

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

6(178)
2011

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Домрачев В. Г., Полещук О. М. Определение рейтинговых оценок совокупностей объектов и мониторинг их состояний в условиях нечеткой исходной информации 2
- Бородащенко А. Ю., Рябцев А. О. Алгоритм оценки массива гипертекстовых документов на семантическое сходство с эталоном 7

РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ

- Топорков В. В., Целищев А. С. Метод критических работ как перспектива эффективной организации распределенных вычислений 13
- Амиршахи Б. GRID-технологии решения больших систем линейных уравнений на вычислительной сети и на суперкомпьютере кластерного типа 17

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Елтаренко Е. А. Аппроксимация предпочтений в многокритериальных задачах операторами агрегирования 23
- Кухаренко Б. Г., Пономарев Д. И. Нелинейная байесовская фильтрация многомерных временных рядов 33
- Филиппова А. С., Фролов Р. В. О создании оперативных сетевых графиков работ бригад текущего и капитального ремонта скважин 39
- Бронштейн Е. М., Заико Т. А. Задача маршрутизации с запретами 42
- Семин В. В. Моделирование поведения стохастической триангуляции на квазистационарных решетках 45

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Пекунов В. В. Метаслой моделирования алгоритма, данных и функциональных характеристик последовательных и параллельных программ 51
- Беневоленский С. Б., Кириллов Ю. И., Пожар В. Э., Пустовойт В. И., Спиридонов И. С. Разработка платформы программного комплекса для акустооптических спектрометров 57

ОБРАБОТКА СИГНАЛОВ

- Замятин А. В. Дифференцированное сжатие аэрокосмических изображений с потерями 60

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

- Куприянов Ю. В., Таратухин В. В. Принципы управления выгодами на проектах внедрения корпоративных информационных систем 66

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Зимин А. М., Шумов А. В., Кривицкий С. Е., Тройнов В. И. Автоматизированная сетевая учебно-научная лаборатория по спектроскопии плазмы 72
- Contents 78

Приложение. Памяти А. С. Нариньяни

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

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

Редакция:

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

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

Журнал включен в систему Российского индекса научного цитирования.

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

УДК 004.942

В. Г. Домрачев, д-р техн. наук, проф.,

О. М. Полещук, д-р техн. наук, проф.,

e-mail: olga.m.pol@yandex.ru

Московский государственный университет леса

Определение рейтинговых оценок совокупностей объектов и мониторинг их состояний в условиях нечеткой исходной информации

Рассматривается достаточно новое направление обработки информации — обработка нечеткой информации. Актуальность этого направления в свое время определилась переходом к изучению организационно-технических задач, процессов принятия решений человеком и систем управления. В работе применяется аппарат теории нечетких множеств для построения рейтинговых оценок совокупностей объектов и их использования для выработки управляющих рекомендаций.

Ключевые слова: совокупность объектов, рейтинговые оценки, лингвистические переменные

Введение

Рейтинговые системы широко применяются в различных областях деятельности человека и играют существенную роль в задачах поддержки принятия решений. Эти системы позволяют получать своевременную информацию в виде некоего интегрального показателя, который используется для принятия управляющих воздействий. Сложность построения рейтинговых систем вытекает из общей сложности количественного оценивания процессов управления. Отчасти эта сложность связана с особенностью измерения качественных характеристик, состоящей в необходимости учета свойств или суждений лиц, проводящих оценивание и принимающих на основе этого субъективного измерения решения. Не меньшую сложность вызывает рейтинговое оценивание состояний объектов в рамках количественных и качественных характеристик, измеренных в разных шкалах и несопоставимых по своей сути. Следствием этого является проблематичность получения

рейтинговых оценок на основе традиционных свертков отдельных показателей [1—3].

Успешное функционирование объектов и совокупностей объектов различных сфер деятельности человека существенным образом зависит от того, какой аппарат используется для обработки данных о состояниях этих объектов. Если рассматриваются объекты с нечеткими описаниями их состояний, то сложность и ответственность задач обработки информации возрастают в силу ряда причин. Одной из основных причин является некорректность арифметических операций для элементов порядковых шкал, используемых для оценивания состояний объектов, которая приводит к неустойчивым конечным выводам.

В последние годы появилось значительное число работ, предлагающих избежать этой некорректности, используя аппарат теории нечетких множеств [4—13]. Отличие результатов авторов статьи от результатов других исследователей состоит в том, что кроме четких (точечных) рейтинговых оценок получены нечеткие рейтинговые оценки и интервальные оценки с заданным уровнем доверия. Подобное расширенное представление результатов соответственно позволяет расширить класс задач, решаемых на их основе: сравнительный анализ выходящих данных, их распознавание в рамках профессионального языка специалистов проблемной области, кластеризация, мониторинг данных, выработка управляющих рекомендаций. В работе [13] разработаны методы определения рейтинговых оценок объектов в рамках одной и нескольких качественных характеристик, а также в рамках количественных и качественных характеристик. Однако на практике возникают задачи нахождения устойчивых рейтинговых оценок не только отдельных объектов, но и совокупностей объектов. Эта задача не получила достойного внимания со стороны исследователей, в результате чего образовался пробел, который предполагается устранить в настоящей работе.

1. Определение рейтинговых оценок совокупностей объектов в рамках одной качественной характеристики

Рассмотрим k совокупностей объектов, у которых оценивается интенсивность проявления качественной характеристики X . Оценивание проводится в рамках вербальной шкалы с уровнями X_j ,

$l = \overline{1, m}$. Обозначим $a_l^j, l = \overline{1, m}, j = \overline{1, k}$ — относительное число объектов j -й совокупности, отнесенных к уровню $X_p, l = \overline{1, m}$, а $a_p, l = \overline{1, m}$ — относительное число объектов совокупностей, отнесенных к уровню $X_j, l = \overline{1, m}$.

Опираясь на данные $a_p, l = \overline{1, m}$, и метод, приведенный в работе [13], построим лингвистическую переменную с названием X , терминами $X_p, l = \overline{1, m}$, и функциями принадлежности $\mu_l(x) \equiv (b_1^l, b_2^l, b_L^l, b_R^l)$,

$l = \overline{1, m}$, соответствующих им нечетких чисел $\tilde{X}_l, l = \overline{1, m}$. Первые два параметра в скобках — абсциссы соответственно левого и правого концов верхнего основания трапеции, которая является графиком функции принадлежности. Последние два параметра — длины соответственно левого и правого крыльев трапеции.

Лингвистической переменной называется пятерка

$$\{X, T(X), U, V, S\},$$

где X — название переменной; $T(X) = \{X_p, i = \overline{1, m}\}$ — терм-множество переменной X , т. е. множество термов или названий лингвистических значений переменной X (каждое из этих значений — нечеткая переменная со значениями из универсального множества U); V — синтаксическое правило, порождающее названия значений лингвистической переменной X ; S — семантическое правило, которое ставит в соответствие каждой нечеткой переменной с названием из $T(X)$ нечеткое подмножество универсального множества U .

Определим нечеткую рейтинговую оценку j -й совокупности объектов в рамках проявления качественной характеристики X в виде нечеткого числа \tilde{C}_j с функцией принадлежности

$$\beta_j = a_1^j \otimes \mu_1 \oplus a_2^j \otimes \mu_2 \oplus \dots \oplus a_m^j \otimes \mu_m$$

или

$$\beta_j \equiv \left(\sum_{l=1}^m a_l^j b_1^l, \sum_{l=1}^m a_l^j b_2^l, \sum_{l=1}^m a_l^j b_L^l, \sum_{l=1}^m a_l^j b_R^l \right), \\ j = \overline{1, k}.$$

Первые два параметра в скобках — соответственно абсциссы верхнего основания трапеции, которая является графиком функции принадлежности нечеткого числа \tilde{C}_j . Два оставшихся параметра — длины соответственно левого и правого крыльев трапеции.

Определим доверительный интервал для четкой рейтинговой оценки z_j , характеризующей проявление характеристики X у j -й совокупности объектов. При уровне доверия $\beta_j(z_j) \geq \alpha, 0 < \alpha < 1$, рейтинговая оценка z_j проявления характеристики X у j -й совокупности объектов лежит в интервале

$$\sum_{l=1}^m a_l^j b_1^l - L^{-1}(\alpha) \sum_{l=1}^m a_l^j b_L^l \leq z_j \leq \\ \leq \sum_{l=1}^m a_l^j b_2^l + R^{-1}(\alpha) \sum_{l=1}^m a_l^j b_R^l.$$

Дефазифицируем нечеткие числа с функциями принадлежности $\beta_j, j = \overline{1, k}, \mu_1, \mu_m$. Полученные числа обозначим через $C_j, j = \overline{1, k}, B_1, B_m$.

Число C_j назовем точечной рейтинговой оценкой проявления качественной характеристики X у j -й совокупности объектов, $j = \overline{1, k}$.

Нормированную рейтинговую оценку $E_j, j = \overline{1, k}$, проявления качественной характеристики X у j -й совокупности объектов найдем по формуле

$$E_j = \frac{C_j - B_1}{B_m - B_1}, j = \overline{1, k}.$$

Назовем оценку $E_j, j = \overline{1, k}$, средней степенью интенсивности проявления характеристики X у j -й совокупности объектов. Область изменения $E_j, j = \overline{1, k}$, есть отрезок $[0, 1]$.

2. Определение рейтинговых оценок совокупностей объектов в рамках нескольких качественных характеристик

Рассмотрим k совокупностей объектов, у которых оцениваются интенсивности проявлений качественных характеристик X_1, \dots, X_m . Оценивание характеристик проводится в рамках вербальных шкал с числом уровней соответственно k_1, k_2, \dots, k_m .

Обозначим через $k_{ij}^p, i = \overline{1, k}, j = \overline{1, k_p}, p = \overline{1, m}$, — относительное число объектов i -й совокупности, отнесенных при оценивании характеристики $X_p, p = \overline{1, m}$, к j -му уровню вербальной шкалы, а через $k_j^p, j = \overline{1, k_p}, p = \overline{1, m}$, — относительное число объектов всех совокупностей, отнесенных при оценивании характеристики $X_p, p = \overline{1, m}$, к j -му уровню вербальной шкалы. Опираясь на данные $k_j^p, j = \overline{1, k_p}, p = \overline{1, m}$, и метод [13], построим m лингвистических

переменных с названиями $X_p, p = \overline{1, m}$, и функциями принадлежности терм-множеств $\mu_j^p, j = \overline{1, k_p}, p = \overline{1, m}$.

Определим функцию принадлежности нечеткой рейтинговой оценки, определяющей интенсивность проявления качественной характеристики $X_p, p = \overline{1, m}$, у i -й совокупности объектов формулой

$$\lambda_i^p = k_{i1}^p \otimes \mu_1^p \oplus k_{i2}^p \otimes \mu_2^p \oplus \dots \oplus k_{ik_p}^p \otimes \mu_{k_p}^p, \\ i = \overline{1, k}.$$

Обозначим через $\omega_p, p = \overline{1, m}$, весовые коэффициенты (коэффициенты важности) характеристик $X_p, p = \overline{1, m}$, $\sum_{p=1}^m \omega_p = 1$.

Функцию принадлежности нечеткой рейтинговой оценки, определяющей интенсивность проявления качественных характеристик X_1, \dots, X_m у i -й совокупности объектов, определим следующим образом:

$$\lambda_i = \omega_1 \otimes \lambda_i^1 \oplus \dots \oplus \omega_m \otimes \lambda_i^m, i = \overline{1, k}.$$

Определим доверительный интервал для четкой рейтинговой оценки z_p , характеризующей проявление качественных характеристик X_1, \dots, X_m у i -й совокупности объектов, $i = \overline{1, k}$. Пусть $\lambda_i \equiv (\Delta_1^i, \Delta_2^i, \Delta_L^i, \Delta_R^i)$, тогда при уровне доверия $\lambda_i(z_i) \geq \alpha, 0 < \alpha < 1$, рейтинговая оценка z_i проявления характеристик X_1, \dots, X_m у i -й совокупности объектов лежит в интервале

$$\Delta_1^i - L^{-1}(\alpha)\Delta_L^i \leq z_i \leq \Delta_2^i + R^{-1}(\alpha)\Delta_R^i, i = \overline{1, k}.$$

Дефазифицируем нечеткие числа с функциями принадлежности

$$\lambda_p, i = \overline{1, k}, \eta = \omega_1 \otimes \mu_1^1 \oplus \dots \oplus \omega_m \otimes \mu_1^m, \\ \delta = \omega_1 \otimes \mu_{k_1}^1 \oplus \dots \oplus \omega_m \otimes \mu_{k_p}^m.$$

Полученные нечеткие числа обозначим через $A_p, i = \overline{1, k}, B, C$.

Число A_i называется точечной рейтинговой оценкой проявления качественных характеристик X_1, \dots, X_m у i -й совокупности объектов, $i = \overline{1, k}$.

Нормированную рейтинговую оценку i -й совокупности объектов найдем по формуле

$$E_i = \frac{A_i - B}{C - B}, i = \overline{1, k}.$$

Назовем оценку $E_i, i = \overline{1, k}$, средней степенью интенсивности проявления качественных характеристик X_1, \dots, X_m у i -й совокупности объектов.

Область изменения $E_i, i = \overline{1, k}$, есть отрезок $[0, 1]$.

Следует отметить, что работая с разными качественными характеристиками, авторы статьи оперируют абстрактными понятиями — функциями принадлежности лингвистических значений этих признаков. Операции для лингвистических значений, как известно, определяются на основе треугольных норм и треугольных конорм и соответственно не являются привычными арифметическими операциями [14].

3. Определение рейтинговых оценок объектов в рамках нескольких количественных и качественных характеристик

Рассмотрим N совокупностей объектов, у которых оцениваются количественные характеристики $X_j, j = \overline{1, l}$, и интенсивности проявления качественных характеристик $X_v, v = \overline{1+1, k}$. В совокупности оцениваемые характеристики оказывают существенное влияние на характеристику Y — успешность функционирования совокупностей объектов, которая оценивается в рамках следующей шкалы: Y_1 = "предельно неуспешно"; Y_2 = "неуспешно"; Y_3 = "средне успешно"; Y_4 = "относительно успешно"; Y_5 = "предельно успешно".

Построим, используя информацию экспертов, лингвистическую переменную с названием Y и термами $Y_i, i = \overline{1, 5}$ [13]. Обозначим через $\tilde{Y}_i, i = \overline{1, 5}$, нечеткие числа с функциями принадлежности $\mu_i(x), i = \overline{1, 5}$, которые соответствуют термам $Y_i, i = \overline{1, 5}$.

Областями значений количественных характеристик $X_j, j = \overline{1, l}$, могут являться несчетные множества точек действительной прямой — $R_j, j = \overline{1, l}$. Построим, используя информацию экспертов, лингвистические переменные с названиями $X_j, j = \overline{1, l}$ [13]. Если рост характеристики $X_j, j = \overline{1, l}$, сопровождается ростом характеристики Y , то "очень малое значение характеристики X_j ", "малое значение характеристики X_j ", "среднее значение характеристики X_j ", "большое значение характеристики X_j ", "очень большое значение характеристики X_j " являются термами лингвистической переменной, а $\mu_{ij}(x), i = \overline{1, 5}, j = \overline{1, l}$, — их функциями принадлежности. Если рост характеристики $X_j,$

$j = \overline{1, l}$, сопровождается уменьшением характеристики Y , то "очень большое значение характеристики X_j ", "большое значение характеристики X_j ", "среднее значение характеристики X_j ", "малое значение характеристики X_j ", "очень малое значение характеристики X_j " являются термами лингвистической переменной с названием $X_j, j = \overline{1, l}$, а $\mu_{ij}(x), i = \overline{1, 5}, j = \overline{1, l}$, — их функциями принадлежности.

Обозначим через $x_j^n, n = \overline{1, N}, j = \overline{1, l}$, среднее арифметическое значение количественной характеристики $X_j, j = \overline{1, l}$, у n -й совокупности объектов, $n = \overline{1, N}$, а через $\mu_{ij}(x_j^n), i = \overline{1, 5}, j = \overline{1, l}, n = \overline{1, N}$, степень принадлежности этого значения к термам лингвистической переменной с названием $X_j, j = \overline{1, l}$.

Построим $k - l$ лингвистических переменных с названиями $X_v, v = \overline{l+1, k}$ [13]. Если рост характеристики $X_v, v = \overline{l+1, k}$, сопровождается ростом характеристики Y , то "очень малое значение характеристики X_v ", "малое значение характеристики X_v ", "среднее значение характеристики X_v ", "большое значение характеристики X_v ", "очень большое значение характеристики X_v " являются термами лингвистической переменной $X_v, v = \overline{l+1, k}$, а $\mu_{iv}(x), i = \overline{1, 5}, v = \overline{l+1, k}$, — их функциями принадлежности. Если рост характеристики $X_v, v = \overline{l+1, k}$, сопровождается уменьшением характеристики Y , то "очень большое значение характеристики X_v ", "большое значение характеристики X_v ", "среднее значение характеристики X_v ", "малое значение характеристики X_v ", "очень малое значение характеристики X_v " являются термами лингвистической переменной с названием $X_v, v = \overline{l+1, k}$, а $\mu_{iv}(x), i = \overline{1, 5}, v = \overline{l+1, k}$, — их функциями принадлежности. Обозначим через $\tilde{X}_{iv}, i = \overline{1, 5}, v = \overline{l+1, k}$, нечеткие числа с функциями принадлежности $\mu_{iv}(x), i = \overline{1, 5}, v = \overline{l+1, k}$.

Обозначим через \tilde{X}_v^n (с функцией принадлежности $\mu_v^n(x), n = \overline{1, N}, v = \overline{l+1, k}$), рейтинговую оценку n -й совокупности объектов в рамках характеристики $X_v, v = \overline{l+1, k}$, определенную в разд. 1 настоящей статьи.

Обозначим через $\delta_i(x_v^n), n = \overline{1, N}, v = \overline{l+1, k}, i = \overline{1, 5}$, функцию, которая принимает значение 1, если рейтинговая оценка n -й совокупности объектов в рамках характеристики $X_v, v = \overline{l+1, k}$, равна нечеткому числу $\tilde{X}_{iv}, i = \overline{1, 5}$, и значение 0, если рейтинговой оценкой n -й совокупности объектов в рамках характеристики $X_v, v = \overline{l+1, k}$, является нечеткое число $\tilde{X}_{pv}, p = \overline{1, 5}, p \neq i$.

Обозначим через $\omega_j, j = \overline{1, k}, \sum_{j=1}^k \omega_j = 1$ весовые коэффициенты оцениваемых характеристик. Вычислим следующие коэффициенты:

$$\lambda_i^n = \sum_{j=1}^l \omega_j \mu_{ij}(x_j^n) + \sum_{v=l+1}^k \omega_v \delta_i(x_v^n),$$

$$i = \overline{1, 5}, n = \overline{1, N}.$$

Найдем сумму этих коэффициентов, используя свойства лингвистических переменных и определение функций $\delta_i(x_v^n), n = \overline{1, N}, v = \overline{l+1, k}, i = \overline{1, 5}$:

$$\sum_{i=1}^5 \lambda_i^n = \omega_1(\mu_{11}(x_1^n) + \mu_{21}(x_1^n) + \dots + \mu_{51}(x_1^n)) +$$

$$+ \dots + \omega_l(\mu_{1l}(x_l^n) + \mu_{2l}(x_l^n) + \dots + \mu_{5l}(x_l^n)) +$$

$$+ \omega_{l+1}(\delta_1(x_{l+1}^n) + \delta_2(x_{l+1}^n) + \dots + \delta_5(x_{l+1}^n)) +$$

$$+ \dots + \omega_k(\delta_1(x_k^n) + \delta_2(x_k^n) + \dots + \delta_5(x_k^n)) = \sum_{i=1}^k \omega_j = 1.$$

Исходя из этого коэффициенты $\lambda_i^n, i = \overline{1, 5}, n = \overline{1, N}$, можно считать весовыми коэффициентами термов характеристики Y для n -й совокупности объектов, $n = \overline{1, N}$.

Нечеткая рейтинговая оценка n -й совокупности объектов, $n = \overline{1, N}$, в рамках характеристик $X_j, j = \overline{1, k}$, определяется в виде нечеткого числа

$$\tilde{A}_n = \lambda_1^n \otimes \tilde{Y}_1 \oplus \dots \oplus \lambda_5^n \otimes \tilde{Y}_5$$

с функцией принадлежности

$$\mu_n(x) =$$

$$\equiv \left(\sum_{i=1}^5 \lambda_i^n a_{i1}, \sum_{i=1}^5 \lambda_i^n a_{i2}, \sum_{i=1}^5 \lambda_i^n a_{iL}, \sum_{i=1}^5 \lambda_i^n a_{iR} \right),$$

$$n = \overline{1, N},$$

где $\tilde{Y}_i \equiv (a_{i1}, a_{i2}, a_{iL}, a_{iR}), i = \overline{1, 5}$.

Определим доверительный интервал для четкой рейтинговой оценки y_n . При уровне доверия $\mu_n(y_n) \geq \alpha, 0 < \alpha < 1$, рейтинговая оценка y_n n -й совокупности объектов, $n = \overline{1, N}$, лежит в интервале

$$\begin{aligned} \sum_{i=1}^5 \lambda_i^n a_{j1} - (1 - \alpha) \sum_{i=1}^5 \lambda_i^n a_{jL} &\leq y_n \leq \\ &\leq \sum_{i=1}^5 \lambda_i^n a_{j2} + (1 - \alpha) \sum_{i=1}^5 \lambda_i^n a_{jR}. \end{aligned}$$

Дефазифицируем нечеткое число $\tilde{A}_n, n = \overline{1, N}$, по методу центра тяжести, полученное четкое число обозначим через $A_n, n = \overline{1, N}$.

4. Распознавание успешности функционирования совокупностей объектов на основе рейтинговых оценок

Пусть нечеткое число $\tilde{A}_n, n = \overline{1, N}$, с функцией принадлежности $\mu_n(x), n = \overline{1, N}$, является нечеткой рейтинговой оценкой n -й совокупности объектов, $n = \overline{1, N}$, полученной в разд. 1—3 настоящей статьи. Для распознавания успешности функционирования совокупностей объектов необходимо идентифицировать нечеткое число $\tilde{A}_n, n = \overline{1, N}$, с одним из термов лингвистической переменной с названием Y (с одним из нечетких чисел $\tilde{Y}_i, i = \overline{1, 5}$, с функциями принадлежности $\mu_i(x), i = \overline{1, 5}$). Для этого вычислим идентификационные показатели:

$$\beta_n^i = \frac{\int_0^1 \min(\mu_i(x), \mu_n(x)) dx}{\int_0^1 \max(\mu_i(x), \mu_n(x)) dx}, i = \overline{1, 5}, n = \overline{1, N}.$$

Если $\beta_n^p = \max_i \beta_n^i$, то состояние n -й совокупности объектов определяется p -м уровнем шкалы $Y_1 =$ "предельно неуспешно", $Y_2 =$ "неуспешно", $Y_3 =$ "средне успешно", $Y_4 =$ "относительно успешно", $Y_5 =$ предельно успешно", $p = \overline{1, 5}$.

Обозначим соответственно через A_n^1, A_n^2 рейтинговые оценки n -й совокупности объектов за периоды 1 и 2. В зависимости от соотношений между

A_n^1, A_n^2 делаются следующие выводы: если $A_n^1 > A_n^2$, то состояние n -й совокупности объектов ухудшилось; если $A_n^1 < A_n^2$, то состояние n -й совокупности объектов улучшилось; если $A_n^1 = A_n^2$, то состояние n -й совокупности объектов не изменилось.

Заключение

Предложен новый подход к определению рейтинговых оценок совокупностей объектов, который позволяет оперировать со значениями разнородных характеристик, формализованными на основе лингвистических переменных.

В работе определены четкие (точечные), нечеткие и интервальные рейтинговые оценки с заданным уровнем доверия. Возможность определения нечетких рейтинговых оценок позволяет оценить успешность функционирования совокупностей объектов путем идентификации ее нечетких значений с полученными рейтинговыми оценками. Таким образом, полученные в статье результаты позволяют анализировать как количественные изменения состояний совокупностей объектов, так и качественные изменения.

Список литературы

1. Хубка В. Теория технических систем. М.: Мир, 1987. 208 с.
2. Панин М. Морфология рейтинга // Высшее образование в России. 1998. № 1. С. 90—94.
3. Литвак Б. Г. Экспертные оценки и принятие решений. М.: Патент, 1996. 271 с.
4. Cheng C. H., Wang J. W., Tsai M. F., Huang K. C. Appraisal support system for high school teachers based on fuzzy linguistic integrating operation // Journal of Human Resource Management. 2004. N 4. P. 73—89.
5. Wang C. H., Chen S. M. Appraising the performance of high school teachers based on fuzzy number arithmetic operations // Soft Computing — A Fusion of Foundations, Methodologies and Applications. 2008. N 12. P. 919—934.
6. Wang Y. H., Yang J. B., Xu D. L., Chin K. S. On the centroids of fuzzy numbers // Fuzzy Sets and Systems. 2006. Vol. 157. P. 919—926.
7. Echaiz J. R., Vachtsevanos G. J. Fuzzy grading system // IEEE Trans. Educ. 1995. Vol. 38, № 2. P. 158—164.
8. Ranjit B. An application of fuzzy set in students' evaluation // Fuzzy Sets and Systems. 1995. Vol. 74. P. 187—194.
9. Biswas R. An application of fuzzy sets in student's evaluation // Fuzzy Set and systems. 1995. Vol. 74. P. 194—197.
10. Capaldo G., Zollo G. Applying fuzzy logic to personnel assessment: A case study // Omega The International Journal. 2001. N 29. P. 585—597.
11. Chen S. M., Lee C. H. New methods for students' evaluation using fuzzy sets // Fuzzy Sets and Systems. 1999. Vol. 104. P. 209—218.
12. Poleshchuk O., Komarov E. The determination of students' fuzzy rating points and qualification levels // Proc. of the 1st International Fuzzy Systems Symposium — FUZZYSS'09. 2009. P. 218—224.
13. Домрачев В. Г., Комаров Е. Г., Полещук О. М. Мониторинг функционирования объектов на основе нечеткого описания их состояний // Информационные технологии. 2007. № 11. С. 46—52.
14. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 165 с.

А. Ю. Бородащенко,
канд. техн. наук, преподаватель,
e-mail: bay55@mail.ru

А. О. Рябцев, слушатель,
e-mail: andrewcs@mail.ru
Академия ФСО России, г. Орел

Алгоритм оценки массива гипертекстовых документов на семантическое сходство с эталоном

Предложен алгоритм оценки массива гипертекстовых документов на семантическое сходство с эталоном, позволяющий повысить качество информационного поиска в сетях Интернет и интранет. Алгоритм реализует функцию отбора гипертекстовых документов из массива данных путем анализа информационного окружения указанного пользователем документа-эталона.

Ключевые слова: семантическое расстояние, гипертекстовая модель, мера близости, алгоритм фильтрации текстов, гипертекстовая связь

В настоящее время наблюдаются высокие темпы роста информатизации по всему миру, которые вызывают галопирующий рост объемов хранимой и передаваемой информации различных форматов. Разнородность информации, сложность ее релевантного поиска и обработки создают проблему формирования у лица, принимающего решение, своевременного, точного и актуального представления об интересующей его предметной области. В этих условиях наиболее эффективным способом организации доступа к необходимой информации является автоматизированный поиск, реализованный на основе интеллектуального анализа данных [1].

Основной проблемой, возникающей при работе в сети Интернет, является поиск документов по их содержанию. Ставшие традиционными средства контекстного поиска по вхождению слов и словосочетаний в документ, представленные привычными поисковыми машинами, зачастую не обеспечивают по запросу пользователя отбор релевантной информации.

Одна из причин заключается в сложности точной формулировки запроса — подбора ключевых слов, которые предстоит искать в телах документов. Это может быть связано с рядом факторов: недостаточным знанием пользователем терминологии предметной области, наличием в языке многозначных и синонимичных слов и даже ор-

фографическими ошибками в написании искомым слов, которые могут встречаться как в текстах, так и в самом запросе.

Другая фундаментальная причина заключается в том, что иногда пользователь не знает точно, какую именно информацию ему хотелось бы получить, имея лишь общее представление о границах своих интересов. Так, например, пытаясь расширить познания в области компьютерной лингвистики, на поисковом сервере Yandex.Ru можно получить список из тысяч документов, содержащих слова "computer" и "linguistic". Возможность классифицировать найденный материал по тематическим группам, отражающим, к примеру, основные события и разработки в этой области, либо отсутствует вообще, либо представлена в крайне примитивном виде.

Наконец, рассмотрение текстов осуществляется в изоляции от их информационного окружения, в качестве которого выступают другие текстовые документы информационного массива, формируемого традиционными средствами поиска. Анализ "соседних" документов позволяет существенно повысить полноту и точность запросов на поиск информации при условии использования в качестве поискового запроса семантического эталона, отражающего информационные потребности пользователя [2].

В информационно-поисковых системах (ИПС) анализ гиперссылок используется для ранжирования результатов информационного поиска, при котором результаты выстраиваются по наибольшему соответствию конкретному запросу. Наиболее распространенные алгоритмы ранжирования гипертекстовых документов — HITS (*Hyperlink induced topic search*) и PageRank — были разработаны в ИВМ Дж. Клейнбергом [3] и в Стенфордском университете С. Брином и Л. Пейджем [4]. Алгоритм HITS обеспечивает выбор из информационного массива лучших первоисточников, на которые ведут множество ссылок ("авторов"), и документов, от которых идут ссылки цитирования ("посредников"), путем вычисления их рангов. Увеличение количества и степени связности документов соответствующего сообщества приводит к увеличению рангов документов, в результате чего выбирается большое число документов по темам, отличным от информационных потребностей пользователя. Алгоритм PageRank близок по идеологии к литературному индексу цитирования и рассчитывается для произвольного документа с учетом числа ссылок из других документов на данный документ. В статье анализ гиперссылок применен для поиска подобных документов.

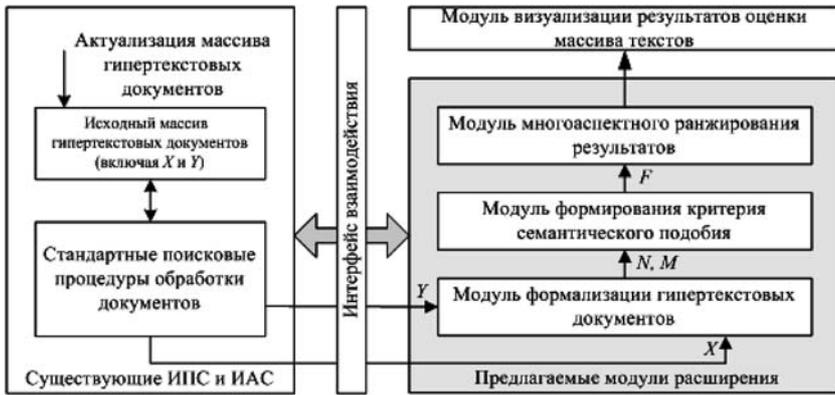


Рис. 1. Обобщенная функциональная схема оценки массива текстов на семантическое сходство с эталоном: ИПС — информационно-поисковая система; ИАС — информационно-аналитическая система

Задача оценки массива гипертекстовых документов на семантическое сходство с эталоном может быть разделена на несколько этапов.

На первом этапе осуществляется выделение гипертекстовых ссылок из загруженных массивов эталонных (X) и произвольных (Y) документов. Для каждой двух публикаций определяются:

- число переходов (N_{ij}) между ними через другие тексты; другими словами, сколько документов необходимо просмотреть, чтобы перейти от публикации X_i к публикации Y_j ;
- число гипертекстовых ссылок (M_{ij}), общих для публикации X_i и публикации Y_j .

$$(N;M) = \begin{array}{c|cccccc|c} & 1 & 2 & 3 & 4 & 5 & \dots & n \\ \hline 1 & 0;6 & 1;4 & -;0 & 3;0 & 2;1 & \dots & 7;0 \\ 2 & 1;4 & 0;12 & 4;2 & 1;1 & 3;3 & \dots & 2;2 \\ 3 & -;0 & 4;2 & 0;7 & 1;2 & 4;1 & \dots & 8;0 \\ 4 & 3;0 & 1;1 & 1;3 & 0;5 & 3;2 & \dots & 1;1 \\ 5 & 2;1 & 3;3 & 4;1 & 3;7 & 0;15 & \dots & 2;1 \\ \hline n & 7;0 & 2;2 & 8;0 & 1;1 & 2;1 & \dots & 0;8 \end{array}$$

Рис. 2. Пример формализации массива текстов

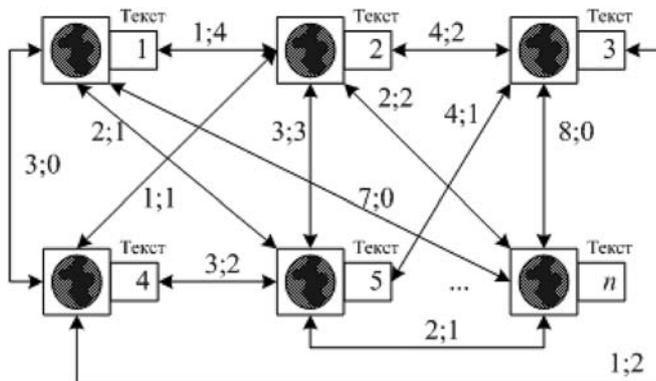


Рис. 3. Графическая иллюстрация гипертекстовых связей в массиве текстов

Обобщенная функциональная схема оценки массива гипертекстовых документов на семантическое сходство с эталоном представлена на рис. 1.

Пример формализации массива текстов представлен на рис. 2, соответствующая ему графическая иллюстрация гипертекстовых связей — на рис. 3.

Экспериментальные исследования гипертекстовой модели на программном макете показали, что мера близости двух произвольных текстов является функцией от N_{ij} и M_{ij} следующего вида:

$$F = f(\min N_{ij}; \max M_{ij}).$$

Рассматриваемую модель предлагается использовать для разбиения исходного массива публикаций по классам U относительно эталонной публикации:

$$U = (u_1, u_2, u_3).$$

Пример разбиения на три класса приведен на рис. 4, где u_1 — сильное подобие: $k < k_H$; u_2 — среднее подобие: $k_H \geq k \leq k_B$; u_3 — слабое подобие: $k > k_B$.

Для решения задачи формирования процедуры классификации применен метод кластерного анализа k -средних [5].

Алгоритм k -средних строит k кластеров, расположенных на возможно больших расстояниях друг от друга. Основной тип задач, которые решает алгоритм k -средних, — наличие предположений (гипотез) относительно числа кластеров, при этом они должны быть различны настолько, насколько это возможно. Выбор числа k может базироваться на результатах предшествующих исследований, теоретических соображениях или интуиции. В случае оценки разбиения массива публикаций по степени близости к эталонной публикации число кластеров $k = 3$ (сильное подобие, среднее подобие, слабое подобие).

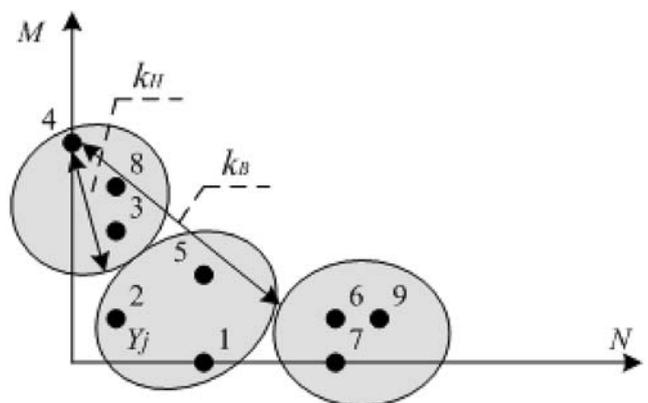


Рис. 4. Пример разбиения публикаций на три класса

Общая идея алгоритма k -средних: заданное фиксированное число k кластеров наблюдения сопоставляются кластерам так, что средние в кластере (для всех переменных) максимально возможно отличаются друг от друга. Правила для классификации формируются в процессе формализации текстов и являются пространством признаков классификации.

По завершении процедуры классификации осуществляется расчет итогового расстояния до центров сформированных кластеров от каждого из анализируемых документов, которое и характеризует степень подобия анализируемых документов к документу-эталону.

Анализ существующих процедур обработки текстовой информации, которые могут быть использованы при решении задачи фильтрации, а также оригинальных авторских разработок, позволил сформировать соответствующий функциональной схеме (см. рис. 1) алгоритм оценки массива гипертекстовых документов на семантическое сходство с эталоном (рис. 5).

Алгоритм включает в себя следующие этапы. В блоке 1 осуществляется загрузка гипертекстовых документов для анализа. Выделение гиперссылок из каждого документа реализуется в блоке 2. В блоке 3 попарно определяется число гипертекстовых переходов между всеми загруженными документами. В блоке 4 реализован попарный подсчет общих гиперссылок для всех загруженных документов. Выбор документа-эталона из массива загруженных страниц реализован в блоке 5. В блоке 6 реализуется кластерный анализ исходного массива загруженных страниц относительно выбранного эталона. В блоке 7 осуществляется визуализация результатов анализа.

Рассмотрим (табл. 1) выборку значений 10 формализованных гипертекстовых документов относительно документа-эталона "4", для которых определены параметры N и M (блок 3, 4) на рис. 5.

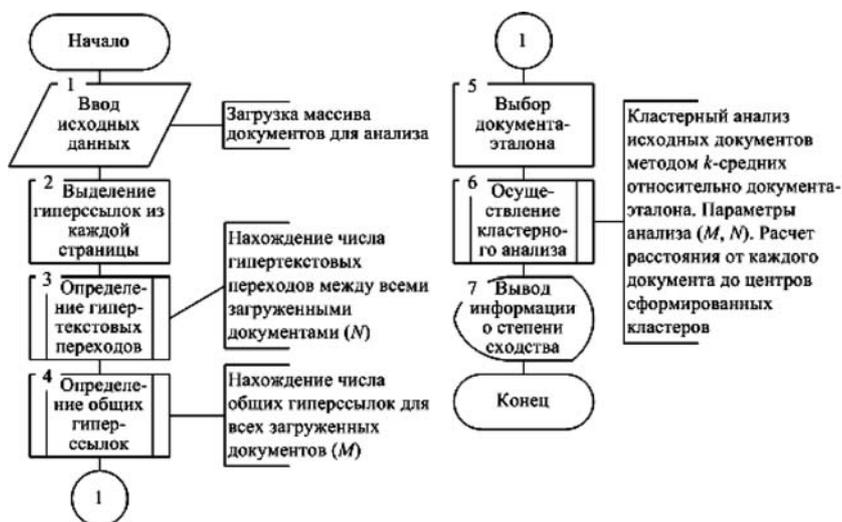


Рис. 5. Алгоритм оценки массива гипертекстовых документов на семантическое сходство с эталоном

Осуществим разбиение исходного множества гипертекстовых документов на три класса (кластера) с помощью метода k -средних (блок 6 на рис. 5).

На нулевом шаге в качестве эталонов центров классов случайным образом выберем три документа, например, "1", "2" и "3". Согласно правилу классификации метода k -средних запишем исходные значения их координат и весов (нулевая итерация):

$$\begin{cases} E_1^0 = 1 = (3; 0), & w_1^0 = 1; \\ E_2^0 = 2 = (1; 1), & w_2^0 = 1; \\ E_3^0 = 3 = (1; 3), & w_3^0 = 1. \end{cases}$$

На первом шаге берем четвертый документ и определяем расстояние от него до каждого из эталонов по евклидовой метрике:

$$d_{41} = \sqrt{(0-3)^2 + (5-0)^2} = 5,831;$$

$$d_{42} = \sqrt{(0-1)^2 + (5-1)^2} = 4,123;$$

$$d_{43} = \sqrt{(0-1)^2 + (5-3)^2} = 2,236.$$

Результаты расчетов показывают, что четвертый объект должен быть включен в третий кластер, а следовательно, третий эталон необходимо пересчитать, а первый и второй оставить без изменения. Координаты нового центра первого кластера будут определяться формулой

$$E_3^1 = \frac{w_3^0 E_3^0 + "4"}{w_3^0 + 1} = \left(\frac{1 \cdot 1 + 0}{1 + 1}; \frac{1 \cdot 3 + 5}{1 + 1} \right) = (0,5; 4).$$

Таблица 1
Описание страниц в системе признаков (N ; M)

Номер документа	Значения признаков	
	N	M
1	3	0
2	1	1
3	1	3
4	0	5
5	3	2
6	6	1
7	6	0
8	1	4
9	7	1
n	1	1

Вес третьего кластера после включения нового объекта будет равен $w_3^1 = w_3^0 + 1 = 1 + 1 = 2$. Веса и координаты первого и второго кластера останутся прежними:

$$E_1^1 = E_1^0 = (3; 0), w_1^1 = w_1^0 = 1,$$

$$E_2^1 = E_2^0 = (1; 1), w_2^1 = w_2^0 = 1.$$

На втором шаге проверяем, к какому эталону ближе всего находится пятый документ:

$$d_{51} = \sqrt{(3-3)^2 + (2-0)^2} = 2,000;$$

$$d_{52} = \sqrt{(3-1)^2 + (2-1)^2} = 2,236;$$

$$d_{53} = \sqrt{(3-0,5)^2 + (2-4)^2} = 3,202.$$

Так как $d_{51} = \min\{d_{51}, d_{52}, d_{53}\}$, то пятый объект должен быть приписан первому кластеру. Координаты и вес первого кластера примут значения

$$E_1^2 = \frac{w_1^1 E_1^1 + "5"}{w_1^1 + 1} = \left(\frac{1 \cdot 3 + 3}{1+1}; \frac{1 \cdot 0 + 2}{1+1} \right) = (3; 1);$$

$$w_1^2 = w_1^1 + 1 = 1 + 1 = 2.$$

Координаты и веса второго и третьего кластера после второго шага не изменяются:

$$E_2^2 = E_2^1 = (1; 1), w_2^2 = w_2^1 = 1,$$

$$E_3^2 = E_3^1 = (0,5; 4), w_3^2 = w_3^1 = 2.$$

На последующих шагах все процедуры повторяются для оставшихся документов: "6", "7", "8", "9", "n".

После того как просмотрены все документы, кроме первых трех, процесс "зацикливается", т. е. по тому же правилу осуществляются просмотр и присоединение к соответствующему эталону каждого из 10 документов. При этом при включении документов происходит перерасчет координат эталонов и весов соответствующих кластеров. Результаты расчета представлены в табл. 2.

Как следует из данных табл. 2, после тридцать седьмого шага выполнилось условие устойчивости разбиения, так как результат итерации 28—37 совпал с результатами предыдущих итераций 18—27.

После получения устойчивого разбиения определяются центры тяжести образовавшихся кластеров:

$$S_3 = \{"6", "7", "9"\},$$

$$S_2 = \{"1", "2", "5", "n"\},$$

$$S_1 = \{"3", "4", "8"\},$$

координаты которых в общем случае могут не совпадать с координатами эталонов. Так, в нашем примере имеем:

$C_1 = (6,333; 0,667)$ — координаты центра тяжести первого кластера;

$C_2 = (2; 1)$ — координаты центра тяжести второго кластера;

$C_3 = (0,667; 4)$ — координаты центра тяжести третьего кластера.

После этого строится окончательное разбиение, для чего каждая многомерная точка (гипертекстовый документ) включается в тот кластер, центр тяжести которого к ней ближе всего. Для рассматриваемого примера расстояния от каждой из точек до центров каждого из трех кластеров приведены в табл. 3.

