять параллельно на одной или нескольких архитектурах, то после заголовка цикла будет выведен список причин, по которым это невозможно. Возможны следующие причины:

- тело цикла содержит вызовы функций с побочными эффектами;
- тело цикла имеет больше одной точки входа или выхода;
- цикл не является каноническим;
- тело цикла содержит операторы, отличные от операторов присваивания;
- цикл порождает зависимость по данным.

Эта информация может быть использована в дальнейшем для повышения параллельности путем выполнения преобразований над программой.

Если цикл можно выполнять параллельно при определенных условиях, то после метки архитектуры перечисляются все такие условия. На данный момент в программе реализовано автоматическое определение следующих условий:

- необходима предварительная рассылка данных;
- необходима приватизация переменных (указывается список переменных);

- необходима редукция (указывается переменная и операция).

На рис. 3 (см. четвертую сторону обложки) приведена экранная форма с фрагментом отчета о распараллеливаемости циклов в программе `composeV4_100_corr_phi`. На ней видно, что цикл под номером 4 помечен метками `MISM` и `MIDM`. Ниже для метки `MIDM` указано, что для распараллеливания требуется предварительная рассылка данных. Для меток `SISM` и `SIDM` выведен список причин, по которым распараллеливание невозможно: цикл не простой; в теле цикла содержатся операторы, отличные от оператора присваивания, цикл порождает информационную зависимость.

Список литературы

1. **Kennedy K., Allen J.** Optimizing Compilers for Modern Architectures: A Dependence-Based Approach. San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., 2001.
2. **Наседкин А. В., Скалиух А. С., Соловьев А. Н.** Пакет ACELAN и конечно-элементное моделирование гидроакустических пьезопреобразователей // Известия вузов. Северо-Кавказский регион. Естеств. науки. 2001. Спецвыпуск. Математическое моделирование. С. 122—125.
3. **Lamport I.** The parallel execution of DO loops / Commun. ACM. — 1974. V. 17, N 2. P. 83—93.
4. **Петренко В. В.** Внутреннее представление Reprise распараллеливающей системы // Труды Четвертой международной конференции "Параллельные вычисления и задачи управления" РАСО'2008. Москва, 27—29 октября 2008. С. 1241—1245.

CONTENTS

Kazakov P. V. *The Genetic Algorithms for Multi-Objective Optimization. Review* 2

The special modifications of genetic algorithm for multi-objective optimization are considered. Their abilities, features and state-of-the-art analyzing are presented.

Keywords: multi-objective optimization, Pareto's principles, Pareto front, multi-objective genetic algorithms

Eltarenko E. A. *Operators of Soft Logic in Multicriteria Problems* 8

Classes of preferences in multicriteria problems, named soft logic are defined. For them axioms of preferences are formulated. Procedures of construction of operators of aggregation are developed for the description of such preferences, based on generating functions. The theorems are proved, allowing to define requirements to generating functions for maintenance of performance of axioms of soft logic. Examples of operators of soft logic are resulted.

Keywords: multicriteria problems, aggregation of criteria, preferences, the soft logic

Norenkov I. P. *Reduction of Direct Acyclic Graph in DAG Scheduling Problems* 18

The main approach to solving the DAG Scheduling Problems is genetic algorithms, but there are some difficulties for application of them if the problems have significant size. Efficiency of solving is increased by clusterization, i. e. model transformation from the DAG to the graph of clusters.

Keywords: scheduling problem, direct acyclic graph, genetic algorithms, clusterization, DAG reduction

Strijov V. V., Krymova E. A. *Model Selection in Linear Regression Analysis* 21

To obtain an adequate regression model one often has to enlarge the feature set by generating of derivative features. So the regression problem must be reformulated as the problem of the feature selection. Hereby we assume that the number of features is almost equal of exceeds the number of samples in the data set and present a comparative study of classical and new feature selection algorithms. The study is illustrated by the problem of European option volatility modelling.

Keywords: regression analysis, model selection, linear models, parameter estimation, algorithm comparison

Bobkov S. G. *Design and Architecture Decisions for 1-100 Tflops Computer* 26

Base decisions for designing high-performance computer with performance 1—100 Tflops are consider.

Keywords: supercomputer, supercomputer technologies, computer architecture

Glivenko E. V., Pryadko S. A., Fomochkina A. S., Levonyan K. A. <i>Parallel Methods of Linear and Non-Linear Algebraical Equations Systems Solution</i>	31
--	-----------

New parallel methods of linear and non-linear algebraical equations systems solution are proposed. Examples of realizations these methods by parallel structures.