Таблица 2

Результаты классификации страниц

Номер итерации	Координаты эталонов и веса кластеров		
	1	2	3
0	$E_1^0 = (3; 0), w_1^0 = 1$	$E_2^0 = (1; 1), w_2^0 = 1$	$E_3^0 = (1; 3), w_3^0 = 1$
1	$E_1^1 = (3; 0), w_1^1 = 1$	$E_2^1 = (1; 1), w_2^1 = 1$	$E_3^1 = (0,5; 4), w_3^1 = 2$
2	$E_1^2 = (3; 1), w_1^2 = 2$	$E_2^2 = (1; 1), w_2^2 = 1$	$E_3^2 = (0,5; 4), w_3^2 = 2$
3	$E_1^3 = (4; 1), w_1^3 = 3$	$E_2^3 = (1; 1), w_2^3 = 1$	$E_3^3 = (0,5; 4), w_3^3 = 2$
4	$E_1^4 = (4,5; 0,75), w_1^4 = 4$	$E_2^4 = (1; 1), w_2^4 = 1$	$E_3^4 = (0,5; 4), w_3^4 = 2$
5	$E_1^5 = (4,5; 0,75), w_1^5 = 4$	$E_2^5 = (1; 1), w_2^5 = 1$	$E_3^5 = (0,667; 4), w_3^5 = 3$
6	$E_1^6 = (5; 0,8), w_1^6 = 5$	$E_2^6 = (1; 1), w_2^6 = 1$	$E_3^6 = (0,667; 4), w_3^6 = 3$
7	$E_1^7 = (5; 0,8), w_1^7 = 5$	$E_2^7 = (1; 1), w_2^7 = 2$	$E_3^7 = (0,667; 4), w_3^7 = 3$
Состав кластера	"1", "5", "6", "7", "9"	"2", "n"	"3", "4", "8"

Номер итерации	Координаты эталонов и веса кластеров		
	1	2	3
8	$E_1^8 = (4,667; 0,667), w_1^8 = 6$	$E_2^8 = (1; 1), w_2^8 = 2$	$E_3^8 = (0,667; 4), w_3^8 = 3$
9	$E_1^9 = (4,667; 0,667), w_1^9 = 6$	$E_2^9 = (1; 1), w_2^9 = 3$	$E_3^9 = (0,667; 4), w_3^9 = 3$
10	$E_1^{10} = (4,667; 0,667), w_1^{10} = 6$	$E_2^{10} = (1; 1), w_2^{10} = 3$	$E_3^{10} = (0,75; 3,75), w_3^{10} = 4$
11	$E_1^{11} = (4,667; 0,667), w_1^{11} = 6$	$E_2^{11} = (1; 1), w_2^{11} = 3$	$E_3^{11} = (0,563; 4,063), w_3^{11} = 5$
12	$E_1^{12} = (4,389; 0,889), w_1^{12} = 7$	$E_2^{12} = (1; 1), w_2^{12} = 3$	$E_3^{12} = (0,563; 4,063), w_3^{12} = 5$
13	$E_1^{13} = (4,657; 0,907), w_1^{13} = 8$	$E_2^{13} = (1; 1), w_2^{13} = 3$	$E_3^{13} = (0,563; 4,063), w_3^{13} = 5$
14	$E_1^{14} = (4,881; 0,756), w_1^{14} = 9$	$E_2^{14} = (1; 1), w_2^{14} = 3$	$E_3^{14} = (0,563; 4,063), w_3^{14} = 5$
15	$E_1^{15} = (4,881; 0,756), w_1^{15} = 9$	$E_2^{15} = (1; 1), w_2^{15} = 3$	$E_3^{15} = (0,672; 4,047), w_3^{15} = 6$
16	$E_1^{16} = (5,234; 0,797), w_1^{16} = 10$	$E_2^{16} = (1; 1), w_2^{16} = 3$	$E_3^{16} = (0,672; 4,047), w_3^{16} = 6$
17	$E_1^{17} = (5,234; 0,797), w_1^{17} = 10$	$E_2^{17} = (1; 1), w_2^{17} = 3$	$E_3^{17} = (0,672; 4,047), w_3^{17} = 6$
Состав кластера	"1", "5", "6", "7", "9"	"2", "n"	"3", "4", "8"
18	$E_1^{18} = (5,234; 0,797), w_1^{18} = 10$	$E_2^{18} = (1,667; 0,667), w_2^{18} = 5$	$E_3^{18} = (0,672; 4,047), w_3^{18} = 6$
19	$E_1^{19} = (5,234; 0,797), w_1^{19} = 10$	$E_2^{19} = (1,5; 0,75), w_2^{19} = 6$	$E_3^{19} = (0,672; 4,047), w_3^{19} = 6$
20	$E_1^{20} = (5,234; 0,797), w_1^{20} = 10$	$E_2^{20} = (1,5; 0,75), w_2^{20} = 6$	$E_3^{20} = (0,754; 3,785), w_3^{20} = 7$
21	$E_1^{21} = (5,234; 0,797), w_1^{21} = 10$	$E_2^{21} = (1,5; 0,75), w_2^{21} = 6$	$E_3^{21} = (0,565; 4,089), w_3^{21} = 8$
22	$E_1^{22} = (5,234; 0,797), w_1^{22} = 10$	$E_2^{22} = (1,875; 1,063), w_2^{22} = 7$	$E_3^{22} = (0,565; 4,089), w_3^{22} = 8$
23	$E_1^{23} = (5,426; 0,848), w_1^{23} = 11$	$E_2^{23} = (1,875; 1,063), w_2^{23} = 7$	$E_3^{23} = (0,565; 4,089), w_3^{23} = 8$
24	$E_1^{24} = (5,569; 0,636), w_1^{24} = 12$	$E_2^{24} = (1,875; 1,063), w_2^{24} = 7$	$E_3^{24} = (0,565; 4,089), w_3^{24} = 8$
25	$E_1^{25} = (5,569; 0,636), w_1^{25} = 12$	$E_2^{25} = (1,875; 1,063), w_2^{25} = 7$	$E_3^{25} = (0,674; 4,067), w_3^{25} = 9$
26	$E_1^{26} = (5,927; 0,727), w_1^{26} = 13$	$E_2^{26} = (1,875; 1,063), w_2^{26} = 7$	$E_3^{26} = (0,674; 4,067), w_3^{26} = 9$
27	$E_1^{27} = (5,927; 0,727), w_1^{27} = 13$	$E_2^{27} = (1,7; 1,05), w_2^{27} = 8$	$E_3^{27} = (0,674; 4,067), w_3^{27} = 9$
Состав кластера	"6", "7", "9"	"1", "2", "5", "n"	"3", "4", "8"
28	$E_1^{28} = (5,927; 0,727), w_1^{28} = 13$	$E_2^{28} = (1,96; 0,84), w_2^{28} = 9$	$E_3^{28} = (0,674; 4,067), w_3^{28} = 9$
29	$E_1^{29} = (5,927; 0,727), w_1^{29} = 13$	$E_2^{29} = (1,768; 0,872), w_2^{29} = 10$	$E_3^{29} = (0,674; 4,067), w_3^{29} = 9$
30	$E_1^{30} = (5,927; 0,727), w_1^{30} = 13$	$E_2^{30} = (1,768; 0,872), w_2^{30} = 10$	$E_3^{30} = (0,756; 3,800), w_3^{30} = 10$
31	$E_1^{31} = (5,927; 0,727), w_1^{31} = 13$	$E_2^{31} = (1,768; 0,872), w_2^{31} = 10$	$E_3^{31} = (0,567; 4,100), w_3^{31} = 11$
32	$E_1^{32} = (5,927; 0,727), w_1^{32} = 13$	$E_2^{32} = (2,0144; 1,098), w_2^{32} = 11$	$E_3^{32} = (0,567; 4,100), w_3^{32} = 11$
33	$E_1^{33} = (5,945; 0,795), w_1^{33} = 14$	$E_2^{33} = (2,0144; 1,098), w_2^{33} = 11$	$E_3^{33} = (0,567; 4,100), w_3^{33} = 11$
34	$E_1^{34} = (5,959; 0,596), w_1^{34} = 15$	$E_2^{34} = (2,0144; 1,098), w_2^{34} = 11$	$E_3^{34} = (0,567; 4,100), w_3^{34} = 11$
35	$E_1^{35} = (5,959; 0,596), w_1^{35} = 15$	$E_2^{35} = (2,0144; 1,098), w_2^{35} = 11$	$E_3^{35} = (0,675; 4,075), w_3^{35} = 12$
36	$E_1^{36} = (6,219; 0,697), w_1^{36} = 16$	$E_2^{36} = (2,0144; 1,098), w_2^{36} = 11$	$E_3^{36} = (0,675; 4,075), w_3^{36} = 12$
37	$E_1^{37} = (6,219; 0,697), w_1^{37} = 16$	$E_2^{37} = (1,812; 1,078), w_2^{37} = 12$	$E_3^{37} = (0,675; 4,075), w_3^{37} = 12$
Состав кластера	"6", "7", "9" слабое подобие	"1", "2", "5", "n" среднее подобие	"3", "4", "8" сильное подобие

Расстояние до центров кластеров

Номер страницы	Центры кластеров		
	C_1	C_2	C_3
1	3,399	1,414	4,631
2	5,343	1	3,018
3	5,821	2,236	1,054
4	7,673	4,472	1,202
5	3,590	1,414	3,073
6	0,471	4	6,119
7	0,746	4,123	6,666
8	6,289	3,162	0,333
9	0,746	5	7,008
<i>n</i>	5,343	1	3,018

Так как данные табл. 3 подтверждают полученное разбиение, то оно и принимается за решение задачи классификации. В противном случае процесс распределения объектов по кластерам продолжается до стабилизации состава кластеров.

Полученное разбиение соответствует графическому представлению результатов классификации, представленному на рис. 4. Таким образом, документу-эталону "4" наиболее подобны тексты "3" и "8", наименее — "6", "7" и "9". В средний класс попали документы "1", "2", "5", "и". Степень подобия анализируемых документов к выбранному документу-эталону характеризует расстояние от анализируемых документов до центров сформированных кластеров.

Экранные формы пользовательского интерфейса программного модуля оценки массива гипертекстовых документов на семантическое сходство с эталоном представлены на рис. 6, 7.

Таким образом, в статье предложен алгоритм оценки массива текстов на семантическое сходство с эталоном, получивший практическую проверку на программном макете и позволяющий повысить качество информационного поиска.

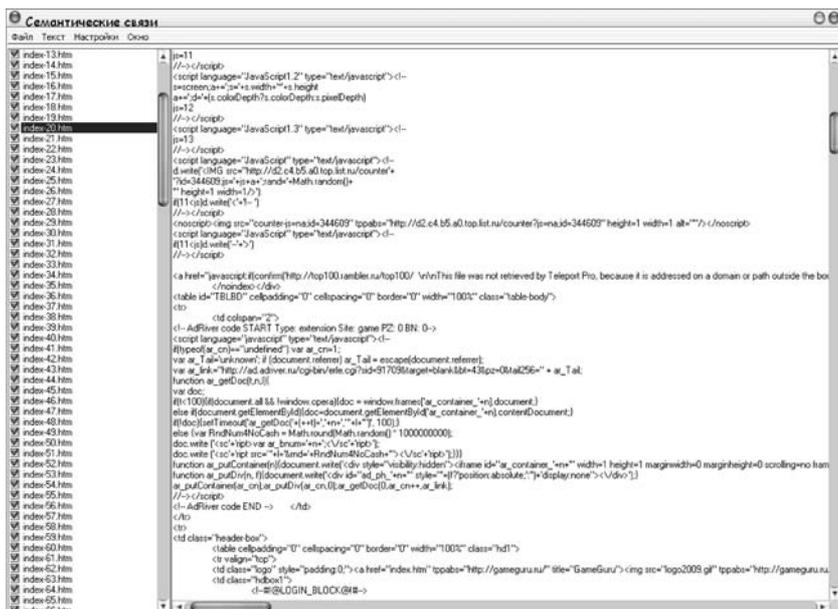


Рис. 6. Экранная форма выделения гиперссылок из каждой страницы

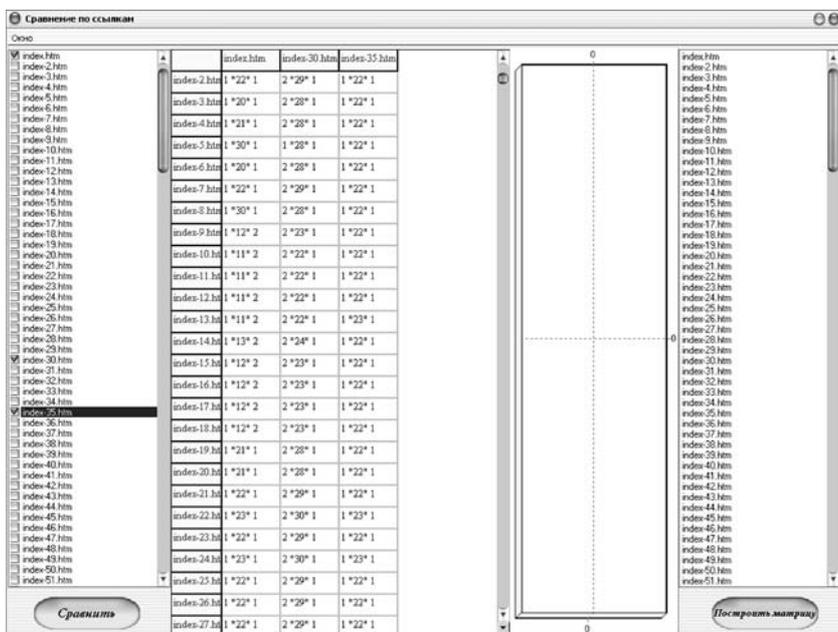


Рис. 7. Экранная форма результата сравнения массива текстовых документов с эталоном

Список литературы

1. Чубукова И. А. Data Mining. Основы информационных технологий: учеб. пособие. М.: БИНОМ. Лаборатория знаний. Интернет-университет информационных технологий, 2006. 382 с.
2. Бородащенко А. Ю., Бочков М. В., Салбиев А. Л. Алгоритм оценки массива текстов на семантическое сходство с эталоном // Информационные технологии. 2008. № 12. С. 8—11.
3. Kleinberg J. M. Authoritative sources in a hyperlink environment // In Processing of ACM-SIAM Symposium on Discrete Algorithms, 1998. N 46(5). P. 604—632.
4. Brit S., Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. WWW7. 1998. P. 107—117.
5. Ландэ Д. В. Основы интеграции информационных потоков. К.: Инжиниринг, 2006. 240 с.

УДК 004.021

В. В. Топорков, д-р техн. наук, проф.,
e-mail: ToporkovVV@mpei.ru,

А. С. Целищев, мл. науч. сотр.,
e-mail: AlexeyTselishev@cern.ch,

Московский энергетический институт
(технический университет)

Метод критических работ как перспектива эффективной организации распределенных вычислений

Описывается оригинальный метод критических работ, позволяющий осуществлять планирование распределенных вычислений на уровне приложений с учетом информационных зависимостей. Приведены основные алгоритмы и эвристики, рассмотрена модификация метода, учитывающая динамику загрузки вычислительных ресурсов, а также некоторые результаты работы программной модели с заданиями, представленными в виде ориентированных параметризованных бесконтурных графов.

Ключевые слова: распределенные вычисления, метапланирование, планировщик, стратегии планирования, расписание, коллизии

Введение

В современном мире распределенные вычисления играют важную роль в решении трудоемких вычислительных задач, а также занимают особое место среди направлений исследований в области информационных технологий. Существование большого многообразия подходов, предлагаемых моделей организации и проведения вычислений, а также практическая реализация многочисленных методов планирования распределенных вычислений не позволяют однозначно ответить на вопрос о выборе наиболее эффективного способа решения задачи оптимизации всего процесса или отдельных его компонентов. На сегодняшний день большинство систем распределенных вычислений используют примитивные методы оптимизации распределения потоков задач либо и вовсе не учитывают интересы как пользователей вычислительных ресурсов, так и пользователей системы, в то время как эффективное распределение задач

и ресурсов могло бы позволить увеличить экономическую эффективность использования вычислительных узлов и всей системы в целом.

Существующие планировщики распределенных вычислений чаще всего лишь частично поддерживают планирование вычислений для трудоемких заданий с существенными зависимостями по данным между составляющими их подзадачами. Так, например, планировщик Maui [1] поддерживает описание составного задания лишь в виде *очереди* однородных подзадач с возможным назначением их приоритетов. Очевидно, что при представлении рассматриваемых задач в виде ориентированных бесконтурных графов [2] успешному распределению поддадутся лишь те задачи, которые можно представить в виде деревьев (рис. 1, а, см. третью сторону обложки). Несмотря на наличие в планировщике Maui механизмов backfilling-а, которые в итоге обеспечат построение корректного расписания, при наличии двух разных путей в графе к одной и той же вершине неизбежна конкуренция за ресурсы или же их простой. В ситуации, когда простой ресурсов или использование быстродействующих вычислительных узлов равносильны потере времени или большим финансовым затратам (например, при условии взимания платы за использование ресурсов), подобные корректно составленные расписания не будут оптимальными.

На рис. 1, б изображен граф, структура и параметры которого не только не позволяют автоматически составить оптимальное расписание с помощью имеющихся программных средств, но и потребуют от пользователя разбиения графа на подграфы для осуществления загрузки задания в систему. Разбиение графа на подграфы-деревья наложит дополнительные условия на моменты начала и окончания выполнения соответствующих им подзадач, а без информации о быстродействии имеющихся ресурсов невозможно будет правильно определить моменты старта и завершения обработки частей задания. Подобные расчеты, очевидно, не должны касаться пользователя, не имеющего представления о технических характеристиках вычислительных узлов и, более того, практически невозможны при более сложной структуре задания, так как поиск деревьев и определение информационных зависимостей становятся труднорешаемой и нетривиальной задачей, не относящейся к решению основного вопроса — собственно получению результата вычислений.

Комбинированное планирование и метод критических работ

Для разрешения описанных проблем предлагается оригинальный подход, в основе которого закладывается иерархическая структура, состоящая из метапланировщика потоков заданий, подконтрольных ему менеджеров заданий, которые, в свою очередь, взаимодействуют с локальными менеджерами управления ресурсами, например, системами пакетной обработки заданий [3–5]. Преимущества иерархически организованных структур управления заданиями хорошо известны. Таким образом строится комплекс GrADS, реализующий централизованную диспетчеризацию заданий на основе прогноза загрузки локальных узлов и метода опережающего планирования. Иерархия промежуточных серверов в системе X-Com позволяет снизить простой узлов из-за заметных задержек в передаче данных и занятости управляющего сервера обслуживанием других узлов. Древовидная структура менеджеров в сетевой среде распределенных вычислений на базе ресурсов МСЦ РАН позволяет избежать тупиков при доступе к разделяемым ресурсам. Еще один немаловажный аспект при организации вычислений в неоднородных средах — объединение под управлением одного менеджера тех вычислительных узлов, которые являются схожими по архитектуре, составу и политике администрирования.

Особенностью предлагаемого подхода является *динамическое* перераспределение потоков заданий в среде на основе стратегий. Под стратегией выполнения составного задания будет пониматься список возможных назначений на ресурсы и планов для каждой из *задач*, входящих в *задание*. Для порождения стратегий предлагается использовать метод критических работ, два основных этапа которого предложены в работе [6]: планирование и распределение последовательно выделяемых критических работ графа, представляющего задание, и последующее разрешение возникающих коллизий при назначении. При практической реализации метода в целях моделирования процессов планирования вычислений были решены важные, неучтенные ранее вопросы, касающиеся конкретных алгоритмов, а также промежуточных этапов вычислений. Сам метод критических работ был развит для планирования составных заданий в целях порождения многофакторных и многокритериальных стратегий распределения потоков заданий. Остановимся подробнее на описании разработанных алгоритмов, а также реализованной имитационной модели.

Каждое из заданий в модели представляется соответствующим параметризованным графом. Степень информационной связности вершин-задач может варьироваться. Число базовых процессор-

ных узлов принимается кратным максимальной ширине яруса графа. Оценки длительности выполнения задач, объемов вычислений, времени доступа к данным — целые случайные числа, равномерно распределенные в заданном диапазоне. Конфликты между задачами, конкурирующими за один и тот же узел, разрешаются за счет незадействованных базовых узлов так, что обеспечивается минимум функции штрафа, например, суммы отношений объемов вычислений к оценкам длительности выполнения задач.

Планирование состоит в итеративном применении двухэтапной процедуры: планирование и назначение последовательно выделяемых критических работ — путей наибольшей длительности, разрешение возникающих коллизий путем изменения назначения соответствующей операции.

Первый этап предполагает поиск и ранжирование критических работ программной модели, которые сводятся к поиску критических путей во взвешенном ориентированном графе, отражающем модель. Тривиальным выглядит способ поиска критических путей в графе — так называемый поиск в глубину, который, однако, требует большого количества вычислений на графах большой размерности и требует введения ограничений для предотвращения зацикливаний. Для программной реализации составления списка критических путей в графе предложен оригинальный гибридный алгоритм, объединяющий в себе элементы алгоритма Форда и алгоритма Йена [7], состоящий из двух этапов (подробные выкладки представлены в работе [8]):

1. Рекуррентное построение корневого дерева критических работ графа с вершинами-работами.
2. Формирование массива критических работ путем перебора вершин дерева.

После получения массива критических путей можно приступать к планированию и назначению вычислительных ресурсов. Планировщик должен принимать на вход ориентированный граф, предварительно разбитый на критические работы, а на выходе формировать вектор априорных оценок времени для каждой из задач-вершин графа на основе многокритериальной оптимизации.

Требования ко входным данным заключаются в следующем: критические работы должны быть ранжированы, т. е. упорядочены по априорному времени исполнения. За оценку времени исполнения работы принимается сумма времен исполнения каждой из задач работы на самых быстрых вычислительных узлах плюс сумма времен информационных обменов данными между задачами.

Критические работы задаются задачами, входящими в них, и информационными связями, выражающими отношение предшествования. Кроме этого, ко входным данным относятся значения производительности процессоров для каждой из задач.

Алгоритм работы планировщика включает в себя цикл, проходящий по всем критическим работам (подробнее в [8]). В результате одной итерации получается один или несколько вариантов распределения, которые являются подвекторами искомого вектора, являющегося результатом решения всей задачи планирования.

На каждой итерации в подвектор включаются значения априорных оценок времени для соответствующих задач из текущей критической работы. Если в начале итерации имеется несколько вариантов, то рассматривается каждый из них. В течение этого процесса в список могут добавляться новые варианты распределения, однако они будут рассматриваться уже на следующей итерации для следующей критической работы. В результате работы алгоритма формируется множество векторов априорных оценок времени (вариантов распределения) для всех задач. Эти данные используются в дальнейшем для назначения вычислительного ресурса.

Второй этап процедуры масштабирования состоит в разрешении коллизий, возникающих при последовательном назначении вычислительных ресурсов для задач каждой критической работы. Представление коллизий двудольным графом (X, Y, A) , множество ребер A которого отражает попарную конкуренцию за ресурс, а также методы их оптимального и условно оптимального разрешения предложены в работах [8 и 9]. В процессе реализации методов были предложены эвристики, позволяющие в некоторых случаях получать более близкие к оптимальному варианты решений.

Модифицированный метод критических работ и среда моделирования

Описанные выше идеи положены в основу программной реализации среды имитационного моделирования MIMAPR, которая позволила доработать предлагаемые алгоритмы, дополнив их вспомогательными механизмами и эвристиками ([8]), а также собрать и проанализировать статистику исходов для разных стратегий планирования ([10]). На основе полученных результатов предлагаемая изначальная концепция была расширена до масштабов планирования на метауровне.

Проблема интеграции более сложных алгоритмов локального планирования заключалась в специфике лежащего в основе подхода метода критических работ, который рассматривает упрощенную ситуацию наличия неограниченного числа ресурсов заданного типа, не беря в расчет вероятную занятость конкретного экземпляра другими задачами. Как выяснилось на практике, в реальной ситуации планировщик должен учитывать информацию об ограниченности ресурсов, а выделяемый набор вычислительных узлов не всегда будет доступен в течение всего времени, отведенного на

выполнение задания. Исходя из этого было принято решение модифицировать среду моделирования и дополнить методы планирования дополнительными функциями, которые позволяют динамически выделять ресурсы и проводить планирование с использованием уже имеющихся наработок.

Для решения этой задачи для начала необходимо хранить и динамически обновлять информацию о вычислительных узлах и интервалах их занятости и учитывать эти данные при любом назначении и переназначении задач в рамках планирования и разрешения коллизий. Только лишь внеся модификации в реализованные алгоритмы, можно сохранить близкие к оптимальным расписания, которые учитывают структуру заданий и дополнительные критерии. Это обусловлено тем, что в системе с частично занятым набором ресурсов построенное оригинальным, немодифицированным методом критических работ расписание неизбежно подвергнется временным сдвигам и уже не будет соответствовать полученным ранее модулем планировщика значениям критериальных функций. Для визуальной оценки допустимости построенных планов в среду имитационного моделирования был введен модуль отображения временных диаграмм, отображающих занятость каждого конечного узла с течением времени. Примеры работы модифицированного метода критических работ и подробное описание внесенных изменений представлено в работе [11].

Программная модель позволяет выбирать описанные параметры случайных сценариев, критерии составления расписаний, функции штрафа, а также частоту подачи заданий. В качестве результатов моделирования отображаются временные диаграммы загрузки ресурсов и распределения критических работ сценария, а также подробная информация о встреченных коллизиях и ходе их разрешения. Разработанная система моделирования позволила оценить качество обслуживания конечных пользователей и создать механизмы априорной оценки вероятности успеха выполнения задания на заданном наборе ресурсов за заданное время. Новые модули мониторинга позволили проанализировать статистику исходов работы планировщика, загрузки ресурсов, а также сформировать эмпирические требования к ресурсам со стороны поступающего задания. Далее эти результаты рассмотрены подробнее.

Временной диапазон, в рамках которого можно составить расписание для сложноструктурированной задачи, можно определить как $T = [0; [T_{k1min}; \Sigma t_j]]$, где T_{k1min} — минимальное время выполнения самой длинной (по сумме весов вершин и ребер графа) критической работы на определенном наборе ресурсов, а Σt_j — сумма времен выполнения всех подзадач. Иными словами, при отсутствии параллелизма время выполнения всего сценария равно сумме времен выполнения каждой из подзадач

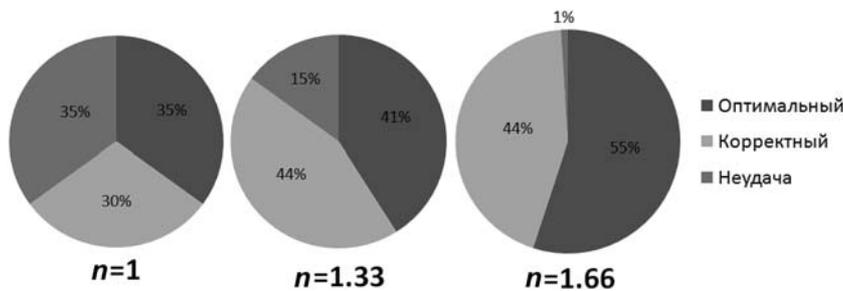


Рис. 2. Статистика планов для разных интервалов

(верхняя граница), а при максимально возможном параллелизме время выполнения сценария ограничено длиной максимальной критической работы (нижняя граница), т. е. самым длинным последовательным участком графа.

Дополнительным подтверждением процента успешно спланированных сценариев из работы [12] послужила новая статистика, отражающая тот факт, что при назначении диапазона планирования одного сценария, равному сумме весов вершин (соответствующих временам выполнения подзадач на определенных ресурсах) и ребер самого длинного последовательного участка графа, успешно и оптимально распределяется около 35 % сценариев, 30 % сценариев не имеют оптимального плана (но для них существует план, близкий к оптимальному), а для остальных 35 % корректных планов не существует. Проблема выбора интервала планирования вызвана тем, что для случайно генерируемых сценариев сложно дать оценку необходимого времени планирования ввиду абстрактности самого сценария (в реальной ситуации это является задачей пользователя). Таким образом, предлагаемый подход позволит сразу отклонить запросы пользователей с заведомо некорректными требованиями к интервалу выполнения сценария (предлагаемый короче $[0; T_{k1min}]$), а также дать вероятностную оценку успеха распределения еще до его начала. На рис. 2 представлена статистическая оценка исхода планирования для интервала

$$T = [0; [n * T_{k1min}]] \text{ для } n = 1, 1,33, 1,66$$

(10 000 прогонов для каждого n).

Таким образом, эксперимент показал, что при длине интервала в 1,66 раз больше, чем длина максимальной критической работы графа, вероятность неудачи при планировании менее 0,01. При этом в 55 % случаев планировщик выдает план, близкий к оптимальному, обеспечивающий минимальный простой ресурсов и минимальную стоимость их использования (возможно использование и других критериев). Эксперименты также показали, что при успешном составлении расписания подзадачи каждого сценария конкурируют за один и тот же ресурс в среднем 2,25 раза (1800 коллизий на 800 сцена-

риев). Более подробные результаты испытаний представлены в работе [13].

Очевидно, что не для каждого сценария возможно составить близкое к оптимальному расписание и обеспечить его выполнение, однако предлагаемые подходы позволяют вплотную подойти к решению этой задачи, а также дать априорные оценки вероятности успешного распределения. На текущий момент примерно в половине рассмотренных случаев (47 %) для случайно сгенерированных графов планировщик предлагает близкий к оптимальному план с разрешенными возможными коллизиями. В настоящий момент идет подготовка к интеграции модели в параллельный проект моделирования планирования на уровне потоков независимых заданий.

Заключение

Полученные результаты позволяют считать, что описанная разработка вплотную подходит к решению задачи создания планировщика, который позволит составлять близкие к оптимальным планы для связанных по данным сложно структурированных сценариев распределенной обработки данных с учетом динамического состава доступных вычислительных ресурсов.

На сегодняшний день, когда важность эффективного использования распределенных ресурсов необычайно высока, очевидно, что успешное завершение разработки позволит создать мощный инструмент, позволяющий избежать значительных экономических потерь — как временных, так и финансовых — в самых различных областях деятельности.

На различных этапах проведения исследований работа была поддержана грантами Российского фонда фундаментальных исследований (РФФИ № 06-01-00027, № 09-01-00095), а также Советом по грантам Президента Российской Федерации для поддержки ведущих научных школ (грант НШ-7239.2010.9) и Министерством образования и науки (проект 2.1.2/6718 и государственные контракты № П2227, 16.740.11.0038).

Список литературы

1. <http://www.clusterresources.com>
2. **Kretsis A., Kokkinos P., Varvarigos E. A.** Developing scheduling policies in gLite middleware, CCGRID'09 IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009.
3. **Buyya R., Abramson D., Giddy J.** Economic Models for Resource Management and Scheduling in Grid Computing // Concurrency and Computation: Practice and Experience. 2002. Vol. 14. N 5. P. 1507—1542.
4. **Ernemann C., Hamscher V., Yahyapour R.** Economic Scheduling in Grid Computing // Proceedings of the 8th Job Scheduling Strategies for Parallel Processing, Springer, Heidelberg, LNCS. 2002. Vol. 2537. P. 128—152.
5. **Garg S. K., Buyya R., Siegel H. J.** Scheduling Parallel Applications on Utility Grids: Time and Cost Trade-off Management //

Pr. of the 32nd Australasian Computer Science Conference, Wellington. 2009. P. 151–159.

6. **Топорков В. В.** Модели распределенных вычислений. М.: ФИЗМАТЛИТ, 2004.

7. **Lestor R., Ford J. R., Fulkerson D. R.** Flows in Networks // Princeton University Press. 1962.

8. **Топорков В. В., Целищев А. С., Бобченков А. В., Рычкова П. В.** Метапланировщик — генератор стратегий распределенных вычислений // Тр. XV Междунар. научно-техн. конф. "Информационные средства и технологии" Т. 1. М.: МЭИ, 2007. С. 28–35.

9. **Топорков V. V., Yemelyanov D., Toporkova A., Tselishchev A.** Scalable co-scheduling strategies in distributed computing // Pr. IEEE/ACS AICCSA International Conference/ Hammamet, 2010. P. 1–8.

10. **Топорков В. В., Топоркова А. С., Целищев А. С.** Стратегии коаллокаций для решения больших задач в распределенных

средах // Тр. Всероссийской науч. конф. (22–27.09.2008, Новосибирск). М.: Изд-во МГУ, 2008. С. 3–6.

11. **Целищев А. С., Топорков В. В.** Модификация метода критических работ с учетом динамически изменяемого состава ресурсов в распределенных вычислениях // Тр. Всероссийской суперкомпьютерной конференции (21–26 сентября 2009 г., Новосибирск). М.: Изд-во МГУ, 2009. С. 340–345.

12. **Топорков V. V., Tselishchev A.** Safety Strategies of Scheduling and Resource Co-allocation in Distributed Computing // Proc. of the 3rd Int. conf. on Dependability of Computer Systems DepCoS-RELCOMEX 2008 IEEE Society Publishing. Wroclaw, 2008. P. 152–159.

13. **Топорков В. В., Целищев А. С.** Оценка эффективности модифицированного метода критических работ в планировании распределенных вычислений и формировании стратегий коаллокаций для решения больших задач // Методы и средства обработки информации. Тр. 3-й Всероссийской науч. конф. / Под ред. Л. Н. Королева. М.: Изд. фак. ВМК МГУ им. Ломоносова. 2009. С. 299–304.

УДК 004.75.021

Б. Амиршахи, аспирант,
Московский государственный университет
путей сообщения (МИИТ),
e-mail: bita.amirshahi@gmail.com

GRID-технологии решения больших систем линейных уравнений на вычислительной сети и на суперкомпьютере кластерного типа

Исследуется оптимизированное распределенное решение "больших" систем линейных уравнений методом Крамера на выделенных GRID-системой сетевых ресурсах, а также на суперкомпьютере МИИТ Т-4700 (СКИФ). Выбор задачи обусловлен типовыми требованиями к организации вычислительного процесса и, в частности, к организации внутрисистемного обмена данными.

Ключевые слова: GRID-вычисления, вычислительная сеть, кластерный суперкомпьютер, метод Крамера

Введение

Развивающаяся технология GRID-вычислений предполагает использование вычислительных сетей — как локальных, так и глобальных, а также современных суперкомпьютеров кластерного типа. Для успешного применения этой технологии необходимо разработать пакет прикладных программ, являющихся основным средством удовлетворения запросов пользователей за минимальное время.

Основная трудность, возникающая при программировании задач пакета, заключается в необходимости интенсивного обмена данными между компьютерами вычислительной сети. Сетевая техноло-

гия лежит в основе комплексирования и кластерного суперкомпьютера. Вычислительный процесс должен быть оптимизирован так, чтобы максимально снизить негативное влияние неизбежного внутрисистемного обмена на производительность вычислительных средств.

Ниже обсуждается проблема достижения максимальной производительности вычислительных средств (минимального времени счета) при решении систем линейных уравнений в общем случае применения сетевых технологий, а также при выполнении этой задачи на суперкомпьютере МИИТ Т-4700 (СКИФ). Выбор задачи — метод Крамера — следует воспринимать, как конструирование теста, концентрирующего все основные требования к организации распределенных вычислений и обмена данными.

1. Проблемы распределенных вычислений

До сих пор основная концепция теории сложности алгоритмов базируется на сравнительном расчёте числа выполняемых операций. Таким образом, критерием сложности каждого разрабатываемого алгоритма традиционно считается число операций. Такой критерий не может быть чисто механически перенесен на анализ параллельных алгоритмов. При распараллеливании первостепенное значение приобретает "длина критического пути" в информационном или информационно-логическом графе, условно или явно отображающем вычислительный процесс [1]. Это означает, что дублирование некоторых операций разными процессорами вычислительной системы или компьютерами сети не только не имеет существенного значения, но даже может служить минимизации времени вычислений в целом.

Это положение обусловило выбор метода Крамера для решения систем линейных уравнений,

B — столбец свободных членов,

$$\mathbf{B} = \begin{pmatrix} b_1 \\ \dots \\ b_n \end{pmatrix}.$$

Процесс распределенных вычислений реализуется по SPMD-технологии [1] при наличии главного процессора.

Представим алгоритм организации вычисления по шагам:

Шаг 1. Инициализация параллельной программы. Начало фазы раздачи заданий.

Шаг 2. Ввод количества процессоров n .

Шаг 3. Каждый процессор получает свой номер и общее число p процессоров, участвующих в вычислении. Главным процессором считается тот, номер которого равен нулю.

Шаг 4. Запускается алгоритм кластеризации. Для этого вычисляются значения s — числа подграфов минимального покрывающего дерева, и определяются главные процессоры, соответствующие каждому подграфу (этот главный процессор называется *хедер* подграфа).

Шаг 5. Если номер процессора, на котором выполняется данная копия программы, равен нулю, осуществляется переход на следующий шаг, иначе — на шаг 11.

Шаг 6. Ввод исходной матрицы **A**.

Шаг 7. Ввод исходной матрицы **B**.

Шаг 8. Проверка условия $\Delta \neq 0$. Для этого с помощью функции "определитель" процессор вычисляет определитель матрицы **A**. Если он равен нулю, на экране появится сообщение "No Answer!". В противном случае выполняется следующий шаг.

Шаг 9. Отправка с помощью процедуры широковещательной рассылки исходных данных всем процессорам (в общем случае исходные данные содержат не только матрицы, но и определитель Δ , хедер, а также программный код для исполнения).

Шаг 10. Установление времени начала вычислений.

Шаг 11. Каждый процессор с не нулевым номером выходит из режима ожидания, получив исходные данные для дальнейшей обработки. Алгоритм входит в фазу параллельного вычисления.

Шаг 12. Вызов функции "определитель".

Шаг 13. Каждый процессор по исходной матрице **A** находит столбец, номер которого соответствует номеру этого процессора, и на его место вставляет столбец **B**, формируя определитель **A'**.

Шаг 14. Вычисление значения полученного определителя.

Шаг 15. Деление значения полученного определителя на определитель Δ — получение одного корня системы уравнений.

Шаг 16. Выделение памяти для хранения всех массивов результатов и смещение для результата вычисления каждого процессора. Это необходимо

для выполнения фазы сбора результатов, в которую входит параллельный алгоритм кластеризации.

Шаг 17. Каждый хедер *последовательно* собирает результаты на всех процессорах своего подграфа, вычисленные *параллельно*.

Шаг 18. Сбор результатов продолжается *параллельно-последовательно* по дереву таким образом, пока они все не соберутся в нулевом процессоре, т. е. в центре GRID-технологий.

Шаг 19. Как только все результаты оказались на центральном процессоре, этот процессор фиксирует время окончания вычислений.

Шаг 20. Определение продолжительности вычисленного процесса с помощью времени окончания и начал.

Шаг 21. Вывод на экране результатов и времени вычислений.

Шаг 22. Завершение параллельной программы.

Шаг 23. Выход.

3. Результаты моделирования сетевой реализации метода Крамера на кластерном суперкомпьютере

Ниже изложена попытка оптимизации GRID-вычислений при решении систем линейных уравнений с помощью метода Крамера на суперкомпьютере кластерной архитектуры МИИТ СКИФ Т-4700 [5]. Для запуска параллельных программ на кластере установлена одна из версий системы PBS (*Portable Batch System* — система пакетной обработки заданий) — Torque [6], которая управляет загрузкой вычислительных комплексов, состоящих из определенного числа выделенных вычислительных узлов, работающих под управлением операционной системы семейства Unix. Для обмена данными между процессорами используется интерфейс MPI [7].

Выполним распараллеливание алгоритма решения задач по указанным выше этапам. Код программы написан на языке Си.

Этап кластеризации и раздачи заданий. Стандарт MPI требует, чтобы общая схема программы на языке Си выглядела, как представлено на рис. 1.

```
#include "mpi.h"
main(int argc, char **argv)
{
...
MPI_Init(&argc, &argv);
...
MPI_Finalize();
...
}
```

Рис. 1. Общая схема MPI программы

Большинство процедур MPI может быть вызвано только в параллельной части программы: между вызовом процедуры инициализации `MPI_Init(&argc, &argv)` и процедурой завершения `MPI_Finalize()`.

После объявления переменных в программе теста для соблюдения требований стандарта MPI выполняется инициализация параллельной части программы, после которой каждый процессор получает свой номер и общее число процессоров, участвующих в вычислительном процессе. Листинг программы приведен на рис. 2. Стандарт не требует обязательного сохранения номера и числа процессоров, однако эта информация крайне необходима для работы программы.

```
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &myRank);
MPI_Comm_size(MPI_COMM_WORLD, &procColl);
```

Рис. 2. Инициализация параллельной части программы

После того как исходные матрицы появились на главном процессоре (процессор, имеющий номер 0), необходимо выполнить их рассылку всем остальным процессорам, которые участвуют в вычислениях. Но предварительно по алгоритму ПАК проводится кластеризация процессоров, участвующих в вычислениях. Для этого используется процедура MPI_GRAPH_CREATE (рис. 3).

```
MPI_GRAPH_CREATE(COMM, NNODES, INDEX, EDGES,
REORDER, COMM_GRAPH, IERR)
INTEGER COMM, NNODES, INDEX(*), EDGES(*),
COMM_GRAPH, IERR LOGICAL REORDER
```

Рис. 3. MPI_GRAPH_CREATE

Подробное описание процедуры можно найти в Приложении или в работе [7].

Для рассылки необходимых данных предназначена групповая процедура широковещательной рассылки, которая позволяет передать блок данных от одного процессора нескольким другим одновременно. Информация, поясняющая параметры вызова функции MPI_Bcast, представлена на рис. 4.

```
MPI_BCAST(BUF, COUNT, DATATYPE, ROOT, COMM, IERR)
<type> BUF(*)
INTEGER COUNT, DATATYPE, ROOT, COMM, IERR
```

Рис. 4. MPI_Bcast

Подробное описание процедуры можно найти в Приложение или в работе [7].

После выполнения широковещательной рассылки каждый процессор располагает копией исходной матрицы **A**, столбцом **B** и значением определителя Δ .

При моделировании необходимо вручную контролировать, на каких узлах запускается программа, т. е. в скрипте Torque следует прописать, например, инструкцию

```
#PBS-1 nodes=node-33:ppn=
=3+node-57:ppn=2+node-60:ppn=5
```

Это означает, что на 33-м узле запустятся 3 процессора, на 57-м — 2, на 60-м — 5.

Перед началом вычислений сохраняется время их начала (рис. 5). Позднее, по окончании вычислительного процесса, будет сохранено время конца вычислений. Разница между временем конца и начала покажет продолжительность решения задачи.

```
//Сохранение времени начала вычислений
MPI_Barrier(COMM, IERR);
integerCOMM, IERR.
...
startTime=MPI_Wtime();
```

Рис. 5. Сохранение времени начала вычислений

Процедура MPI_Barrier используется для общей синхронизации процессоров, т. е. для одновременного начала работы. Работа каждого процессора блокируется до тех пор, пока все остальные процессоры коммуникатора COMM не выполнят эту процедуру. Только после того, как последний процессор коммуникатора выполнит данную процедуру, все процессоры будут разблокированы и продолжат работу со следующей инструкцией.

После синхронизации выполняется проверка номера процессора, и в случае если номер равен 0, выполняется вызов процедуры MPI_Wtime(), которая возвращает на вызвавшем процессоре астрономическое время в секундах.

Так как каждый процессор будет вычислять лишь часть элементов матрицы, для их хранения необходимо выделить память. Для этого использовались процедуры calloc().

Этап параллельных вычислений. На этом этапе все просто. Каждый процессор вычисляет только один корень абсолютно *независимо* от других процессоров. Для этого каждый процессор в исходной матрице **A** находит столбец, номер которого равен его номеру, и заменяет этот столбец столбцом **B**. Так будет получена матрица **A'**.

Затем процессор с помощью вызова функции "определитель" вычисляет определитель матрицы **A'** и делит его на определитель Δ .

Таким образом вычисляется корень системы *n* линейных уравнений с *n* неизвестными.

Этап сборки результатов. Для сборки данных вызывается функция MPI_Gatherv, рис. 6 иллюстрирует параметры вызова MPI_Gatherv и их тип.

Подробное описание процедуры можно найти в Приложении или в работе [7].

Здесь используется параллельный алгоритм кластеризации (ПАК) для сбора данных. Предполагается, что нулевой процессор располагается в центре GRID-технологий, а (s-1) хедеры являются

```
MPI_GATHERV(SBUF, SCOUNT, STYPE, RBUF, RCOUNTS, DISPLS, RTYPE, ROOT, COMM, IERR)
<type> SBUF(*), RBUF(*)
INTEGER SCOUNT, STYPE, RCOUNTS(*), DISPLS(*), RTYPE, ROOT, COMM, IERR
```

Рис. 6. MPI_Gatherv

процессорами ROOT для остальных процессоров своего подграфа.

По окончании вычислительного процесса следует сохранить время конца вычислений (рис. 7).

```
//сохранение времени конца вычислений.
stopTime=MPI_Wtime();
```

Рис. 7. Сохранение времени конца вычислений

Корректное завершение программы предполагает освобождение памяти и вызов функции MPI, указывающей на завершение параллельной части программы (рис. 8).

```
free();
free();
free();
free(displs);
free(rcounts);
MPI_Finalize();
return 0;
}
```

Рис. 8. Корректное завершение программы

В результате выполнения программы на различном числе процессоров было получено время решения задачи системы n линейных уравнений с n неизвестными, результаты измерений представлены в таблице, где $t_{\text{выч}}$ — время вычисления в секундах, n — число процессоров.

Здесь значение $t_{\text{выч}} = 0$ условно отображает пренебрежительно малую (для суперкомпьютера) величину.

Примерный график зависимости времени вычислений от числа процессоров, с учетом обязательного выполнения операции сборки результатов на головном процессоре, имеет вид, показанный на рис. 9.

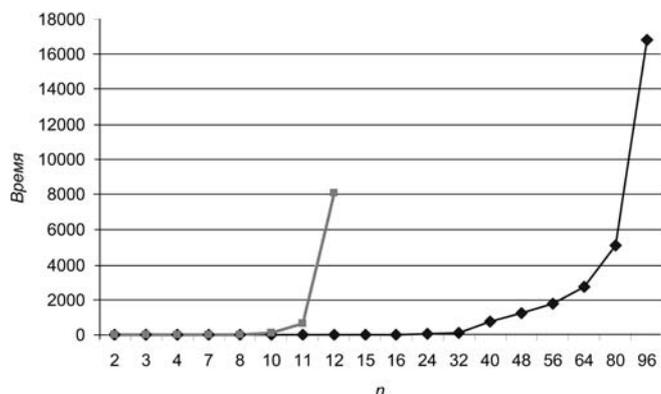


Рис. 9. Примерный вид графика зависимости времени выполнения распределенных вычислений от числа участвующих процессоров с выполнением и без выполнения операции кластеризации

Результаты измерений

n	$t_{\text{выч}}$, с, без кластеризации	$t_{\text{выч}}$, с, с кластеризацией
2	0	0
3	0	0
4	0	0
7	0,686	0
8	2,215	0
10	80,979	0
11	669,13	0,241
12	8045,09	3,844
15	—	5,013
16	—	21,741
24	—	43,182
32	—	64,193
40	—	941,535
48	—	1151,124
56	—	1749,69
64	—	2753,219
80	—	5122,683
96	—	16843,863

Оказалось, что сокращение времени вычислений $t_{\text{выч}}$ с ростом числа n процессоров, участвующих в счете, без применения каких-либо мер по оптимизации обмена прекращается по достижении $n \approx 15$. Далее с ростом n обмен начинает настолько превалировать, что время $t_{\text{выч}}$ начинает расти. Хотя некоторый рост времени счета следует считать "естественным". Далее, с ростом n обмен начинает настолько превалировать, что время $t_{\text{выч}}$ возрастает резко, что становится недопустимым.

Левая часть графика соответствует последовательному обмену каждого процессора с головным.

Целью оптимизации этого обмена является, во-первых, дальнейшее уменьшение времени решения задачи при вовлечении большего числа процессоров, а во-вторых, — максимальное смещение вправо, т. е. на большое число процессоров, точки достижения минимального времени решения задачи.

Применение предложенного в работе [2] алгоритма кластеризации позволило получить дальнейшее снижение времени распределенных вычислений при привлечении гораздо большего числа процессоров (порядка 90), как показано на правом графике рис. 9. При этом наблюдается "естественный" рост времени вычислений с ростом размерности задачи. Однако при достижении некоторого числа процессоров этот рост времени счета определяется превалирующим временем обмена и становится недопустимым.

Заключение

При проведении экспериментов на суперкомпьютере предполагалось, что число используемых процессоров совпадало с размерностью задачи. Это обусловлено тем, что увеличение абсолютно-

го времени занятости процессоров при счете не одного, а нескольких (порядка n/p) неизвестных практически не влияет на логарифмический рост времени обмена при сборке данных от p процессоров (и то — с учетом их кластеризации). Этот обмен превалирует настолько, что допускает любые ограничения на собственно вычислительные операции. Это, в частности, определило выбор метода вычислений. Важно то, что общая схема вычислительного процесса остается типичной и неизменной.

Выводы

1. Распределенные вычисления в сети имеют низкую границу эффективности параллельного использования процессоров (компьютеров, рабочих станций) сети. Это обусловлено потерями времени, затрачиваемыми на сбор результатов на головном процессоре. Такой сбор требует единовременной передачи данных от всех участвующих процессоров. Сетевые технологии, предполагающие только последовательный попарный обмен, не обеспечивают требуемой скорости обмена данными.

Так, при решении системы n уравнений методом Крамера уже при $n = 15$, где n — число участвующих процессоров, между которыми по SPMD-технологии распределяется счет корней, распределенные вычисления следует считать нецелесообразными. С этого значения время вычислений начинает расти экспоненциально.

2. Применение разработанного автором алгоритма кластеризации процессоров ПАК (отличающегося параллельными вычислениями и лучшими характеристиками по сравнению с известными ранее алгоритмами), где сбор данных имеет "древесную" структуру, позволило существенно повысить границу эффективности распределенных вычислений, доведя ее до ~90 процессоров.

3. Суперкомпьютер МИИТ Т-4700 (СКИФ), где обмен осуществляется на основе сетевой технологии *Infiniband* [5], не обладает средством кластеризации при обмене "от многих к одному". Применение ПАК здесь также позволило повысить границу эффективности распределенных вычислений почти в 10 раз.

Указанные выше результаты исследований определяют рекомендации разработчику программ.

Список литературы

1. Барский А. Б. Параллельные информационные технологии: учеб. пособие. М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2007.
2. Амиршахи Б. Кластеризация GRID-ресурсов для оптимизации информационного обмена при совместной обработке результатов распределенных вычислений // Информационные технологии. 2011. № 2. С. 22—28.

3. Метод Крамера. URL: http://ru.wikipedia.org/wiki/Метод_Крамера

4. Минимальное остовное дерево. URL: http://wiki.pedia.org/wiki/Минимальное_остовное_дерево

5. Т-Платформы. Суперкомпьютер МИИТ Т-4700, URL: <http://www.t-platforms.ru/ru/clusters/clusters/unique/miit-t-4700.html>

6. Т-Платформы. Система пакетной обработки заданий torque: Руководство пользователя. 2008.

7. Антонов А. С. Параллельное программирование с использованием технологии MPI. М.: Изд-во МГУ, 2004.

Приложение

MPI_GRAPH_CREATE () — создание на основе коммуникатора COMM нового коммуникатора COMM_GRAPH с топологией графа. Параметр NNODES задает число вершин графа, INDEX(I) содержит суммарное число соседей для первых I вершин. Массив EDGES содержит упорядоченный список номеров процессоров — соседей всех вершин. Параметр REORDER при значении .TRUE. означает, что системе разрешено менять порядок нумерации процессоров.

MPI_BCAST () — рассылка COUNT элементов данных типа DATATYPE из массива BUF от процессора ROOT всем процессорам данного коммуникатора COMM, включая сам рассылающий процессор. При возврате из процедуры содержимое буфера BUF процессора ROOT копируется в локальный буфер каждого процессора коммуникатора COMM. Значения параметров COUNT, DATATYPE, ROOT и COMM должны быть одинаковыми у всех процессоров.

MPI_GATHERV () — сборка SCOUNT элементов данных типа STYPE из массивов SBUF со всех процессоров коммуникатора COMM в буфере RBUF процессора ROOT. Каждый процессор, включая ROOT, посылает содержимое своего буфера SBUF процессору ROOT. Собирающий процессор сохраняет данные в буфере RBUF, располагая их в порядке возрастания номеров процессоров.

На собирающем процессоре существенными являются значения всех параметров, а на остальных процессорах — только значения параметров SBUF, SCOUNT, STYPE, ROOT и COMM. Значения параметров ROOT и COMM должны быть одинаковыми у всех процессоров. Параметр RCOUNT у процессора ROOT обозначает число элементов типа RTYPE, принимаемых не от всех процессоров в целом, а от каждого процессора.

Как уже было упомянуто выше, для вызова функции *MPI_Gatherv* каждый процессор должен иметь два массива: первый (*rcounts*) для хранения длин всех массивов результатов, которые будут посланы процессорами, второй (*displs*) для хранения смещений результатов каждого из процессоров относительно начала буфера.

УДК 519.816;004.89

Е. А. Елтаренко,

Национальный исследовательский
ядерный университет "МИФИ", г. Москва
e-mail: EAEltarenko@mephi.ru

Аппроксимация предпочтений в многокритериальных задачах операторами агрегирования

Предлагается подход, основанный на выявлении предпочтений лица, принимающего решения (ЛПР), на основе единой процедуры опроса. Подход предполагает выполнение формально сформулированных аксиом предпочтений: монотонность предпочтений; веса критериев отражают важность. Приводится аппарат генерации операторов агрегирования критериев, отвечающих аксиомам предпочтений. Известные функции полезности являются частными случаями операторов агрегирования. Процедура идентификации оператора позволяет по результатам опроса ЛПР определить оператор, аппроксимирующий индивидуальные предпочтения ЛПР.

Ключевые слова: многокритериальные задачи, описание предпочтений, веса критериев, операторы агрегирования

Введение

Решение многокритериальных задач (МКЗ) основывается на учете предпочтений лица, принимающего решения (ЛПР). В каждом подходе и методе решения МКЗ предпочтения задаются в том или ином виде.

В многомерной теории полезности (*multi-attribute utility theory*) [1] они задаются *прямо*. На основе изучения свойств предпочтений определяется функция полезности, которая описывает предпочтения ЛПР. При решении конкретной задачи проводится работа с ЛПР по изучению предпочтений, и затем проверяются условия применения функции полезности. Число различных функций полезности не так велико, так в работе [1] рассматриваются три функции: полилинейная, мультипликативная и аддитивная. Проверка условий — весьма трудоемкий процесс, требует подготовленности ЛПР. Надо сказать, что строгая теория этого подхода плохо сочетается с практикой его применения. При решении МКЗ многомерные функ-

ции полезности выполняют роль агрегирования векторов критериев в скаляр.

Широко используются на практике интерактивные методы вследствие своей простоты, наглядности. Наибольшее распространение получили методы группы *outranking*: ELECTRE, PROMETHEE, ORESTE и их модификации [2—4], а также методы группы "смещенного идеала" [5]. Агрегирование критериев в интерактивных методах осуществляется по фиксированной процедуре. Для задания предпочтений ЛПР используются определенные параметры метода, среди которых чаще всего присутствуют веса критериев, определяющие их относительную важность. Можно сказать, что предпочтения в них задаются *косвенно через параметры* метода. Причем такое косвенное задание неполно описывает предпочтения ЛПР. В интерактивных методах у ЛПР нет возможности точно задать свои предпочтения, даже если он хотел бы это сделать. Неподготовленных ЛПР такие методы вполне устраивают, но для решения задач принятия ответственных решений интерактивные методы недостаточно строги, так как в них нельзя задать предпочтения в явном виде.

Использование в методах теории нечетких множеств (*fuzzy set*) [6] позволяет отразить неуверенность, неточности в задании ЛПР данных, но не изменяет сути описания предпочтений. Более того, для неподготовленных ЛПР использование понятийного аппарата нечетких множеств усложняет задание своих предпочтений.

Следует подчеркнуть, что на получаемые решения МКЗ в наибольшей степени влияют процедуры агрегирования критериев, именно они определяют предпочтения ЛПР. В настоящей работе предлагается использовать *операторы агрегирования* [7] для описания предпочтений. Операторы агрегирования переменных представляют собой процедуры агрегирования данных. Их нельзя относить к многомерным функциям полезности, хотя по свойствам они похожи на них, так как функции полезности изначально разрабатываются в целях изучения предпочтений ЛПР и определения функции на основе выявленных свойств. Операторы агрегирования ориентированы не только на многокритериальные методы и могут применяться для других целей.

Постановка задачи. Необходимо разработать инструмент, с помощью которого ЛПР сможет

прямо задавать свои предпочтения. Наряду с этим ЛПР может свои предпочтения отразить и в традиционном для интерактивных методов виде, задав веса критериев. Заданные предпочтения должен описывать один из операторов агрегирования.

Для решения поставленной задачи необходимо разработать:

а) аппарат генерации операторов агрегирования; требования к операторам определяются свойствами предпочтений;

б) план проведения опроса ЛПР, чтобы в результате опроса получить оценки предпочтений в точках, позволяющих идентифицировать оператор;

в) процедуру идентификации оператора на основе результатов опроса ЛПР, т. е. определить оператор, который наиболее точно аппроксимирует заданные предпочтения.

1. Генерация операторов агрегирования

Пусть имеется множество критериев u_1, u_2, \dots, u_m , измеряемых в относительных единицах в интервале $[0; 1]$ (вопросы перевода критериев в относительные величины в статье не рассматриваются).

Прежде всего, укажем свойства предпочтений $P(u_j, \dots, u_m)$, которые рассматриваются в качестве аксиом предпочтений. Они должны выполняться и для операторов агрегирования. Считаем заданными коэффициенты относительной важности критериев V_1, \dots, V_m (веса критериев), причем

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

Нормирующее условие (1) говорит о том, что веса измеряются в шкале отношений.

Введем следующие обозначения: \bar{u}_j — дополнение критерия u_j , все множество агрегируемых критериев, кроме j -го критерия; $\bar{u}_{j,s}$ — дополнение критериев u_j и u_s . Выберем два произвольных уровня изменения значений критериев: $u_j^+ = u^+$ и $u_j^- = u^-$, причем $u^+ > u^-$. Запись \bar{u}_j^- означает — все критерии, кроме u_j , принимают значение u^- .

Аксиомы предпочтений многокритериальных объектов

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

Эта аксиома обеспечивает выполнение условий доминирования многокритериальных объектов. Оптимальность по Парето также основана на этой аксиоме. Если ЛПР считает, что условие доминирования выполняется во всей области определения переменных, то аксиома выполняется. Выполнение аксиомы означает, что поверхности безразличия не имеют общих точек (не пересекаются).

Зафиксируем дополнение критерия \bar{u}_j^1 , определим изменение предпочтений при изменении критерия в интервале $u_j \in [0; 1]$, назовем ее условной функцией предпочтений $P(u_j, \bar{u}_j^1)$.

Аксиома 2. Веса V_j отражают предпочтения критериев. Для любых двух критериев с $V_j > V_s$ с любым фиксированным дополнением $\bar{u}_{j,s}^1$ условная функция предпочтения $P(u_j^+, u_s^-, \bar{u}_{j,s}^1)$ больше $P(u_j^-, u_s^+, \bar{u}_{j,s}^1)$. Из $V_j = V_s$ следует $P(u_j^+, u_s^-, \bar{u}_{j,s}^1) = P(u_j^-, u_s^+, \bar{u}_{j,s}^1)$.

В соответствии с аксиомой 2 веса критериев определяют соотношения условных функций предпочтений $P(u_j, \bar{u}_j^-)$, где $u_j \in [0; 1]$, и $u^- = 0$: из $V_j > V_s$ следует $P(u_j, \bar{u}_j^-) > P(u_s, \bar{u}_s^-)$. Если принять $u^+ = 1$, в соответствии с аксиомой из $V_j > V_s$ следует $P(u_j, \bar{u}_j^+) < P(u_s, \bar{u}_s^+)$, где $u_j, u_s \in [0; 1]$. Более того, при $u^- > 0$, $u^+ < 1$ и $u^- > u^+$ из $V_j > V_s$ следует $P(u_j, \bar{u}_j^-) > P(u_s, \bar{u}_s^-)$ и $P(u_j, \bar{u}_j^+) < P(u_s, \bar{u}_s^+)$, где $u_j, u_s \in [u^-; u^+]$.

Аксиома 3. Существует единственный объект с минимальным нулевым предпочтением. Приняв $u^- = 0$, выполнение этой аксиомы можно записать в виде $P(u_j^-, \bar{u}_j^-) = 0$ и для любого $u_j > 0$ выполняется $P(u_j, \bar{u}_j^-) > 0$.

Аксиома 4. Существует единственный объект с максимальным предпочтением, равным 1. Приняв $u^+ = 1$, выполнение данной аксиомы можно записать в виде: $P(u_j^+, \bar{u}_j^+) = 1$ и $P(u_j, \bar{u}_j^+) < 1$ для любого $u_j < 1$.

Для случая, когда значения всех критериев изменяются на двух уровнях $u^- = 0$ и $u^+ \in (0, 1]$, на основании аксиом сформулируем несколько утверждений, которые потребуются в дальнейшем.

Утверждение 1. Порядок предпочтения объектов $P(u_j^+, \bar{u}_j^-)$ ($j = 1, 2, \dots, m$) определяется порядком весов критериев.

Утверждение 2. Для трех критериев с весами V_j, V_s, V_t из $V_j + V_s > V_j + V_t$ следует $P(u_j^+, u_s^+, \bar{u}_{j,s}^-) > P(u_j^+, u_t^+, \bar{u}_{j,t}^-)$, $V_j + V_s = V_j + V_t \Rightarrow P(u_j^+, u_s^+, \bar{u}_{j,s}^-) = P(u_j^+, u_t^+, \bar{u}_{j,t}^-)$.

Замечание 1. Для четырех критериев в общем случае из $V_j + V_s \geq V_r + V_t$ не следует $P(u_j^+, u_s^+, \bar{u}_{j,s}^-) \geq$

$\geq P(u_r^+, u_t^+, \bar{u}_{r,t}^-)$, но в некоторых случаях такие соотношения выполняются.

Утверждение 3. Начиная с объекта $\{u_j^-, \bar{u}_j^-\}$, предпочтение объектов в последовательности, образованной заменой u_s^- на u_s^+ , возрастает:

$$P(u_j^-, \bar{u}_j^-) < P(u_j^+, \bar{u}_j^-) < P(u_j^+, u_s^+, \bar{u}_{j,s}^-) < < P(u_j^+, u_s^+, u_r^+, \bar{u}_{j,sr}^-) < \dots < P(\bar{u}_t^+, u_t^-) < P(\bar{u}_t^+, u_t^+).$$

Это справедливо для любого критерия j , начиная с которого строится последовательность.

Перейдем к требованиям, которым должны удовлетворять операторы агрегирования $h(u_1, \dots, u_m)$:

а) входными переменными оператора являются значения агрегируемых критериев (переменных) u_1, u_2, \dots, u_m . Результат агрегирования (значение оператора) принадлежит интервалу $[0; 1]$;

б) параметрами оператора являются веса критериев;

в) для оператора должны выполняться все аксиомы предпочтений.

Рассмотрим основную процедуру генерации операторов.

Пусть заданы две функции $\varphi(x)$ и $\psi(x)$ — непрерывные, монотонно возрастающие на участке $x \in [0; 1]$, на границах $\varphi(0) = 0, \psi(0) = 0$ и $\varphi(1) = 1, \psi(1) = 1$. Функции $\varphi(x)$ и $\psi(x)$ назовем *генерирующими функциями оператора агрегирования*. Чтобы сформировать генерирующую функцию, достаточно взять монотонно возрастающую функцию $f(x)$ и пронормировать ее: $\varphi(x) = \frac{f(x) - f(0)}{f(1) - f(0)}$. Используя

разных видов $f(x)$ приводит к формированию отдельных *видов генерирующих функций*. Каждый вид генерирующих функций может иметь свой параметр, с помощью которого формируется семейство генерирующих функций одного вида. Например, степенная функция $\varphi(x) = x^p$ имеет параметр p , позволяющий формировать семейство функций.

На основе пары генерирующих функций можно предложить два варианта формирования операторов агрегирования, отличающихся способом учета весов критериев.

Способ 1. Для каждого критерия u_j определим значение обратной функции $\varphi^{-1}(u_j)$ (рис. 1), затем вычислим сумму $Fh(u_1, \dots, u_m) = \sum_{j=1}^m V_j \varphi^{-1}(u_j)$. В качестве оператора агрегирования используем

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

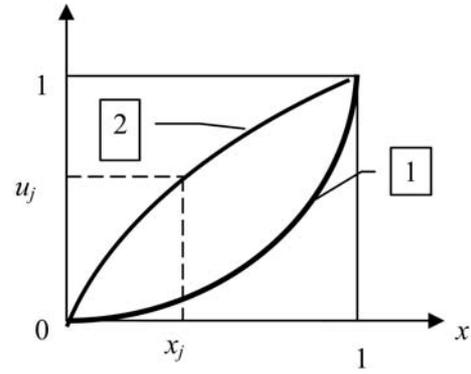


Рис. 1. Построение оператора типа h :
1 — $\psi(x)$; 2 — $\varphi(x), x_j = \varphi^{-1}(u_j)$

Определим оператор (2) как *оператор типа h*.

Способ 2. Для каждого критерия определим $\varphi^{-1}(V_j u_j)$, а затем найдем сумму $Fg(u_1, \dots, u_m) = \sum_{j=1}^m \varphi^{-1}(V_j u_j)$. Оператором агрегирования будет сле-

дующая функция: $g(u_1, \dots, u_m) = \psi \left(\sum_{j=1}^m \varphi^{-1}(V_j u_j) \right)$.

Следует сказать, что граничные условия $g(0, \dots, 0) = 0, h(0, \dots, 0) = 0$ (аксиома 3) и $h(1, \dots, 1) = 1$ (аксиома 4) выполняются исходя из требований к генерирующим функциям. Для выполнения условия $g(1, \dots, 1) = 1$ необходимо провести масштабирование $Fg(u_1, \dots, u_m)$, т. е. обеспечить выполнение условия $Fg(1, \dots, 1) = 1$. Это условие выполняется при следующем равенстве:

$$\sum_{j=1}^m \varphi^{-1}(V_j) = 1. \quad (3)$$

Обеспечить выполнение (3) можно введением масштабного коэффициента оператора либо в качестве множителя $\varphi(x)$ и тогда условие (3) можно записать в следующем виде:

$$\sum_{j=1}^m d \varphi^{-1}(V_j) = 1, \quad (4)$$

либо в качестве множителя V_j и тогда условие (3) переписывается в виде

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

Обозначим оператор, для которого масштабирование осуществляется в соответствии с (5), как оператор типа g_a :

$$g_a(u_1, \dots, u_m) = \psi \left(\sum_{j=1}^m \varphi^{-1}(k V_j u_j) \right), \quad (6)$$

а оператор с параметром d , определяемым из уравнения (4), назовем оператором типа g_b :

$$g_b(u_1, \dots, u_m) = \psi \left(d \sum_{j=1}^m \varphi^{-1}(V_j u_j) \right). \quad (7)$$

Отметим, что коэффициент d оператора g_b определяется по формуле

$$d = 1 / \sum_{j=1}^m \varphi^{-1}(V_j), \quad (8)$$

а для определения коэффициента k оператора g_a требуется решить уравнение (5).

Заметим также, что при $\sum_{j=1}^m \varphi^{-1}(V_j) > 1$ масштабные коэффициенты d и k меньше 1, а при $\sum_{j=1}^m \varphi^{-1}(V_j) < 1$ больше 1, причем $k < 1/\max(V_j)$.

В чем принципиальное отличие операторов разного типа? Операторы отличаются функциями Fh , Fg_a , Fg_b . В операторах типа h при изменении переменной $u_j \in [0; 1]$ слагаемое $V_j \varphi^{-1}(u_j)$ в Fh использует всю функцию $\varphi(x)$. В операторах типа g_b в Fg_b только часть $\varphi(x)$ в интервале $V_j u_j \in [0; V_j]$, а в операторах типа g_a в Fg_a используется часть $\varphi(x)$ в интервале $k V_j u_j \in [0; k V_j]$. Можно сказать, что в операторе g_b каждая переменная имеет собственную генерирующую функцию $\varphi_j(x)$, определяемую из $\varphi(x)$ с учетом веса V_j , а в операторе g_a функции $\varphi_j(x)$ определяются из $\varphi(x)$ и весов всех критериев, так как коэффициент k рассчитывается на основе всех V_j .

Для операторов всех типов выполняются аксиомы предпочтений. Монотонность генерирующих функций обеспечивает монотонность операторов, тем самым выполняется аксиома 1. Выполнение аксиомы 3 обеспечивается требованием к генерирующим функциям $\varphi(0) = 0$, $\psi(0) = 0$. Для выполнения аксиомы 4 для операторов введены масштабные коэффициенты.

Подробнее остановимся на выполнении аксиомы 2. Условная функция оператора при фиксированном дополнении равна $h(u_j, \bar{u}_j^1)$. Выполнение аксиомы 2 для оператора записывается следующим образом: $h(u_j, u_s^-, \bar{u}_{j,s}^1) > h(u_j^-, u_s, \bar{u}_{j,s}^1)$ при $V_j > V_s$. Она будет выполняться, если в функциях Fh , Fg_a , Fg_b веса будут определять соотношения между слагаемыми по каждой переменной.

В операторе типа h слагаемое Fh для критерия j равно $V_j \varphi^{-1}(u_j)$, т. е. вес определяет условную функцию оператора $h(u_j, u_s^-, \bar{u}_{j,s}^1)$, соответственно,

для критерия s слагаемое $V_s \varphi^{-1}(u_s)$ определяет $h(u_j^-, u_s, \bar{u}_{j,s}^1)$. Все слагаемые Fh , входящие в дополнение $\bar{u}_{j,s}^1$, остаются неизменными. Поэтому порядок $h(u_j, u_s^-, \bar{u}_{j,s}^1) > h(u_j^-, u_s, \bar{u}_{j,s}^1)$ определяется весами V_j и V_s .

В операторе типа g_a слагаемое Fg_a для критерия j равно $\varphi^{-1}(k V_j u_j)$. Чтобы выполнялась аксиома, необходимо выполнить условие: $\varphi^{-1}(k V_j u) > \varphi^{-1}(k V_s u)$ при $V_j > V_s$. Это условие выполняется для всех монотонно-возрастающих функций.

Аналогично чтобы выполнялась аксиома 2 для операторов типа g_b , необходимо выполнить требование: $d \varphi^{-1}(V_j u) > d \varphi^{-1}(V_s u)$ при $V_j > V_s$, которое справедливо для монотонно-возрастающих функций.

Остановимся на вопросах влияния параметров оператора на его функционал. Отметим, что параметрами оператора являются генерирующие функции и веса переменных.

Функция $\varphi(x)$ вместе с весами определяет поверхности безразличия оператора (поверхностей одинакового уровня). Действительно, задание поверхности безразличия означает, что значение оператора

$$\begin{aligned} & \text{фиксируется, что равнозначно } Fh = \sum_{j=1}^m V_j \varphi^{-1}(u_j) = \\ & = \text{const (для операторов типа } h) \text{ или } Fg(u_1, \dots, u_m) = \\ & = \sum_{j=1}^m \varphi^{-1}(V_j u_j) = \text{const (для операторов типа } g). \end{aligned}$$

Видим, что поверхности безразличия определяют только генерирующей функцией $\varphi(x)$ и весами.

Утверждение 4. Если генерирующая функция выпукла вниз/вверх, то и поверхности безразличия оператора агрегирования типа h выпуклы вниз/вверх.

Доказательство.

Поверхность безразличия оператора h определяется неявной функцией $Fh(u_1, \dots, u_m) = \sum_{j=1}^m V_j \varphi^{-1}(u_j) = \text{const}$. Функция $Fh(u_1, \dots, u_m)$ выпукла вниз,

если гессиан $\left\| \frac{\partial^2 Fh}{\partial u_j \partial u_s} \right\|$ неотрицательно определен

[8, стр. 44]. Вторые производные $\frac{\partial^2 Fh}{\partial u_j \partial u_s} = 0$, если

$$j \neq s, \text{ а знак } \frac{\partial^2 Fh}{\partial^2 u_j} = \frac{V_j \varphi''(u_j)}{\varphi'^2(u_j)} \text{ определяется знаком}$$

$\varphi''(u_j)$. Таким образом, в гессиане только диагональные элементы для выпуклой вниз генерирующей функции $\varphi(x)$ больше нуля, т. е. гессиан неот-

рицательно определен. Следовательно, функция $Fh(u_1, \dots, u_m)$ выпукла вниз. Сечение выпуклой функции плоскостью образует в результате выпуклую функцию. Значит поверхность безразличия $Fh(u_1, \dots, u_m) = \text{const}$ — выпуклая функция.

Утверждение 5. Если для максимального веса V_1 генерирующая функция $\varphi(x)$ на участке $[0; V_1]$ выпукла вниз/вверх, то поверхности безразличия оператора агрегирования типа g_b выпуклы вниз/вверх. Доказательство аналогично утверждению 4.

Утверждение 6. Если для максимального веса V_1 генерирующая функция $\varphi(x)$ выпукла вниз/вверх на участке $[0; kV_1]$, где k — масштабный коэффициент оператора типа g_a , то и поверхности безразличия оператора g_a выпуклы вниз/вверх. Доказательство аналогично утверждению 4.

Вторая генерирующая функция $\psi(x)$ определяет совместное влияние всех переменных вместе. Так, если функция Fh , Fg_a или Fg_b близка к 1 (это зависит от значений всех переменных), то значение оператора будет достаточно большое. Насколько оно близко к 1 определяется функцией $\psi(x)$. Например, если $\psi(x)$ выпукла вниз ($\psi(x) < x$), то чтобы достигнуть значения оператора, равного 0,9, необходимо, чтобы Fh , Fg_a или Fg_b были больше 0,9. Наглядно влияние $\psi(x)$ на значения оператора отражает одномерная функция оператора, вычисляемая при равных значениях переменных $u_j = u$, назовем такую функцию — *функцией равных значений переменных* (ФРЗП) оператора и будем обозначать $h(u)$, $g_a(u)$, $g_b(u)$.

Сформулируем ряд утверждений, касающихся ФРЗП разных типов операторов. Ввиду простоты доказательства не приводим.

Утверждение 7. Для оператора типа h ФРЗП, равная $h(u) = \varphi(\varphi^{-1}(u))$, не зависит от весов критериев, а определяется только генерирующими функциями.

Утверждение 8. Для оператора типа h при равенстве генерирующих функций (см. ниже рис. 4) ФРЗП линейная: $h(u) = u$.

Для операторов типа g_a и g_b ФРЗП зависит от весов критериев:

$$g_a(u) = \psi\left(\sum_{j=1}^m \varphi^{-1}(kV_j u)\right), \quad g_b(u) = \psi\left(d \sum_{j=1}^m \varphi^{-1}(V_j u)\right).$$

Таким образом, обе генерирующие функции определяют поведение оператора, используя разные генерирующие функции, получаем конкретный оператор. При построении операторов разного типа были рассмотрены все возможные варианты использования весов в качестве параметров, это говорит о полноте множества типов операторов h , g_a , g_b , для которых выполняется аксиома 2. Вопрос о пол-

ноте множества операторов, для которых выполняются все аксиомы предпочтений, сводится к полноте множества *видов генерирующих функций*.

Веса критериев V_j определяют значения оператора в вершинах куба, образуемого сочетаниями $u^- = 0$ и $u^+ = 1$ всех u_j (план полного факторного эксперимента).

Для трех критериев вершины куба определяются следующими точками: $\{u_1^+, \bar{u}_1^-\}$, $\{u_2^+, \bar{u}_2^-\}$, $\{u_3^+, \bar{u}_3^-\}$, $\{u_1^+, u_2^+, u_3^-\}$, $\{u_1^+, u_2^-, u_3^+\}$, $\{u_1^-, u_2^+, u_3^+\}$, все типы операторов в вершине $\{u_1^-, u_2^-, u_3^-\}$ рав-

ны 0, а в вершине $\{u_1^+, u_2^+, u_3^+\}$ равны 1. *Множество значений оператора в вершинах куба* (МЗВК) определяются генерирующими функциями и весами критериев, поэтому будем обозначать значения МЗВК с использованием весов: $h(v_j) = h(u_j^+, \bar{u}_j^-)$,

$h(v_j, v_s) = h(u_j^+, u_s^+, \bar{u}_{j,s}^-)$ и т. д. Следует сказать, что в качестве u^- и u^+ при построении куба можно использовать $u^- > 0$ и $u^+ < 1$, значения u^- и u^+ определяют ребро куба.

Утверждение 9. Если обе генерирующие функции линейные $\varphi(x) = \psi(x) = x$, то все типы операторов агрегирования принимают одинаковый вид:

$$h(u_1, \dots, u_m) = \sum_{j=1}^m V_j u_j, \quad (9)$$

это аддитивная функция полезности или ее обычно называют *линейной сверткой*.

Линейный оператор описывает предпочтения, для которых выполняется условие взаимной компенсации критериев: любое уменьшение на $-\Delta_j$ одного критерия можно компенсировать увеличением другого на $+\Delta_s$, причем изменения Δ_j и Δ_s не зависят от исходных значений u_j и u_s . В качестве условий применения аддитивного оператора можно использовать и другие теоремы о существовании аддитивной функции полезности. Поверхностями безразличия такого оператора являются плоскости. Для линейного оператора ФРЗП — линейная, и МЗВК — тоже линейная относительно весов: $h(v_j) = V_j$, $h(v_j, v_s) = V_j + V_s$.

Утверждение 10. Для оператора типа h значения МЗВК равны $h(v_j) = \psi(V_j)$, $h(v_j, v_s) = \psi(V_j + V_s)$, следовательно, МЗВК, по существу, является одной для всех критериев функцией весов $h(v)$, которая определяется генерирующей функцией $\psi(x)$. Аргументами $h(v)$ будут суммы весов, соответствующих вершинам u_j^+ , $h(v)$ — монотонная функция.

Назовем $h(v)$ *весовой функцией* оператора типа h .

Утверждение 11. Для оператора типа g_a с одинаковыми генерирующими функциями $\varphi(x) = \psi(x)$ значения в m вершинах куба линейно связаны с весами критериев: $g_a(v_j) = kV_j$.

Для доказательства достаточно записать значение оператора в указанной вершине куба: $g_a(v_j) = \psi(\varphi^{-1}(kV_j u_j^+) + \sum_{s \neq j} \varphi^{-1}(kV_s u_s^-)) = \psi(\varphi^{-1}(kV_j)) = kV_j$.

Утверждение 12. При фиксированном дополнении $\bar{u}_j^- = 0$ оператор типа g_a с одинаковыми генерирующими функциями $\varphi(x) = \psi(x)$ является линейной функцией u_j : $g_a(u_j, \bar{u}_j^-) = kV_j u_j$. Это видно на рис. 2 (см. третью сторону обложки).

Утверждение 13. МЗВК для операторов типа g описывается множеством из m монотонных весовых функций.

Действительно, в соответствии с утверждением 3 можно для каждого критерия сформировать монотонную весовую функцию $g_a(v_j)$, например, $g_a(v_j) = \psi(\varphi^{-1}(kV_j))$, $g_a(v_j, v_s) = \psi(\varphi^{-1}(kV_j) + \varphi^{-1}(kV_s))$ и т. д.

Объединение всех весовых функций в одну $g(v)$ может привести к тому, что она будет не монотонной, более того, с учетом замечания 1 будет неоднозначной. Для операторов типа g возможны случаи, когда при равенстве $V_j + V_s = V_r + V_t$ весовые функции отличны $g(v_j, v_s) \neq g(v_r, v_t)$. Но для визуализации МЗВК $g(v)$ весьма полезна.

Замечание 2. Немонотонность МЗВК характерна только для операторов типа g .

На рис. 3 приведены две генерирующие функции: $q_1(x)$ выпуклая вниз, $q_2(x)$ не выпуклая. На основе их для случая двух критериев, имеющих веса 0,6 и 0,4, сгенерированы операторы разного типа, они представлены на рис. 2, 4, 5 (см. третью сторону обложки). На этих рисунках выделены линии безразличия.

На рис. 6 изображены ФРЗП операторов, приведенных на рис. 2, 4, 5. Понятно, что для аддитивного оператора ФРЗП линейная. На рис. 5 представлен оператор с одинаковыми невыпуклыми

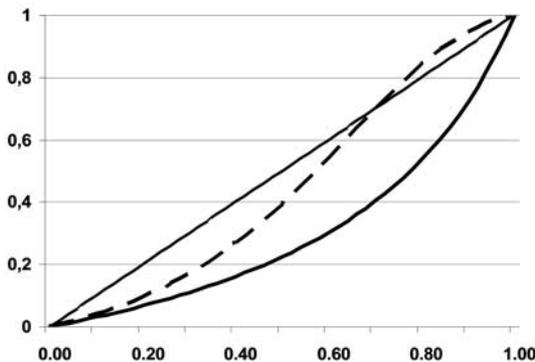


Рис. 3. Генерирующие функции:
— $q_1(x)$, - - - $q_2(x)$, — $q(x) = x$

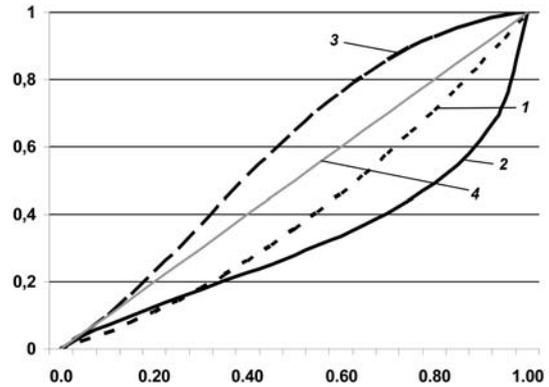


Рис. 6. ФРЗП операторов, приведенных на рисунках:
1 — рис. 2; 2 — рис. 4; 3 — рис. 5; 4 — $q(x) = x$

генерирующими функциями. Несмотря на это в соответствии с утверждением 5 линии безразличия выпуклы вниз. ФРЗП операторов определяется в основном генерирующей функцией $\psi(x)$.

2. Примеры генерирующих функций и операторов агрегирования

Ранее уже определили аддитивный оператор (9) с линейными генерирующими функциями. Рассмотрим вопросы генерации операторов на основе конкретных видов генерирующих функций. В качестве примеров рассмотрим выпуклые функции.

Так как на свойства оператора существенно влияет выпуклость генерирующих функций $\varphi(x)$ и $\psi(x)$, введем для них специальные параметры λ_φ и λ_ψ , которые будут определять степень выпуклости функции. Обозначим генерирующие функции через $q(x)$. Параметр λ_q определяет точку пересечения функции и линейной функции $1 - x$ (рис. 7). Координаты этой точки равны $(1 - \lambda_q; \lambda_q)$. Уравнение для определения λ_q имеет вид:

$$\lambda_q = q(1 - \lambda_q). \quad (10)$$

Если $\lambda_q < 0,5$, то функция выпукла вниз, при $\lambda_q > 0,5$ функция выпукла вверх, значению $\lambda_q = 0,5$ соответствует линейная функция. Каждый вид генерирующей функции $q(x)$ имеет свой параметр, который и определяется из уравнения (10).

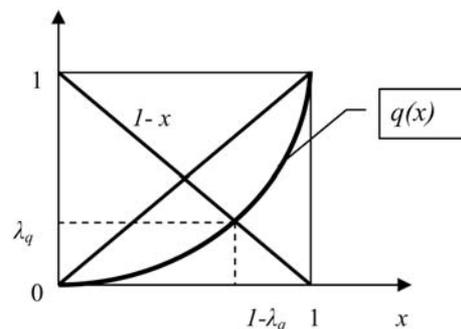


Рис. 7. Параметр выпуклости λ_q генерирующей функции

2.1. Степенная генерирующая функция

Степенная выпуклая генерирующая функция с параметром $p > 0$ имеет вид $q(x) = x^p$. Обратная функция $q^{-1}(u) = u^{1/p}$. Чтобы выразить p через λ_q , составим уравнение (10): $\lambda_q = (1 - \lambda_q)^p$, откуда получаем значение параметра $p = \ln(\lambda_q)/\ln(1 - \lambda_q)$.

Пусть заданы генерирующие функции $\varphi(x) = x^{p_\varphi}$ с параметром выпуклости λ_φ и $\psi(x) = x^{p_\psi}$ с параметром λ_ψ . На основе λ_φ и λ_ψ определяются соответственно p_φ и p_ψ .

Оператор типа h имеет вид:

$$h(u_1, \dots, u_m) = \left(\sum_{j=1}^m V_j u_j^{1/p_k} \right)^{p_\psi}.$$

ФРЗП степенного оператора $h(u) = u^{p_\psi/p_\varphi}$ определяется отношением p_ψ/p_φ .

Степенные операторы типов g_a и g_b с генерирующими функциями $q(x) = x^p$ совпадают.

Действительно, выражение для оператора g_a имеет вид

$$\begin{aligned} g_a(u_1, \dots, u_m) &= \left(\sum_{j=1}^m (kV_j u_j)^{1/p_\psi} \right)^{p_\psi} = \\ &= \left(k^{1/p_\psi} \sum_{j=1}^m (V_j u_j)^{1/p_\psi} \right)^{p_\psi}, \end{aligned}$$

а для оператора типа g_b

$$g_b(u_1, \dots, u_m) = \left(d \sum_{j=1}^m (V_j u_j)^{1/p_\psi} \right)^{p_\psi}.$$

Видим, что выражения для g_a и g_b совпадают, а коэффициент $d = k^{1/p_\psi}$.

ФРЗП степенного оператора $g(u) = u^{p_\psi/p_\varphi}$ совпадает с $h(u)$.

При равенстве λ_ψ и λ_φ , что эквивалентно $p_\psi = p_\varphi$, оператор имеет вид $g(u_1, \dots, u_m) = d^p \left(\sum_{j=1}^m (V_j u_j)^{1/p} \right)^p$,

что совпадает с L^p -метрикой, используемой в многокритериальных методах (см. [5]).

Утверждение 14. Для степенных операторов всех типов h , g_a и g_b с одинаковыми генерирующими функциями ФРЗП — линейная.

2.2. Показательные генерирующие функции

Показательную функцию b^x удобнее для дальнейшего рассмотрения представить в виде $q(x) = e^{ax}$.

Из приведенной показательной функции можно сгенерировать две генерирующие функции: выпуклую вниз $q_1(x) = \frac{e^{ax} - 1}{e^a - 1}$ и выпуклую вверх

$$q_2(x) = \frac{1 - e^{-bx}}{1 - e^{-b}},$$

имеющих параметры a и b , которые определяются по заданному параметру λ_q . Рассмотрим каждую из них.

С использованием уравнения (10) определяется параметр a ($\lambda_q < 0,5$): $\lambda_q = \frac{e^{a(1-\lambda_q)} - 1}{e^a - 1}$.

Обратная функция имеет вид:

$$q_1^{-1}(u) = \frac{1}{a} \ln((e^a - 1)u + 1).$$

Нетрудно показать, что параметр a функции $q_1(x)$ при $\lambda_{q1} < 0,5$ совпадает с параметром b функции $q_2(x)$ для $\lambda_{q2} = 1 - \lambda_{q1} > 0,5$. Обратная функция $q_2(x)$ имеет вид:

$$q_2^{-1}(u) = -\frac{1}{b} \ln(1 - (1 - e^{-b})u).$$

Выражение для показательного оператора типа h имеет вид:

$$\begin{aligned} h(u_1, \dots, u_m) &= \frac{1}{e^{a_\psi} - 1} \times \\ &\times \left\{ \left[\prod_{j=1}^m ((e^{a_\varphi} - 1)u_j + 1)^{V_j} \right]^{a_\psi/a_\varphi} - 1 \right\}. \end{aligned} \quad (11)$$

Если обозначить коэффициенты для генерирующих функций $C_\varphi = e^{a_\varphi} - 1$ и $C_\psi = e^{a_\psi} - 1$, то выражение (11) примет вид:

$$h(u_1, \dots, u_m) = \frac{1}{C_\psi} \left\{ \left[\prod_{j=1}^m (C_\varphi u_j + 1)^{V_j} \right]^{a_\psi/a_\varphi} - 1 \right\}. \quad (12)$$

Если $a_\psi = a_\varphi$, соответственно, $C_\psi = C_\varphi = C$, то (12) будет иметь вид:

$$\begin{aligned} h(u_1, \dots, u_m) &= \frac{1}{C} \left[\prod_{j=1}^m (Cu_j + 1)^{V_j} - 1 \right] \\ \text{или } Ch(u_1, \dots, u_m) + 1 &= \prod_{j=1}^m (Cu_j + 1)^{V_j}. \end{aligned}$$

Для оператора g_a с использованием коэффициентов C_φ и C_ψ оператор имеет вид:

$$g_a(u_1, \dots, u_m) = \frac{1}{C_\psi} \left\{ \left[\prod_{j=1}^m (C_\varphi k V_j u_j + 1) \right]^{a_\psi/a_\varphi} - 1 \right\}. \quad (13)$$

При $a_\psi = a_\varphi$ (13) будет иметь вид

$$Cg_a(u_1, \dots, u_m) + 1 = \prod_{j=1}^m (CkV_j u_j + 1). \quad (14)$$

На рис. 3 генерирующая функция $q_1(x)$ показательная с параметром $\lambda = 0,35$, соответственно на рис. 2 приведен оператор (14).

Выражение (14) совпадает с формулой мультипликативной функции полезности [1, стр. 281]:

$$Cu(u_1, \dots, u_m) + 1 = \prod_{j=1}^m (CW_j u_j + 1). \quad (15)$$

В мультипликативной функции полезности $W_j = u(u_j^+, \bar{u}_j^-)$, а в соответствии с утверждением 11 $W_j = kV_j$. Как отмечалось ранее, при $\varphi(x)$ выпуклой

вниз, $k < 1$ и тогда $\sum_{j=1}^m W_j = k < 1$.

Если в качестве генерирующих функций использовать $q(x) = \frac{1 - e^{-bx}}{1 - e^{-b}}$ (выпуклую вверх), вместо выражения (14) получим

$$1 - Cg_a(u_1, \dots, u_m) = \prod_{j=1}^m (1 - CkV_j u_j), \quad (16)$$

причем $k > 1$ для выпуклой вверх $\varphi(x)$, значит

$$\sum_{j=1}^m W_j = k > 1. \quad (17)$$

Выражение (16) также совпадает с формулой мультипликативной функции полезности (15), так как при условии (17) масштабный коэффициент мультипликативной функции меньше нуля, а в операторах агрегирования $C > 0$ и $k > 0$.

Следовательно, мультипликативная функция полезности есть оператор типа g_a , сформированный на основе показательных генерирующих функций при $\lambda_\psi = \lambda_\varphi$.

Для оператора g_b выражение имеет вид:

$$g_b(u_1, \dots, u_m) = \frac{1}{C_\psi} \left\{ \left[\prod_{j=1}^m (C_\varphi V_j u_j + 1) \right]^{d a_\psi/a_\varphi} - 1 \right\}.$$

В случае равенства a_ψ и a_φ , соответственно $C_\psi = C_\varphi = C$ и оператор g_b запишется в виде

$$Cg_b(u_1, \dots, u_m) + 1 = \left[\prod_{j=1}^m (C V_j u_j + 1) \right]^d.$$

Рассмотренные два вида генерирующих функций иллюстрируют многообразие возможных операторов. Можно использовать и другие выпуклые функции в качестве генерирующих. Кроме того, для генерации операторов можно использовать в качестве генерирующих функций $\varphi(x)$, $\psi(x)$ разные пары видов функций, например, взять $\varphi(x)$ степенную, а $\psi(x)$ показательную.

Возникает вопрос, как из этого многообразия операторов выбрать тот, который следует использовать для агрегирования критериев в конкретной задаче. Этому вопросу посвящен следующий раздел.

3. Идентификация операторов агрегирования

Виды генерирующих функций и их параметры определяются МЗВК и ФРЗП, поэтому для их идентификации от ЛПР следует получить оценки МЗВК и ФРЗП.

Для оценки МЗВК формируется анкета по форме табл. 1, в которой $u_j^+ = 1$ и $u_j^- = 0$, нумерация критериев соответствует порядку их весов по убыванию: $V_1 > V_2 > V_3$.

Число псевдообъектов определяется числом вершин гиперкуба 2^m за вычетом двух эталонов. Оценки ЛПР следует проверить на выполнение аксиом 1—4 и утверждений 1—3, касающихся предпочтений. Для случая трех критериев должны выполняться

Таблица 1
Опросная таблица для оценки лицом, принимающим решения, МЗВК оператора для трех критериев

Оцениваемые объекты	Веса критериев			Оценки ЛПР МЗВК
	V_1	V_2	V_3	
	u_1	u_2	u_3	
Объект эталон B^+	u_1^+	u_2^+	u_3^+	1,0
Псевдообъект B^1	u_1^+	u_2^-	u_3^-	W_1
Псевдообъект B^2	u_1^-	u_2^+	u_3^-	W_2
Псевдообъект B^3	u_1^-	u_2^-	u_3^+	W_3
Псевдообъект B^4	u_1^+	u_2^+	u_3^-	W_{12}
Псевдообъект B^5	u_1^+	u_2^-	u_3^+	W_{13}
Псевдообъект B^6	u_1^-	u_2^+	u_3^+	W_{23}
Объект эталон B^-	u_1^-	u_2^-	u_3^-	0,0

Таблица 2

Форма таблицы для оценки лицом, принимающим решение, ФРЗП оператора

Оцениваемые объекты	Значения критериев			Оценки ЛПР ФРЗП
	u_1	u_2	u_3	
Объект эталон B^+	1,0	1,0	1,0	1,0
Псевдообъект B^1	0,8	0,8	0,8	$R^1(h(0,8))$
Псевдообъект B^2	0,7	0,7	0,7	$R^2(h(0,7))$
...
Псевдообъект B^n	0,2	0,2	0,2	$R^n(h(0,3))$
Объект эталон B^-	0,0	0,0	0,0	0,0

следующие условия: $0 < W_j < 1$; $0 < W_{j,s} < 1$; порядок W_j совпадает с порядком весов V_j ; $W_{1,2} > W_{1,3} > W_{2,3}$; $W_1 < W_{1,3}$; $W_2 < W_{2,3}$; $W_3 < W_{2,3}$. Если оценки МЗВК не отвечают аксиомам, следует обсудить с ЛПР выявленные противоречия и, в первую очередь, выполнение аксиомы 2 о неизменности весов критериев во всей области определения критериев.

Для оценки функции равных значений переменных тоже формируется табличная анкета, где в каждой строке значения критериев принимают одинаковое значение (табл. 2).

Следует сказать, что оценивание псевдообъектов в обеих таблицах ЛПР выполняет в одной шкале интервалов, так как в обеих таблицах заданы два одинаковых эталонных объекта.

Идентификацию генерирующих функций следует начать с анализа полученных МЗВК и ФРЗП. Рассмотрим сначала случаи, которые позволяют идентифицировать или сузить поиск генерирующих функций.

Случай 1: а) МЗВК — линейная функция весов критериев; $W_j = V_j$, $W_{j,s} = V_j + V_s, \dots$; б) ФРЗП — тоже линейная функция $R^i = u^i$. Этот случай соответствует аддитивному оператору (9).

Случай 2: а) МЗВК — линейная функция весов критериев; $W_j = V_j$; $W_{j,s} = V_j + V_s, \dots$; б) ФРЗП — нелинейная. Тогда имеет место оператор типа h с линейной генерирующей функцией $\psi(x) = x$, а ФРЗП определяет генерирующую функцию $\varphi(x)$.

Случай 3: а) МЗВК является нелинейной монотонной весовой функцией, причем $W_j = \psi(v_j)$, $W_{j,s} = \psi(v_j + v_s), \dots$; б) ФРЗП — линейная функция $R^i = u^i$. В соответствии с утверждением 10 условие (а) определяет тип h оператора, при этом МЗВК определяет генерирующую функцию $\psi(x)$. В соответствии с утверждением 8 выполнение условия (б) для оператора типа h означает, что $\varphi(x) = \psi(x)$. Вывод о типе h оператора следует дополнительно проверить независимостью ФРЗП от весов (см. утверждение 7).

Случай 4: а) часть МЗВК W_j является линейной функцией $W_j = kV_j$; б) ФРЗП — линейная функция $R^i = u^i$. На основании утверждения 11 из условия (а) можно говорить, что имеет место оператор типа g_a с одинаковыми генерирующими функциями. Из (б) с учетом утверждения 14 следует, что оператор типа g_a степенной.

Случай 5: а) часть МЗВК W_j является линейной функцией $W_j = kV_j$; б) ФРЗП — нелинейная функция. На основании утверждения 11 из условия (а) можно говорить, что оператор типа g_a с одинаковыми генерирующими функциями.

Случай 6: а) оценки единой весовой функции оператора немонотонны или не однозначны. Учитывая замечание 2, условие (а) означает, что оператор типа g_a или g_b .