Keywords: geometrical interpretation, grid, node

Andreev D. V. <i>Analog Multiplexers for Logical Conversion of Continual Data</i>	35
--	-----------

In article produced two methods of structural organization of analog multiplexers. These methods are based on the offered decompositions of functions of continual logic and give simple circuit decisions. Features of adjustment of these multiplexers on realization of continual logical functions of a corresponding class are considered.

Keywords: continual logic, functional decomposition, universal logical devices

Bernikov A. R., Grafov R. P., Lysenko S. N., Savenko O. S. <i>Search Malware in Distributed Simulator with Technology of Fuzzy Logic</i>	42
---	-----------

In this paper the methodology of solving the problem of malware detection in distributed information systems, characterized by fuzzy information about the current processes. Used the mechanism of fuzzy inference based on production rules decision. The results can be used for diagnosing antivirus software of different computer systems.

Keywords: fuzzy inference, finding malware, the fuzzy model of decision making, fuzzy production rules and strategies, semantic variables, the function of accessory

Borisov V. V., Senkov A. V. <i>The Complex Managing of Risks in a Organizational-Technical Systems</i>	47
---	-----------

A method of intelligent managing of risks in a complex organizational-technical systems suggested, which is allows: complex risk-management of professional and industrial risks in several levels of hierarchy of complex system; hybridization of intelligent technologies at each stage of risk management; adaptability in the process of risk management.

Keywords: intelligent managing of risks, professional and industrial risks, fuzzy modeling

Zak Yu. A. <i>System Probabilistic Criteria and Limitations in the Problems of Choice of Optimal Portfolio Investment</i>	52
--	-----------

Target formation effective portfolio securities is formulated as a multiobjective mathematical programming problems with probability-governmental restrictions. Based on the properties set the left sides of the restrictions and local performance criteria problem, are the operators and iterative process of elimination of subsets of variables that do not contain admissible solutions, and effective assessment of the prospects of development of different options. The algorithms for solving the problem based on modifications of the "branch and bound method" which en-carton pack made to solve the problem for a fairly broad class of practically applications.

Keywords: efficient portfolio asset securities, probabilistic constraints, multicriteria optimization, separable monotone functions, modification of the "branch and bound method"

Benevolenskiy S. B., Kirillov Yu. I., Kolosyanko D. M., Pozhar V. E., Pozhar K. V., Pustovoyt V. I. <i>Improvement of the Software for Processing Data Received from Acousto-Optic Spectrometers</i>	60
---	-----------

This article describes the problems encountered in the processing of information received from the acousto-optic spectrometer, as well as possible solutions. It is necessary to improve approaches to processing of spectral information and it is shown that specific features of the developed software will make it possible to increase convenience and efficiency of data processing with acousto-optic spectrometer.

Keywords: software for acousto-optic spectrometer, spectral analysis, spectrum processing, data storage, graphical display of information, the information core

Trachtengerts E. A. <i>Information Technologies Correction Goals and Strategic Decision in the Dynamics of Public Opinion Manipulation</i>	64
---	-----------

Considered computer correction methods and algorithms goals and strategic decisions in the dynamics of public opinion manipulation

Keywords: information technologies, correction algorithms, goals, strategic decisions

Malafeev S. S., Chernyaev A. V. System for Visualizing of Work Process of Excavator. 70

This paper considering system for visualizing of work process of excavator, based on technology of three-dimension computer graphic and using data from digital control system of main movement drives for determination of position, direction and speed of movement of mechanisms of excavator. Proposed system uses only existing hardware units and allows to improve efficiency of monitoring for excavators of mining complex.

Keywords: excavator, monitoring, visualizing, computer graphic

Prokhorov A. I., Tumarov V. E., Psika B. L., Polikarpov K. I. Information-Analytical System for Management Accounting of Results of Intellectual Activity in the Scientific Organization 72

In article the information-analytical system of monitoring and management accounting of results of intellectual activity in the academic institute, including management of results of scientific and technical activity is considered. The database system accumulates in itself data on all results of intellectual activity received by scientific divisions in the institute. This work was supported by grant from the Russian Foundation for Basic Research 09-00481-a.

Keywords: the automated information system, online access, management accounting, objects of intellectual property

Morylev R. I. Parallel Loop Determination Tool. 76

The paper consider the conditions of parallel execution of do loops on SIMD and MIMD machines both with shared and distributed memory. The description of the automatic tool implementing described analysis in High-level dialogic optimizing parallelizer is presented.

Keywords: loop parallelization, information dependency, SIMD, MIMD, shared memory, distributed memory

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.

Корректор *М.Г. Джавадян*.

Сдано в набор 05.08.2011. Подписано в печать 19.09.2011. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.

Усл. печ. л. 9,8. Уч.-изд. л. 10,95. Заказ 691. 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