В общем случае для определения видов генерирующих функций и их параметров λ_φ и λ_ψ необходимо, перебирая все виды генерирующих функций и изменяя параметры λ_φ и λ_ψ , выбрать $\varphi(x)$, $\psi(x)$ такие, которые обеспечивают минимальное отличие МЗВК и ФРЗП оператора от оценок W_j , $W_{j,s}, \dots$ и R^i . Полный перебор проводится ввиду нелинейности операторов агрегирования, соответственно, возможности нескольких локальных экстремумов меры S^2 (см. ниже).

В качестве меры отличия значений оператора от оценок МЗВК используем $S^2(\text{МЗВК}) = \frac{1}{n'} \sum_{t=1}^{n'} \left(\psi \left[\sum_{j=1}^m \varphi^{-1}(kV_j u_j^t) \right] - W^t \right)^2$, где n' — количество данных во множестве МЗВК, аналогично для

ФРЗП: $S^2(\text{ФРЗП}) = \frac{1}{n} \sum_{s=1}^n \left(\psi \left[\sum_{j=1}^m \varphi^{-1}(kV_j u_j^s) \right] - R^s \right)^2$.

Выражения записаны для типа оператора g_a , для других типов операторов они такие же. Критерии $S^2(\text{МЗВК})$ и $S^2(\text{ФРЗП})$ вычисляются из расчета на одну точку множества, поэтому общий критерий $S^2 = S^2(\text{МЗВК}) + S^2(\text{ФРЗП})$ в равной степени учитывает отклонения от обоих множеств. Можно использовать и другие меры отличия экспертных оценок МЗВК и ФРЗП и операторов.

Чтобы иметь возможность подбирать генерирующие функции, которые бы обеспечили небольшие значения $S^2(\text{МЗВК})$ и $S^2(\text{ФРЗП})$, необходимо разнообразие генерирующих функций. В разд. 2 рассмотрены два вида функций в качестве примеров, для решения практических задач число видов $\psi(x)$ и $\varphi(x)$ следует расширить.

На практике при получении оценок МЗВК по форме табл. 1 у ЛПР могут возникать трудности, связанные с тем, что псевдообъекты неправдоподобны (значения критериев принимают значения

Заключение

$u_j^+ = 1$ или $u_j^- = 0$). Чтобы приблизить псевдо-объекты к реальным, ребро куба следует изменить, использовать в качестве u_j^+ значение меньше 1, а в качестве u_j^- больше 0, например, $u_j^+ = 0,7$ и $u_j^- = 0,3$.

При переходе к новому ребру куба необходимо контролировать единство шкалы интервалов для оценок МЗВК и ФРЗП. Единство шкалы определяется следующим образом. После перехода к новым u_j^+ и u_j^- для эталонных объектов B^+ и B^- в табл. 1 оценки проставляет сам ЛПР. В табл. 2 должны присутствовать B^+ и B^- , причем их оценки должны совпадать в обеих таблицах. Одинаковые значения этих оценок обеспечивает единство шкалы оценки МЗВК и ФРЗП. Если оценки объектов B^+ и B^- не совпадают, надо обратить внимание ЛПР на это, так как расхождение может быть объяснено только его невнимательностью или недобросовестностью.

Отметим, что при переходе к ребру с $u_j^+ < 1$ и $u_j^- > 0$, оператор типа h в вершине, определяемой эталонным объектом B^- , принимает значение, равное ФРЗП $h(u^-) = \psi(\varphi^{-1}(u^-))$, а для B^+ $h(u^+) = \psi(\varphi^{-1}(u^+))$, откуда получаем:

$$\varphi^{-1}(u^-) = \psi^{-1}(h(u^-)) \text{ и } \varphi^{-1}(u^+) = \psi^{-1}(h(u^+)). \quad (18)$$

При переходе к новому ребру куба утверждения 10 и 11, касающиеся МЗВК, не выполняются, но вместо утверждения 10 выполняется следующее.

Утверждение 15. Для операторов типа h МЗВК с ребром куба ($u_j^+ < 1$ и $u_j^- > 0$) является функцией от весов, которая определяет генерирующую функцию $\psi(x)$.

Действительно, значения оператора в вершинах куба равны $h(v_j) = \psi(\varphi^{-1}(u_j^+)V_j + \varphi^{-1}(u_j^-)(1 - V_j))$, а с учетом (18) получаем $h(v_j) = \psi(\psi^{-1}(h(u_j^+))V_j + \psi^{-1}(h(u_j^-))(1 - V_j))$, не зависящую от $\varphi(x)$. Аналогично получим $h(v_j, v_s) = \psi(\psi^{-1}(h(u_j^+))(V_j + V_s) + \psi^{-1}(h(u_j^-))(1 - (V_j + V_s)))$.

Предложен подход к агрегированию критериев в МКЗ, основанный на аппроксимации предпочтений ЛПР.

Сформулирована система аксиом предпочтений ЛПР, включающая: а) монотонность предпочтений; б) веса критериев, отражающие важность критериев; в) единственность объектов с минимальным и максимальным предпочтениями.

Разработан аппарат генерации операторов агрегирования критериев. Все операторы обеспечивают выполнение аксиом предпочтения. Известные функции полезности являются частными случаями операторов агрегирования.

Предложена процедура опроса ЛПР, позволяющая выявить его индивидуальные предпочтения: оценку весов критериев, МЗВК и ФРЗП.

Разработана процедура идентификации оператора на основе результатов опроса ЛПР по оценкам весов критериев, МЗВК и ФРЗП. Процедура заключается в подборе оператора, который наиболее точно аппроксимирует индивидуальные предпочтения ЛПР.

Важно отметить, что введенные понятия, связанные с генерацией операторов агрегирования, имеют содержательную интерпретацию, определяют свойства предпочтений. Это позволяет предметно обсуждать и анализировать с ЛПР полученные операторы агрегирования.

Список литературы

1. Кини Р. Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения: Пер. с англ. М.: Радио и связь, 1981. 560 с.
2. Ларичев О. И. Теория и методы принятия решений: учеб. М.: Логос, 2002. 392 с.
3. Brans J. P., Vincke Ph. A preference ranking organization method: the PROMETHEE method // Management science. 1985. Vol. 31. P. 647–656.
4. Pastun M., Leysen J. Construction an outranking relation with ORESTE // Math. Comput. modeling. 1989. Vol. 12, N 10–11. P. 1255–1268.
5. Opricovich S., Gho Hshiong Tzeng. Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS // European Journal Operation Research. 2004. Vol. 156, N 2. P. 445–455.
6. Kahraman C. Fuzzy multi-criteria decision making: theory and applications with recent developments. New York: Springer, 2008. 591 p.
7. Елгаренко Е. А. Оценка и выбор решений по многим критериям. М.: Изд-во МИФИ, 1995. 112 с.
8. Рокафеллар Р. Выпуклый анализ: Пер. с англ. М.: Мир, 1973. 470 с.

Б. Г. Кухаренко, канд. физ.-мат. наук,
ст. науч. сотр., вед. науч. сотр.,

Институт машиноведения РАН, г. Москва,
e-mail: kukharenko@imash.ru,

Д. И. Пономарев, аспирант,

Московский физико-технический институт (ГУ),
e-mail: ponomarev-102@mail.ru

Нелинейная байесовская фильтрация многомерных временных рядов

Описан метод идентификации нелинейной системы на основе временных рядов, обеспечивающий фильтрацию этих рядов. Алгоритм ожидания и максимизации правдоподобия итеративно оценивает множество состояний локально линеаризованной нелинейной системы как Гауссово множество данных. Нелинейная регрессия этих множеств использует Гауссовы радиальные базисные функции. Оценки параметров определяются аналитически. Инициализация алгоритма выполняется на основе факторного анализа и обучения линейной динамической системы. Эксперименты по нелинейной байесовской фильтрации используют многомерные временные ряды и пространства состояний большой размерности.

Ключевые слова: многомерные временные ряды, сглаживатель Рауха, нелинейные системы, алгоритм ожидания и максимизации правдоподобия, Гауссовы множества данных, радиальные базисные функции

Введение

В настоящей статье описывается метод идентификации динамической системы на основе наблюдаемых зашумленных временных рядов, обеспечивающий фильтрацию этих временных рядов (т. е. определяющий скрытые состояния системы). Основными моделями для оценки состояния являются линейные динамические системы с Гауссовыми шумами, поскольку для них разработан точный и эффективный метод определения скрытых состояний. Пусть $\mathbf{X} = \mathbf{X}[\overline{1}, \overline{L}; \overline{1}, \overline{N}]$ — многомерный временной ряд для вектора состояния размерности $L \times N$, $\mathbf{U} = \mathbf{U}[\overline{1}, \overline{L}; \overline{1}, \overline{N}]$ — управляющее воздействие, а $\mathbf{Y} = \mathbf{Y}[\overline{1}, \overline{M}; \overline{1}, \overline{N}]$ — многомерный наблюдаемый временной ряд измерений этого вектора состояний размерности $M \times N$, причем в общем случае $L \neq M$ (например, некоторые компоненты вектора состояния \mathbf{X} не измеряются). Как шум состояния $\mathbf{W} = \mathbf{W}[\overline{1}, \overline{L}; \overline{1}, \overline{N}]$, так и шум измерений $\mathbf{V} = \mathbf{V}[\overline{1}, \overline{M}; \overline{1}, \overline{N}]$ — многомерные вре-

менные ряды для случайных переменных с нулевым средним, распределенных по нормальному закону с ковариационными матрицами \mathbf{Q} и \mathbf{R} соответственно. Исходное распределение состояния в начальный момент времени также предполагается Гауссовым со средним $\boldsymbol{\pi}[1]$ и вариацией $\mathbf{V}[1]$. Линейные инвариантные относительно времени динамические системы описываются двумя уравнениями для векторов-столбцов $\mathbf{x}[k] = \mathbf{X}[\overline{1}, \overline{L}; k]$, $\mathbf{y}[k] = \mathbf{Y}[\overline{1}, \overline{M}; k]$, $\mathbf{u}[k] = \mathbf{U}[\overline{1}, \overline{L}; k]$, $\mathbf{v}[k] = \mathbf{V}[\overline{1}, \overline{M}; k]$ и $\mathbf{w}[k] = \mathbf{W}[\overline{1}, \overline{L}; k]$ (индекс k представляет дискретное время):

$$\left. \begin{aligned} \mathbf{x}[k+1] &= \mathbf{F} \cdot \mathbf{x}[k] + \mathbf{A} \cdot \mathbf{u}[k] + \mathbf{w}[k]; \\ \mathbf{y}[k] &= \mathbf{G} \cdot \mathbf{x}[k] + \mathbf{B} \cdot \mathbf{u}[k] + \mathbf{v}[k], \end{aligned} \right\} \quad (1)$$

где \mathbf{F} — матрица переходов; \mathbf{G} — матрица измерений; \mathbf{A} , \mathbf{B} — матрицы управления.

Если линейная модель (1) динамической системы известна, то в качестве средства фильтрации временных рядов используется фильтр Калмана [1]. Он обеспечивает оптимальную оценку состояния динамической системы на основе последовательности наблюдений ее выхода. Перечислим три ключевых момента для фильтра Калмана. Во-первых, фильтр Калмана — это метод для реализации правила Байеса. Во-вторых, отсутствует необходимость обращения матрицы переходов \mathbf{F} или матрицы измерений \mathbf{G} , поскольку используются нормализуемые распределения скрытых состояний \mathbf{X} . Это дает уравнения фильтра Калмана. В-третьих, процедуры оценки состояния реализуются рекурсивно. Апостериорная оценка на предыдущем временном шаге преобразуется через динамическую модель (1) и становится априорной для текущего временного шага. Затем эта априорная оценка конвертируется в новую априорную оценку на основе текущего наблюдения. Алгоритм вычисления апостериорного среднего и ковариации состояния $\mathbf{x}[k]$ при некоторой заданной последовательности наблюдений состоит из двух частей: прямой рекурсии, которая использует наблюдения $\{\mathbf{y}[1], \dots, \mathbf{y}[k]\}$ (т. е. фильтра Калмана), и обратной рекурсии, которая использует наблюдения $\{\mathbf{y}[N], \mathbf{y}[N-1], \dots, \mathbf{y}[k+1]\}$. Комбинированная прямая и обратная рекурсия называются сглаживателем Рауха [1].

Предположение о знании линейной модели (1) не является существенным ограничением, поскольку алгоритм ожидания и максимизации правдоподобия — Expectation-Maximization (EM) algorithm — оценивает параметры динамических моделей (1) при наличии скрытых состояний \mathbf{X} . Обозначим все параметры модели (1) матриц \mathbf{A} , \mathbf{B} , \mathbf{F} и \mathbf{G} , а также статистические характеристики \mathbf{R} , \mathbf{Q} ,

$\pi[1]$ и $\mathbf{V}[1]$ посредством вектора $\boldsymbol{\theta}$. В теории линейной байесовской фильтрации состояния в последовательные (дискретные) моменты времени на основе модели (1) объединяются в Гауссову случайную переменную с ковариацией \mathbf{Q} шума состояния [2, 3]. Аналогичная комбинация формируется на основе модели (1) для одновременных векторов состояния и измерения системы [2, 3]. Состояние системы (1) зависит только от предыдущего состояния $\mathbf{x}[k+1] \leftarrow \mathbf{x}[k]$, и имеется текущая связь входа $\mathbf{x}[k]$, $\mathbf{u}[k]$ с выходом $\mathbf{y}[k]$. Отсюда следует свойство Марковости всех условных вероятностей. Поэтому логарифм совместной вероятности (правдоподобия) $P(\mathbf{X}, \mathbf{Y})$ является квадратичной формой векторов-столбцов состояния и измерения системы. Поскольку $\log(P(\mathbf{X}, \mathbf{Y}))$ зависит от ненаблюдаемых состояний \mathbf{X} , он применяется условно по отношению к наблюдаемым временным рядам \mathbf{Y} , т. е. в EM-алгоритме используется логарифм правдоподобия $\mathcal{L} = E[\log(P(\mathbf{X}, \mathbf{Y}|\mathbf{Y}))]$. Поэтому \mathcal{L} зависит от трех условных средних (ожиданий) $E[\mathbf{x}[k]|\mathbf{y}]$, $E[\mathbf{x}[k] \cdot \mathbf{x}[k]^T|\mathbf{y}]$ и $E[\mathbf{x}[k] \cdot \mathbf{x}[k-1]^T|\mathbf{y}]$, где τ — символ транспонирования. E-шаг (EM-алгоритма) использует сглаживатель Рауха для оценки скрытых состояний \mathbf{X} при заданных последовательностях наблюдений \mathbf{Y} , входов \mathbf{U} и фиксированных значениях параметров $\boldsymbol{\theta}$. M-шаг включает максимизацию правдоподобия как функцию $\boldsymbol{\theta}$, используя оценку состояний \mathbf{X} на E-шаге посредством сглаживателя Рауха. Для моделей нелинейных систем с не Гауссовым шумом оценка состояния на основе известных входов и наблюдаемых выходов оказывается более сложной [4, 5].

Алгоритм ожидания и максимизации правдоподобия для нелинейной системы

Нелинейная динамическая система в дискретном времени описывает эволюцию состояния $\mathbf{x}[k+1] \leftarrow \mathbf{x}[k]$ на одном временном шаге и текущую связь входа $\mathbf{x}[k]$, $\mathbf{u}[k]$ с выходом $\mathbf{y}[k]$:

$$\left. \begin{aligned} \mathbf{x}[k+1] &= \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k]) + \mathbf{w}[k]; \\ \mathbf{y}[k] &= \mathbf{g}(\mathbf{x}[k], \mathbf{u}[k]) + \mathbf{v}[k], \end{aligned} \right\} \quad (2)$$

где $\mathbf{w}[k]$ и $\mathbf{v}[k]$ — Гауссовы шумовые процессы с нулевым средним. Вектор состояния $\mathbf{x}[k]$ эволюционирует в соответствии с нелинейной, но стационарной Марковской динамикой, производимой входом $\mathbf{u}[k]$ в присутствии шума $\mathbf{w}[k]$. Выход $\mathbf{y}[k]$ — нелинейный с шумом, но стационарный и является функцией текущего состояния $\mathbf{x}[k]$ и текущего входа $\mathbf{u}[k]$. Нелинейные вектор-функции $\mathbf{f}(\cdot)$ и $\mathbf{g}(\cdot)$ предполагаются дифференцируемыми. Ниже на E-шаге (EM-алгоритма) для оценки приблизительного распределения скрытых состояний нелинейной системы (2) используется расширенный сглаживатель Рауха, а на M-шаге для нелинейной регрес-

сии вектор-функций $\mathbf{f}(\cdot)$ и $\mathbf{g}(\cdot)$ — радиальные базисные функции (radial basis function — RBF) [6, 7].

Два условных распределения вероятности

$$\left. \begin{aligned} P(\mathbf{x}[k]|\mathbf{u}[1], \dots, \mathbf{u}[N], \mathbf{y}[1], \dots, \mathbf{y}[N], k = \overline{1, N}); \\ P(\mathbf{x}[k], \mathbf{x}[k+1]|\mathbf{u}[1], \dots, \mathbf{u}[N], \mathbf{y}[1], \dots, \mathbf{y}[N]), \end{aligned} \right\} \quad (3)$$

$$k = \overline{1, (N-1)},$$

являются основными на E-шаге для определения последовательности скрытых состояний нелинейной системы (2) на основе последовательностей наблюдаемых входов $\{\mathbf{u}[1], \dots, \mathbf{u}[N]\}$ и выходов $\{\mathbf{y}[1], \dots, \mathbf{y}[N]\}$. Условные распределения (3) являются не Гауссовыми, поэтому уравнения вывода распределений не могут быть представлены в замкнутой форме. Более того, объемы вычислений растут экспоненциально с увеличением длины наблюдаемых временных рядов. Расширенный сглаживатель Рауха аппроксимирует стационарную нелинейную динамическую систему (2) нестационарной, но линейной системой [1]. Он применяет стандартный сглаживатель Рауха к локальной линеаризации нелинейной системы. В каждой точке $\bar{\mathbf{x}}$ в пространстве состояний \mathbf{x} производные вектор-функций $\mathbf{f}(\cdot)$ и $\mathbf{g}(\cdot)$ определяют матрицы

$$\mathbf{F}_{\bar{\mathbf{x}}} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x} = \bar{\mathbf{x}}} \quad \text{и} \quad \mathbf{G}_{\bar{\mathbf{x}}} = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x} = \bar{\mathbf{x}}}$$

соответственно. Уравнения (2) линеаризуются в окрестности $\bar{\mathbf{x}}[k]$ средней текущей отфильтрованной (а не сглаженной) оценки состояния $\mathbf{x}[k]$ (в момент времени k). Аналогично линеаризуется уравнение для выхода. Эти линеаризации дают уравнения

$$\left. \begin{aligned} \mathbf{x}[k+1] &\approx \mathbf{f}(\bar{\mathbf{x}}[k], \mathbf{u}[k]) + \\ &+ \mathbf{F}_{\bar{\mathbf{x}}[k]} \cdot (\mathbf{x}[k] - \bar{\mathbf{x}}[k]) + \mathbf{w}[k]; \\ \mathbf{y}[k] &\approx \mathbf{g}(\bar{\mathbf{x}}[k], \mathbf{u}[k]) + \\ &+ \mathbf{G}_{\bar{\mathbf{x}}[k]} \cdot (\mathbf{x}[k] - \bar{\mathbf{x}}[k]) + \mathbf{v}[k]. \end{aligned} \right\} \quad (4)$$

Если распределения шума и априорные распределения скрытого состояния при $k = 1$ Гауссовы, то в линеаризованной системе (4) условное распределение вероятности скрытого состояния в произвольный момент времени при заданной последовательности входов и выходов также будет Гауссовым. Таким образом, сглаживатель Рауха может использоваться на линеаризованной системе (4) для вывода этого условного распределения. В противоположность линейному сглаживателю Рауха в расширенном сглаживателе Рауха ошибка ковариации для оценки состояния и матрицы усиления Калмана зависит не только от наблюдаемых данных с текущим временным индексом. Более того, если система (4) стационарная, матрица усиления Калмана не сходится к значению, при котором гла-

живатель Рауха функционирует как оптимальный фильтр Винера в стационарном состоянии.

На М-шаге (ЕМ-алгоритма) проблема состоит в том, что вектор-функции $\mathbf{f}(\cdot)$ и $\mathbf{g}(\cdot)$ должны обучаться, используя неопределенные оценки состояния посредством сглаживателя Рауха [6, 7]. Это затрудняет применение стандартных методов регрессии. Рассмотрим оценку $\mathbf{f}(\cdot)$, которая принимает как вход $\mathbf{x}[k]$ и $\mathbf{u}[k]$, так и выход $\mathbf{x}[k+1]$. Для каждого k оцениваемое сглаживателем Рауха условное распределение является Гауссовым с полной ковариацией в пространстве $\{\mathbf{x}[k], \mathbf{x}[k+1]\}$. Таким образом, $\mathbf{f}(\cdot)$ приспособляется не к набору точек, а к смеси Гауссовых распределений с полной ковариацией (т. е. к Гауссовым множествам данных — Gaussian clouds of data). Следуя ЕМ-алгоритму, при оценке вектор-функции $\mathbf{f}(\cdot)$ эти Гауссовы распределения исключаются посредством интегрирования. Однако исключение Гауссовых распределений посредством интегрирования является нетривиальным практически для любой формы $\mathbf{f}(\cdot)$. Поэтому ниже нелинейности $\mathbf{f}(\cdot)$ и $\mathbf{g}(\cdot)$ представляются как соответствие набора Гауссовых радиальных базисных функций Гауссовым множествам данных.

Рассмотрим следующее нелинейное отображение входных векторов \mathbf{x} и \mathbf{u} на выходной вектор \mathbf{z} :

$$\mathbf{z} = \sum_{i=1}^I \mathbf{h}[i] \cdot \rho_i(\mathbf{x}) + \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} + \mathbf{b} + \mathbf{w}, \quad (5)$$

где \mathbf{w} — Гауссов шум с нулевым средним и ковариацией \mathbf{Q} . Параметрами отображения (5) являются: коэффициенты $\mathbf{h}[i]$ при скалярных радиальных базисных функциях $\rho_i(\mathbf{x})$, матрицы \mathbf{A} , \mathbf{B} и вектор \mathbf{b} смещения выхода. В формуле (5) используются Гауссовы радиальные базисные функции в пространстве векторов \mathbf{x} с центром $\mathbf{c}[i]$ и ковариационной матрицей $\mathbf{S}[i]$:

$$\rho_i(\mathbf{x}) = |2\pi\mathbf{S}[i]|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c}[i])^T(\mathbf{S}[i])^{-1}(\mathbf{x} - \mathbf{c}[i])\right), \quad (6)$$

где $|\mathbf{S}[i]|$ — детерминант матрицы $\mathbf{S}[i]$. Отображение (5) может использоваться несколькими способами для представления систем (4) в зависимости от того, какое из отображений вход—скрытые состояния—выход считается нелинейным. Три примера этого:

1) $\mathbf{f}(\cdot)$ представляется посредством отображения (5) с подстановками $\mathbf{x} \leftarrow \mathbf{x}[k]$, $\mathbf{u} \leftarrow \mathbf{u}[k]$ и $\mathbf{z} \leftarrow \mathbf{x}[k+1]$;

2) $\mathbf{f}(\cdot)$ использует подстановки $\mathbf{x} \leftarrow (\mathbf{x}[k], \mathbf{v}[k])$, $\mathbf{u} \leftarrow \mathbf{0}$ и $\mathbf{z} \leftarrow \mathbf{x}[k+1]$;

3) $\mathbf{g}(\cdot)$ использует подстановки $\mathbf{x} \leftarrow \mathbf{x}[k]$, $\mathbf{u} \leftarrow \mathbf{u}[k]$ и $\mathbf{z} \leftarrow \mathbf{y}[k]$.

Поскольку набор данных $\{\mathbf{x}, \mathbf{z}, \mathbf{u}\}$ представляется смесью Гауссовых распределений, эта смесь Гауссовых распределений аналитически исключается посредством интегрирования, чтобы обеспечить соответствие RBF-модели (5). Распределение данных имеет вид

$$P(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \frac{1}{J} \sum_j P_j(\mathbf{x}, \mathbf{z}) \cdot \delta(\mathbf{u} - \mathbf{u}[j]).$$

То есть наблюдаются образцы переменной \mathbf{u} , и каждый имеет пару в виде множества данных с Гауссовым распределением $P_j(\mathbf{x}, \mathbf{z}) \equiv \mathcal{N}(\mathbf{x}, \mathbf{z} | \boldsymbol{\mu}[j], \mathbf{C}[j])$ переменных (\mathbf{x}, \mathbf{z}) , где среднее $\boldsymbol{\mu}[j] = \{\boldsymbol{\mu}_x[j], \boldsymbol{\mu}_z[j]\}$ и ковариационная матрица

$$\mathbf{C}[j] = \begin{bmatrix} \mathbf{C}_{xx}[j] & \mathbf{C}_{xz}[j] \\ \mathbf{C}_{zx}[j] & \mathbf{C}_{zz}[j] \end{bmatrix}.$$

Пусть по определению

$$\bar{\mathbf{z}}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^I \mathbf{h}[i] \cdot \rho_i(\mathbf{x}) + \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} + \mathbf{b},$$

где $\boldsymbol{\theta} = [\mathbf{h}[1], \dots, \mathbf{h}[I], \mathbf{A}, \mathbf{B}, \mathbf{b}]$ — вектор параметров отображения (5). Поскольку в (5) шум \mathbf{w} — Гауссов с нулевым средним и ковариацией \mathbf{Q} , логарифм правдоподобия одной полностью наблюдаемой точки \mathbf{z} , полностью удовлетворяющей RBF-модели, имел бы вид

$$-\frac{1}{2}(\mathbf{z} - \bar{\mathbf{z}}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}))^T \mathbf{Q}^{-1}(\mathbf{z} - \bar{\mathbf{z}}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})) - \frac{1}{2} \log|\mathbf{Q}| + c, \quad (7)$$

где c — некоторая константа. Поскольку набор $\{\mathbf{x}, \mathbf{z}\}$ — это Гауссово множество данных, максимум ожидаемого логарифма правдоподобия RBF-модели (50) дается минимизацией проинтегрированной (со знаком минус) квадратичной формы:

$$\min_{\boldsymbol{\theta}, \mathbf{Q}} \left\{ \sum_j \iint_{\mathbf{xz}} P_j(\mathbf{x}, \mathbf{z}) \cdot (\mathbf{z} - \bar{\mathbf{z}}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}))^T \times \right. \\ \left. \times \mathbf{Q}^{-1}(\mathbf{z} - \bar{\mathbf{z}}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})) d\mathbf{x}d\mathbf{z} + J \log|\mathbf{Q}| \right\}. \quad (8)$$

Пусть угловые скобки $\langle \cdot \rangle_j$ обозначают ожидание по Гауссову распределению P_j .

Определим вектор переменных $\boldsymbol{\Phi} = [\rho_1(\mathbf{x}), \dots, \rho_I(\mathbf{x}), \mathbf{x}^T, \mathbf{u}^T, \mathbf{1}]^T$. С учетом тождества

$$(\mathbf{z} - \boldsymbol{\theta}\boldsymbol{\Phi})^T \mathbf{Q}^{-1}(\mathbf{z} - \boldsymbol{\theta}\boldsymbol{\Phi}) = \text{tr}(\mathbf{Q}^{-1}(\mathbf{z} - \boldsymbol{\theta}\boldsymbol{\Phi})(\mathbf{z} - \boldsymbol{\theta}\boldsymbol{\Phi})^T),$$

где $\text{tr}(\cdot)$ обозначает след (трек) матриц, формула (8) принимает вид

$$\min_{\boldsymbol{\theta}, \mathbf{Q}} \left\{ \text{tr} \left(\mathbf{Q}^{-1} \sum_j \langle (\mathbf{z} - \boldsymbol{\theta}\boldsymbol{\Phi})(\mathbf{z} - \boldsymbol{\theta}\boldsymbol{\Phi})^T \rangle_j \right) + J \log|\mathbf{Q}| \right\}. \quad (9)$$

Беря частную производную по θ^T и приравнявая ее нулю, получаем уравнения

$$\sum_j \langle (\mathbf{z} - \theta \Phi) \Phi^T \rangle_j = 0,$$

которые решаются относительно θ :

$$\theta = \left(\sum_j \langle \mathbf{z} \Phi^T \rangle_j \right) \left(\sum_j \langle \Phi \Phi^T \rangle_j \right)^{-1}. \quad (10)$$

Аналогично выражение для ковариации \mathbf{Q} имеет вид

$$\mathbf{Q} = \frac{1}{J} \left(\sum_j \langle \mathbf{z} \mathbf{z}^T \rangle_j - \theta \sum_j \langle \Phi \mathbf{z}^T \rangle_j \right). \quad (11)$$

То есть при известных ожиданиях в угловых скобках $\langle \cdot \rangle_j$ значения параметров оцениваются на основе аналитических формул (10)–(11).

Перечислим ожидания, которые необходимо вычислить, чтобы оценить θ и \mathbf{Q} по формулам (10)–(11): $\langle \mathbf{x} \rangle_j$, $\langle \mathbf{z} \rangle_j$, $\langle \mathbf{x} \mathbf{x}^T \rangle_j$, $\langle \mathbf{z} \mathbf{z}^T \rangle_j$, $\langle \mathbf{x} \mathbf{z}^T \rangle_j$, $\langle \rho_i(\mathbf{x}) \rangle_j$, $\langle \mathbf{x} \rho_i(\mathbf{x}) \rangle_j$, $\langle \mathbf{z} \rho_i(\mathbf{x}) \rangle_j$ и $\langle \rho_i(\mathbf{x}) \rho_l(\mathbf{x}) \rangle_j$. Ожидания выражаются через параметры Гауссовых распределений $\mathcal{P}_j(\mathbf{x}, \mathbf{z}) \equiv \mathcal{N}(\mathbf{x}, \mathbf{z} | \boldsymbol{\mu}[j], \mathbf{C}[j])$:

$$\begin{aligned} \langle \mathbf{x} \rangle_j &= \boldsymbol{\mu}_x[j]; \quad \langle \mathbf{z} \rangle_j = \boldsymbol{\mu}_z[j]; \\ \langle \mathbf{x} \mathbf{x}^T \rangle_j &= \boldsymbol{\mu}_x[j] (\boldsymbol{\mu}_x[j])^T + \mathbf{C}_{xx}[j]; \\ \langle \mathbf{z} \mathbf{z}^T \rangle_j &= \boldsymbol{\mu}_z[j] (\boldsymbol{\mu}_z[j])^T + \mathbf{C}_{zz}[j]; \\ \langle \mathbf{x} \mathbf{z}^T \rangle_j &= \boldsymbol{\mu}_x[j] (\boldsymbol{\mu}_z[j])^T + \mathbf{C}_{xz}[j]. \end{aligned}$$

При умножении Гауссовой RBF-функции $\rho_i(\mathbf{x})$ (6) на Гауссово распределение $\mathcal{P}_j(\mathbf{x}, \mathbf{z})$ получаются Гауссово распределение переменных (\mathbf{x}, \mathbf{z}) со средним

$$\boldsymbol{\mu}[i; j] = \mathbf{C}[i; j] \left((\mathbf{C}[j])^{-1} \boldsymbol{\mu}[j] + \begin{bmatrix} (\mathbf{S}[i])^{-1} \mathbf{c}[i] \\ 0 \end{bmatrix} \right)$$

и ковариацией

$$\mathbf{C}[i; j] = \left((\mathbf{C}[j])^{-1} + \begin{bmatrix} (\mathbf{S}[i])^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right)^{-1}$$

и дополнительный скаляр (учитывающий отсутствие нормализации)

$$\begin{aligned} \beta[i; j] &= (2\pi)^{-d_x/2} |\mathbf{S}[i]|^{-1/2} |\mathbf{C}[j]|^{-1/2} \times \\ &\times |\mathbf{C}[i; j]|^{1/2} \exp\left(-\frac{1}{2} \delta[i; j]\right), \end{aligned}$$

где

$$\begin{aligned} \delta[i; j] &= (\mathbf{c}[i])^T (\mathbf{S}[i])^{-1} \mathbf{c}[i] + (\boldsymbol{\mu}[j])^T (\mathbf{C}[j])^{-1} \boldsymbol{\mu}[j] - \\ &- (\boldsymbol{\mu}[i; j])^T (\mathbf{C}[i; j])^{-1} \boldsymbol{\mu}[i; j]. \end{aligned}$$

Используя $\beta[i; j]$, можно оценить ожидания $\langle \rho_i(\mathbf{x}) \rangle_j = \beta[i; j]$, $\langle \mathbf{x} \rho_i(\mathbf{x}) \rangle_j = \beta[i; j] \cdot \boldsymbol{\mu}_x[i; j]$, $\langle \mathbf{z} \rho_i(\mathbf{x}) \rangle_j = \beta[i; j] \cdot \boldsymbol{\mu}_z[i; j]$. Наконец, ожидание

$$\begin{aligned} \langle \rho_i(\mathbf{x}) \rho_l(\mathbf{x}) \rangle_j &= ((2\pi)^{-d_x} |\mathbf{C}[j]|^{-1/2} |\mathbf{S}[i]|^{-1/2} \times \\ &\times |\mathbf{S}[l]|^{-1/2} |\mathbf{C}[i; l; j]|^{1/2} \exp\left(-\frac{1}{2} \gamma[i; l; j]\right), \end{aligned}$$

где

$$\mathbf{C}[i; l; j] = \left((\mathbf{C}[j])^{-1} + \begin{bmatrix} (\mathbf{S}[i])^{-1} + (\mathbf{S}[l])^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right)^{-1};$$

$$\begin{aligned} \gamma[i; l; j] &= (\mathbf{c}[i])^T (\mathbf{S}[i])^{-1} \mathbf{c}[i] + (\mathbf{c}[l])^T (\mathbf{S}[l])^{-1} \mathbf{c}[l] + \\ &+ (\boldsymbol{\mu}[j])^T (\mathbf{C}[j])^{-1} \boldsymbol{\mu}[j] - \\ &- (\boldsymbol{\mu}[i; l; j])^T (\mathbf{C}[i; l; j])^{-1} \boldsymbol{\mu}[i; l; j]; \end{aligned}$$

$$\begin{aligned} \boldsymbol{\mu}[i; l; j] &= \mathbf{C}[i; l; j] \left((\mathbf{C}[j])^{-1} \boldsymbol{\mu}[j] + \right. \\ &\left. + \begin{bmatrix} (\mathbf{S}[i])^{-1} \mathbf{c}[i] + (\mathbf{S}[l])^{-1} \mathbf{c}[l] \\ 0 \end{bmatrix} \right). \end{aligned}$$

Практический успех EM-алгоритма зависит от его инициализации. Для систем с нелинейной динамикой, но линейной функцией выхода, инициализация использует факторный анализ по максимуму правдоподобия, обучаемый на наборе выходных наблюдений. Факторный анализ предполагает, что выходные переменные генерируются небольшим числом независимых Гауссовых скрытых состояний, и к каждой выходной переменной добавляется независимый Гауссов шум [8]. Матрица весов (называемая матрицей нагрузок) обучается посредством факторного анализа для инициализации матрицы наблюдений \mathbf{G} динамической системы (4). Это дает оценки состояния в каждый момент времени. Эти оценки используются в нелинейной регрессии на основе RBF-функций. Перед инициализацией нелинейной регрессии проводится обучение линейной системы.

Эксперименты на временных рядах показывают ограничения нелинейной байесовской фильтрации при использовании одномерных динамических моделей. В настоящей статье нелинейная байесовская фильтрация применяется для двумерных временных рядов, что позволяет использовать нелинейные динамические модели в пространстве состояний достаточно большой размерности $L \geq 4$. Это повышает эффективность нелинейной байесовской фильтрации. Если имеется только один наблюдаемый временной ряд, то, как и в случае линейной байесовской фильтрации, увеличение размерности наблюдаемого временного ряда обеспечивается добавлением его первой разности (возможно до-

бавление разностей более высоких порядков) [9]. Таким образом, дополнительную компоненту дает разность исходного временного ряда

$$\Delta y = y[\overline{2, N}] - y[\overline{1, (N-1)}]. \quad (12)$$

Численное дифференцирование временного ряда $y[\overline{1, N}]$ по формуле (12) делает его амплитудно-частотную характеристику более высокочастотной. После этого выполняется нелинейная байесовская фильтрация двумерного наблюдаемого временного ряда

$$Y = \{y; \Delta y\}^T. \quad (13)$$

Пример нелинейной байесовской фильтрации

Нелинейная байесовская фильтрация демонстрируется на примере записи y сигнала одной из трех осей прецизионного акселерометра, являющегося чувствительным элементом удаленного манипулятора (рис. 1, *a*). Первая разность наблюдаемой последовательности y (рис. 1, *б*) показывает, что сигнал удаленного манипулятора содержит значительный шум, который препятствует устойчивому функционированию управляемого им устройства. При использовании нелинейной модели (2) наблюдаемая последовательность y (рис. 1, *a*) позволяет определить одномерную последовательность x скрытых состояний. Для наглядности на рис. 2 для наблюдаемой последовательности y (см. рис. 1, *a*) при размерности пространства состояний $L = 1$ на фоне определенных посредством EM-алгоритма последовательностей пар скрытых состояний $\{x[k], x[k+1]\}$, $k = \overline{1, (N-1)}$, показана нелинейная регрессия Гауссова множества данных с использованием десяти радиальных базисных функций (6).

Для повышения эффективности нелинейной байесовской фильтрации сигнала удаленного манипулятора необходимо увеличить размерность наблюдаемой последовательности. Это достигается добавлением к наблюдаемой последовательности y (см. рис. 1, *a*) ее первой разности Δy (12), представленной на рис. 1, *б*. Двумерный временной ряд $Y = \{y; \Delta y\}$ обеспечивает байесовскую нелинейную фильтрацию при размерности пространства состояний $L = 4$. Определенные в результате байесовской фильтрации компоненты $x[i] = \{X[i; k], k = \overline{1, N}\}$, $i = \overline{1, 4}$, матрицы X состояний нелинейной модели (2) представлены на рис. 3, *a—г* соответственно.

Зависимость ожидаемого логарифма правдоподобия RBF-модели (5) для числа итераций EM-алгоритма представлена на рис. 4.

В качестве критерия близости компонент (рис. 3, *a—г*), определенных в результате нелинейной байесовской фильтрации, к исходному сигна-

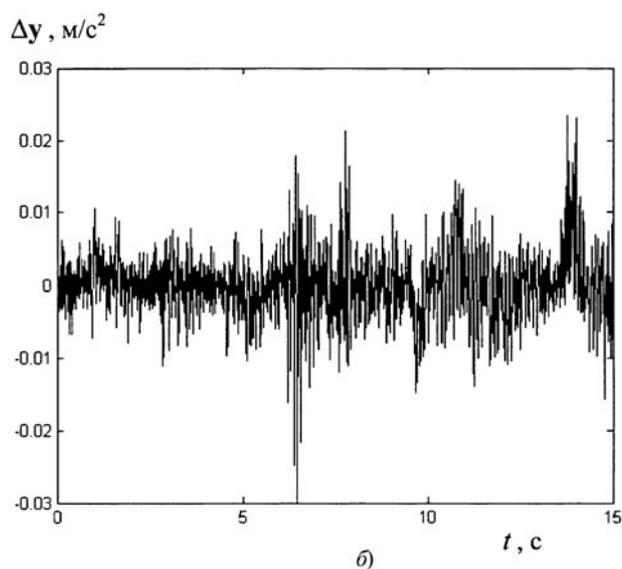
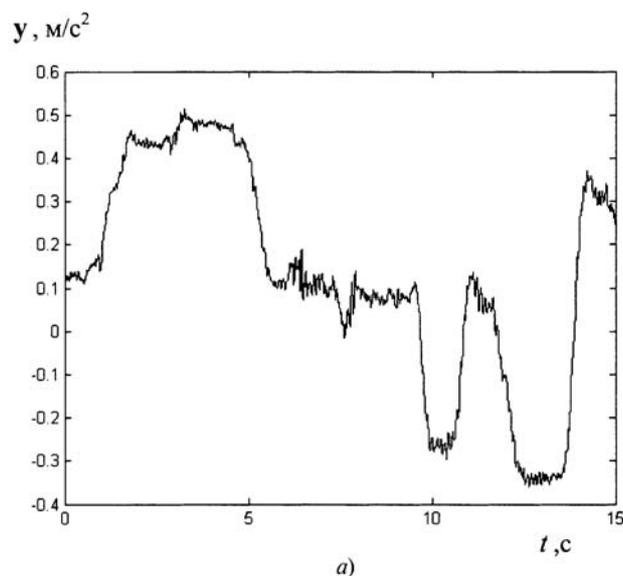


Рис. 1. Сигнал чувствительной оси прецизионного акселерометра (*a*); первая разность наблюдаемой последовательности y (*б*)

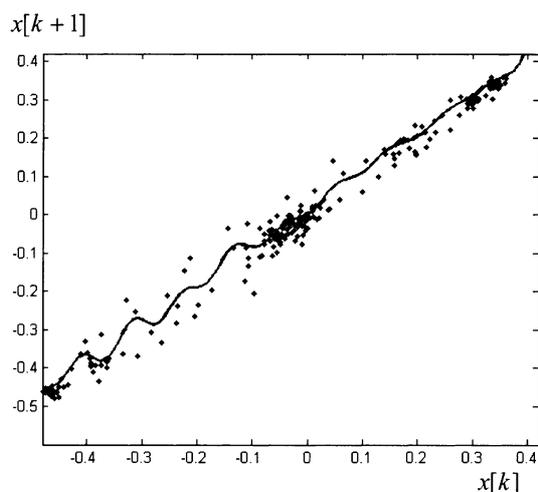


Рис. 2. Пары последовательных состояний (точки) и результат нелинейной регрессии (сплошная линия)

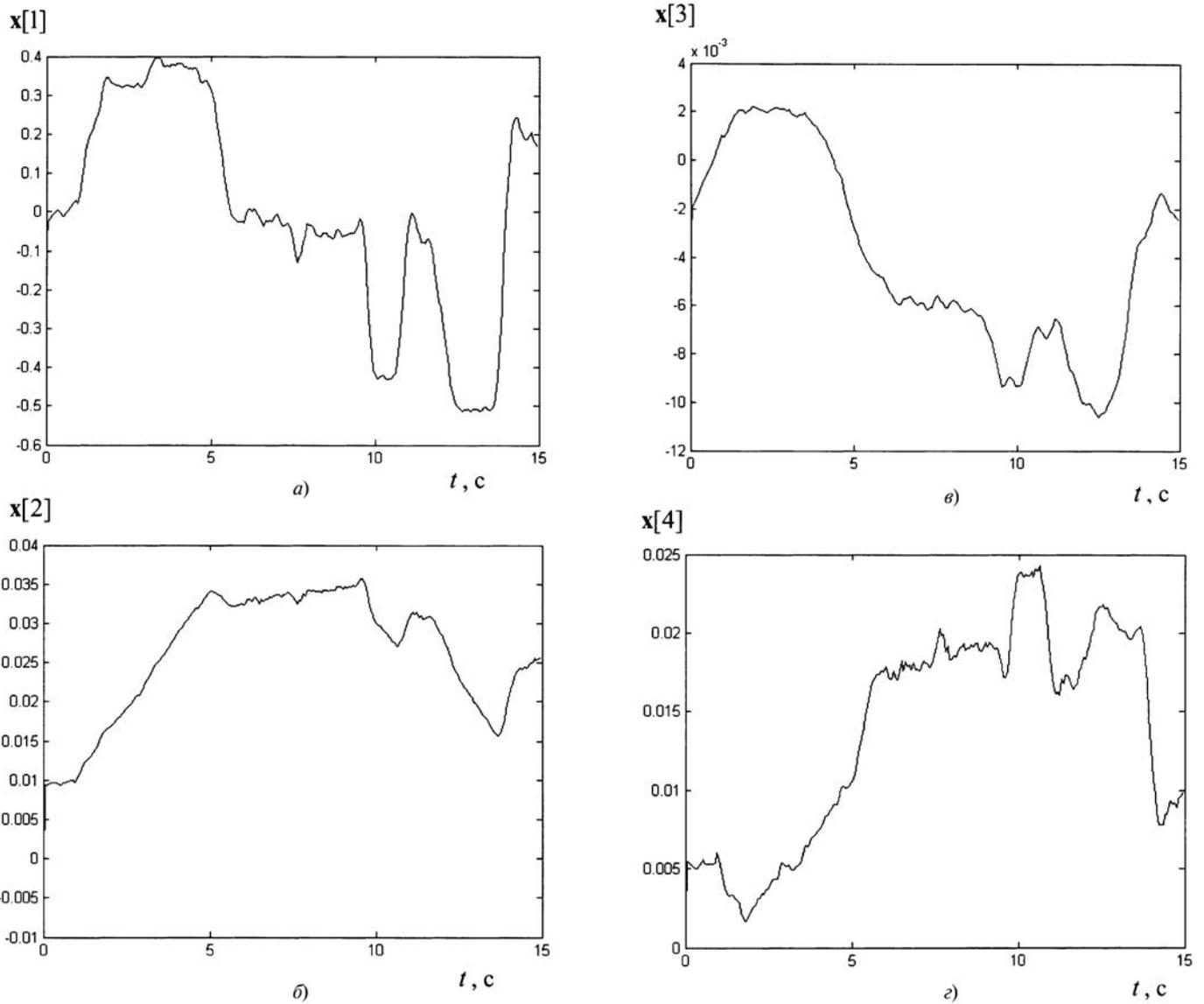


Рис. 3. Компоненты матрицы X состояний нелинейной динамической модели

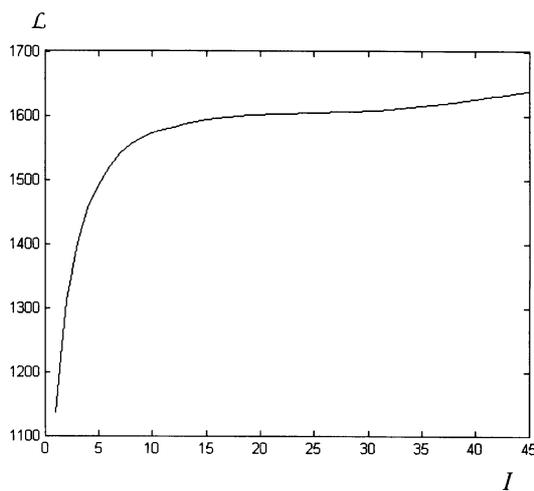


Рис. 4. Зависимость логарифма правдоподобия от числа итераций

лу (см. рис. 1, а) используется их кросскорреляция без запаздывания. Для временных рядов x и y кросскорреляция без запаздывания определяется как усредненная оценка

$$c_{xy}(0) = E\{(x[i] - E\{x[i]\})(y[i] - E\{y[i]\})\}, \quad (14)$$

где $E\{x[i]\}$, $E\{y[i]\}$ — оценки (ожидания) средних значений этих временных рядов [10]. Значения кросскорреляции (14) каждой из компонент $x[i]$, $i = \overline{1, 4}$, (рис. 3, а–г) и исходного сигнала y (см. рис. 1, а) приведены в таблице.

Компонента состояния	x[1]	x[2]	x[3]	x[4]
$c_{xy}(0)$	16,9732	6,7003	-6,4847	1,5776

Таким образом, компонента $x[1]$ на рис. 3, *a* наиболее коррелирована с исходным сигналом y на рис. 1, *a*. Подобие этих временных зависимостей подтверждается визуально. Поэтому компонента $x[1]$ на рис. 3, *a* представляет результат нелинейной байесовской фильтрации исходного сигнала y на рис. 1, *a*.

Список литературы

1. **Simon D.** Optimal State Estimation: Kalman, H_∞ and Non-linear Approaches. Hoboken, New Jersey: John Wiley & Sons, Inc. 2006.
2. **Shumway R. H., Stoffer D. S.** An approach to time series smoothing and forecasting using the EM algorithm // Journal of Time Series Analysis. 1982. V. 3. N 4. P. 253–264.

3. **Shumway R., Stoffer D.** Dynamic linear models with switching // Journal of the American Statistical Association. 1992. V. 86. P. 763–769.
4. **Ghahramani Z., Roweis S.** Learning nonlinear dynamical systems using an EM algorithm / Kearns M. S., Solla S. A., Cohn D. A., eds. // Advances in Neural Information Processing Systems. V. 11. Cambridge, MA: The MIT Press. 1999. P. 599–605.
5. **Roweis S., Ghahramani Z.** Learning nonlinear dynamical systems using the Expectation-Maximization algorithm / Haykin S., ed. Kalman Filtering and Neural Networks. John Wiley & Sons. 2001. P. 175–220.
6. **Moody J., Darken C.** Fast learning in networks of locally-tuned processing units // Neural Computation. 1989. V. 1. P. 281–294.
7. **Broomhead D. S., Lowe D.** Multivariable functional interpolation and adaptive networks // Complex Systems. 1988. V. 2. P. 321–355.
8. **Roweis S., Ghahramani Z.** A unifying review of linear Gaussian models // Neural Computation. 1999. V. 11. N 2. P. 305–345.
9. **Кухаренко Б. Г.** Байесовская фильтрация в технологии спектрального анализа на основе быстрого преобразования Фурье // Информационные технологии. 2010. № 8. С. 36–42.
10. **Orfanidis S. J.** Optimum Signal Processing. An Introduction. 2nd Edition. Englewood Cliff, NJ: Prentice-Hall. 1996.

УДК 004.031.43

А. С. Филиппова, д-р техн. наук, проф.,
Уфимский государственный авиационный
технический университет,
Р. В. Фролов, зам. нач. отдела,
ООО РН-УфаНИПИнефть"
e-mail: frolovrv@mail.ru

О создании оперативных сетевых графиков работ бригад текущего и капитального ремонта скважин*

Рассматривается ресурсосберегающая проблема, систематически возникающая при эксплуатации нефтяных месторождений. Одним из факторов, повышающих реальную добычу нефти, является своевременный ремонт нефтяных скважин. Предлагается разрабатывать сетевые графики выполнения ремонтных работ и создать для их поддержки мобильную систему маршрутизации.

Ключевые слова: нефтяные месторождения, ремонт скважин, сетевой график, расписание, маршруты, движение бригад

Введение

Одним из основных видов деятельности при эксплуатации нефтяных месторождений является поддержание работы эксплуатационного фонда скважин, связанной с проведением ремонтных геологических мероприятий. В настоящее время в большинстве нефтяных компаний на территории

России применяется сервисный подход в организации ремонтных работ на скважинах. Подход основан на передаче выполнения работ внешним организациям, которые осуществляют ремонтные работы, по договорам подряда. Расчет расценок за проведение работ осуществляется исходя из нормированной продолжительности работ.

Одной из основных задач является корректное составление сетевых графиков работ большого числа ремонтных бригад. Автоматизация процесса позволит существенно повысить качество планирования и снизить производственные издержки за счет расширения перечня анализируемых показателей.

В настоящее время создание графиков работ бригад на добывающих нефтяных предприятиях проводится вручную силами круглосуточных централизованных служб.

Автоматизированные системы, содержащие различные элементы планирования движения бригад, эксплуатируются в нефтяной компании "ТНК-ВР" и ОАО "Лукойл" [1]. Однако данные системы предполагают ручное назначение работ скважинам, что не решает общей проблемы анализа широкого перечня исходных показателей.

В качестве примера может служить автоматизированная система, построенная на базе математической модели организации ремонтных работ. Она была разработана на кафедре моделирования и управления процессами нефтегазодобычи Тюменского государственного нефтегазового университета [2]. Система позволяет провести моделирование деятельности бригад и оценить достаточность их числа с точки зрения экономической эффективности, но не решает задачи оперативного планирования.

* Работа поддержана грантом Президента РФ господдержки ведущих научных школ РФ, грант РФНШ-65497.2010.9

1. Постановка задачи

Решение о назначении бригады на выполнение работы зависит от ряда параметров, а именно — длительности и сложности переезда на новое место, наличия необходимых приборов и оборудования в оснастке бригады, готовности скважины к ремонту на момент приезда бригады, планируемой длительности ремонта, показателей работы скважины до и после ремонта (дебит нефти, обводненность). Также решение о перемещении бригады зависит от утвержденного долгосрочного плана мероприятий.

Задача состоит из двух подзадач — подзадачи нахождения путей на графе и подзадачи распределения заданий по бригадам.

При нахождении путей на графе следует учитывать возможность выбора не только минимальных путей, но и более длинных путей, так как итоговая оценка зависит от затрат на перемещение бригады и эффекта от планируемых работ.

Среди критериев работы ремонтной бригады выделим основные, характеризующиеся следующими параметрами:

- протяженностью пути, проходимого бригадой;
- затратами на досрочное прекращение ремонта;
- затратами на возобновление незавершенного ремонта;
- затратами, связанными с простоем ремонтных бригад.

Многокритериальность задачи позволяет ввести общий затратный критерий оптимальности или использовать схему решения с выделением отдельных критериев, предоставив пользователю выбор одного из решений. Ситуация осложняется тем, что решение задачи путем составления полного перечня вариантов перестановок бригад с последующей оценкой каждого из вариантов и выбора наиболее подходящего, не представляется возможным в связи с большим числом вариантов. Задача может быть решена с использованием какого-либо эвристического алгоритма.

Наиболее перспективной представляется разработка алгоритма перебора большого числа случайных вариантов с использованием выбранной метаэвристики (1 + 1)-EA [3], так как позволяет быстро получить допустимое решение.

Общий эволюционный алгоритм, обозначаемый как $(\mu + \lambda)$ -EA, был предложен И. Рехенбергом в 1973 г. в связи с решением задач непрерывной оптимизации [4]. Общие схемы EA принято называть эволюционными стратегиями (ES). В общем алгоритме $(\mu + \lambda)$ -EA предполагается существование на каждой итерации λ особей. Оператор селекции S_μ определяется как $S_\mu(s_1, s_2, \dots, s_\lambda) = (s_{i_1}, s_{i_2}, \dots, s_{i_\mu})$, $0 < \mu < \lambda$, т. е. S_μ выбирает μ лучших из λ возможных потомков. Алгоритм $(\mu + \lambda)$ -EA — частный случай общего эволюцион-

ного алгоритма, когда μ и λ равны 1 (одна особь в популяции и один потомок). Его называют еще односточечным эволюционным алгоритмом. П. Борисовский и А. Еремеев сформулировали условия монотонности для мутации и воспроизведения, при выполнении которых стратегия (1 + 1)-EA оказывается лучшей среди других эволюционных процессов [3]. Доказать или опровергнуть выполнение монотонности для конкретного класса задач пока не удастся. Применяется экспериментально исследование алгоритмов. Подробно исследован этот алгоритм для задач раскроя—упаковки А. Филипповой [5] и для решения транспортной классической задачи П. Борисовским [6]. В этой работе впервые описан подход к применению стратегии односточечного эволюционного алгоритма для комплексной транспортировки и выполнения ремонтных работ.

2. Описание алгоритма

Для оптимизации процесса составления сетевых графиков работ был разработан программный модуль, предоставляющий пользователям возможные варианты сетевых графиков работ в оперативном режиме.

Программный модуль реализует следующий алгоритм.

Шаг 1. Подготовка информации.

Составляется полный перечень скважин, находящихся в ремонте, и скважин, подлежащих ремонту.

Шаг 2. Расчет стоимости.

Для каждой пары скважин выполняется расчет стоимости переезда ремонтной бригады. Для этого применяется алгоритм Дейкстры [7] поиска кратчайшего пути между двумя вершинами.

Шаг 3. Метка скважин.

Каждой скважине (вершине), на которой фактически проводится ремонт, выставляется признак охваченности ремонтом и длительность оставшихся ремонтных работ.

Шаг 4. Построение варианта движения.

4.1. Выбирается скважина, ремонт которой заканчивается раньше всего.

4.2. Планируется перемещение бригады на новую позицию (вершина помечается как охваченная ремонтом, ей выставляется оценочная продолжительность ремонта). Перемещение проводится на первую скважину в списке, не охваченную ремонтом.

4.3. Осуществляется циклическое выполнение п.п. 4.1 и 4.2.

4.4. Построение варианта завершается при обходе всех вершин, вариант сохраняется как текущий.

Шаг 5. Оценка варианта.

Проводится оценка текущего варианта и выбор рекордного.

5.1. Вариант оценивается как разница между прибылью от добычи отремонтированных скважин

за время от окончания ремонта до максимальной глубины планирования и суммарной стоимостью затрат на переезды бригад.

5.2. Оценка текущего варианта сравнивается со значением оценки рекордного; в случае, если оценка текущего варианта лучше оценки рекордного, текущий вариант сохраняется как рекордный. Для первой итерации текущий вариант сохраняется как рекордный, сравнение оценок не проводится.

Шаг 6. Построение текущего варианта.

Проводится построение следующего варианта на основе текущего. Для этого выполняется случайная замена перемещений на выбранном числе шагов (число заменяемых переходов задается).

Алгоритм (1 + 1)-EA в итоге перебирает различные варианты и позволяет выбрать лучший из них [3].

Шаг 7. Построение следующего варианта.

Построение вариантов осуществляется в течение заданного интервала времени. При изменении текущей ситуации (отказ скважины, фактическое завершение ремонтных работ) проводится перезапуск алгоритма, который осуществляет изменение в списке скважин. В любой момент времени по запросу пользователя предоставляется рекордный вариант.

3. Численный эксперимент

Численный эксперимент проводился на тестовых наборах данных. Число скважин и бригад было выбрано таким образом, чтобы задача могла быть решена полным перебором за приемлемое время (до двух—трех суток). На каждом тестовом наборе проводился полный перебор вариантов, при котором находился оптимальный вариант, и серия испытаний разработанного на базе (1 + 1)-EA алгоритма. Для параметров, используемых при построении графика движения, были взяты случайные значения в интервалах, типичных для реальной ситуации:

- дебит жидкости скважины 100 м³/сут;
- средняя продолжительность ремонтов — 6 суток;
- среднее время переезда 7 ч.

Полное число вариантов

$$N = (m + n - 1)! / (m - 1)!,$$

где m — число бригад; n — число скважин.

Реализация алгоритма с полным перебором показала производительность 100 000 вариантов в секунду, что позволило находить точное решение на наборах, содержащих до 12 скважин и трех бригад (43 589 145 600 вариантов). Полученные при испытаниях разработанного на базе (1 + 1)-EA алгоритма значения приведены в таблице.

Эксперименты показывают быстрое улучшение результата в самом начале работы алгоритма и выдают приемлемый вариант уже через 2 мин работы. Были проведены эксперименты, дающие аналогич-

Отклонение в % от оценки оптимального плана

Время, мин	Число итераций	Отклонение, %, от оптимума
0,1	600 000	25,0
0,5	3 000 000	20,0
1	6 000 000	8,0
2	12 000 000	4,0
5	30 000 000	2,0
10	60 000 000	1,5

ные результаты, в которых изменялись заложенные критерии оценки (минимизировалось время переездов бригад). Это показывает возможность расширения критериев оценки без ухудшения параметров.

Также были проведены эксперименты на других размерностях (11 скважин 3 бригады, 10 скважин 4 бригады, 10 скважин 3 бригады), которые показали слабое ухудшение результатов при увеличении числа скважин (ухудшение на 0,5—1 % при добавлении одной скважины).

Заключение

Разработка алгоритма перебора большого числа случайных вариантов с использованием выбранной метаэвристики и реализация данного решения дает удовлетворительные результаты, которые могут быть использованы в практической деятельности нефтяных компаний при планировании работ ремонтных бригад.

Прикладные решения, использующие описанные в статье алгоритмы, могут дать существенный эффект и позволят снизить операционные издержки в добыче нефти.

Список литературы

1. ЗАО "ОТ-ОИЛ" Разработка системы "GTMDB". URL: http://www.ot-oil.com/project_gtmdb.php
2. Мамчистова Е. И., Кучумов Р. Я. Оптимизация экономической эффективности организации ремонтных работ на скважинах. // Алгоритмизация и моделирование процессов разработки нефтегазовых месторождений: Сб. науч. трудов. Тюмень: Вектор-Бук, 2005. Вып. 2. С. 37—44.
3. Борисовский П. А., Еремеев А. В. О сравнении некоторых эволюционных алгоритмов // Автоматика и телемеханика. 2004. № 3. С. 3—9.
4. Rechenberg I. Evolutionsstrategie — Optimierung technischer Systeme nach Prinzipien der Biologischen Evolution / Rechenberg I. Stuttgart-Bad Gannstatt: Frommann-Holzboog, 1973. 169 p.
5. Филиппова А. С. Моделирование эволюционных алгоритмов решения задач прямоугольной упаковки на базе технологии блочных структур // Приложение. Информационные технологии. 2006. № 6. 32 с.
6. Борисовский П. А. Исследование эволюционных алгоритмов решения некоторых задач дискретной оптимизации. Дисс. ... на соискание ученой степени кандидата физ.-мат. наук. 2005. 109 с.
7. Ху Т. Ч., Шинг М. Т. Комбинаторные алгоритмы / Пер. с англ. Нижний Новгород: Изд-во Нижегородского гос. университета им. Н. И. Лобачевского. 2004. 330 с.

Е. М. Бронштейн, д-р физ.-мат. наук, проф.,

Т. А. Заико, аспирант,

Уфимский государственный авиационный

технический университет

e-mail: bro-efim@yandex.ru

Задача маршрутизации с запретами

Рассматривается задача маршрутизации, когда предусмотрены запреты на перевозку некоторых видов грузов отдельными транспортными средствами. Разработана соответствующая математическая модель.

Предложены эвристические алгоритмы решения, проведен вычислительный эксперимент для сравнительной оценки эффективности различных алгоритмов.

Ключевые слова: маршрутизация, транспортные средства, пункты потребления, математическое моделирование, эвристические алгоритмы

Введение

Задачи маршрутизации (в частности транспортной логистики) являются важным классом задач оптимизации как для теории, так и для практики. Наиболее известной задачей такого типа является задача маршрутизации транспортных средств — VRP (Vehicle Routing Problem) [1], обзоры по данной проблеме приведены в работах [2—4].

В данной работе рассматривается новая оптимизационная задача маршрутизации, в которой учитывается ассортимент перевозимых грузов. В частности, учитывается, что некоторые виды грузов запрещено перевозить тем или иным транспортным средством. Это ограничение выполняется для некоторых химических веществ, источников радиации, для определенных грузов (в частности продовольственных) предусматривается особый температурный режим и т. д.

Поставленная задача относится к классу дискретно-непрерывных нелинейных оптимизационных задач и является обобщением двух известных задач: задачи коммивояжера и задачи о загрузке рюкзака. Тем самым рассматриваемая задача является NP-сложной.

В работе приводится формализация поставленной задачи, намечены пути ее решения, сравнивается достаточно широкое семейство эвристических подходов к решению.

Математическая модель

Пусть на базе имеются грузы k видов. Грузы необходимо доставить в n пунктов потребления (ПП), в j -й ПП необходимо доставить заданное количество груза j -го вида. Доставка груза осуществляется m транспортными средствами (ТС), каждое из которых характеризуется грузоподъемностью, транспортными затратами на 1 км пути, а также указанием, какие виды грузов не разрешается перевозить данным ТС.

Требуется организовать доставку грузов таким образом, чтобы суммарные расходы на транспортировку были минимальными.

Введем следующие обозначения:

$N = \{1, \dots, n\}$, $K = \{1, \dots, k\}$, $M = \{1, \dots, m\}$ — номера ПП, виды грузов, ТС, соответственно; x_{ij}^p — масса груза i -го вида, перевозимого p -м ТС в j -й ПП ($i \in K, j \in N, p \in M$); $y_i^p = \sum_{j=1}^n x_{ij}^p$ — общая масса

груза i -го вида, перевозимого p -м ТС; S^p — грузоподъемность p -го ТС; L_{ij} — масса груза i -го вида, который необходимо доставить в j -й ПП ($i \in K,$

$j \in N$); $w_i = \sum_{j=1}^n L_{ij}$ — суммарная потребность в грузе

i -го вида; $A^p \subset K$ — множество видов грузов, недопустимых к перевозке p -м ТС; B^p — затраты на проезд p -го ТС на единицу расстояния; $c(U)$, ($U \subset N$) — минимальная из длин циклов, содержащих ПП из U и нулевой пункт (депо);

$F^p = \left\{ j \in N: \sum_{i=1}^k x_{ij}^p > 0 \right\}$ — множество ПП, в которые груз доставляется p -м ТС.

Задача имеет следующий вид.

Найти числа x_{ij}^p такие, что:

$$x_{ij}^p \geq 0, x_{ij}^p = 0 \text{ при } i \in A^p, j \in N; \quad (1)$$

$$\sum_{p=1}^m x_{ij}^p = L_{ij}; \quad (2)$$

$$\sum_{i=1}^k \sum_{j=1}^n x_{ij}^p \leq S^p; \quad (3)$$

$$\sum_{p=1}^m c(F^p) B^p \rightarrow \min. \quad (4)$$

Условие (1) отражает ограничения на использование ТС для перевозки определенных видов груза, условие (2) означает полное удовлетворение

потребностей во всех ПП, условие (3) — это ограничения грузоподъемностей ТС, целевая функция (4) — стоимость перевозок.

Рассмотрим следующую систему уравнений и неравенств:

$$\sum_{p=1}^m y_i^p = w_i; \quad (5)$$

$$\sum_{i=1}^k y_i^p \leq S^p; \quad (6)$$

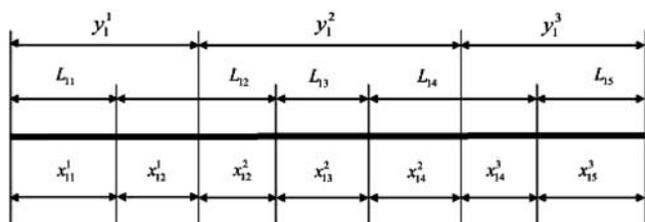
$$y_i^p \geq 0; \quad (7)$$

$$y_i^p = 0 \text{ при } i \in A^p. \quad (8)$$

Очевидное необходимое условие совместности системы (5)—(8) $\sum_{i=1}^k w_i \leq \sum_{p=1}^m S^p$, которое является

и достаточным в отсутствии условия (8), в общем случае таковым не является. Приведем соответствующий пример. Пусть имеются два ТС с грузоподъемностями 20 и 5 единиц, с помощью которых надо перевезти груз двух видов: 15 единиц первого и 10 второго. Пусть при этом первым ТС запрещено перевозить груз второго вида. Легко проверить, что организовать перевозку невозможно.

В то же время из совместности системы (5)—(8) следует допустимость задачи (1)—(4). Для доказательства рассмотрим какое-нибудь решение системы (5)—(8). Для каждого вида груза проведем построение, проиллюстрированное на рисунке для груза первого вида.



Формирование плана перевозок

Здесь нижние индексы величин x наследуются от L , а верхние — от y .

Тем самым определяется план перевозок: сколько груза того или иного вида каждое ТС должно доставить в каждый ПП.

Эвристический алгоритм решения задачи

Для решения задачи (1)—(4) использовался трехэтапный эвристический алгоритм.

На *первом этапе* в случае совместности найдем какое-нибудь решение системы (5)—(8). Для этого к системе добавляем линейную целевую функцию

$$\sum_{i=1}^k \sum_{p=1}^m \mu_i^p y_i^p \rightarrow \max, \text{ где } \mu_i^p \text{ — числа, последова-$$

тельно генерируемые датчиком псевдослучайных чисел из интервала $[-1; 1]$, и решаем полученную задачу линейного программирования.

Затем упорядочиваем ТС и ПП.

Мы использовали следующие способы упорядочения ТС:

1) по убыванию удельных стоимостей перевозок B^p/S^p ;

2) по возрастанию удельных стоимостей перевозок B^p/S^p ;

3) по убыванию грузоподъемности S^p ;

4) по возрастанию грузоподъемности S^p ;

5) по возрастанию допустимого набора грузов.

ПП упорядочивали следующими способами.

А. Индуктивно: первым является ПП, ближайший к депо, следующим является ПП, расстояние от которого до объединения уже занумерованных ПП (включая депо) минимальное среди занумерованных ПП. Это упорядочение реализовано на основе алгоритма Прима построения минимального остова дерева неориентированного связного графа.

В. Аналогично А, но с использованием правила ближайшего соседа.

С. По возрастанию суммарной массы заказа.

Д. По убыванию суммарной массы заказа.

Е. По возрастанию отношения суммарной массы заказа к расстоянию до начального пункта.

Ф. По убыванию отношения суммарной массы заказа к расстоянию до начального пункта.

Всего получаем 30 вариантов.

На *втором этапе* по значениям y_i^p и L_{ij} определяются значения x_{ij}^p . Для каждого вида груза выполняется построение плана перевозок (см. рисунок), в ходе которого для каждого ТС определяется множество ПП, которые ему необходимо посетить. При этом поочередно используются все 30 способов упорядочения.

На *третьем этапе* для каждого ТС решается задача коммивояжера, это позволяет вычислить

$$\text{значение целевой функции } \sum_{p=1}^m c(F^p)B^p. \text{ Естественно, третий этап оказался самым трудоемким.}$$

Численный эксперимент

Генерация исходных данных проводилась с учетом следующих условий:

1. Число видов грузов, ПП, ТС — псевдослучайные числа из интервала $[1, 10]$. Аналогичным

Способы упорядочения ПП	Способы упорядочения ТС				
	1	2	3	4	5
A	1; 15;	13; 14'	3; 13; 14; 15; 17;	1;	1; 2; 8; 15
B	1; 4; 15;	14; 16;	3; 14; 15; 16;	1;	1; 2; 15;
C	19;	14; 16; 19; 20	14; 16; 20;	15; 25;	
D	1; 5; 6; 11; 22; 24;	12; 14;	7; 12; 14; 15;	1; 5; 6; 11; 18; 21	1; 2; 9; 10; 18
E		14; 15; 16; 20;	7; 14; 16; 20	15; 25;	2;
F	1; 15; 19;	13; 14; 19;	13; 14; 15; 23;	1; 15	1; 8; 15

образом генерируется матрица потребностей ПП в грузах различного вида, матрица расстояний между ПП (и депо), грузоподъемность ТС, значения затрат на проезд ТС на единицу расстояния.

2. Возможность ТС перевозить тот или иной груз назначается путем псевдослучайной генерации нулей и единиц для каждого вида груза (1 — возможность перевозки груза данного вида, 0 — запрет). При этом каждое ТС может перевозить хотя бы один вид груза, и каждый вид груза, в свою очередь, может быть перевезен хотя бы одним ТС.

Всего генерировалось 25 задач следующих размерностей:

1 — 4×3×3 8 — 6×6×6 15 — 6×4×3 22 — 3×4×6
 2 — 4×4×4 9 — 6×6×6 16 — 7×4×3 23 — 3×4×7
 3 — 5×5×5 10 — 7×7×7 17 — 8×4×3 24 — 3×4×8
 4 — 5×5×5 11 — 4×4×4 18 — 9×4×3 25 — 3×4×9
 5 — 5×5×5 12 — 4×4×4 19 — 3×4×3
 6 — 5×5×5 13 — 4×4×3 20 — 3×4×4
 7 — 5×5×5 14 — 5×4×3 21 — 3×4×5

Здесь первая компонента — число различных видов грузов, вторая — число ПП, третья — число ТС.

В таблице приведены результаты численного эксперимента. Отмечены сочетания способов упорядочения, при которых достигается лучший результат для каждой задачи. Повторы означают, что полученные при различных упорядочениях значения целевой функции совпадают.

Результаты численного эксперимента позволяют сделать вывод о том, что в среднем лучший результат достигается при упорядочении пунктов потребления по убыванию суммарной массы заказа, а транспортных средств — по возрастанию удель-

ной стоимости перевозки, по убыванию грузоподъемности и по возрастанию допустимого набора грузов.

Заключение

Рассмотрена задача маршрутизации, в которой предусмотрены ограничения на перевозку некоторых видов грузов теми или иными транспортными средствами. Для решения задачи предложена трехэтапная эвристическая процедура, в основе которой упорядочение транспортных средств и пунктов потребления. В результате проведенного вычислительного эксперимента выявлены способы упорядочения пунктов потребления и транспортных средств, при сочетании которых в среднем достигается лучший результат.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект 10-06-00001) и гранта Президента Российской Федерации для государственной поддержки ведущих научных школ Российской Федерации № НШ-65497.2010.9.

Список литературы

1. **Dantzig G. B., Ramser J. H.** The Truck Dispatching Problem // Management Science. 1959. V. 6, N 1. P. 80—91.
2. **Parragh S., Doerner K., Hartl R.** A survey on pickup and delivery problems. Part I: Transportations between customers and depot // Journal fur Betriebswirtschaft. 2008. V. 58. P. 21—51.
3. **Parragh S., Doerner K., Hartl R.** A survey on pickup and delivery problems. Part II: Transportations between customers and depot // Journal fur Betriebswirtschaft. 2008. V. 58. P. 82—117.
4. **Бронштейн Е. М., Заико Т. А.** Детерминированные оптимизационные задачи транспортной логистики // Автоматика и телемеханика. 2010. № 6.

В. В. Семи́н, аспирант,
Московский государственный институт
электроники и математики (МГИЭМ),
e-mail: Noric-12@yandex.ru

Моделирование поведения стохастической триангуляции на квазистационарных решетках

Задача представления топологических моделей в виде триангулированных кубических комплексов связана с характером и поведением триангуляции в n -мерном пространстве. Основой для анализа данных особенностей может служить фундаментальная статистическая информация, позволяющая проводить синтез характеристик особенностей триангуляции в модельных пространствах различной размерности. В статье предложен подход к решению этой задачи на примере анализа статистических особенностей стохастической триангуляции для пространств размерности не выше чем 5, с помощью бинарного кодирования, а также свойства перестроек примитивной триангуляции.

Ключевые слова: триангуляция, симплекс, комплекс, единичный куб, топология, решетка

Введение

Основой для представления различных геометрико-топологических моделей в приложениях, связанных с задачами гидро- и газодинамики, с расчетами с помощью методов конечных элементов и объемов, в приложениях САПР являются решетчатые структуры. Как правило, такие объекты и модели достаточно сложны, что затрудняет их анализ и синтез. Поэтому необходимо иметь строгое формальное представление подобных объектов или их составляющих, позволяющее однозначно определить структуру объекта и легко модифицируемое в процессе работы с ним.

Несмотря на разнообразие требований и особенностей применения подобных структур данных в различных областях, существует ряд общих черт.

1. В больших научно-технических проектах (климатический прогноз, аэродинамика летательных аппаратов, задачи обработки томограмм, конструирование новых лекарственных средств и т. д.) размеры сеток и различных геометрико-топологических структур в настоящее время достигают диапазона $10^8 \dots 10^9$ узлов, а актуальными считаются размеры из диапазона $10^{10} \dots 10^{13}$ узлов.

Примером подобного научно-технического проекта может служить глобальная модель атмосферы MITgcm [1], успешно применяемая, но требующая больших ресурсов для моделирования структур и объектов указанных размеров.

2. В самих структурах возрастает необходимость учета все большего числа особенностей геометрико-топологического характера.

3. В ряде случаев требуются оперативные, часто в процессе непосредственно компьютерной обработки, преобразования структур с сохранением определенных инвариантов.

С этими положениями связаны проблемы непосредственного представления соответствующих данных, удобного для обработки и визуализации с помощью ЭВМ, а также вопросы хранения и преобразования с учетом специфических свойств (таких как связность, непрерывность и симметрия). В связи с этим важным является наличие программно-аппаратной поддержки для решения перечисленных выше задач. Представляется, что роль такой поддержки может играть топологический процессор [2].

Для понимания принципов работы топологического процессора необходимо знать поведение и структуру триангуляции в n -мерном пространстве, основой которой служит триангулированный куб соответствующей размерности. Важным для понимания также является динамика поведения симплицальных комплексов с жестко зафиксированными узлами решетки.

Основой для анализа данных особенностей может служить статистическая информация о характерных особенностях триангуляции в рассматриваемых пространствах различной размерности.

1. Основные понятия

Рассмотрим множество всех целых точек n -мерного евклидова пространства Z^n , т. е. точек с целочисленными координатами. На нем определим решеточную структуру как множество всех вершин из Z^2 , Z^3 и Z^4 множества простых ребер-отрезков V_p нормы p , инцидентных вершинам, где $p = \max|X_i|$, $i = 1 \dots n$, если $\chi = (X_1, \dots, X_n)$ — вектор, описывающий ребро. Если потребовать, чтобы евклидова длина простого ребра не превышала единицы, т. е. $p = 1$, то для пространств размерности не выше 4 это будет означать построение решетки соответствующей размерности. Определение решеточной структуры, данное выше, следует логике определений, приводимых в работе [2].

Для каждой такой решетки, покрывающей пространство соответствующей размерности, можно задать ее триангуляцию как проведение диагонали в каждой двумерной грани. Полученная таким образом триангуляция тривиальна для пространства размерности 2. Но для пространств высшей размерности триангуляция может называться правильной, только если она приводит к разбиению всей структуры на множество правильно располо-

женных симплексов. Основой такой решетки может служить n -мерный куб I^n .

Процесс правильной триангуляции I^n приводит к выделению ограниченного набора конфигураций, которые служат основой для триангуляции I^{n+1} . При такой модели триангуляции достигается эргодичность поведения триангуляции во всем пространстве. Что дает возможность на основе поведения триангуляции I^n сделать вывод о поведении триангуляции всего пространства.

2. Триангуляция в двумерном пространстве

Для построения триангуляции двумерной решетки с узлами в целых точках может послужить модель *плотного плоского покрытия* P_1^2 .

Определение 1. Будем называть *плотным плоским покрытием* P_1^2 множество $P = \{1 \leq x \leq X, 1 \leq y \leq Y\}$ с простыми ребрами индекса k , и таким соединением ребрами-отрезками всех целых точек из P , когда нет ни одного пересечения ребер-отрезков и нельзя добавить ни одного дополнительного ребра без пересечения. Здесь индекс k означает, что для любого ребра-отрезка с координатами $\{(x_0, y_0), (x_1, y_1)\}$ выполняются неравенства $|x_0 - x_1| \leq k$, $|y_0 - y_1| \leq k$ и $|x_0 - x_1|, |y_0 - y_1|$ взаимно просты.

Далее будем рассматривать покрытие P_1^2 (рис. 1). Очевидно, что такая триангуляция тривиальна. В данной решетке можно выделить всего два вида ячеек — клетку с диагональю, ориентированной влево, и клетку с диагональю, ориентированной вправо. Такая структура легко описывается бинарным кодом, состоящим из 0 и 1. Единица записывается в соответствующую ячейку, если диагональ имеет правую ориентацию, и записывается ноль, если диагональ имеет левую ориентацию.

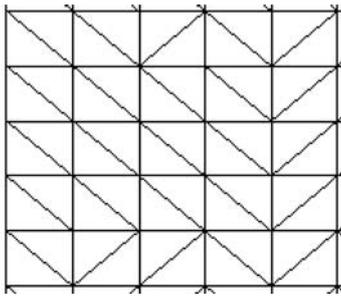


Рис. 1. Покрытие P_1^2

3. Триангуляция в трехмерном пространстве

Рассмотрим множество (Z^3, V_1) , где вершины являются узлами трехмерной целочисленной решетки, а множество простых ребер-отрезков имеет норму $p = 1$, образуя множество ребер решетки, инцидентных вершинам. Основным элементом полученной структуры является единичный куб

I^3 с вершинами в целых точках и простыми ребрами-отрезками в качестве одномерных граней.

Определение 2. Определим *произвольную триангуляцию (стохастическую)* на множестве (Z^3, V_1) как триангуляцию двумерных граней трехмерной решетки, получаемую проведением диагонали в соответствующей грани.

Введем понятие правильной триангуляции, основывающееся на требовании правильной расположения симплексов по Понтрягину [3].

Определение 3. Триангуляцию будем называть *правильной*, если она порождает разбиение выбранного пространства на множество правильно расположенных симплексов по Понтрягину.

Очевидно, что произвольная триангуляция (Z^3, V_1) не всегда будет являться правильной, т. е. будут существовать такие триангуляции двумерных граней I^3 , которые не порождают разбиения куба на правильно расположенные симплексы.

Опишем формально требование правильного разбиения I^3 на правильно расположенные симплексы.

Рассмотрим вектор диагональных индексов вершин $x = (x_0, x_1, x_2, x_3)$, где x_i — целое неотрицательное число — число вершин, имеющих число инцидентных диагоналей соответственно 0, 1, 2 и 3.

Тогда необходимым условием выполнения требования триангуляции I^3 будет выполнение двух диофантовых уравнений:

$$\begin{cases} \sum_{i=0}^3 x_i = 8 \\ \sum_{i=0}^3 ix_i = 12 \end{cases} \quad (1)$$

Вектор диагональных индексов вершин будет являться решением системы, однозначно характеризующим тип триангуляции I^3 .

Очевидно рассматриваемой системе, с учетом требования разбиения куба на правильно расположенные симплексы, удовлетворяют следующие наборы решений:

- 1) $x = (0, 6, 0, 2)$ — тип 1,
- 2) $x = (1, 3, 3, 1)$ — тип 2,
- 3) $x = (2, 0, 6, 0)$ — тип 3,
- 4) $x = (2, 2, 2, 2)$ — тип 4,
- 5) $x = (4, 0, 0, 4)$ — тип 5.

Таким образом, соответствие конфигурации куба одному из выше перечисленных пяти типов является достаточным условием выполнения требования правильной триангуляции.

На рис. 2 приведено графическое изображение всех пяти типов.

Типы неконгруэнтны и других правильных триангуляций I^3 не существует. Полная триангуляция I^3 для типов 1—4 завершается проведением большой диагонали внутри куба. При этом положение ее однозначно для типов 1, 2 и 4. Тип 3 до-

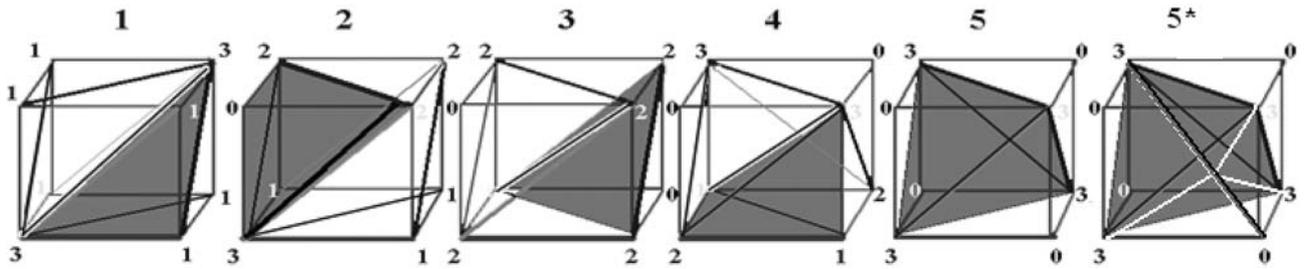


Рис. 2. Типы правильной триангуляции I^3

пускает три различных положения большой диагонали, в то время как триангуляция граней остается неизменной. При триангуляции типа 5 большая диагональ внутри куба не проводится.

Однако системе (1) удовлетворяет также решение $(0, 4, 4, 0)$. Данный вектор порождает неоднозначную триангуляцию двумерных граней I^3 , что приводит к двум новым типам 6 и 7 (рис. 3), которые не удовлетворяют требованию разбиения куба на правильно расположенные симплексы. Эти типы будем считать неполно триангулированными.

Цифрами в вершинах кубов на рис. 3 обозначено число диагональных ребер, инцидентных данной вершине.

Исходя из приведенных выше определений и рассуждений легко видеть, что стохастическая триангуляция (Z^3, V_1) порождает разбиение кубов решетки в соответствии с одним из семи типов.

Полную информацию о поведении триангуляции в данном множестве можно получить, построив

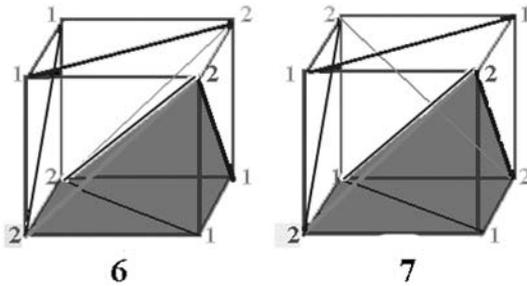


Рис. 3. Типы неправильной триангуляции I^3

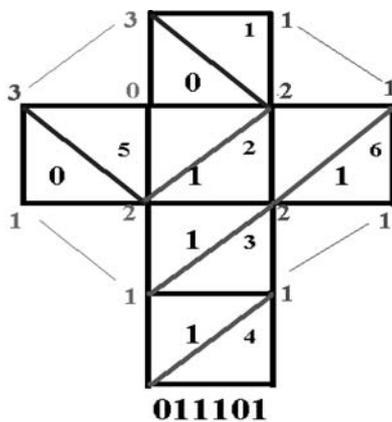


Рис. 4. Развертка куба с двоичным кодом (011101)

распределение по типам, встречающимся при произвольной триангуляции I^3 .

Для этого закодируем каждую возможную конфигурацию шестью двоичными разрядами в соответствии с числом двумерных граней куба. Не уменьшая общности рассуждений, зафиксируем порядок развертки куба. Будем считать, что 0 записывается в соответствующий разряд, если диагональ грани имеет левую ориентацию, и 1 — в противном случае. Тогда каждой такой развертке может быть поставлен в соответствие вектор $a = (a_1, \dots, a_6)$, $a \in \{0, 1\}$. Пример подобной развертки показан на рис. 4.

Применяя такой способ кодирования, легко определить число всех возможных конфигураций I^3 . Оно равно 2^6 . При этом каждый такой код однозначно соответствует одному из шести решений системы (1). Очевидно, что разные бинарные коды могут соответствовать одному типу триангуляции. Поэтому необходима общая система кодирования на основе числа диагональных ребер, инцидентных вершинам куба.

Примером такой системы кодирования может служить набор решений системы (1), далее система кодирования будет расширена для случая пространства размерности 4.

Рассмотрим множество элементарных событий Ω с элементами — кодами развертки. Мощность Ω составляет 2^6 . Определим также сигма-алгебру A случайных событий как множество событий появления куба определенного типа, тогда $A = \{A_1, \dots, A_7\}$. Вероятностную меру P определим стандартным образом, $P(\Omega) = 1$.

Получим искомые вероятности простым перечислением всех событий. Среди всех возможных исходов (64 элементарных события) только 48 являются случаями появления правильно триангулированного куба, а 18 — неполно триангулированного. Таким образом, общая вероятность получить правильную триангуляцию I^3 составляет $p = 0,71875$, в то время как на долю неполной триангуляции приходится $q = 0,28125$.

При стохастической триангуляции это означает, что именно с такими вероятностями можно получить ту или иную триангуляцию куба в выбранной ячейке трехмерной решетки. Однако полное представление о поведении триангуляции на множестве (Z^3, V_1) может дать только распределение по типам кубов с указанием веса (вероятности собы-

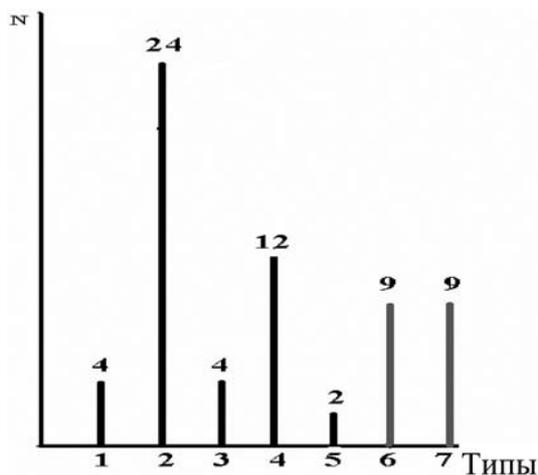


Рис. 5. Распределение по типам

тия A_i). Графически распределение представлено на рис. 5, где по оси абсцисс указаны типы соответствующих триангуляций кубов, а по оси ординат отложено число событий A_i , где i — номер триангуляции в соответствии с классификацией, предложенной в этой статье. Анализ всех бинарных кодов a показывает, что каждый из пяти типов триангуляции I^3 можно описать вдвое меньшим числом бинарных кодов, чем число событий A_i , представленных на графике (см. рис. 5), так как для любого кода a , соответствующего некоторому типу I^3 , всегда найдется код \bar{a} , соответствующий тому же типу I^3 . Сформулируем это свойство.

Свойство 1. Для каждого набора $a = (a_1, \dots, a_6)$, $a_i \in \{0, 1\}$, соответствующему одному из типов триангуляции I^3 , существует набор \bar{a} , однозначно соответствующий тому же типу триангуляции I^3 .

4. Триангуляция в четырехмерном пространстве

Рассмотрим множество (Z^4, V_1) , где вершины являются узлами уже четырехмерной целочисленной решетки, а множество простых ребер-отрезков имеет норму $p = 1$, образуя множество ребер решетки, инцидентных вершинам. Основным элементом полученной структуры является единичный куб I^4 с вершинами в целых точках и простыми ребрами-отрезками в качестве одномерных граней.

По аналогии зададим произвольную триангуляцию (стохастическую) на множестве (Z^4, V_1) как триангуляцию двумерных граней четырехмерной решетки, получаемую проведением диагонали в соответствующей грани.

Понятие правильной триангуляции, основывающееся на требовании правильного расположения симплексов по Понтрягину, оставим в приведенной ранее формулировке без изменений.

Таким образом, неполной будет называться такая триангуляция, когда в развертку I^3 входит хотя бы один неполно триангулированный трехмерный куб.

Как и в трехмерном случае, очевидно, что произвольная триангуляция (Z^4, V_1) не всегда будет являться правильной. В то же время классификация правильных типов триангуляции I^4 достаточно велика, чтобы можно было наглядно, как для I^3 , привести ее в графическом виде.

Поэтому будем использовать способ классификации на основе вектора количества типов I^3 , входящих в развертку I^4 . Формально $x = (x_0, x_1, x_2, x_3, x_4)$, где x_i — число трехмерных кубов $i + 1$ типа, входящих в развертку четырехмерного куба.

Получим статистическую информацию, аналогичную распределению для типов I^4 .

Четырехмерный куб имеет 24 двумерных грани, являющиеся одновременно гранями трехмерных кубов его развертки. Фиксируем порядок развертки, тогда каждую конфигурацию можно закодировать 24 двоичными разрядами, по аналогии с трехмерным случаем. На рис. 6, а представлена трехмерная модель развертки I^4 , развертка каждого куба выполняется аналогично трехмерному случаю (см. рис. 4).

Тогда число всех возможных конфигураций I^4 будет равно 2^{24} . Рассмотрим множество элементарных событий Ω с элементами — кодами развертки. Мощность Ω составляет 2^{24} . Определим сигма-алгебру A случайных событий, как множество событий появления I^4 определенного типа, тогда $A = \{A_1, \dots, A_{295}\}$, где A_{295} — обобщенное событие появления неполной триангуляции I^4 , а $P(\Omega) = 1$.

Введя таким образом вероятностное пространство, получим искомые вероятности событий и распределение по типам.

Число перебранных вариантов событий — 16777216, что соответствует числу всех возможных бинарных кодов длины 24, где 24 — число двумерных граней четырехмерного куба. Всего различных правильных конфигураций I^4 — 1099404, соответственно, всего неполных конфигураций — 15677812.

Таким образом, вероятность выделить правильно триангулированный четырехмерный куб в четырехмерном пространстве составляет 0,000304579, неполно триангулированный — 0,99969542. По

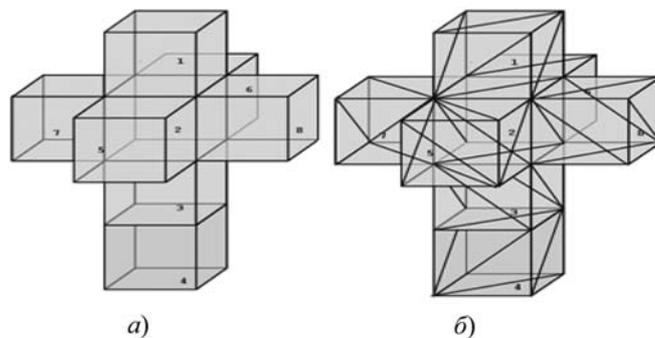


Рис. 6. Развертка четырехмерного куба (а) и триангулированная развертка (8, 0, 0, 0, 0) (б)

сравнению со случаем правильной триангуляции I^3 , вероятность получить правильную триангуляцию I^4 достаточно мала, что дает основания утверждать, что при повышении размерности эта вероятность будет только уменьшаться.

На рис. 6, а приведено изображение триангулированной развертки I^4 , целиком состоящей из кубов первого типа и соответствующей вектору количества типов $(8, 0, 0, 0, 0)$.

Детальную картину можно получить, построив распределение по векторам количества типов.

Оценим общее число N всевозможных векторов количества типов. Напомним, что вектор выглядит следующим образом: $x = (x_0, x_1, x_2, x_3, x_4)$, где x_i — число трехмерных кубов $(i + 1)$ типа, входящих в развертку четырехмерного куба. При этом выполняется равенство

$$\sum_{i=0}^4 x_i = 8. \quad (2)$$

По определению распределение n одинаковых частиц по m ячейкам, называется статистической моделью Бозе—Энштейна [4]. Таким образом, число N способов размещения n частиц по m ячейкам равно

$$N = \frac{(n + m - 1)!}{n!(m - 1)!}. \quad (3)$$

Тогда из выведенной формулы имеем

$$N = \frac{(8 + 4)!}{8! \cdot 4!} = 495. \quad (4)$$

В данном случае распределение полностью подходит для описания распределения восьми частиц по пяти ячейкам — элементам вектора количества типов.

Если упорядочить все N векторов в лексикографическом порядке, то распределение будет иметь вид, представленный на рис. 7. Максимум приходится на вектор $(1, 3, 1, 2, 1)$ с весом 2987.

Заметим также, что для I^4 при использовании двоичного кодирования также справедливо свойство 1 в новой формулировке (свойство 2).

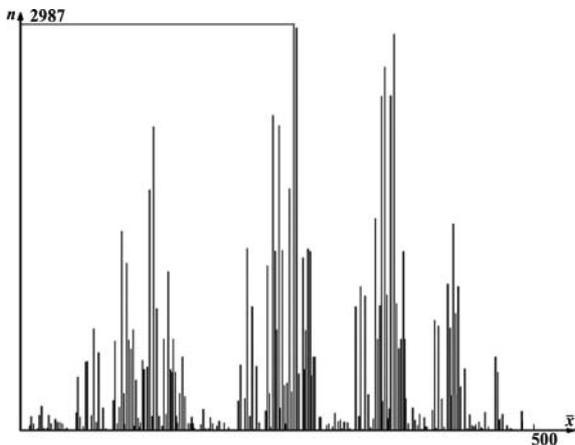


Рис. 7. Распределение по векторам числа типов

Свойство 2. Для каждого набора $a = (a_1, \dots, a_{24})$, $a_i \in \{0, 1\}$, соответствующего одному из типов триангуляции I^4 , существует набор \bar{a} , однозначно соответствующий тому же типу триангуляции I^4 .

Рассмотрим теперь задачу, аналогичную задаче разбиения I^3 на симплексы — задачу разбиения I^4 на 4-симплексы с учетом его правильной триангуляции.

Утверждение 1. Для того чтобы существовало правильное разбиение правильно триангулированных конфигураций I^4 на 4-симплексы, необходимо, чтобы число 3-симплексов, входящих в развертку, делилось без остатка на 4.

Действительно, так как в 4-симплексе существуют четыре трехмерные грани, представляющие собой 3-симплекс, то необходимо, чтобы хотя бы число 3-симплексов в развертке I^4 делилось без остатка на 4.

Очевидно, что не все конфигурации I^4 , характеризующиеся векторами количества типов, представляемые на графике распределения, удовлетворяют данному условию.

Рассмотрим все возможные конфигурации I^4 . Для этого оценим сначала число 3-симплексов, входящих в каждый тип I^3 :

Тип	1	2	3	4	5
Число симплексов	6	6	6	6	5

Таким образом, условию необходимости удовлетворяют все конфигурации I^4 , в векторах которых первые четыре типа присутствуют четное число раз и/или тип 5 присутствует число раз, кратное 4.

Попытаемся теперь на качественном уровне оценить возможность разбиения на 4-симплексы. В первую очередь это влечет, по аналогии с I^3 и I^4 , выделение ограниченного набора конфигураций, которые будут характеризовать тип триангуляции I^3 . Однако в действительности разбиения I^4 на правильно расположенные 4-симплексы для предложенных разбиений на симплексы не существует. Но возможно построение неполных разбиений на 4-симплексы триангуляции I^4 , например для конфигурации $(8, 0, 0, 0, 0)$.

Приведенные выше результаты легко переносятся на широкий класс симплициальных комплексов, приводимых к кубическим комплексам.

Определение 4. Под малыми локальными деформациями симплициального комплекса будем подразумевать такие деформации, при которых любая вершина симплициального комплекса может перемещаться внутри сферы такого радиуса, который исключает возможность совпадения с любой другой вершиной и изменения положения вершины относительно гиперплоскости основания данного симплекса.

Последнее требование обусловлено тем, что в процессе деформации не допускается самопересечения граней симплекса, кроме как по ребрам симплекса. Необходимым условием для осуществления такой деформации является совпадение числа вершин симплициального комплекса с числом вершин

в целевом кубическом комплексе, а также совпадение числа симплексов.

Определение 5. Будем называть симплицальный комплекс *приводимым*, если он может быть однозначно преобразован в кубический путем малых локальных деформаций без нарушения связности и выпуклости.

Таким образом, любой симплицальный комплекс, приводимый к кубическому, описывается с помощью соответствующего распределения.

5. Процессы стохастической триангуляции

В работе [5] приводятся основные положения, касающиеся марковских процессов, индуцируемых перестройками примитивной триангуляции на целочисленных решетках в R^2 и R^3 . Там же приведены матрицы переходных вероятностей, графические представления полученных цепей, изучаются их простейшие свойства. Кратко приведем основные результаты, обобщая их исходя из рассуждений, приведенных в предыдущих частях статьи.

Случайные перестройки диагоналей в гранях примитивной триангуляции (Z^n, V_1) индуцируют марковскую цепь с конечным числом состояний.

При допущении всех типов триангуляции I^n и одиночной перестройки диагонали грани в каждый дискретный момент времени процесс является периодическим.

При допущении только правильных типов триангуляции процесс для нечетного числа меняющихся диагоналей в единицу времени является эргодическим с одинаковым стационарным распределением. Для четного числа меняющихся диагоналей процесс распадается на подцепи с эргодическими свойствами.

Однако при допущении только правильных типов триангуляции для перестроек, индуцированных на некоторой подобласти (Z^n, V_1) , является актуальным вопрос о их возможности. Связан он с тем, что в результате перестройки диагонали и перехода соответствующего I^n в новое состояние меняется диагональ прилежащего I^n , по условию также находящегося в состоянии правильной триангуляции. Если предположить, что при переходе некоторого I^n в новое правильное состояние прилежащие I^n также стремятся перейти в правильное состояние, то ответ на него дает следующее утверждение.

Заметим, что данный процесс уже не является марковским в смысле приведенных выше результатов, однако соответствует процессам стохастической триангуляции.

Утверждение 2. Рассмотрим правильно триангулированную непустую подобласть (Z^n, V_1) . В каждый дискретный момент времени для перестройки одной диагонали одного произвольного $I^n \in (Z^n, V_1)$ при допущении только правильных типов триангуляции всегда существует, по крайней мере, одно правильное конечное состояние данной подобласти (Z^n, V_1) .

Доказательство данного утверждения напрямую следует из свойств (1) и (2), приведенных выше. Представим подобласть (Z^n, V_1) в виде бинарного

кода согласно методике, описанной выше, и зафиксируем порядок обхода и развертки для каждого $I^n \in (Z^n, V_1)$. Тогда любая одиночная перестройка диагонали некоторого $I^n \in (Z^n, V_1)$ вызывает изменение некоторого разряда бинарного кода a , соответствующего триангуляции рассматриваемого I^n . Согласно свойствам (1) и (2) для каждой конфигурации I^n существует код \bar{a} , который может быть получен как последовательная перестройка всех диагоналей I^n . Учитывая требование, что подобласть должна всегда оставаться в состоянии правильной триангуляции, процесс перестройки проведем для всех $I^n \in (Z^n, V_1)$. Таким образом, рассматриваемая подобласть должна прийти в состояние, соответствующее полной инверсии всех диагоналей на гранях. При этом данное состояние будет правильным.

Заключение

Свойства стохастической триангуляции, описанные в данной статье, позволяют проводить анализ геометрико-топологических триангулированных моделей с помощью предложенных статистических оценок. Ключевую роль в данном случае играет распределение Бозе—Эйнштейна, описывающее динамику и характер триангуляции в кубических симплицальных комплексах, являющихся основой структуры данных для широкого круга приложений и проблемных областей. При этом получены качественно новые данные о характере описываемого распределения по сравнению с работой [5]. Эти данные позволяют получить новые количественные и качественные оценки процессов триангуляции. Эти оценки и характер распределения необходимы для понимания процессов триангуляции в среде "Топологический процессор".

Также в работе сформулирован критерий разбиения на 4-симплексы кубических комплексов размерности 4 и показана его невозможность. Это означает, что основным элементом для построения геометрико-топологических моделей в среде "Топологический процессор" может являться только 3-симплекс.

Список литературы

1. Adcroft A., Campin J.-M. Massachusetts Institute of Technology General Circulation Model, online documentation. URL: http://mitgcm.org/sealion/online_documents/node2.html
2. Рябов Г. Г. Алгоритмические основы топологического процессора // Методы и средства обработки информации: Труды Второй всероссийской научной конференции. М.: Издательский отдел Факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова, 2005. С. 53—58. URL: <http://lvk.cs.msu.su/old/mco/mso2005-full.doc>
3. Понтрягин Л. С. Основы комбинаторной топологии. М.: Наука, 1986. 180 с.
4. Мартинсон Л. К., Смирнов Е. В. Квантовая физика. М.: Изд-во МГТУ им. Н. Э. Баумана, 2006. 528 с. URL: <http://www.lgrflab.ru/phybook/tom5/content.htm>
5. Рябов Г. Г. Марковские процессы в динамике примитивных триангуляций в пространствах R^3 и R^4 // Вычислительные методы и программирование. 2009. Т. 10, № 1. С. 1—8.

УДК 004.42::[519.876.2:519.218.82]::004.272.2

В. В. Пекунов, канд. техн. наук, доц.,
Ивановский государственный
энергетический университет (г. Иваново)
e-mail: pekunov@mail.ru

Метаслой моделирования алгоритма, данных и функциональных характеристик последовательных и параллельных программ

Рассматриваются проблемы моделирования и предикции данных, состояния алгоритма и времени исполнения программы. Обосновано применение формализма объектно-событийных моделей в задачах построения и интерпретации предикационных моделей. Предложена концепция абстрагированного от программы метаслоя, осуществляющего моделирование. Дана парадигма применения предикационных моделей для динамической оптимизации и приближения/восстановления алгоритмов как обычных, так и параллельных программ.

Ключевые слова: метаслой программы, моделирование алгоритма, прогнозирование данных, прогнозирование времени исполнения, объектно-событийная модель, параллельное программирование, оптимизация программы

Введение

Существует ряд нетривиальных задач, эффективное решение которых требует разработки специфических программ, нуждающихся в предикции времени исполнения тех или иных своих фрагментов, их внутреннего состояния или формируемых ими данных.

Предикция времени исполнения путем построения профилирующей модели необходима при решении ряда задач, связанных с *параллельной обработкой данных*, особенно при работе в гетерогенной вычислительной среде:

- определения потребности в распараллеливании и его потенциальной эффективности;
- выбора схем распараллеливания [1];
- оптимальной балансировки загрузки процессоров.

Последняя проблема характерна для решений, связанных с применением гранулярного параллелизма,

характерного для Т-систем, МС#, процедур с повторным входом [2], OpenMP 3.

Предикция *внутреннего состояния* путем построения модели хода исполнения может потребоваться при *выборе оптимального алгоритма*, предполагающем косвенную оценку времени исполнения через предикцию значений переменных состояния, характеризующих исполнение алгоритма. Если время исполнения является сложноопределяемой функциональной зависимостью от переменных состояния, то такая косвенная оценка может быть достаточной и менее трудозатратной в сравнении с прямой оценкой. Это может быть полезно, например, при выборе алгоритма поиска/сортировки информации в сложноструктурированных данных по их характеристикам на небольшой выборке.

Интересным вариантом применения такой предикции является *интеллектуализация разработки и отладки*, а именно — сверка получаемых графов переходов алгоритма с целевым, определяемым изначально или в процессе построения программы. В работе [3] отмечено применение схожего подхода для оптимизации исходной программы.

Предикция *формируемых данных* может использоваться при *параллельном решении ряда задач математической физики*. Возможно исключение части пересылок путем предикции недостающих данных [4]. Также можно говорить о *копировании или приближении алгоритмов обработки данных без копирования реализующего их кода*, что представляет определенный интерес, например, если имеется ряд программных модулей, написанных на иных языках, взаимодействие с которыми затратно по времени.

Предикция времени исполнения, внутреннего состояния и/или данных должна быть внешним по отношению к основной программе фоновым процессом, не увеличивающим время ее исполнения, функционирующим на отдельном ядре или группе ядер центрального процессора (ЦП) или графического процессора (ГП) видеокарт класса nVidia GeForce с поддержкой технологии CUDA, содержащих матрицы однородных SIMD-процессоров, обладающих высокой производительностью обработки однородных данных. Задача сводится к группе подзадач построения математических моделей данных в сочетании с вероятностными и/или логическими моделями алгоритмов, с последующей интерпретацией моделей. Осуществляется дина-

мический анализ потоков данных и трассы исполнения [3].

Введем понятие *метаслоя* (скрытого слоя, концептуально несколько схожего с пространством моделирования ЕСО [5]) основной программы, собирающего сведения о ходе ее исполнения и значениях контролируемых переменных при прохождении через заданные контрольные точки, осуществляющего построение и интерпретацию предикционных моделей алгоритма/данных в фоновом режиме и представляющего результаты предикции основной программе по запросу. Принятие решений о том или ином варианте исполнения осуществляется основной программой, хотя интересным представляется и вариант, при котором решение может приниматься непосредственно в метаслое, который в таком случае должен включать соответствующую систему дедуктивного вывода и/или продукционную систему.

Построение предикционных моделей может состоять в индукции алгоритма "входы → переменные состояния" и/или индукции функциональных моделей, записанных в терминах "входы → выходы" или "входы + переменные состояния → выходы". Второй вариант позволяет упростить как функциональные модели, так и процесс их построения. Упрощение объясняется тем, что после корреляционного анализа в пространстве признаков с наибольшей вероятностью остаются преимущественно переменные состояния, обычно имеющие более ярко выраженную корреляцию с выходами. Их ввод в функциональную модель позволяет сократить анализ и исключить некоторые фрагменты комплекса зависимостей "входы → переменные состояния".

Для представления моделей целесообразен выбор такого формализма, в рамках которого были бы строго определены не только операции логического синтеза модели, адекватно отражающей структуру и алгоритм основной программы, но и операции ее интерпретации (параметризацию и, возможно, непосредственное исполнение в целях предикции) и трансформации/компиляции в шаблонный код, решающий задачу предикции. Очевидно, что это должен быть некий класс дедуктивно-индуктивных синтезирующих моделей. Указанным требованиям в полной мере отвечает формализм объектно-событийных моделей (ОСМ) [4, 6].

Представление модели алгоритма и данных

Входными данными при синтезе модели являются: трасса исполнения (список узлов-контрольных точек в порядке их прохождения), текущие значения контролируемых переменных программы и времени прохождения для каждого узла. Пусть синтез структуры модели алгоритма (фактически —

графа переходов) осуществляется путем логического анализа трассы исполнения (может использоваться машина дедуктивного вывода), содержащей перечень всех совершенных переходов. Как и при построении функциональных моделей, целесообразно искать закономерности переходов как в значениях входных контролируемых переменных, так и в значениях внутренних переменных состояния, что упрощает модель алгоритма.

Внутренними переменными могут быть счетчики активизации узлов, которые рассчитываются путем прохождения по узлам модели в порядке, определяемом трассой исполнения. Значения времени и контролируемых переменных присоединяются к значениям внутренних переменных, таким образом получаем журнал исполнения, анализ которого позволяет осуществить параметризацию моделей алгоритма и данных.

Как уже отмечалось ранее, для представления таких моделей удобно использовать формализм ОСМ с классической двуслойной схемой интерпретации [4]:

- последовательным достраиванием и трансформацией модели в решающем слое с применением аппарата логического анализа и синтеза;
- исполнением (непосредственным или через компиляцию в программный код).

На первом этапе модель может представлять собой единственный дедуктивный объект-анализатор, относящийся к решающему классу, принимающий на вход данные трассы исполнения и синтезирующий по ней индукционно-аналитическую ОСМ — граф переходов. На втором уровне данная ОСМ исполняется: проводит индукцию правил перехода и трансформации данных, после чего переходит (планируется новое событие) к предикции в режиме прямого исполнения правил или через генерацию исполняющего кода. Режим генерации кода является исходным режимом работы ОСМ и детально описан в работе [4] для блок-схемы, имеющей взаимно однозначное соответствие графу переходов. Режим прямой интерпретации может быть реализован с передачей управления по элементарной событийной схеме.

Структура ОСМ-алгоритма и данных

Структуру ОСМ можно представить графом (P, E) , где P — множество объектов (узлов-контрольных точек) различных классов, E — множество дуг, представляющих направления возможных переходов. Дуги, замыкающие цикл, целесообразно представить информационными связями (без прямой активизации узла, инцидентного дуге по входу), прочие дуги — основными. Объекты (узлы) модели относятся к одному из классов, входящих в двухуровневую иерархию, родительским элементом

которой является базовый класс S — узлы-инкременторы без предикции, его наследники: P_t — узел с предикцией времени, P_d — с предикцией данных, P_{td} — с предикцией времени и данных.

В терминах ОСМ любой класс из множества классов модели $C = \{S, P_t, P_d, P_{td}\}$ описывается пятеркой (N_c, I, O, F, M) , где N_c — идентификатор (имя) класса, I и O — наборы входных и выходных контактов, $F = F(N_c)$ — поля, $M = M(N_c)$ — методы. Объекту любого из этих классов достаточно иметь один активизационный вход и один активирующий выход.

В ОСМ любой контакт является шестеркой (N_t, T, L, Min, Max, D) , где N_t — идентификатор контакта, T — тип контакта (входной, выходной, анонимный), L — множество пар (класс, контакт), определяющих контакты классов и их наследников, к которым можно провести выходную связь, Min — минимальная степень контакта, Max — максимальная степень, D — ассоциативный кортеж, отражающий состояние контакта. Таким образом,

$$\forall c \in C: I = \{I_0\} = \{("вход", \text{входной}, \emptyset, 0, \infty, D)\};$$

$$\forall c \in C: O = \{O_0\};$$

$$O_0 = ("выход", \text{выходной}, \{(S, I_0)\}, 0, \infty, D).$$

Входные связи множеством L не специфицируются, поскольку легко выводятся из множества выходных связей. Минимальное множество полей, позволяющих описать семантику модели:

$$F(S) = \{ID, \text{counter}, R_{\text{trans}}, B_{\text{trans}}\},$$

где ID — идентификатор текущего объекта (совпадающего с идентификатором узла-контрольной точки); counter — поле-счетчик, используемое исключительно в режиме исполнения модели (счетчик инкрементируется при каждом входе в узел, т. е. при активизации соответствующего объекта); $R_{\text{trans}} = R_{\text{trans}}[i]$ — ассоциативный кортеж *взаимоисключающих* правил перехода (предикатов) по исходящим дугам; $B_{\text{trans}} = B_{\text{trans}}[i]$ — ассоциативный кортеж булевых значений, определяющих наличие/отсутствие инициализации счетчика узла, инцидентного дуге с номером i по входу. Ее наличие определяет существование цикла, начало которого обозначено данным узлом. Если $R_{\text{trans}}[i] = \emptyset$, то дуга с номером i является отрицательной, т. е. переход по ней осуществляется, если не выполняется ни одно из правил для прочих дуг (отрицательными по определению являются единственные безусловные дуги). Таковой целесообразно сделать одну дугу из числа инцидентных узлу по выходу, для которой существуют наиболее трудозатратные в расчете предикаты перехода.

С явным учетом наследования можно записать

$$F(P_t) = F(S) \cup \{T_{\text{trans}}\}; F(P_d) = F(S) \cup \{D_{\text{trans}}\}; \\ F(P_{td}) = F(P_t) \cup F(P_d),$$

где $T_{\text{trans}} = T_{\text{trans}}[i]$ и $D_{\text{trans}} = D_{\text{trans}}[i]$ — ассоциативные кортежи моделей (правил) расчета времени и трансформации данных при переходах по исходящим дугам.

Параметризация правил модели

Любое из правил расчета времени и трансформации данных, сопоставленных переходам, является тройкой $(V_{\text{inp}}, \text{out}, f(X))$, где V_{inp} — вектор идентификаторов входных переменных, являющихся аргументами расчетной функции $f(X)$, причем компоненты векторов V_{inp} и X имеют прямое позициональное соответствие, out — идентификатор выходной переменной.

В настоящее время $f(X)$ являются полиномиальными зависимостями от одной или многих переменных, построенными, соответственно, с применением линейной (в рамках критерия метода наименьших квадратов) или нелинейной регрессии в соответствии с подходом метода группового учета аргументов [7]. Отбор переменных-аргументов реализуется классическим корреляционным анализом.

Правила (предикаты) перехода находятся путем комплексного логического анализа журнала исполнения A_{pv}^{ks} — матриц значений переменных $v \in V$ при p -м по счету переходе программы из точки k в точку s , причем

$$k, s \in K; p = \overline{1, M_{ks}};$$

$$V = \{\text{time}\} \cup R; R = Z \cup K,$$

где K — множество идентификаторов узлов-контрольных точек, совпадающих с идентификаторами соответствующих внутренних переменных; M_{ks} — число наличествующих в трассе случаев прохождения из k в s ; time — идентификатор переменной времени; Z — множество идентификаторов контролируемых входных переменных.

Пусть из рассматриваемого узла f возможны переходы в N_f узлов, идентификаторы которых содержатся в векторе F . Правило перехода rule_{fg} в g -й узел ($g \in F$) может быть определено с помощью простых алгоритмов поиска классификационных правил по типу алгоритма Кора [7] настолько, насколько это возможно в текущем контексте, т. е. при имеющемся наборе классифицирующих признаков R . На каждом этапе поиска определяется переменная b с наибольшей дискриминативностью d_j , т. е. такая, которой соответствует:

а) наибольшее число строк матрицы A_{pb}^{fg} , у которых значения в столбце b отличаются от значений в том же столбце матриц A_{pb}^{fs} , $s \neq g$;

б) наибольшее число разных значений такого рода.

Соответственно,

$$b = \arg(\max_j (d_j)); d_j = \alpha P_j^{fg} + \beta \text{card}(G_j^{fg} / U_j^{fg});$$

$$U_j^{fg} = \bigcup_{s \neq g} \{A_{pj}^{fs} | p = 1, M_{fs}\}; G_j^{fg} = \{A_{pj}^{fg} | p = 1, M_{fg}\};$$

$$P_j^{fg} = \sum_{s \neq g} \text{card}\{p \in 1, M_{fs} | A_{pj}^{fs} \in (G_j^{fg} / U_j^{fg})\},$$

где α и β — коэффициенты настройки, которые целесообразно выбирать из условия $\beta > \alpha$. При ненулевом значении $\max(d_j)$ из матрицы A_{pb}^{fs} исключаются строки, в которых переменная b имеет любое из дискриминирующих значений из множества G_j^{fg} / U_j^{fg} . В правило вводится соответствующее условие $\text{check}(b)$, покрывающее данные значения, представляющее собой либо единственный шаблонный предикат $\text{cond}_0(b)$, либо дизъюнкцию нескольких таких предикатов $\text{cond}_i(b)$, таких что

$$\bigcup_i T_i \equiv G_j^{fg} / U_j^{fg}; T_i = \{t \in U_j^{fg} \cup G_j^{fg} | \text{cond}_i(t)\};$$

$$\bigcup_i F_i \equiv U_j^{fg}; F_i = \{f \in U_j^{fg} \cup G_j^{fg} | \neg \text{cond}_i(f)\},$$

причем в качестве $\text{cond}_i(b)$ рассматриваются *шаблонные выражения* для проверки отдельных значений и значений, входящих в ряды с арифметической или геометрической прогрессией, наиболее часто встречающиеся в классических алгоритмах.

В случае нулевого значения $\max(d_j)$ в конъюнктивное правило перехода в g -й узел вводится остаточное вероятностное условие

$$\text{check}(b) = \sum_{s=1}^{g-1} M_{fs} / \sum_{s=1}^{N_F} M_{fs} \xi_1 q \xi_2 \sum_{s=1}^g M_{fs} / \sum_{s=1}^{N_F} M_{fs},$$

где ξ_1, ξ_2 — знаки отношения строгого или нестрогого следования; q — равномерно распределенная случайная величина в диапазон $[0; 1]$. При этом из всех матриц A_{pv}^{fg} исключаются все оставшиеся в них строки. Таким образом, модель алгоритма принимает либо вероятностно-логическую, либо чисто вероятностную форму.

Процедура поиска условий повторяется, пока хотя бы одна из матриц A_{pv}^{fg} не пуста. Результирующее правило перехода rule_{fg} определяется дизъюнкцией всех найденных для данного перехода условий $\text{check}(b)$.

Исполнение ОСМ

Как прямое, так и опосредованное компиляцией исполнение объектно-событийной модели алгоритма и данных осуществляются через ее интерпретацию, выполняемую по обычной двухуровневой схеме [4]. Режим интерпретации (прямое исполнение или опосредованное компиляцией) модели определяется специализацией классов или параметрами входящих в нее объектов.

Рассмотрим *режим прямого исполнения*, который подразумевает последовательную передачу управления от одного узла к другому. Для наиболее эффективного исполнения целесообразно передавать управление в ходе обработки одного и того же события, если же это невозможно (при замыкании цикла) — планировать следующие события исполнения. Такая передача осуществляется по достаточно тривиальной схеме с хранением необходимых для передачи управления данных в общем кортеже "почтовый ящик" MAIL[имя_ячейки]. Для текущего события с идентификатором $EV_{\text{тек}}$ хранятся $ID_{\text{акт}}$ — идентификатор исполняемого (получающего управление) при наступлении данного события объекта-узла и ссылка на кортеж VARS текущих значений всех переменных модели, т. е.

$$\text{MAIL}[EV_{\text{тек}}] = (ID_{\text{акт}}, \text{VARS}),$$

причем имеет место биекция

$$\text{var_refs:VARS} \leftrightarrow V.$$

Как только календарь событий исчерпывается, интерпретация ОСМ завершается. При наличии вероятностных правил интерпретация проводится несколько раз, полученные временные характеристики осредняются.

Рассмотрим реализацию методов-обработчиков. Введем операцию применения правила $T = (V_{\text{inp}}, \text{out}, f(X))$, обозначив ее $\text{apply}(T)$ со следующим содержанием:

$$\text{VARS}[\text{out}] = f(X);$$

$$\forall i: X_i = \text{VARS}[V_{\text{inp}}]_i.$$

Операцию инкремента переменной x обозначим $\text{inc}(x)$.

Запишем основное содержание методов, активируемых при обработке любого из событий прямого исполнения модели:

$$\tilde{M}(S) = \left\{ \begin{array}{l} \text{inc(counter)} \\ \text{link} = \left\{ \begin{array}{l} i \in \tilde{N}_1 \text{ if } R_{\text{trans}}[i] \vee \\ \left(R_{\text{trans}}[i] = \emptyset \wedge \right. \\ \left. \vee \left(\wedge \forall y \in \tilde{N}_1, y \neq i : \neg R_{\text{trans}}[i] \right) \right); \\ \emptyset \text{ if } (\tilde{N}_1 = \emptyset) \vee \\ \left. \vee \left[\begin{array}{l} \forall i \in \tilde{N}_1, : \neg R_{\text{trans}}[i] \wedge \\ \wedge (\neg \exists i \in \tilde{N}_1 : R_{\text{trans}}[i] = \emptyset) \right] \right); \end{array} \right. \end{array} \right.$$

$$\tilde{M}(P_t) = \text{apply}(T_{\text{trans}}[\text{link}]);$$

$$\tilde{M}(P_d) = \text{apply}(D_{\text{trans}}[\text{link}]);$$

$$\tilde{M}(P_{td}) = \tilde{M}(P_t) \circ \tilde{M}(P_d);$$

$$\forall c \in C, \\ \forall j \in N_{ev} : (M(c))_j =$$

$$= \left\{ \begin{array}{l} \emptyset \text{ if } ID \neq ID_{\text{акт}}; \\ \tilde{M}(c) \circ \text{activate}(\text{link}) \text{ if } ID = ID_{\text{акт}}, \end{array} \right.$$

где \tilde{N}_1 — множество всех допустимых номеров выходных дуг; link — номер дуги, инцидентной по входу узлу, которому будет передано управление; N_{ev} — множество номеров всех событий. Знаком "o" обозначена конкатенация операций, подразумевающая их строго последовательное исполнение слева направо. Все обозначения относятся к активному в данный момент объекту. Операция activate(link) передает управление узлу, инцидентному по входу дуге link.

В режиме исполнения через компиляцию ОСМ в процессе интерпретации генерирует код в соответствии с четырехэтапной схемой (размещение, инициализация, вызов, деинициализация) [4], при которой каждый объект порождает фрагменты кода C++, реализующие инкремент счетчика и условные переходы с исполнением правил трансформации, с модификацией переменной времени и иными действиями. Полученный код компилируется в функцию в составе динамически подгружаемого модуля.

В любом из режимов исполнения ОСМ результатами являются значения времени и/или внешних переменных, полученные при завершении моделирования.

Исследование основных вариантов применения метаслоя

1. Предикция времени исполнения позволяет, в частности, легко определить необходимость в распараллеливании расчета. Например, при исследовании динамики брюсселятора решается система из двух дифференциальных уравнений для различных сочетаний начальных значений переменных, хранимых в матрице размером $K \times K$. Проведя несколько экспериментов для различных $K = 2^p$, $p \in Z$, с помощью метаслоя легко построить предикционную модель времени time(K). Зная издержки $P(N)$ на запуск параллельного расчета на N процессорах, учитывая закон Амдала [8], легко получить условие эффективности распараллеливания

$$P(N) < \frac{N-1}{N} \text{time}(K).$$

В таблице приведены данные экспериментальных замеров¹ времени параллельного (с распределением витков цикла по ядрам с помощью OpenMP) и однопроцессорного решения указанной задачи при различных K , а также результаты проверки условия эффективности.

Очевидно, что предикция времени исполнения в метаслое позволила избежать непроизводительных затрат на распараллеливание в двух последних случаях ($K = 6; 1$). Важно, что абстракция метаслоя от кода позволяет ввести такой прием профилирующей оптимизации в любой алгоритмический язык программирования с различными интерфейсами распараллеливания.

2. Предикция данных часто встречается при решении задач математической физики в частных производных с распараллеливанием по пространству. Требование непрерывности производных на стыках блоков обрабатываемых областей подразумевает выполнение обменов данными, что может быть длительной операцией. Иногда удается сократить временные затраты, заменив обмен экстраполяцией требуемых данных. Данная задача рассмотрена в работе [4], в которой показана возможность повышения эффективности распарал-

Замеры времени (с) решения с проверкой условия эффективности

K	800	160	32	6	1
Решение на одном ядре	1059	212	8,25	0,3	0,009
Решение на двух ядрах	539	107	4,47	0,47	0,414
Условие эффективности	+	+	+	—	—

¹ Результаты получены в системе с двухъядерным процессором Intel T4400, 2,2 ГГц. Компилятор Visual C++ 2008 Professional.

леливания некоторых задач моделирования пространства загрязнений на 4...11 % за счет такого сокращения числа обменов. Построение соответствующей регрессионной модели данных вполне может быть реализовано в метаслое.

3. Комплексная предикция алгоритма и данных подразумевает параметризацию механизмов исполнения, которая вводит внутренние переменные состояния и, тем самым, упрощает как процесс построения, так и структуру функциональных моделей данных и времени исполнения, иногда повышает точность предсказания. Улучшение прогностических свойств возможно, например, в случае алгоритмов, в которых искомые величины являются такими функциями от входных параметров, что сложность их интерполяционного приближения превышает возможности встроенного прогностера.

Для некоторых итерационных алгоритмов с простыми внутренними преобразованиями данных, таких как численное интегрирование систем дифференциальных уравнений, возможно даже фактическое копирование логики алгоритма обработки и аппроксимации выполняемых в нем преобразований с закономерным итогом — абсолютная погрешность на уровне машинной $\approx 10^{13}$. Прямое приближение "начальные значения \rightarrow конечные значения" сводится к несомненно более трудной задаче интерполяции численного решения со вполне заурядной погрешностью порядка $10^{-3} \dots 10^{-1}$, однако при меньшей трудоемкости по сравнению с расчетом.

Учет внутренних параметров путем моделирования алгоритма позволяет повысить точность прогноза времени в задачах, в которых оно является, например, алгоритмически нелинейной или даже разрывной функцией от входных переменных. Это многие расчетные задачи со сложными алгоритмическими зависимостями, например, внутренним ветвлением в одном или нескольких вложенных циклах. Ввод внутренних параметров позволяет в несколько раз снизить погрешность предикции, доведя ее до нескольких процентов. При распараллеливании решения таких задач предикция позволяет заблаговременно более равномерно распределить нагрузку процессоров, корректируя алгоритмическую нелинейность счета.

Заключение

Предложена концепция метаслоя программы, осуществляющего фоновое динамическое построение моделей алгоритмов, данных и времени исполнения такой программы. Показано, что адекватным формализмом для представления, построения и интерпретации указанных моделей являются ОСМ. Построение структуры ОСМ может проводиться в решающем слое, параметризация и исполнение осуществляются в ходе ее интерпретации. Предложены стратегии построения и интерпретации.

В настоящее время разработана и испытана базовая версия системы поддержки метаслоя. Применение предикционных моделей алгоритмов и времени исполнения, формируемых в его рамках, позволяет проводить динамические профилировку и оптимизацию распараллеливаемых программ независимо от конкретного алгоритмического языка программирования и стратегии распараллеливания. Предикция алгоритма, проецирующая его состояние на функциональные модели времени и данных, в ряде случаев упрощает такие модели и снижает погрешность предсказания. Предикция данных иногда может заменять их передачу при параллельном решении некоторых задач математической физики, уменьшая время расчета.

Список литературы

1. Пекунов В. В. Модель образования и распространения твердых, жидких и газообразных загрязнителей. Оптимальное распараллеливание // Математическое моделирование. 2009. Т. 21. № 3. С. 69—82.
2. Пекунов В. В. Процедуры с планированием повторного входа в языках высокого уровня при традиционном и параллельном программировании // Информационные технологии. 2009. № 8. С. 63—67.
3. Аветисян А. И., Иванников В. П., Гайсарян С. С. Анализ и трансформация программ // Всероссийский конкурс отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы", 2008. 78 с.
4. Пекунов В. В. Автоматизация параллельного программирования при моделировании многофазных сред. Оптимальное распараллеливание // Автоматика и телемеханика. 2008. № 7. С. 170—180.
5. Бобровский С. И. Технологии Delphi 2006. Новые возможности. СПб.: Питер, 2006. 288 с.
6. Пекунов В. В. Дедуктивный вывод объектно-событийных моделей. Применение при решении задач динамики многофазных сред // Вестник ИГЭУ. Иваново, 2008. Вып. 4. С. 81.
7. Дюк В., Самойленко А. Data mining: учебный курс. СПб.: Питер, 2001. 368 с.
8. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.

С. Б. Беневоленский¹, д-р техн. наук, зав. каф.;
Ю. И. Кириллов¹, ассистент;
В. Э. Пожар², д-р физ.-мат. наук, зав. лаб.;
В. И. Пустовойт², академик РАН, директор;
И. С. Спиридонов¹, студент,
¹"МАТИ" — Российский государственный
технологический университет
им. К. Э. Циолковского,
²Научно-технологический центр
уникального приборостроения РАН
E-mail: sbb13@mail.ru

Разработка платформы программного комплекса для акустооптических спектрометров

Рассматриваются требования к разрабатываемому программному комплексу для акустооптических спектрометров и выбор средств его реализации.

Ключевые слова: программное обеспечение для акустооптического спектрометра, обработка спектра, анализ спектра, информационное ядро

В настоящее время акустооптика является одной из наиболее интенсивно развивающихся областей создания приборов на новых физических принципах. Обработка спектральной информации, получаемой с помощью акустооптического спектрометра, позволяет изучать состав различных веществ, сводя обработку к разложению этих спектров на более простые (эталонные спектры различных веществ) [1]. При этом решение задачи анализа акустооптических спектров автоматизируется с использованием специализированных систем и программных продуктов. В настоящее время в этой сфере используются различные стандартные пакеты, такие как Origin и другие [2]. Их совершенствование и практическое применение связано с преодолением существующего на сегодняшний день ряда недостатков, в частности, неудобства работы со спектральными данными при их переносе из файла в среду обработки данных; неудобства интерфейса, выраженного в отсутствии русскоязычного меню; отсутствия русскоязычной справочной системы, созданной специально под программное обеспечение для работы со спектральной информацией.

При выборе средств разработки программного обеспечения главным условием была кроссплатформенность, т. е. возможность запуска программы

на более чем одной аппаратной платформе и(или) операционной системе. Также немаловажным фактором было наличие бесплатной лицензии на использование библиотек команд (лицензия GNU GPL). Кроме того, реализация платформы разрабатываемого ПК должна подразумевать возможность объединения в одном ядре разнородных программных компонентов, таких как база данных и информационно-справочная подсистема.

Кроссплатформенными можно назвать большинство современных высокоуровневых языков программирования [2]. Например, C, C++ и Free Pascal — кроссплатформенные языки на уровне компиляции, т. е. для этих языков есть компиляторы под различные платформы. Java и C# — кроссплатформенные языки на уровне выполнения, т. е. их исполняемые файлы можно запускать на различных платформах без предварительной перекомпиляции. PHP, ActionScript, Perl, Python, Tcl и Ruby — кроссплатформенные интерпретируемые языки, их интерпретаторы существуют для многих платформ.

Язык Pascal является одним из самых распространенных и известных языков высокого уровня, однако на сегодняшний день считается устаревшим ввиду отсутствия нормальных средств работы с динамической памятью, ограниченной библиотеки ввода-вывода, отсутствия средств подключения функций, написанных на других языках, отсутствия средств отдельной компиляции и т. п. [3].

Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (Java Virtual Machine, JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор, но с тем отличием, что байтовый код, в отличие от текста, обрабатывается значительно быстрее [4].

Достоинство подобного способа выполнения программ — в полной независимости байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят то, что исполнение байт-кода виртуальной машиной может снижать производи-

тельность программ и алгоритмов, реализованных на языке Java [3]. Данное утверждение было справедливо для первых версий виртуальной машины Java, однако в последнее время оно практически потеряло актуальность [4]. Этому способствовал ряд усовершенствований, таких как применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде; широкое использование платформенно-ориентированного кода (*native*-код) в стандартных библиотеках; аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например технология Jazelle, поддерживаемая некоторыми процессорами фирмы ARM).

По данным сайта shootout.alioth.debian.org, для семи разных задач время выполнения на Java составляет в среднем в 1,5–2 раза больше, чем для C/C++, в некоторых случаях Java быстрее, а в отдельных случаях в 7 раз медленнее [4]. Вместе с тем для большинства из них потребление памяти Java-машиной было в 10–30 раз больше, чем программой на C/C++.

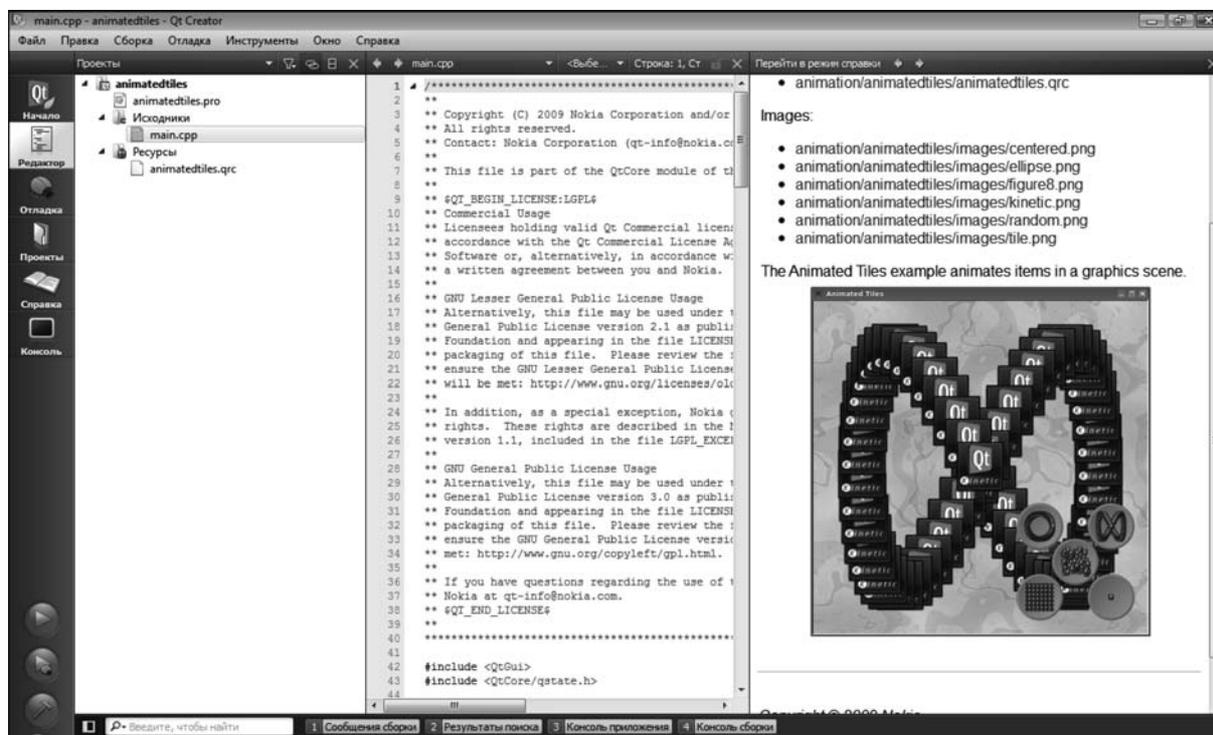
PHP (англ. Hypertext Preprocessor — PHP: пре-процессор гипертекста; англ. Personal Home Page Tools (устар.) — инструменты для создания персональных веб-страниц) — скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений. Язык и его интерпретатор разрабатываются

группой энтузиастов в рамках проекта с открытым кодом, однако главным недостатком применения является его распространение под собственной лицензией, несовместимой с GNU GPL.

ActionScript — объектно-ориентированный язык программирования, который добавляет интерактивность, обработку данных и многое другое в содержимое *Flash*-приложений. ActionScript исполняется виртуальной машиной (ActionScript Virtual Machine), которая является составной частью Flash Player. ActionScript компилируется в байт-код, который включается в SWF-файл.

SWF-файлы исполняются Flash Player, что затрудняет лицензирование разрабатываемого программного обеспечения и исключает использование GNU GPL. Flash Player существует в виде плагина к веб-браузеру, а также как самостоятельное исполняемое приложение. Во втором случае возможно создание исполняемых *exe*-файлов (*projector*), когда SWF-файл включается во Flash Player.

Из перечисленных выше языков предпочтение было отдано C++, так как он является универсальным, предусматривающим статические типы данных. Язык C++ не зависит от платформы, а также не требует слишком усложненной среды программирования [5]. Для возможности применения лицензии GNU GPL была выбрана библиотека Qt и среда разработки для нее QtCreator (см. рисунок). Рассмотрим эту библиотеку подробнее.



Главное окно среды QtCreator

Qt является кросс-платформенным инструментарием разработки ПО на языке программирования C++. Есть также "привязки" ко многим другим языкам программирования: Python—PyQt, PySide; Ruby — QtRuby; Java—Qt Jambi; PHP—PHP-Qt и другие.

Qt позволяет запускать написанное с его помощью ПО в большинстве современных операционных систем путем простой компиляции программы для каждой ОС без изменения исходного кода. Библиотека включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

Существуют версии библиотеки для Microsoft Windows, систем класса UNIX с графической подсистемой X11, Mac OS X, Microsoft Windows CE, встраиваемых Linux-систем и платформы S60. Также идет портирование на HaikuOS.

До недавнего времени библиотека Qt также распространялась еще в одной версии: Qt/Embedded. Теперь эта платформа переименована в Qt/Qt for Embedded Linux и распространяется как отдельный продукт. Qt for Embedded Linux обеспечивает базовую функциональность для всей линейки платформ, предназначенных для разработки приложений для встраиваемых и мобильных устройств (КПК, смартфонов и т. п.).

С версии 4.5 Qt распространяется в двух редакциях: Qt Commercial (для разработки коммерческих приложений); Qt Open Source (под лицензией GNU GPL и GNU LGPL).

Со времени своего появления в 1996 г. библиотека Qt легла в основу тысяч успешных проектов во всем мире [5]. Кроме того, Qt является фундаментом популярной рабочей среды KDE, входящей в состав многих дистрибутивов GNU/Linux.

Отличительная особенность Qt от других библиотек — использование Meta Object Compiler (MOC) — предварительной системы обработки исходного кода (в общем-то, Qt — это библиотека не для чистого C++, а для его особого наречия, которого и "переводит" MOC для последующей компиляции любым стандартным C++ компилятором). MOC позволяет во много раз увеличить мощь библиотек, вводя такие понятия, как слоты и сигналы. Кроме того, это позволяет сделать код более лаконичным. Утилита MOC ищет в заголовочных файлах на C++ описания классов, содержащие макрос Q_OBJECT, и создает дополни-

тельный исходный файл на C++, содержащий мета-объектный код.

Qt позволяет создавать собственные плагины и размещать их непосредственно в панели визуального редактора. Также существует возможность расширения привычной функциональности виджетов, связанной с размещением их на экране, отображением, перерисовкой при изменении размеров окна.

Qt комплектуется визуальной средой разработки графического интерфейса "Qt Designer", позволяющей создавать диалоги и формы "мышью" (в режиме WYSIWYG). В поставке Qt есть "Qt Linguist" — графическая утилита, позволяющая упростить локализацию и перевод программы на многие языки; и "Qt Assistant" — справочная система Qt, упрощающая работу с документацией по библиотеке, а также позволяющая создавать кросс-платформенную справку для разрабатываемого на основе Qt ПО. Начиная с версии 4.5.0 в комплект Qt включена среда разработки "Qt Creator", которая содержит редактор кода, справку, графические средства "Qt Designer" и дает возможность отладки приложений. "Qt Creator" может использовать GCC или Microsoft VC++ в качестве компилятора и GDB в качестве отладчика. Для Windows-версий библиотека комплектуется компилятором, заголовочными и объектными файлами MinGW.

Таким образом, платформа ПК, объединяющая все его логические и функциональные блоки, реализуется с использованием языка C++ и бесплатной библиотеки операций Qt, отличающейся обширным функционалом, позволяющим объединять разнородные программные компоненты, создавать средства визуализации данных и элементы графического интерфейса пользователя. Такой подход к созданию ПК позволил создать кросс-платформенную систему, объединяющую в себе такие подсистемы как СУБД и справочная подсистема, а также использующую лицензию GNU GPL.

Список литературы

1. 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 рентгенооптические методы и средства измерений в науке и технике. М.: Изд. ВНИИФТРИ, 2005. С. 16—26.
2. <http://www.crossplatform.ru/>
3. Емельянова Н. З., Паргыка Т. Л., Попов И. И. Основы построения автоматизированных информационных систем: учеб. пособие. М.: Инфа-М, 2008. 416 с.
4. Монахов В. Язык программирования Java и среда NetBeans. 2-е изд. СПб.: БХВ-Петербург, 2009. 720 с.
5. Шлее М. Qt 4.5. Профессиональное программирование на C++. СПб.: БХВ-Петербург, 2010. 896 с.

УДК 004.627

А. В. Замятин, канд. техн. наук, доц.,
Национальный исследовательский
Томский политехнический университет
e-mail: zamyatin@tpu.ru

Дифференцированное сжатие аэрокосмических изображений с потерями

Предложен подход к сжатию аэрокосмических изображений, основанный на дифференцированном сжатии сегментов с различным уровнем качества в зависимости от уровня их значимости. Найдены способы определения ключевых параметров подхода — зависимости уровня качества сжатия от уровня значимости сегмента и от степени сегментации, позволяющие варьировать в широких диапазонах степень сжатия, уровень качества и точность классификации восстановительного аэрокосмического изображения. Приведены результаты численных экспериментов, полученные с использованием аэрокосмических изображений с различными характеристиками, позволяющие сделать выводы об эффективности и пределах применимости предложенного подхода к сжатию и способов определения его ключевых параметров.

Ключевые слова: аэрокосмические изображения, дифференцированное сжатие, сжатие с потерями, сегментация, классификация

Введение

В связи со значительным ростом объемов получаемой аэрокосмической информации и необходимостью ее эффективной обработки, хранения и передачи по каналам связи все более актуальной становится задача сжатия этой информации [1, 2]. Учитывая необходимость автоматизированной обработки восстановленных после сжатия данных дистанционного зондирования Земли (ДЗЗ), широкое распространение получили подходы к сжатию без потерь, не искажающие статистические характеристики аэрокосмических изображений (АИ) [3, 4]. Основным недостатком этих подходов является сравнительно невысокая степень сжатия. Существенно более высокие показатели степени сжатия могут быть достигнуты при сжатии с потерями, но при этом существует риск значительного искажения статистических характеристик АИ, ведущего к сложностям применения автоматизированной классификации при оперативном

построении тематических карт исследуемой территории [5, 6]. Поэтому решение задачи сжатия данных ДЗЗ с высокой степенью сжатия (с потерями) с сохранением возможностей автоматизированной классификации восстановленных данных является крайне востребованным направлением исследований.

Одним из решений проблемы высокоэффективного сжатия с сохранением возможностей автоматизированной обработки могло бы быть дифференцированное сжатие, при котором различные фрагменты АИ сжимаются с различным уровнем потерь, обеспечивая минимальные статистические искажения наиболее значимых фрагментов и возможность их автоматизированной обработки (например классификации как одного из наиболее распространенных методов применения АИ) с приемлемым уровнем качества.

В настоящее время существует лишь несколько примеров результатов исследований, реализующих в некоторой степени принципы дифференцированного сжатия [7, 8, 9]. Предлагаемые подходы, главным образом, направлены на решение конкретного класса задач и не отличаются универсальностью (требуют априорной информации о ландшафтно-классовой структуре). В них использованы достаточно сложные принципы сегментации, затрудняющие практическую реализацию и требующие для восстановления АИ, дополнительно к сжимаемым данным, оперировать и результатами сегментации с произвольными границами. Например, в работе [7] предлагается алгоритм, предназначенный исключительно для бортовой реализации, выполняющий сегментацию на основе текстурного анализа, но обеспечивающий сжатие только интересующих фрагментов. Прочие фрагменты, которые в дальнейшем могли бы быть интересны, необратимо исключаются из рассмотрения. Кроме того, в таких исследованиях, как правило, отсутствуют результаты, демонстрирующие влияние сжатия с потерями на результаты автоматизированной классификации восстановленных данных [8, 9].

В связи с изложенным выше предлагается подход к дифференцированному сжатию АИ с потерями, обеспечивающий высокие показатели степени сжатия. Этот подход отличается значительной простотой сегментации, а также позволяет выполнять автоматизированную классификацию восстановленных данных с приемлемым уровнем качества.

Для практического использования доступны *панхроматические АИ*, полученные в одной широкой зоне спектра (одноканальные) и *мультиспектральные АИ*, полученные в нескольких спектральных диапазонах (мультиспектральные) [2, 10]. Рассмотрим сначала подход к дифференцированному сжатию одноканальных АИ.

Дифференцированное сжатие одноканальных АИ

Сегментация. Как отмечено выше, дифференцированный подход к сжатию АИ предполагает *сегментацию* изображения — разбиение на области, сходные по некоторой характеристике или свойству, при этом соседние области (сегменты) значительно отличные по этой характеристике или свойству. В каждом конкретном случае сегментация требует выполнения целого комплекса операций: определения требуемой характеристики или свойства сегментации; выполнения достаточно сложной процедуры обработки; сохранения результатов сегментации с произвольными границами нескольких областей для последующего использования при восстановлении сжатого изображения.

Для существенного упрощения процедуры сегментации и сокращения объема данных (что особенно значимо в задачах сжатия) предлагается в качестве границ сегментов $\mathbf{Sg}[r, c]$ использовать регулярную сеть с равным числом сегментов в строках и столбцах, причем $r = 1, 2, \dots, FR_{row}$, $c = 1, 2, \dots, FR_{col}$, где FR_{row} — число сегментов в строке изображения, FR_{col} — число сегментов в столбце изображения. В этом случае для определения границ сегментов некоторого АИ достаточно знать его коэффициент сегментации FR , определяемый здесь как $FR = FR_{row} = FR_{col}$. Такой подход универсален и при наличии на АИ однородных областей позволит выполнить их успешное выделение.

Информативность сегментов. Каждая из областей $\mathbf{Sg}[r, c]$, полученных в результате сегментации, содержит некоторый объем информации $I^{rc} \sim f_I(\mathbf{Sg}[r, c])$ [11]. При этом, как правило, чем больше значение I^{rc} , тем сегмент является более информативным и значимым при анализе. Однородные области имеют больше избыточной информации, менее информативны и имеют больший потенциал к высокой степени сжатия с потерями без значительного искажения данных [12]. Неоднородные сегменты имеют меньшую избыточность данных и более низкий потенциал к сжатию. Определив потенциал некоторого сегмента $\mathbf{Sg}[r, c]$ к сжатию, можно выполнить его сжатие с качеством $Q^{rc} \in (0; 1, 0]$ и степенью сжатия R^{rc} , т. е. справедливо $I^{rc} \sim Q^{rc} \sim f_{сж}(1/R^{rc})$. Зависимость $f_{сж}$ учитывает как особенности алгоритма сжатия с потерями, так и данных.

В качестве меры информативности используют различные метрики [11]. Применим в данном случае для определения зависимости f_I распространную характеристику — дисперсию D_x^{rc} сегмента. С учетом связи между I^{rc} и D^{rc} справедлива запись

$$I^{rc} \sim D^{rc} \sim Q^{rc} \sim (1/R^{rc}). \quad (1)$$

Для установления зависимости между элементами в выражении (1) при условии $I^{rc} = D^{rc}$ следует определить лишь зависимость f_d

$$D^{rc} = f_d(Q^{rc}), \quad (2)$$

формирующую связь между I^{rc} и Q^{rc} при сжатии данных сегмента.

Зависимость Q^{rc} от I^{rc} . Основными требованиями к зависимости (2) следует считать простоту практической реализации, а также возможность обеспечения в широких пределах параметрического дифференцирования сегментов по Q^{rc} и R^{rc} . Возможны различные варианты определения f_d . Например, с учетом отмеченных выше требований в случае *линейного варианта* зависимость f_d может быть определена как

$$Q^{rc} = \Delta \left[\frac{D_x^{rc} - D_{\min}}{D_{\max} - D_{\min}} \right] k + Q_{\min}, \quad (3)$$

где $Q^{rc} \in [Q_{\min}, Q_{\max}]$; Q_{\min}, Q_{\max} — заданные границы изменения качества сжатия с потерями для изображения в целом; $\Delta = (Q_{\max} - Q_{\min})$, D_{\min} и D_{\max} — минимальное и максимальное значения дисперсий, найденные для всех сегментов изображения; D_x^{rc} — дисперсия сегмента $\mathbf{Sg}[r, c]$; k — коэффициент дифференцирования. Коэффициент k позволяет варьировать силу влияния D^{rc} на Q^{rc} .

Для увеличения степени дифференцирования сегментов, позволяющего добиться более высоких показателей степени сжатия изображения при незначительных потерях, в выражении (2) следует использовать вид зависимости с существенной нелинейностью. Например, при *степенном варианте* такая функциональная зависимость может быть определена как

$$Q^{rc} = \Delta \left[\frac{D_x^{rc} - D_{\min}}{D_{\max} - D_{\min}} \right]^k + Q_{\min}. \quad (4)$$

В этом случае коэффициент k может обеспечивать более значительное дифференцирование сегментов по уровню качества сжатия.

Приведенные примеры могут быть основой для построения семейства аналогичных способов,

обеспечивающих в каждом конкретном случае необходимую связь f_d между D_x^{rc} и Q^{rc} .

Сжатие. Для реализации процедуры сжатия АИ с потерями возможно использование целого ряда алгоритмов, позволяющих для каждого из полученных сегментов $\mathbf{Sg}[r, c]$ применять сжатие с заданным уровнем качества Q^{rc} [12]. Одними из наиболее распространенных и подходящих в данном случае алгоритмов сжатия изображений с потерями являются алгоритмы, основанные на стандарте JPEG [15]. Этот стандарт хорошо известен и не требует здесь подробного изложения. Он предполагает сжатие каждого сегмента $\mathbf{Sg}[r, c]$ в три этапа.

1. *Дискретное косинус-преобразование* исходных данных сегмента, позволяющее сформировать низкочастотные и высокочастотные составляющие.

2. *Квантование*, задающее необходимый уровень качества Q^{rc} .

3. *Вторичное сжатие*, основанное на сжатии без потерь и применяемое к существенно разреженной матрице данных, полученной на предыдущем этапе.

Для получения восстановленного изображения необходимо выполнить аналогичные преобразования для каждого сегмента $\mathbf{Sg}[r, c]$ с заданным уровнем качества Q^{rc} в обратном порядке.

Поиск степени сегментации

Степень сегментации АИ может изменяться в широких пределах ($FR_i = 1, 2, \dots, FR_{\max}$) и существенно влиять на его степень сжатия R . Максимальная степень сегментации некоторого изображения может быть найдена как $FR_{\max} = \min\{\text{rows}, \text{cols}\}/\beta$, где *rows* и *cols* — число элементов изображения по горизонтали и вертикали; β — размер матрицы преобразования в алгоритме сжатия JPEG (как правило, принимают $\beta = 8$).

При значительной $FR = FR_i$ малый размер сегментов способствует тому, что в сегменте представлен один ландшафтный класс со статистически близкими значениями элементов, а значит D_x^{rc} будет невелика. И, наоборот, при небольшой $FR = FR_i$ размер сегмента будет существенным, его значения будут значительно отличаться, и D_x^{rc} будет сравнительно велико. Причем увеличение сжатия $R = R_i$ изображения при увеличении степени сегментации $FR_i = 1, 2, \dots, FR_{\max}$ не будет постоянным — слишком большое число сегментов усложняет оперирование этими данными. Поэтому, если необходимо обеспечить наивысшую степень сжатия, то для любого изображения можно найти та-

кую степень сегментации $FR_i = FR_{\text{опт}}$, при которой

$$R_i = \max_{i=1, FR_{\max}} \{R_i\} = R_{\max}.$$

Способ поиска степени сегментации $FR_{\text{опт}}$ предлагается осуществлять с использованием *метода золотого сечения*, широко используемого в различных задачах оптимизации [16]. Для краткости изложения алгоритм, реализующий этот способ, не приводится. Результатом работы алгоритма будет найденная степень сегментации $FR_{\text{опт}}$, при которой достигается наибольшая степень сжатия R_{\max} .

Дифференцированное сжатие многоканальных АИ

Изложенный подход к сжатию одноканальных АИ может быть адаптирован для применения к мультиспектральным АИ. В этом случае описанные выше преобразования применяются для каждого из L каналов мультиспектрального АИ в отдельности. Связь между информативностью каждого сегмента и уровнем качества его сжатия для каждого канала, по аналогии с выражением (2),

может быть определена как $D_l^{rc} = f_d(Q_l^{rc})$, где D_l^{rc} —

дисперсия, а Q_l^{rc} — качество сжатия для сегмента

$\mathbf{Sg}^l[r, c]$ в канале $l, l = 1, 2, \dots, L$. Следовательно, f_d в случае линейного варианта для канала l изо-

бражения может быть определена как $Q_l^{rc} =$

$$= \Delta^l \left[\frac{D_{x,l}^{rc} - D_{\min}^l}{D_{\max}^l - D_{\min}^l} \right] k^l + Q_{\min}^l, \text{ где } Q_l^{rc} \in [Q_{\min}^l, Q_{\max}^l],$$

Q_{\min}^l, Q_{\max}^l — заданные границы изменения ка-

чества сжатия с потерями; $\Delta^l = (Q_{\max}^l - Q_{\min}^l)$,

D_{\min}^l и D_{\max}^l — минимальное и максимальное

значения дисперсий, найденные для всех сегмен-

тов; $D_{x,l}^{rc}$ — дисперсия текущего сегмента $\mathbf{Sg}^l[r, c]$;

k^l — коэффициент дифференцирования. При степенном варианте зависимость f_d определяется ана-

$$\text{логично } Q_l^{rc} = \Delta \left[\frac{D_{x,l}^{rc} - D_{\min}^l}{D_{\max}^l - D_{\min}^l} \right]^{k^l} + Q_{\min}^l.$$

Эксперименты

Постановка задачи. Основной задачей исследований предлагаемого подхода к сжатию АИ должно быть определение его эффективности и пределов применимости с точки зрения влияния степени сегментации FR и степени дифференци-

рования сегментов на степень сжатия R и качество восстановленных изображений. Для оценки качества воспользуемся двумя критериями. Одним критерием может быть коэффициент корреляции $K_{XY} \in [-1; 1]$, используемый для определения степени соответствия исходного I_X и восстановленного I_Y изображений [10]. Применяемый подход к сжатию должен позволить получить не только высокую степень сжатия, но и сохранить возможности автоматизированной классификации восстановленных данных с приемлемым качеством. Поэтому второй критерий должен позволять проводить оценку результатов классификации исходного и восстановленного АИ, представленных в виде соответствующих тематических карт. Для такой оценки широко используют *каппа-индекс согласия* (КИС), который целесообразно применить при проведении данных исследований — $КИС \in [-1,0; 1,0]$ [10].

В качестве экспериментальных данных используем тестовые АИ растровой геоинформационной системы *Idrisi Kilimanjaro*, характеристики которых представлены в табл. 1, а также следующие параметры сжатия: $FR_{\text{опт}}^l = FR_{\text{опт}}^1$, $k^l = k^1$, $Q_l^{rc} = Q_1^{rc}$ для всех $l = 1, 2, \dots, L$, что оправдано вследствие значительной корреляции между данными, полученными в различных диапазонах. Произвольно зададим границы изменения $Q^{rc} \in [0,4; 0,95]$. Выполним классификацию, не требующую предварительного построения обучающих выборок (неконтролируемую, модулем ISOCLUST системы *Idrisi Kilimanjaro* с числом итераций — 100, числом кластеров — 5, минимальным размером выборки — 50).

Результаты. Анализ результатов экспериментов, полученных для одноканальных АИ при фиксированной $FR = 20$, показывает, что при линейной зависимости f_d существенное изменение R , K_{XY} и $КИС$ достигается при $k > 1$. При этом R уменьшается на 20...25 %, K_{XY} увеличивается на 3...5 %, $КИС$ — на 1...2 %. Очевидно, в этом случае наблюдается небольшая дифференцирующая сила линейной зависимости f_d для $k > 1$, а для $k \in [0; 1,0]$ — она практически незначительна.

Анализ результатов экспериментов, выполненных для степенного варианта зависимости f_d , показывает, что дифференцирующая сила f_d более существенная. В этом случае при $k \in [0,5; 2,0]$ степень сжатия R увеличивается на 30...35 %, а качество восстановленных изображений уменьшается (K_{XY} — на 10...12 %, $КИС$ — на 3...5 %). Исполь-

Таблица 1

Характеристики тестовых АИ

Номер изображения	Название	Размер изображения, пиксель	Размер файла, байт	Число каналов
1	etde	292 × 288	83	1
2	town	2971 × 1829	5307	1
3	etdem12	512 × 400	200	1
4	vinh345	534 × 300	157	1
5	eng	799 × 768	600	1
6	maur	256 × 256	64	7
7	sierra	700 × 934	639	6
8	mad	104 × 202	21	4

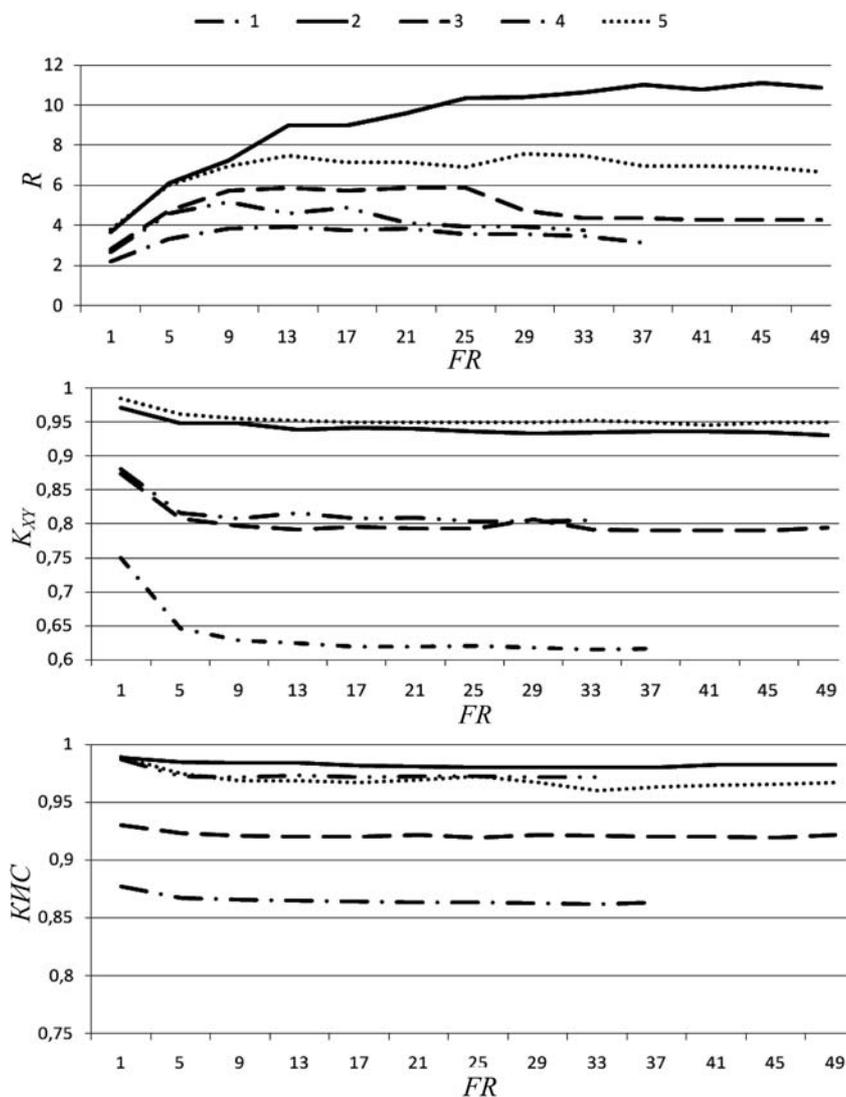


Рис. 1. Зависимость основных характеристик сжатия от FR при степенной f_d

зование степенной зависимости в сравнении с линейной при отмеченных k позволяет в более широких диапазонах варьировать R , K_{XY} и KIS .

Следует отметить, что результаты экспериментов показывают сравнительно небольшое уменьшение точности по KIS при более существенных отличиях I_X и I_Y по K_{XY} . Это объясняется тем, что многие

значения элементов изображения I_Y , восстановленные с некоторым отклонением (фиксируемым K_{XY}) от значений изображения I_X , при классификации будут отнесены к тому же ландшафтному классу, что и значения I_X .

Проведена оценка влияния FR на R , K_{XY} и KIS при линейном и степенном вариантах зависимости. При линейном варианте принято $k = 0,5$, при степенном — $k = 2,0$ (рис. 1).

Отметим, что $FR_i = 1$ соответствует результатам, полученным без использования сегментации. Анализ результатов экспериментов подтверждает, что R существенно зависит от FR . Процедура сегментации позволяет без значительной потери точности как по критерию K_{XY} (3...5%), так и по критерию KIS (1...2%) в 1,5—3 раза увеличить R .

Результаты экспериментов, фрагменты которых приведены на рис. 2, показывают, что предложенный подход применим и для мультиспектральных АИ. При этом степень сжатия и показатели качества при различном числе каналов L остаются практически неизменными.

Результаты оценки эффективности алгоритма поиска наилучшей степени сегментации относительно степени сжатия показали, что степень сжатия $R = R_i$ при различных $FR = FR_i$ может значительно варьироваться, и особенно существенно возрастает относительно результатов, полученных без сегментации. В табл. 2 приведены результаты поиска "наилучшей" степени сегментации для каждого из тестовых изображений.

Видно, что FR_{max} может быть значительной и поиск $FR_{opt} \in [1; FR_{max}]$ потребует многократного выполнения процедуры сжатия с соответствующими вычислительными затратами, что может негативно сказаться на практической применимости способа. Однако экспериментальная оценка вычислительных затрат алгоритма, проведенная для тестовых АИ, показала, что зависимость времени вычислений от размера и числа каналов L изображения близка к линейной и составляет без применения специальных средств оптимизации и снижения вычислительных

Таблица 2

Максимальная и "наилучшая" степень сегментации изображений

№ АИ	1	2	3	4	5	6	7	8
FR_{max}	36	228	50	37	96	32	87	13
FR_{opt}	18	114	25	18	30	16	43	6

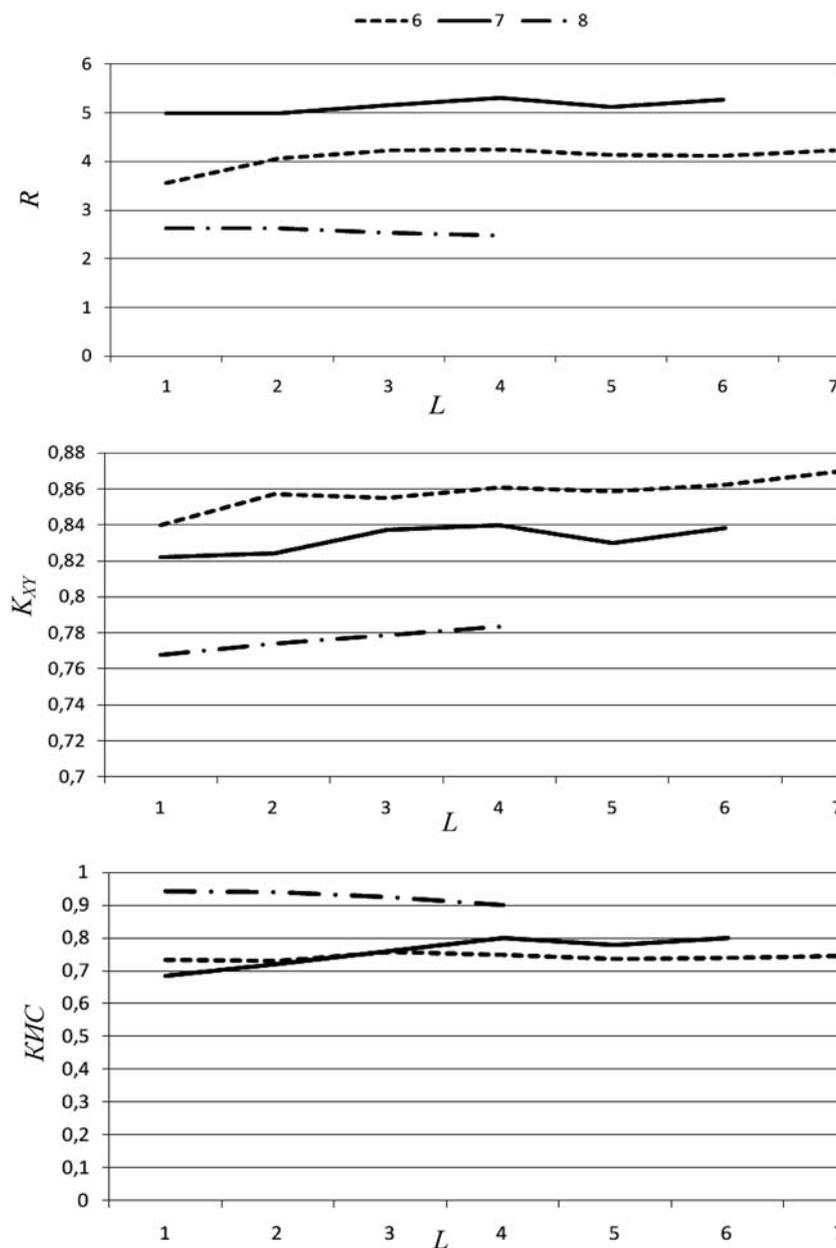


Рис. 2. Результаты экспериментов, полученные с использованием мультиспектральных АИ

затрат десятки секунд, что может быть признано приемлемым для практического применения. Результаты оценки вычислительных затрат алгоритма получены на ПЭВМ с процессором *Intel Pentium IV* 2,8 ГГц и объемом оперативной памяти 1 Гбайт под управлением операционной системы *Windows XP (SP 3)*.

Заключение

Для существенного увеличения степени сжатия значительных объемов аэрокосмической информации, повышающих эффективность ее хранения и передачи данных по каналам связи, предложен подход, основанный на дифференцированном сжатии сегментов с различным уровнем потерь в зависимости от уровня их значимости, и позволяющий осуществлять автоматизированную неконтролируемую классификацию восстановленных изображений с приемлемым уровнем качества.

Предложена упрощенная универсальная процедура сегментации аэрокосмического изображения, основанная на использовании регулярной сети с равным числом сегментов в строках и столбцах, которая оперирует не результатами сегментации с произвольными границами областей, а лишь коэффициентом сегментации и матрицей качества сжатия сегментов.

Найдены способы определения ключевых параметров подхода — зависимости уровня качества сжатия от уровня значимости сегмента, а также степени сегментации, позволяющие варьировать степень сжатия, уровень качества и точность классификации восстановленного аэрокосмического изображения в широких диапазонах.

Разработано программное обеспечение, реализующее предложенный подход к сжатию аэрокосмических изображений с различным числом каналов, в котором сжатие каждого из сегментов на основе найденного уровня его информативности осуществляется алгоритмом на основе стандарта JPEG.

Приведены результаты численных экспериментов, полученные с использованием одноканальных и мультиспектральных аэрокосмических изображений с различными характеристиками, позволившие экспериментально определить, что степенной вариант зависимости уровня качества сжатия от уровня значимости сегмента обеспечивает более существенное дифференцирование сегментов, чем линейный вариант такой зависимости. Показано, что степень сжатия с использованием сегмента-

ции может быть в 1,5–3 раза выше, чем без сегментации, при сопоставимом уровне точности как при сравнении исходного и восстановленного изображений по критерию корреляции, так и при сопоставлении результатов их неконтролируемой классификации с применением критерия каппа-индекс согласия. Определено, что отличия исходного и восстановленного изображений более существенные, чем результатов их классификации. Получены оценки вычислительных затрат алгоритма, реализующего способ поиска наилучшей степени сегментации относительно степени сжатия, составляющие на персональном компьютере десятки секунд, что приемлемо на практике.

Список литературы

1. **Motta G., Rizzo F., Storer J. A.** Hyperspectral Data Compression. Berlin: Springer, 2006. 415 p.
2. **Дистанционное зондирование: количественный подход / Ш. М. Дейвис, Д. А. Ландгребе, Т. Л. Филлипс и др.:** пер с англ. М.: Недра, 1983. 415 с.
3. **Cagnazzo M., Cicala L., Poggi G., Verdoliva L.** Low-complexity compression of multispectral images based on classified transform coding // *Signal Processing: Image Communication*. 2006. Vol. 10, N 21. P. 850–861.
4. **Kiely A., Klimesh M., Xie H., Aranki N.** ICER-3D: A Progressive Wavelet-Based Compressor for Hyperspectral Images // *The Interplanetary Network Progress Report*. 2006. P. 142–164.
5. **Zamyatin A.** Multistage algorithm for lossless compression of multispectral remote sensing images // *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vienna, Austria. 2010. Vol. 37, Part 7A. P. 304–309.
6. **Замятин А. В., Марков Н. Г.** Анализ динамики земной поверхности с использованием данных дистанционного зондирования Земли. М.: Физматлит, 2007. 176 с.
7. **Ancis M., Murrioni M., Giusto D. D., Petrou M.** Region-based remote-sensing image compression in the wavelet domain // *IGARSS'99 Proceedings*. IEEE 1999 International. 1999. Vol. 38, N 22. P. 1335–1337.
8. **Chang L.** Multispectral image compression using eigenregion-based segmentation // *J. of Pattern Recognition*. 2004. Vol. 37, N 6. P. 1233–1243.
9. **Бабкин В. Ф., Книжный И. М., Хрекин К. Е.** Сжатие мультиспектральных изображений для задач дистанционного зондирования Земли из космоса // *Современные проблемы дистанционного зондирования Земли из космоса*. М.: Изд. Института космических исследований РАН, 2004. С. 230–232.
10. **Richards J. A., Xiuping Jia.** Remote Sensing Digital Image Analysis: An Introduction. Berlin: Springer, 1999. 363 p.
11. **Шеннон К.** Работы по теории информации и кибернетике. М.: Наука, 1963. 829 с.
12. **Salomon D.** Data Compression: The Complete Reference Springer (4th ed.). 2007. 1092 p.
13. **Zamyatin A.** Advanced Processing of Remote Sensing Data for Land Use and Land Cover. LAP Lambert Academic Publishing AG & Co. KG, Saarbrücken, Germany, 2010. 232 p.
14. **Plaza A. J., Chang C.-I.** High Performance Computing in Remote Sensing. Chapman & Hall / CRC, 2008. 496 p.
15. **Pennebaker W. B., Mitchell J. L.** JPEG still image data compression standard (3rd ed.). Springer, 1993. 291 p.
16. **Коршунов Ю. М.** Математические основы кибернетики. М.: Энергоатомиздат, 1980. 424 с.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

УДК 65.011.46

Ю. В. Куприянов, аспирант, преподаватель,
Государственный университет —
Высшая школа экономики,
e-mail: yury.kupriyanov@gmail.com,

В. В. Таратухин,
канд. техн. наук, д-р философии, зав. каф.,
Государственный университет —
Высшая школа экономики;
директор программ партнерства
с высшими учебными заведениями
и научно-исследовательскими организациями
СНГ, Скандинавии и Восточной Европы, SAP CIS

Принципы управления выгодами на проектах внедрения корпоративных информационных систем

Внедрение корпоративных информационных систем (КИС) по своей сути есть проект организационной трансформации, требующий соответствующих методов управления. Тем не менее, руководители большинства ИТ-проектов по-прежнему придерживаются подхода, характеризующегося технологическим детерминизмом, и концентрируются на технологической составляющей проекта внедрения КИС. Планирование и получение запланированных выгод остается вне рассмотрения, поскольку по-прежнему сильно убеждение, что информационные технологии обладают набором неотъемлемых характеристик, обеспечивающих бизнес-выгоды.

Проанализированы особенности реализации выгод от инвестиций в информационные технологии, а именно КИС. На основе проведенного анализа формулируются принципы управления выгодами и предлагается концептуальная модель подхода к управлению выгодами на проектах внедрения КИС.

Ключевые слова: управление выгодами, ИТ-проект, бизнес-выгоды, оценка эффективности, внедрение КИС

Введение

Широкое распространение информационных технологий и повышение их значимости до критического уровня как для процессов управления предприятием [2], так и для проектирования и разработки наукоемкой продукции [5] обуславливает все возрастающие объемы инвестиций в ин-

формационные технологии (ИТ). При этом инвестиции в данный вид активов остаются одними из наиболее противоречивых с точки зрения обеспечиваемой ими отдачи [3]. Согласно исследованиям [8], трехкратный рост инвестиций в ИТ, а именно, в корпоративные информационные системы класса ERP, PLM, CRM в период с 1995 по 2005 гг. сказался на уровне монополизации, динамике отраслевого лидерства, а также на конкурентном разрыве участников рынка. Таким образом, за счет возможностей по относительно легкой репликации на всю компанию лучших практик, примененных первоначально на локальном уровне одного подразделения, и за счет предотвращения возврата к предшествующим, менее продуктивным практикам, ИТ являются ключевым фактором перехода к более динамичной и прозрачной экономике, но при этом характеризующейся высокой вероятностью формирования олигополистических рыночных структур с выраженным доминированием ожесточенно конкурирующих между собой компаний-лидеров. Очевидно, что значимая роль, которую играет ИТ на макроуровне, должна быть, в том числе, обусловлена эффектами, наблюдаемыми на микроуровне, уровне предприятий. Эксперты и исследователи в своем большинстве сходятся на том, что ИТ оказывают существенное влияние на принципы работы производственных и сервисных организаций [2, 4, 9]. Например, внедрение систем класса ERP приводит к изменениям организационно-функциональной структуры, корпоративной культуры, общего уровня производительности, а также функциональных обязанностей, системы мотивации и продуктивности работы отдельных сотрудников [6]. Для описания комплексного воздействия корпоративных информационных систем на предприятие используется специальный термин "техноорганизационная трансформация" — совокупность организационных изменений, обусловливаемых внедрением корпоративных информационных систем (КИС) [2, 10].

Особого внимания требует тот факт, что внедрение корпоративных информационных систем в почти идентичном функциональном объеме на схожих (с точки зрения зрелости бизнес-процессов, отраслевой принадлежности, общего уровня компетенций персонала) предприятиях может принести совершенно противоположные результаты [11]. Так, в одной компании ИТ-проект успешен: вне-

дренное решение становится инструментом обеспечения роста бизнеса и повышения эффективности операционной деятельности; в другой — провален: ИТ-решение становится причиной длительной остановки критических для бизнеса операций и последующих финансовых потерь. Число ИТ-проектов, подпадающих под второе описание, в течение последних 30 лет остается стабильно высоким [11]: в конце 70-х годов XX века только о 20 % ИТ-проектов можно было говорить как об успешных с точки зрения достижения бизнес-целей, в конце 80-х — более 70 % ИТ-проектов считались провальными, а на рубеже веков число неудачных ИТ-проектов приблизилось к 90 %. Причиной неудач ИТ-проектов в 75 % случаев становятся проблемы, вызванные не технологическими трудностями, а организационными изменениями, сопровождающими проект [9]. Парадокс состоит в том, что последние есть необходимое условие реализации бизнес-выгод от ИТ-проекта. При этом вопросу управления организационными изменениями уделяется непропорционально мало внимания. В абсолютном большинстве случаев по причине ложного восприятия ИТ как типичного инвестиционного актива, обладающего неотъемлемой ценностью, на проектах отсутствует практика оценки и системного управления выгодами [11, 13].

В связи с этим целью данной статьи является идентификация характеристик ИТ как объекта инвестиций, формулирование принципов реализации бизнес-выгод, разработка на их основе модели управления выгодами на проектах внедрения КИС.

ИТ как объект инвестиций

В то время как наличие ИТ, в частности, систем класса ERP является своего рода входным барьером для ведения крупного бизнеса с большими оборотами, а следовательно, и существенными потоками транзакционной информации [10], довольно часто встречается мнение, что внедренная система уже сама по себе есть источник бизнес-выгод, а в качестве аргумента приводятся результаты исследований, согласно которым внедрение ИС на базе решений SAP в среднем увеличивает капитализацию компании на 5 % (хотя данный рост есть лишь ожидания инвесторов, а не реальные бизнес-выгоды) [7]. О распространенности именно такого восприятия информационных систем (ИС) говорит повсеместное отсутствие системного подхода для управления реализацией выгод, тогда как для технологической части проекта внедрения КИС предусмотрена разработка детальных планов и концептуальных проектов [11, 14]. Такой подход не просто некорректен: он является основной причиной неумения уменьшающегося числа провальных ИТ-проектов [13], ибо игнорирует тот факт, что

бизнес-выгоды состоят не в обладании технологиями, а в их эффективном использовании [6].

Введем ключевые понятия, которыми будем оперировать в процессе формулирования принципов и разработки подхода по управлению выгодами: "бизнес-выгода" и "бизнес-результат". *Бизнес-результатом* ИТ-проекта мы будем называть измеримый результат, который может изменяться как по ходу, так и в результате реализации ИТ-проекта. *Бизнес-выгодой* в рамках данной статьи будем считать положительный бизнес-результат, превышающий средние аналоги и эквиваленты¹, необходимость достижения которого обусловлена причиной реализации ИТ-проекта.

Попытка извлекать бизнес-выгоду из ИТ как из типовых объектов инвестирования (недвижимость, оборудование и т. п.) неприемлема, ибо понятие выгоды, или ценности, как таковое, не применимо к ИТ. Мы можем говорить о потенциальных бизнес-выгодах от выстроенной на основе внедренных ИТ информационной системы (под которой в данном случае мы понимаем: организационно-техническую систему, использующую информационные технологии в целях обучения, информационно-аналитического обеспечения научно-инженерных работ и процессов управления [1]), которая, в свою очередь, обеспечивает ИТ-сервисы, целевое использование которых и приводит к созданию бизнес-выгод через увеличение прибыли или снижение издержек. Целевое использование вновь созданных ИТ-сервисов есть организационно-функциональное изменение, необходимое для реализации бизнес-выгод. Таким образом, бизнес-выгоды могут быть реализованы за счет успешного достижения бизнес-целей, что возможно только путем проведения соответствующих организационных изменений.

Выделение принципов реализации бизнес-выгод ИТ-проекта

Из сказанного выше следует, что должен быть предусмотрен набор действий по сопровождению организационных изменений, обеспечивающих достижение целевого бизнес-результата. Данный набор действий принято называть "управление выгодами" [6, 11, 13]. В рамках ИТ-проекта управление выгодами должно быть представлено как отдельный поток действий, ассоциированных с оптимизацией бизнес-процессов, внедрением новых принципов работы, развитием новых способов взаимодействия как внутри, так и за пределами предприятия. При этом должна быть обеспечена

¹ Дополнительные ограничения на тип бизнес-результата, отвечающему категории "бизнес-выгода", были введены из соображения учитывать альтернативные издержки и результаты конкурентного анализа (benchmarking).

взаимосвязь между развертыванием технологий и управлением выгодами. На основе проведенного обзора литературы и опыта участия в проектах внедрения КИС авторами в рамках данной работы систематизированы принципы реализации бизнес-выгод ИТ-проекта, которым должен отвечать подход по управлению выгодами.

1. Факт владения ИТ не несет ценности: для извлечения бизнес-выгод необходимо обеспечить эффективное использование внедренных технологий.

ИТ, находящиеся в собственности компании, в отличие от традиционных инвестиционных активов (например, производственные мощности, недвижимость, драгоценные металлы), не создают ценности. Более того, ИТ несут в себе затраты на поддержку, что также не типично для традиционных активов. Только эффективное использование ИТ может (опосредованно) обеспечить получение бизнес-выгод.

2. Реализация ключевых бизнес-выгод от внедрения ИТ требует изменений рабочих практик сотрудников и принципов функционирования предприятия.

Выгоды появляются как результат применения более совершенных управленческих методов, основанных на полной, целостной, актуальной и непротиворечивой информации. В этом отношении ИТ (опосредованно через ИС) обеспечивает ИТ-сервисы, позволяющие сотрудникам более эффективно работать с информацией: осуществлять сбор, обработку, хранение, непрерывно иметь доступ и анализировать в необходимом разрезе в режиме настоящего времени [13], а также расширяет их возможности по взаимодействию внутри и за пределами организации. Таким образом, ИС/ИТ выступают источником новых возможностей по работе с информацией, в то время как реализовывать эти возможности могут только сотрудники, готовые внести изменения в свои рабочие привычки и практики.

3. Ответственность за реализацию бизнес-выгод ИТ-проекта имеет "солидарный" характер и должна быть возложена на всех участников.

Исходя из сформулированного выше принципа № 2: для реализации выгод необходимо изменение поведения сотрудников — проектная команда не может нести полную ответственность за реализацию бизнес-выгод ИТ-проекта. В ИТ-проекте должны создаваться определенные условия для взаимодействия сторон, и стороны, участвующие в нем, несут равную ответственность за результаты проекта, признание этого принципа организацией-заказчиком может значительно повысить шансы на достижение заявленных бизнес-выгод [11].

4. Целями реализации ИТ-проекта должны быть только бизнес-выгоды, а не любой бизнес-результат.

Обращая пристальное внимание на реализацию положительных эффектов, не стоит забывать, что

проект может иметь и негативные результаты [13], следовательно, усилия участников проекта должны быть направлены не только на реализацию бизнес-выгод, но и на предупреждения и минимизацию негативных бизнес-результатов (отрицательных бизнес-выгод) проекта.

5. Реализация бизнес-выгод может быть проведена только при условии системного управления выгодами во время и после завершения ИТ-проекта.

Таким образом, пятый принцип привел нас к тому, с чего был начат данный раздел. Реализация бизнес-выгод от ИТ-проекта требует специальных действий со стороны команды проекта и сотрудников организации заказчика. Процесс управления выгодами должен носить системный характер и отвечать соответствующим требованиям [1]. Он не должен завершаться сразу после внедрения технологии; его следует продолжать до момента реализации всех запланированных бизнес-выгод или явного обоснования невозможности их достичь [6]. Логичным продолжением обозначенного принципа является необходимость включения в процесс управления выгодами ключевых точек "отката", когда, несмотря на произведенные инвестиции, целесообразно отменить реализацию ИТ-проекта ввиду накопленных диспропорций в финансовом и операционном плане.

Построение модели управления выгодами ИТ-проекта

На основе выделенных принципов реализации бизнес-выгод авторами предложена модель ИТ-проекта (рис. 1).

Поскольку основным предметом рассмотрения данной статьи является управление выгодами, то мы остановимся на части модели, посвященной элементам этого процесса, а именно: бизнес-кейс, планирование реализаций выгод, управление реализацией выгод и оценка и контроль выгод.

Бизнес-кейс. В контексте предложенной модели бизнес-кейс является отправной точкой для процесса управления выгодами и внедрения самих технологий, так как с помощью бизнес-кейса выполняется обоснование необходимости выделения средств на ведение всего проекта. Основные задачи его построения — формирование исчерпывающего перечня планируемых выгод ИТ-проекта и демонстрация соответствия проекта стратегическим целям компании. Наиболее распространенным методом решения задачи формирования бизнес-кейса является модель отдачи от инвестиций (ROI¹-модели) [2—4, 12], которая, однако, обладает одним существенным недостатком: в подобных моделях учитываются исключительно фи-

¹ROI (англ., *return on investment*) — отдача от инвестиций.

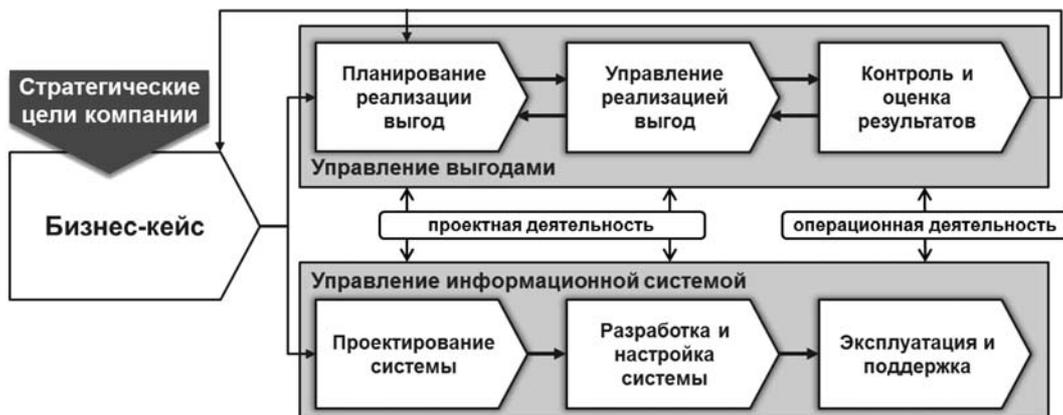


Рис. 1. Концептуальная модель ИТ-проекта

		ХАРАКТЕР ВОЗДЕЙСТВИЯ НА БИЗНЕС		
		Создание новых возможностей	Повышение эффективности	Отказ от операций
СТЕПЕНЬ ИЗМЕРИМОСТИ	Финансовые			
	Количественные			
	Измеримые			
	Качественные			

Рис. 2. Матрица структурирования выгод, адаптировано из [14]

нансовые выгоды. Авторы данной статьи считают, что бизнес-кейс должен отражать различные виды бизнес-выгод, а не единственно финансовые. В то же время все выгоды должны быть представлены и оценены в таком формате, чтобы можно было проконтролировать факт их реализации.

Другой важной характеристикой бизнес-кейса является явное указание на характер изменений в деятельности предприятия, связанных с реализацией соответствующей бизнес-выгоды. В наибольшей степени этим требованиям отвечает так называемая матрица структурирования выгод [14]. В рамках дачного подхода предлагается классифицировать все выгоды в разрезе двух категорий: степени измеримости бизнес-выгоды и характера изменений в бизнесе организации. По первой категории введены следующие типы выгод (в порядке от субъективных качественных к объективным количественным): качественные, измеряемые, количественные, финансовые.

Соответственно, по второй категории выделяют три типа воздействия на бизнес: введение новых операций и создание новых возможностей; повышение эффективности существующих операций; отказ от потерявших актуальность существующих операций.

Таким образом, в результате анализа каждая из идентифицированных бизнес-выгод должна быть размещена в одной из 12 ячеек (рис. 2). Как показывает практика, подобное структурированное представление выгод гораздо нагляднее линейного списка и порождает больший интерес участников проекта, детальные обсуждения и высокое качество оценки бизнес-выгод. Кроме того, выгоды, выраженные в финансовых терминах, могут быть использованы для построения ROI-модели и расчета инвестиционных показателей ИТ-проекта.

Планирование реализации бизнес-выгод. Данный процесс представляет собой совокупность действий, направленных на достижение бизнес-выгод ИТ-проекта, сформулированных в бизнес-кейсе [1, 6]. Ключевым моментом разработки плана является указание полной совокупности организационно-технического обеспечения достижения каждой из заявленных бизнес-выгод [11].

Построение плана реализации выгод осуществляется следующим образом (рис. 3): каждой бизнес-выгоде ставится в соответствие описание необходимого организационного изменения и совокупность действий и технологий (функциональность внедряемой КИС) для его реализации. При этом, в соответствии с выделенным принципом № 3, назначаются ответственные за реализацию орга-

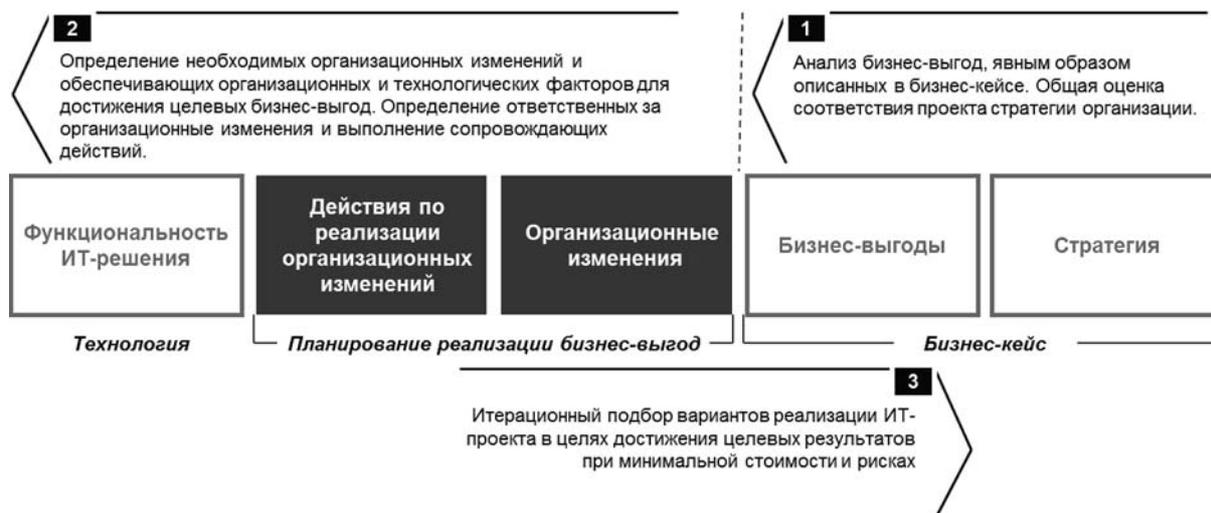


Рис. 3. Модель планирования реализации выгод, адаптировано из [11]

низационных изменений на стороне агента, принимающего проект, и за выполнение обеспечивающих это изменение действие — со стороны агента, реализующего проект. После того как был сформулирован первоначальный набор организационных мер и технологических факторов, рекомендуется провести несколько итераций для формирования оптимального набора мероприятий — шаг 3 (рис. 3).

Важным элементом предлагаемого подхода является наличие явной связи между действиями по управлению выгодами и собственно внедрением технологий, призванных стать проводником планируемых к реализации бизнес-выгод.

Управление реализацией бизнес-выгод. С точки зрения функциональной модели цикла управления, подпроцесс "управление реализацией бизнес-выгод" соответствует традиционному процессу "оперативное управление" [1] и направлен на обеспечение реализации бизнес-выгод в рамках разработанного плана реализации выгод и бизнес-кейса. Исходя из принципа № 2 управление реализацией бизнес-выгод в качестве основного элемента включает в себя осуществление необходимых организационных изменений. Важно отметить, что данный подпроцесс инициируется сразу после утверждения плана по реализации бизнес-выгод и находится в активном состоянии до момента завершения всех организационных изменений, сопровождающих проект внедрения КИС [6].

Данный подпроцесс, так же как и предшествующий, в большей степени характерен для реализации в рамках проекта, а значит, подпадает под типовые проектные ограничения: время, стоимость, объем.

Контроль и оценка результатов. Данный подпроцесс представляет собой систему действий, обеспечивающих определение состояния хода про-

цесса управления выгодами, оценку степени отклонения текущих результатов от запланированных в бизнес-кейсе и плане реализации выгод. Основные задачи, реализуемые в рамках контроля и оценки результатов, — сбор данных, оперативная оценка состояния ИТ-проекта с точки зрения реализуемых бизнес-выгод, а также корректировка плана реализации последующих выгод на основе информации о фактическом ходе проекта.

Стоит особо отметить роль, которую играет бизнес-кейс в рамках данного подпроцесса, — на основе информации из этого документа проводится анализ отклонений фактических значений реализованных бизнес-выгод от плановых. За счет реализации обратной петли управления осуществляется концепция самообучающейся организации, что способствует дальнейшему становлению и развитию практики управления выгодами внутри организации.

Использование бизнес-кейса, в том числе как инструмента контроля, обуславливает еще одно требование к процессу управления выгодами и бизнес-кейсу, в частности адаптируемость. При реализации процесса управления может возникнуть необходимость вносить изменения в исходные предпосылки бизнес-кейса, с тем чтобы ключевой инструмент мониторинга был актуален. В то же время, должно быть четко определено, что является источником производимых изменений, и отражено в финальных расчетах по итогам реализации проекта и оценке достигнутых результатов.

Выводы

При планировании ИТ-проекта необходимо формулировать его результаты в терминах бизнес-выгод, которые должны быть реализованы, а не в терминах технологий, которые должны быть вне-

дрены. Корректная разработка и настройка ИТ-решения является необходимым, но не достаточным условием для реализации запланированных бизнес-выгод ИТ-проекта.

Учитывая представленные в статье данные по числу провальных ИТ-проектов, можно заключить, что рассматриваемая проблема выделения принципов и построения модели управления выгодами ИТ-проектов весьма актуальна. Несмотря на то, что предложенная модель носит концептуальный характер, она дает предельно четкое представление о том:

- каковы основополагающие принципы управления выгодами ИТ-проекта;
- каково место процесса управления выгодами внутри ИТ-проекта и относительно работ по внедрению ИТ-решения;
- какова структура процесса управления выгодами ИТ-проекта;
- каковы ключевые задачи элементов (подпроцессов) управления выгодами и как эти элементы связаны между собой.

Следовательно, предложенная модель может служить базой для формирования подхода к управлению выгодами на реальном проекте внедрения КИС.

Дальнейшее направление развития исследования по данной тематике авторы видят в разработке следующих направлений:

- оценка применимости предложенного метода в рамках ограничений реального проекта;
- создание математической модели выбора оптимального набора методов обеспечения реализации бизнес-выгод проекта.

Список литературы

1. **Анфилов В. С., Емельянов А. А., Кукушин А. А.** Системный анализ в управлении: учебн. пособие / Под ред. А. А. Емельянова. М.: Финансы и статистика, 2009. 368 с.
2. **Бэккер Й., Вилков Л., Таратухин В., Кугелер М., Роземан М.** Менеджмент процессов. М.: Эксмо, 2007. 384 с.
3. **Каргин А. Ю., Лигачева Л. А.** Проблема оценки эффективности инвестирования в информационные технологии // Россия: перспективы развития: Научные труды ИМПЭ им. А. С. Грибоедова. М.: Буквояд, 2009.
4. **Лихварев А. С., Асеева Н. В.** Оценка инвестиционной привлекательности ИТ-проектов с учетом особенностей современных подходов к архитектуре программных комплексов // Бизнес-информатика. 2008. № 4. С. 3—12.
5. **Норенков И. П., Кузьмик П. К.** Информационная поддержка наукоемких изделий. CALS-технологии. М.: МГТУ им. Н. Э. Баумана, 2002. 320 с.
6. **Ashurts C., Doherty N. L., Peppard J.** Improving the impact of IT development projects: the benefits realization capability model // European Journal of Information Systems. 2008. N 7. P. 352—370.
7. **Bharadwaj A., Keil M., Mähring M.** Effects of information technology failures on the market value of firms // Journal of Strategic Information Systems. 2009. N 4. P. 3—16.
8. **Brynjolfsson E., Saunders A.** Wired for Innovation: how information technology is reshaping the economy. Cambridge, MA: MIT Press. 2010. 160 p.
9. **Kohnke O.** The Human Side of SAP Implementations: How Effective Change Management Works, Results of an European Study 2006, SAP AG, 2006.
10. **Markus M. L.** Technochange management: using IT to drive organizational change // Journal of Information Technology. 2004. N 19. P. 4—20.
11. **Peppard J., Ward J., Daniel J.** Managing the realization of business benefits from IT investments // MIS Quarterly Executive. 2007. N 6/1. P. 1—11.
12. **Stefanou C.J.** A framework for the ex-ante evaluation of ERP software // European Journal of Information Systems. 2001. N 4(10). P. 201—215.
13. **Tiernan C., Peppard J.** Information Technology: of Value or a Vulture? // European Management Journal. 2004. N 6(22). P. 609—623.
14. **Ward J., Daniel E., Peppard J.** Building a better business case for IT investment // California Management Review. 2007. N 8. P. 3—27.

2011
CEE-SEC(R)
Разработка ПО
31 октября 1-3 ноября

Конференция
«Разработка ПО 2011/
CEE-SEC(R)»
Москва

Программный Комитет CEE-SEC(R) 2011 объявляет о начале приема заявок на выступления по следующим направлениям:

- Исследования/Технологии
- Человеческий капитал и Образование
- Практика разработки ПО
- Бизнес и Предпринимательство

*Отбор докладов осуществляется Программным Комитетом Конференции

Телефон орг. комитета: +7 (812) 336 93 44 www.secr.ru

УДК 533.9.08;004.771

А. М. Зимин, д-р техн. наук, проф.,
А. В. Шумов, канд. техн. наук, ст. науч. сотр.,
С. Е. Кривицкий, аспирант,
В. И. Тройнов, студент,
Московский государственный технический
университет имени Н. Э. Баумана,
e-mail: lud2002@bmstu.ru

Автоматизированная сетевая учебно-научная лаборатория по спектроскопии плазмы

Представлена интегрированная Интернет-лаборатория, позволяющая дистанционно проводить на сложном оборудовании как лабораторные практикумы, так и полноценные научные исследования, что особенно актуально для федеральных и национальных исследовательских университетов. Описана автоматизированная диспетчерская система, позволяющая осуществлять заказ и выполнение экспериментов, сохранение и обработку их результатов, администрирование работы лаборатории.

Ключевые слова: Интернет-лаборатория, удаленный доступ, спектроскопия плазмы

Введение

В учебном процессе и научных исследованиях, проводимых учреждениями высшего профессионального образования, все большую роль играют современные информационные и телекоммуникационные технологии (ИКТ). Особое значение приобретают ИКТ в федеральных и национальных исследовательских университетах, где в соответствии с основными принципами их построения организуются специализированные центры коллективного пользования (ЦКП) приобретенным уникальным экспериментальным оборудованием. Если при выполнении Приоритетного национального проекта "Образование" в соответствии с разработанной ранее концепцией [1–5] в сетевых ЦКП основное внимание уделялось учебному процессу (см., например [6]), на современном этапе не менее важной является научно-исследовательская составляющая. Поэтому ЦКП, основанные на сетевом использовании уникального оборудования, при разработке методических ма-

териалов и интерфейсов удаленного управления должны предусматривать различные возможности для проведения учебных и научных экспериментов. Минимальный список параметров, присутствующих в инструментах формирования учебных сценариев эксперимента, и их диапазоны изменения при выполнении научных опытов должны быть расширены, однако при этом не должна пострадать степень защиты уникального оборудования от возможных неверных действий удаленного пользователя.

В течение первого десятилетия XXI века в Московском государственном техническом университете имени Н. Э. Баумана (МГТУ им. Н. Э. Баумана) было создано несколько лабораторий удаленного доступа (ЛУД) [3, 5–9]. Их демонстрационные версии размещены на специализированном портале ЛУД МГТУ им. Н. Э. Баумана (<http://lud.bmstu.ru>) и на Всероссийском сервере автоматизированных лабораторных практикумов с удаленным доступом (<http://www.alpud.ru>).

Одна из первых ЛУД [7] была посвящена выполнению сетевого практикума с удаленным доступом по диагностике плазмы, который успешно функционировал в течение 8 лет, однако к настоящему времени его аппаратная база физически и морально устарела. Поэтому в последние годы нами создана интегрированная сетевая лаборатория по спектроскопии плазмы учебно-научного назначения, основанная на распределенном управлении современным спектрометром AvaSpec-2048 с CCD-детекторной линейкой. Рассмотрим особенности организации проведения учебных и научных удаленных экспериментов в этой автоматизированной лаборатории.

Учебная Интернет-лаборатория "Спектрометрия плазмы"

Для проведения сетевых лабораторных практикумов, совместимых с системой ИНДУС [4], разработан программно-аппаратный комплекс [10], ориентированный на удаленное управление экспериментальным оборудованием (рис. 1). Его основу составляет многоканальный спектрометр AvaSpec-2048, связанный с персональным компьютером посредством USB-интерфейса.

Управление электронными системами спектрометра осуществляется с помощью электронной платформы AS-161 со встроенным микропроцес-



Рис. 1. Схема проведения эксперимента по спектроскопии плазмы через глобальную сеть

сором, которая поддерживает также обмен командами и данными с управляющим компьютером. Используемая нами конфигурация прибора рассчитана на видимый диапазон длин волн (355...810 нм) и включает четыре независимых канала, различающихся диапазонами регистрации излучения, характеристиками используемых дифракционных решеток, размерами щелей, фильтрами для обрезания гармоник второго порядка.

Управляющий работой спектрометра компьютер (Lab-сервер) связан локальной сетью с Web-сервером, доступным из сети Интернет. При проведении сетевого эксперимента с помощью интерфейса удаленного управления, генерируемого с помощью PHP-скриптов в виде динамических HTML-страниц, задаются параметры режима работы спектрометра, которые после проверки на корректность заносятся в интегрированную базу данных и затем реализуются.

Структура и взаимодействие компонентов программно-аппаратного комплекса. Разработанное программное обеспечение (рис. 2, см. четвертую сторону обложки) поддерживает функционирование подсистем регистрации и авторизации пользователей, подсистем обучения и контроля знаний, пользовательские интерфейсы удаленного управления, доступа и сетевой обработки результатов проведенных экспериментов, рассчитанные на использование Web-браузера, а также интегрированную базу данных [9], реализованную под управлением MySQL. Windows-служба SpectrService, разработанная в среде Borland Delphi и размещенная на Lab-сервере, периодически опрашивает таблицы базы данных, в которых хранится информация об отправленных удаленными пользователями запросах на регистрацию спектров, и ставит

соответствующие сценарии в очередь на выполнение. При этом удаленному экспериментатору предоставляются оперативные сообщения о ходе проведения эксперимента ("сценарий поставлен в очередь на выполнение", "проводится регистрация спектра" и т. п.).

Windows-служба по USB-интерфейсу задает режим работы спектрометра в соответствии с указанными удаленным пользователем параметрами и по завершении регистрации спектра обеспечивает сохранение результатов в четырех файлах (их число соответствует числу каналов спектрометра) на диске Web-сервера (протокол SMB), а в базу данных заносятся только имена файлов с результатами. Такой подход позволяет реализовать на спектрометрическом комплексе многопользовательский режим работы, что особенно важно при проведении учебных удаленных практикумов для групп студентов, находящихся в компьютерных классах своего вуза.

Доступ к результатам экспериментов осуществляется с помощью соответствующих PHP-скриптов, которые после обращения к базе данных предоставляют экспериментатору результаты в графическом и табличном видах как во время проведения сеанса управления, так и по его завершении. Созданное программное обеспечение также предоставляет дополнительные возможности для обработки данных с учетом специфики источника излучения.

Интегрированная база данных содержит информацию о перечне и атрибутах (название, аннотация, учебная дисциплина, адрес сервера удаленного управления, контактные данные), имеющихся в Интернет-лаборатории автоматизированных лабораторных практикумов с удаленным доступом (АЛП УД); об удаленных пользователях (личные сведения, уровень и права пользователя, принадлежность к какому-либо образовательному учреждению и учебной группе, статистика и атрибуты участия в проведении конкретных АЛП УД, результаты тестирований и полученные оценки); о сеансах удаленного управления оборудованием конкретным пользователем (время, дата и длительность проведенных сеансов, сценарии и условия экспериментов, ссылки на файлы результатов); о статистике загрузки и режимах работы уникального оборудования, входящего в Интернет-лабораторию.

Исследуемые источники излучения. В лабораторных практикумах используют источники излучения, позволяющие регистрировать сплошные, линейчатые атомные или состоящие из спектральных полос и линий молекулярные спектры разрядов без пространственного разрешения. Для анализа спектров применяют различные спектральные лампы (AvaLight-CAL, OSRAM, ДДС-30, набор

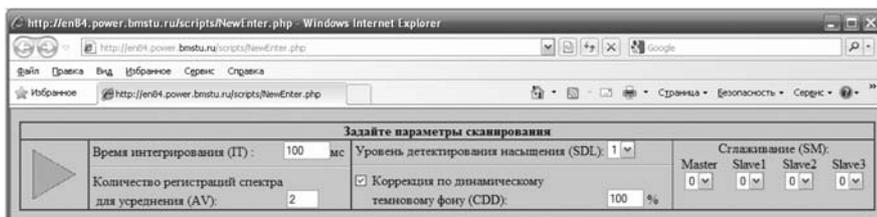


Рис. 3. Интерфейс удаленного управления спектрометром

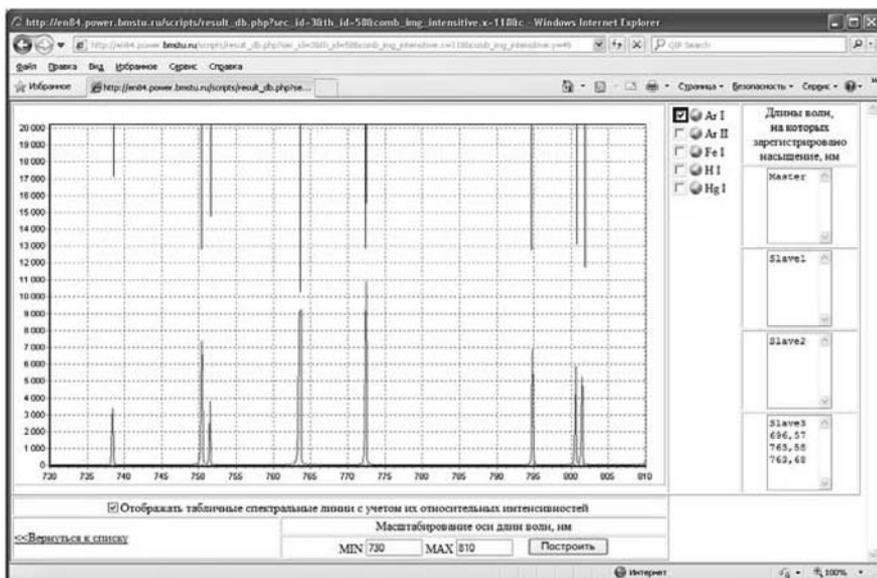


Рис. 4. Интерфейс сетевой обработки результатов эксперимента

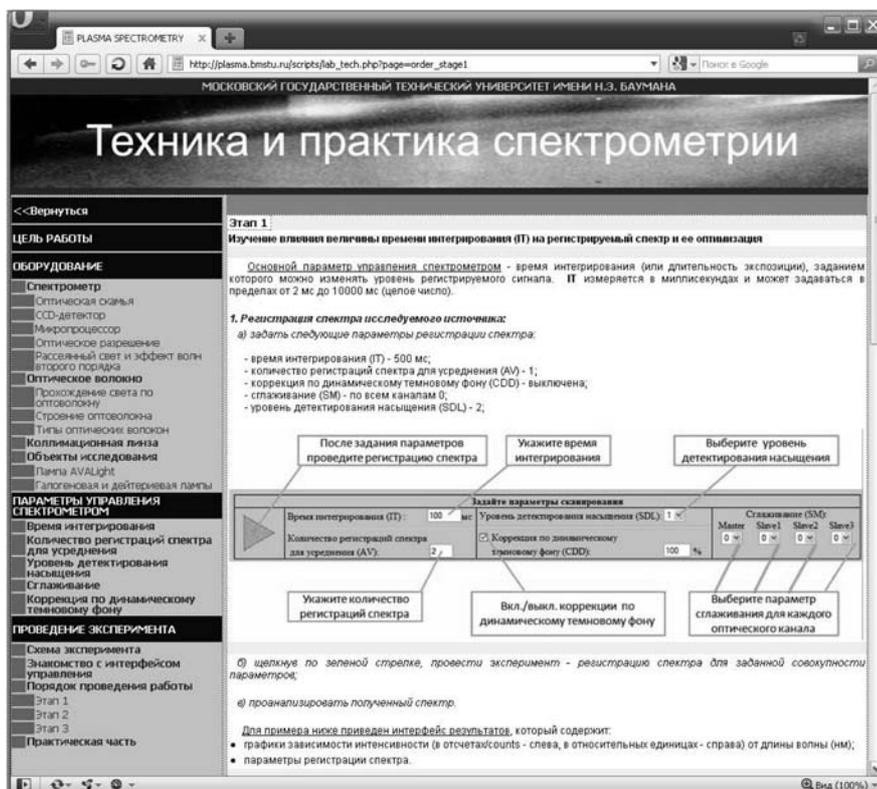


Рис. 6. Страница методики проведения лабораторной работы по технике и практике спектрометрии

ламп ВСБ-2 с различным наполнением), широко используемые для калибровки различных интерферометрических и спектрофотометрических устройств.

Удаленное проведение учебных экспериментов. Сетевое управление подсистемами спектрометра и сетевой анализ результатов регистрации спектров плазмы реализованы с помощью динамического HTML. Сгенерированный с помощью PHP-скриптов интерфейс удаленного пользователя (рис. 3) позволяет изменять основные параметры регистрации спектра: время интегрирования (*Integration time*) и значение усреднения (*Average*), задавать параметры сглаживания (*Smoothing*), включать, выключать и настраивать режимы коррекции по динамическому темновому фону (*Correct for dynamic dark*) и детектирования насыщения (*Saturation Detection Level*).

Интерфейс сетевой обработки результатов (рис. 4) предназначен для анализа первичных данных как во всем регистрируемом диапазоне длин волн, так и на отдельных его участках, что актуально при большой густоте спектральных линий. Пользователю выводится информация о насыщенных спектральных линиях — длины волн, на которых произошла засветка пикселей CCD-линейки (на рис. 4 — справа), а интенсивность сигнала определена с ошибкой. При проведении одного из практикумов (по диагностике плазмы) интерфейс обработки данных предусматривает также отображение табличных спектральных линий из атласа (на рис. 4 — сверху), что значительно облегчает анализ состава излучающей плазмы. Для нанесения линий элементов использована база данных Национального института стандартов и технологии США — <http://www.nist.gov>.

Сайт сетевой лаборатории (<http://plasma.bmstu.ru>) доступен в рабочее время для регистрации пользователей и подачи заявок

на проведение практикумов, а также позволяет проводить в согласованное с администрацией лаборатории время лабораторные практикумы и исследовательские работы и осуществлять обработку первичных данных.

Изучение полного комплекта методических пособий, подготовленных и структурированных с помощью современных Интернет-технологий, может быть выполнено предварительно как на сайте лаборатории, так и с использованием демо-версии. Переход к соответствующим разделам осуществляется через главное меню.

Электронные пособия содержат краткие теоретические основы, необходимую информацию об используемом оборудовании (рис. 5, см. четвертую сторону обложки) и исследуемых источниках излучения, перечне и пределах изменения параметров управления системой диагностики, методиках и последовательности измерений (рис. 6).

Автоматизированная диспетчерская система. Заказ на проведение удаленного учебного эксперимента, поддержка его проведения и анализ результатов обеспечивается специально созданной диспетчерской системой. После регистрации на сайте лаборатории и согласования по электронной почте условий проведения практикума пользователь может через свой личный кабинет оставить заявку на проведение выбранной им лабораторной работы в удобное для него время (рис. 7).

Администратор лаборатории просматривает поступившие заявки и заносит их в график экспериментов, после чего управление выполнением экспериментов и сохранением их результатов происходит в автоматическом режиме. В обусловленное в графике экспериментов время система разрешает формирование сценария и задание параметров удаленного эксперимента. Доступ к сетевому пульту управления стендом становится возможным лишь после успешного прохождения теста об усвоении опубликованных на сайте методических материалов. Контроль за допустимыми значениями

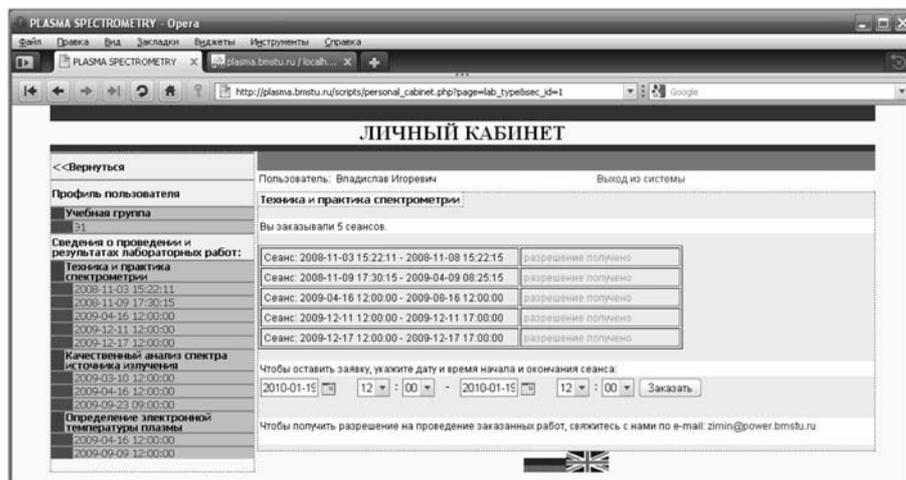


Рис. 7. Заказ сеанса удаленного прохождения практикума в личном кабинете

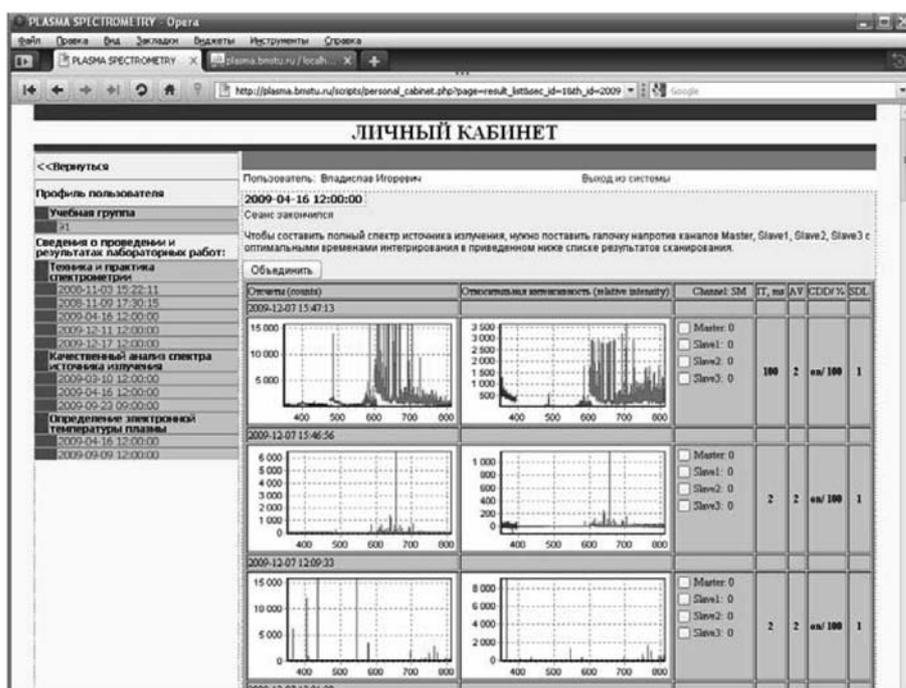


Рис. 8. Личный кабинет удаленного пользователя

условий эксперимента и его осуществимостью также осуществляется автоматически.

По окончании опыта удаленный экспериментатор через личный кабинет получает доступ к его результатам, а также к инструментам для обработки первичных данных (рис. 8).

Личный кабинет преподавателя. Кроме индивидуальной учебной работы в Интернет-лаборатории предусмотрено также и групповое проведение практических занятий. В этом случае преподаватель после регистрации на сайте получает более высокий уровень доступа и у него появляется воз-

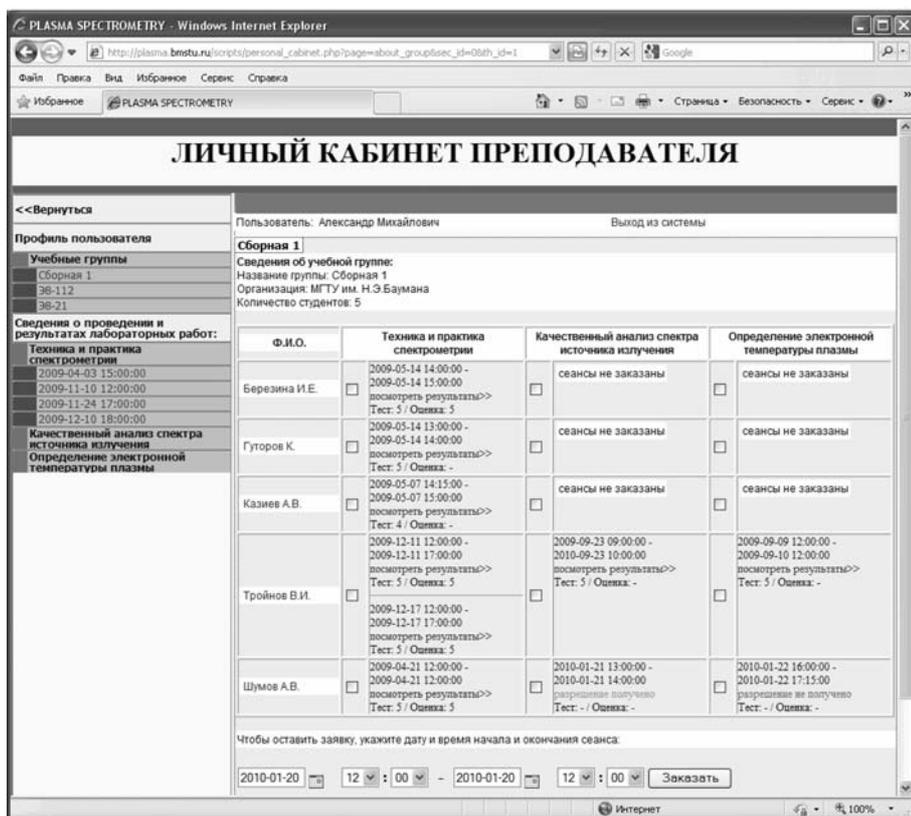


Рис. 9. Личный кабинет преподавателя

возможность формирования заявки на проведение сеансов удаленного доступа как отдельно для каждого из студентов, так и для всех членов группы. В личном кабинете преподавателя предусмотрены возможности создания учебной группы, информация о которой заносится в интегрированную базу данных и становится доступной пользователям с уровнем доступа "обучающийся". Последние получают возможность через свои личные кабинеты зарегистрироваться в созданной учебной группе, после чего преподавателю открывается доступ к информации о ходе прохождения ими лабораторных работ (рис. 9).

Выполняемые сетевые практикумы. В настоящий момент возможно проведение нескольких автоматизированных лабораторных практикумов.

Техника и практика спектрометрии. Цель работы — изучение устройства, технических характеристик и параметров управления многоканального спектрометра, регистрация эмиссионных спектров исследуемого источника излучения для последующего качественного и количественного анализа.

Качественный анализ спектра источника излучения. Цель работы: определение качественного состава источника излучения по зарегистрированному линейчатому эмиссионному спектру.

Определение электронной температуры плазмы. Цель работы: определение температуры по относи-

тельным интенсивностям эмиссионных спектральных линий.

Лаборатория активно используется для проведения практической подготовки по диагностике плазмы студентов как МГТУ им. Н. Э. Баумана, так и родственных университетов (НИЯУ МИФИ, МЭИ (ТУ)).

Демонстрационная версия сетевой лаборатории опубликована на круглосуточном сервере ЛУД МГТУ им. Н. Э. Баумана (<http://lud.bmstu.ru>).

Проведение удаленных научных экспериментов

Для проведения научных экспериментов используется сетевой доступ к комплексу автоматизированной спектральной диагностики для исследования плазмы, генерируемой в магнетронной распылительной системе [15]. Удаленный исследователь получает возможность не только регистрировать и обрабатывать спектры с помощью

описанного выше интерфейса управления спектрометром и сетевых средств обработки данных, но и исследовать пространственные зависимости излучательных характеристик плазмы вдоль оси разряда, управляя также составом и давлением плазмообразующего газа (рис. 10, см. четвертую сторону обложки).

Во время проведения эксперимента удаленный пользователь имеет возможность подключения к Интернет-камере-серверу, позволяющей наблюдать за ходом проведения опыта. Дополнительно к Lab-серверу 2 подключено мультимедиа-оборудование, позволяющее с помощью программы Skype осуществлять аудиовизуальный контакт с оператором стенда.

Разработанное в среде LabVIEW программное обеспечение управления положением головки и давлением газа предусматривает возможность использования удаленного интерфейса. Средствами, предоставляемыми средой LabVIEW, создан еще один Lab-сервер, настройки которого позволяют администрировать доступ удаленных исследователей по IP-адресам. Для реализации возможностей сетевого управления на компьютере удаленного экспериментатора в этом случае дополнительно должна быть установлена свободно распространяемая утилита LabVIEW Run-Time Engine, содержащая необходимые плагины для браузера.

Пространственное сканирование плазмы. Сбор излучения из локальных областей разряда проводится с помощью перемещаемой вдоль оси разряда оптической головки, соединенной с оптоволоком. Пространственное сканирование разряда обеспечивается перемещением системы сбора излучения, для чего используется шаговый двигатель. Формирование сигнала для задания частоты вращения ротора двигателя осуществляется программно через цифровые каналы модуля сопряжения и позволяет обеспечить позиционирование оптической головки с высокой точностью. На вход модуля подаются управляющие цифровые сигналы: включение/выключение питания, направление вращения и угол поворота вала.

Программный интерфейс удаленного управления шаговым двигателем реализован в среде графического программирования LabVIEW. Экспериментатор имеет возможность управлять перемещением оптической головки. В зависимости от исходных данных соответствующая программа вычисляет необходимое число импульсов и генерирует их на цифровом TTL-выходе модуля сопряжения. Электромеханическая защита от ошибочного задания параметров обеспечивается расположением в крайних положениях винтовой передачи концевых выключателей, которые при срабатывании коммутируют соответствующие реле и подают логический сигнал на TTL-вход модуля, что дает возможность программно обрабатывать данную ситуацию.

Управление давлением и составом смеси рабочих газов. Для управления давлением газа в разрядной вакуумной камере экспериментального стенда реализована система автоматического регулирования, работающая в диапазоне давлений 0,01...5,0 Па. Давление в вакуумной камере измеряется с помощью термодатчика манометрического преобразователя ПМТ-2, состоящего из нагревателя и термопары, ее выводы соединены с усилителем, который преобразует входной сигнал 0...10 мВ в выходной 0...10 В. Усиленный сигнал подается на вход аналого-цифрового преобразователя (АЦП) модуля USB-6259. Расход рабочих газов управляется двумя регуляторами РРГ. Управление исполнительными элементами осуществляется сигналом через цифроаналоговый преобразователь (ЦАП) в диапазоне 0...5 В. Использование двух регуляторов позволяет создавать в камере смесь рабочих газов с заданным объемным или массовым соот-

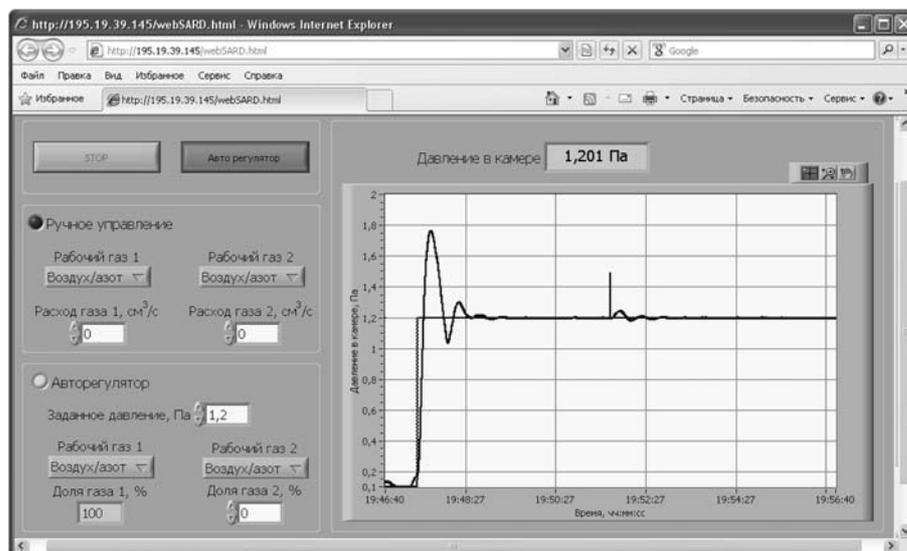


Рис. 11. Интерфейс сетевого управления газовакуумной системой

ношением. Программа управления газовакуумной системой создана в среде LabVIEW (рис. 11).

Экспериментатор имеет возможность либо управлять давлением "вручную", задавая расход рабочего газа, либо использовать автоматический регулятор, указывая значение давления, которое необходимо стабилизировать. В окне отображаются текущие значения расхода, давления, а также график зависимости давления газа в вакуумной камере от времени.

Заключение

В результате организации сетевого доступа к сложному оборудованию создана многоцелевая Интернет-лаборатория с университетским Центром коллективного пользования на ее основе.

Разработано программно-методическое обеспечение для поддержки проведения практикумов, содержащее наряду с подсистемами обучения и тестирования автоматизированную систему доступа к оборудованию, сбора и обработки результатов экспериментов. Предусмотрена возможность организаций групповых практических занятий студентов под контролем преподавателя.

Научная часть лаборатории предусматривает значительно большее количество задаваемых удаленным пользователем параметров и существенное расширение возможностей эксперимента за счет увеличения числа каналов управления.

Использованный комплексный подход особенно актуален для федеральных и национальных исследовательских университетов, уникальное оборудование которых целесообразно использовать в режиме коллективного сетевого доступа как в учебных, так и в научных целях.

Список литературы

1. **Новый** подход к инженерному образованию: теория и практика открытого доступа к распределенным информационным и техническим ресурсам / Ю. В. Арбузов, В. Н. Леньшин, С. И. Маслов и др. М.: Центр-Пресс, 2000. 238 с.
2. **Методика** применения дистанционных образовательных технологий (дистанционного обучения) в образовательных учреждениях высшего, среднего и дополнительного профессионального образования РФ // Поиск. 2003. № 4. С. 8—9.
3. **Норенков И. П., Зимин А. М.** Информационные технологии в образовании. М.: Изд-во МГТУ им. Н. Э. Баумана, 2004. 352 с.
4. **Информатизация** образования: направления, средства, технологии / Под общей ред. С. И. Маслова. М.: Изд-во МЭИ, 2004. 868 с.
5. **Remote Access Computer-Aided Laboratories and Practical Training of XXI Century Engineers** / I. B. Fedorov, A. M. Zimin, S. V. Korshunov et al. // Innovations 2008: World Innovations in Engineering Education and Research / Ed. W. Aung. USA, Arlington, 2008. Chap. 37. P. 415—423.
6. **Учебная Интернет-лаборатория "Испытания материалов"** / Б. В. Букеткин, А. М. Зимин, А. В. Шумов и др. // Информационные технологии. 2006. № 10. С. 58—65.
7. **Лабораторный** практикум по спектральной диагностике плазмы с удаленным доступом через Интернет / А. М. Зимин, В. А. Аверченко, А. В. Шумов и др. // Информационные технологии. 2002. № 3. С. 39—45.
8. **Интернет-лаборатория "Радиотелескоп МГТУ им. Н. Э. Баумана"** / И. Б. Федоров, А. М. Зимин, Б. А. Розанов и др. // Информационные технологии. 2005. № 9. С. 66—72.
9. **Шумов А. В.** Технология распределенной сетевой поддержки удаленных экспериментов в Интернет-лаборатории "Спектрометрия плазмы" // Труды XV Всероссийской научно-методической конференции Телематика'2008. — СПб. 2008. Т. 1. С. 237—238.
10. **Автоматизированный** спектрометрический комплекс для диагностики плазмы магнетронного разряда / В. М. Градов, А. М. Зимин, С. Е. Кривицкий, А. В. Шумов // Вопросы атомной науки и техники. Сер. Термоядерный синтез. 2009. № 1. С. 64—71.

CONTENTS

Domrachev V. G., Polishchuk O. M. *The Determination of Rating Points of Groups of Objects and Monitoring of their States Based on Fuzzy Initial Information* 2

This paper is devoted to the new direction of information processing — fuzzy information processing. Relevance of this direction in its time was defined the transition to the study of organizational and technical problems, decision-making processes and management systems. In this paper we apply fuzzy sets theory to determine rating points of groups of objects and to use them to develop management recommendations.

Keywords: group of objects, rating points, linguistic variables

Borodaschenco A. Yu., Ryabtsev A. O. *Filtering Algorithm Textual Information Based Hypertexts Models* 7

In this paper an algorithm for evaluation of the array of hypertext documents on the semantic similarity with the standard, which allows for vysiquity information retrieval on the Internet and intranet cers fields. The algorithm realizes the function of selecting a hypertext documenting from the array of data by analyzing the information surrounding the user-specified document-reference.

Keywords: semantic distance, hypertexts model, proximity measure, filtering algorithm texts, hyperlink

Toporkov V. V., Tselishchev A. S. *Critical Jobs Method as Means of Effective Scheduling for Distributed Computing* 13

In this paper the basic algorithms and heuristics of the critical jobs method, that allows performing real-time scheduling in distributed computing on application level with data dependencies, are described. The modifications to the basic method allow taking into account CPU nodes load. Finally, some experimental results based on a developed software model concerning the efficiency of this method are shown.

Keywords: distributed computing, meta-scheduling, scheduler, planning strategies, allocation, collision

Amirshahi B. *Parallel Cluster Algorithm for Solving Large Linear Equation Systems, Using Grid-Technology and Cramer's Rule, on a Supercomputer* 17

Large sets of data provide a challenge while solving linear equation systems, especially when we use a network with distributed resources in a noisy background, and that is why a parallel algorithm is so needed. In this paper I will try to optimize the current methods of solving of large linear equation systems using a parallel cluster algorithm and mathematical Cramer's rule. I implemented the algorithm on a clustered supercomputer model T-4700 (Scyth) which is located at Moscow state university of MIIT.

Keywords: GRID-computing, computer network, a cluster supercomputer, the method of Cramer

Eltarenko E. A. *Approximation of Preferences in Multicriteria Problems by Aggregation Operators* 23

Here is described an approach which is based on identification of preferences of the decision maker (DM) on the basis of a common survey. The approach involves the implementation of the formal axioms of preferences: a) the preferences are monotonous; b) the weights of the criteria reflect their importance. There is an apparatus for generating operators of aggregation for the criteria, which are consistent with the axioms of preferences. The known utility functions are special cases of aggregation operators. The procedure of the identification of the operator and the results of the survey allow to determine the operator which approximates the individual preferences of the DM.

Keywords: multicriteria problems, description of preferences, weights of criteria, aggregation operators

Kukhareno B. G., Ponomarev D. I. *Bayes Nonlinear Filtering Multi-Dimensional Time-Series* 33

Method is described of identifying a non-linear system with time-series in use provides filtering the time-series. The Expectation-Maximization algorithm estimates iteratively the states of non-linear system linearized locally as Gaussian clouds of data. The non-linear regression fits Gaussian radial basic functions to the data sets. The parameters estimates are obtained analytically. The algorithm is initialized by the Factor analysis and training a linear system iteratively. The experiments on Bayes nonlinear filtering use multi-dimensional time-series and state-space of high dimensions.

Keywords: multi-dimensional time-series, Rauch smoother, non-linear systems, Expectation-Maximization algorithm, Gaussian clouds of data, radial basis functions

Philippova A. S., Frolov R. V. *Creation of Operative Network Graphics of Works Brigades Reconditioning of Well and Pull out of Hole.* 39

The article considers the resource-saving problem in regularly arising at operation of oil deposits is discussed. One of the factors raising real oil recovery is timely repair of oil wells. The suggested approach is based on working out of network schedules of performance of repair work and creations for their support of mobile system of routing.

Keywords: oil deposits, pull out of hole, the network graphic, the schedule, routes, movement of brigades

Bronshtein E. M., Zaiko T. A. *Vehicle Routing Problem with Restrictions* 42

In the paper there is considered the vehicle routing problem with restrictions on transportation of certain types of products by particular vehicles. Correspondent mathematical model is worked out. Heuristic solution algorithms are proposed, numerical experiments for comparative appraisal of effectiveness of the suggested algorithms are conducted.

Keywords: vehicle routing, vehicles, points of demand, mathematical model approach, heuristic algorithms

Semin V. V. *Modeling the Behavior of the Stochastic Triangulation on Quasi-Stationary Gratings* 45

The task of representations of topological models in the form of triangulated cubic complexes associated with the nature and behavior of triangulation in n-dimensional space. The basis for the analysis of these features can serve as a fundamental statistical information, allowing them to perform the synthesis of the characteristics of triangulation in model spaces of different dimensions. The paper proposed an approach to solving this problem using statistical analysis features of a stochastic triangulation for spaces of dimension not higher than 5, using a binary encoding, as well as the properties of primitive surgery triangulation.

Keywords: triangulation, simplex, complex, unit cube, topology, grid

Pekunov V. V. *The Simulation of Algorithms, Data and Execution Parameters of Sequential and Parallel Programs in the Metalayer* 51

The problems of simulation and of prediction of data, algorithm and execution time of program are considered. It is proposed to apply a formalism of object-event models (OEM) for the constructing and interpretation of predictive models. An idea of the modelling metalayer abstracted from the program is introduced. The ways of the using the predictive models for the dynamic optimization and approximation/copying of algorithms in the cases of usual and parallel programs are given.

Keywords: program metalayer, algorithm simulation, data prediction, prediction of execution time, object-event model, parallel programming, program optimization

Benevolenskiy S. B., Kirillov Yu. I., Pozhar V. E., Pustovoit V. I., Spiridonov I. S. *Development of Software Platform for Acousto-Optic Spectrometers* 57

The article reviews the requirements for the developed software for acousto-optic spectrometers and a choice of means of implementation.

Keywords: the software for acousto-optic spectrometer, processing of the spectrum, spectral analysis, informational core

Zamyatin A. V. *Differentiated Loss Compression of Remote Sensing Images* 60

An approach to loss compression of remote sensing images is proposed. It is based on differentiated compression of segments with different levels of quality depending on levels of its significance. The techniques of the key parameters definition of the approach are found. These parameters are relation of level of quality vs. level of its significance and segmentation degree, which can define the compression ratio, quality of an uncompressed image and its classification accuracy in a wide range. The numerical research results using different remote sensing images are obtained. It shows the effectiveness and validity limits of the proposed compression approach and the techniques of its key parameters definition.

Keywords: remote sensing images, differentiated compression, loss compression, segmentation, classification

Kupriyanov Yu. V., Taratoukhine V. V. *Principles of Benefits Management at Enterprise Information Systems Implementation Projects* 66

One should be perceiving an information system (IS) implementation project as a major [techno]organizational transformation requiring appropriate organizational transformation management techniques. However today a larger portion of IT-project managers continue to follow IS-deterministic approach mainly concentrating on delivering IS functionality and satisfying projects constraints. As a result the importance of activities aimed at business benefits attainment is underestimated.

In this article authors analyze peculiarities of IT-investment business-benefits realization, formulate the principles of IT-value extraction and argue the conceptual model for benefits management at information systems implementation projects.

Keywords: benefits management, IT-project, business-benefits, value assessment, IS implementation

Zimin A. M., Shumov A. V., Krivitsky S. E., Troynov V. I. *The Automated Network Educational-Scientific Laboratory on Plasma Spectroscopy* 72

The article is devoted to the integrated Internet laboratory which allows to remotely carrying out practical trainings as well as full investigation on sophisticated equipment. It is especially important for federal and national research universities. An automated dispatching system is also described. It allows to place an order and perform experiments, preserve and process of their results, administrate of the equipment operation.

Keywords: Internet laboratory, remote access, plasma spectroscopy

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 08.04.2011. Подписано в печать 19.05.2011. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 9,8. Уч.-изд. л. 10,30. Заказ 371. 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