ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

2(198) 2013

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Информация о журнале доступна по сети Internet по адресу **http://novtex.ru/IT.**
Журнал включен в систему Российского индекса научного цитирования.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 519.816;004.89

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

Описание предпочтений в многокритериальных задачах с иерархической системой критериев

Проблема описания предпочтений рассматривается как задача измерения предпочтений. В иерархической системе критериев задача сводится к измерению в шкале интервалов вышестоящих критериев через множество нижестоящих. Для этого подхода определен тип функции предпочтения.

Разработан единый для всех уровней план опроса лица, принимающего решения, для выявления его предпочтений. По результатам опроса для каждой вершины иерархии идентифицируется функция предпочтения и ее параметры.

Ключевые слова: *многокритериальные задачи, иерархия критериев, описание предпочтений, важность критериев, агрегирование критериев, функции предпочтения*

Введение

При большом числе критериев в многокритериальной задаче (МКЗ) их структурируют в виде иерархии (дерева). В результате получаем МКЗ с иерархической системой критериев. Надо сказать, что в большинстве практических задач число критериев больше десяти, и они должны рассматриваться как МКЗ с иерархической системой критериев.

В постановке МКЗ выделим:

- задачи с заданным множеством объектов (альтернатив), которые требуется упорядочить или выбрать из них наиболее предпочтительную, или дать оценку предпочтения;
- задачи описания предпочтений лица, принимающего решения (ЛПР), при этом множество оцениваемых объектов может не задаваться.

Наиболее известными методами решения МКЗ с иерархической системой критериев являются метод анализа иерархий (МАИ) [1, 2] и многомерная теория полезности (МТП) [3].

МАИ применим для задач с заданным множеством объектов, а МТП предназначена для задач описания предпочтений. В статье рассматривается задача описания предпочтений, поэтому предлагаемый подход будем сравнивать с МТП.

В работе [3] по МТП приводятся примеры решения МКЗ с иерархической системой критериев. При этом на всех уровнях иерархии используется ограниченное число функций полезности: мультипликативная, аддитивная. Проверка условий применения этих функций: независимости по полезности, по предпочтению весьма затруднительна, особенно для критериев верхних уровней иерархии. Поэтому для задач с иерархической системой критериев вводят допущения о независимости [3. п. 6.11].

Иногда без достаточного обоснования для агрегирования критериев по иерархии используют аддитивную функцию. В связи с этим отметим, что применение аддитивной функции на всех уровнях иерархии означает использование дерева критериев только для определения весов критериев, а результат — обобщенный критерий (корень дерева) — рассчитывается как взвешенная сумма всех критериев нижнего уровня.

Все вышесказанное указывает на актуальность развития методов решения МКЗ с иерархической системой критериев.

Вопросы построения иерархии критериев подробно рассмотрены в работах [1—3]. Для классификации элементов дерева (корня, ветвей и листьев) используем термины из квалиметрии [4]. Нижние уровни (листья дерева) представляют собой *единичные* (*частные*) *критерии*, которые измеряются в разных единицах. Критерии могут быть не только количественные, но и текстовые, имеющие конечный набор значений. Ветви дерева составляют *комплексные критерии.* Они вычисляются в результате агрегирования нижестоящих критериев. Результаты агрегирования определяют комплексные критерии в шкале интервалов [0; 1]. Верхний уровень (корень дерева) образует обобщенный критерий.

Корень дерева определяет цель, следующий уровень — подцели и т. д. Будем говорить о предпочтениях с точки зрения достижения поставленной цели. Чтобы не вводить множества целей 2-го, 3-го уровней, характеризующихся соответствующими комплексными критериями, будем использовать для всех них один термин "предпочтение". Решая задачу агрегирования множества критериев в вышестоящий, будем говорить об описании предпочтений на данной ветви дерева. Предпочтение — это свойство, которое характеризует вышестоящий комплексный критерий.

1. Постановка задачи

При решении МКЗ выделяют две проблемы: неоднородность пространства критериев (критерии измеряются в разных единицах) и агрегирование критериев в скаляр.

В МКЗ с иерархической системой критериев сначала следует решить задачу перехода к безразмерным величинам для критериев нижнего уровня (единичных критериев). Затем надо обосновать процедуры агрегирования критериев для вышестоящих уровней (комплексных критериев). Обе проблемы будут рассматриваться с точки зрения описания предпочтений ЛПР. Эти вопросы излагаются в п. 2.

В работах автора [5, 6] рассмотрены вопросы описания предпочтений с помощью операторов агрегирования. В них рассмотрены ситуации, когда агрегируемые критерии измеряются в относительных (безразмерных) величинах. Поэтому операторы могут использоваться для агрегирования комплексных критериев в иерархической структуре.

Эти вопросы обсуждаются в п. 3.

2. Описание предпочтений на нижних уровнях иерархии

Описание предпочтений начинается с нижних уровней иерархии (листьев). По существу необходимо решить проблему неоднородности пространства критериев.

Для каждого единичного критерия k_j определим допустимые значения в виде интервала $[k_{j\min}; k_{j\max}]$ либо как множество текстовых значений.

Иллюстрировать основные положения будем на примере построения функции предпочтения при поиске места работы. В качестве критериев выберем два: k_1 — размер заработной платы (тыс. руб.) и k_2 — время поездки до места работы (мин). Допустимый интервал для k_1 — [25; 125], нижняя граница определяет порог, ниже которого вариант исключается, верхняя граница означает, что заработная плата ЛПР полностью устраивает, а превышение ее не увеличивает предпочтение. Аналогично, зададим интервал для критерия k_2 — [10; 90]. В качестве примера текстового критерия можно рассмотреть использование вместо k_2 (время поездки) перечень территориальных районов, в которых должно быть место работы.

В статье будем использовать терминологию монографии [3]. Дополнение критерия $\overline{k}_j$ — все множество критериев, исключая k_j, дополнение пары критериев k_j и k_s – $\overline{k}_{js}$.

Определение 1. *Функция предпочтения* (ФП) — оператор агрегирования (свертка) критериев в скаляр $u(\mathbf{k}) = u(k_1, ..., k_m)$, определяющий предпочтения в шкале интервалов.

С учетом понятия дополнения критериев вариантами записи ФП могут быть $u(k_j, \overline{k}_j)$ или $u(k_j, k_s, \overline{k}_{js})$. ФП принимает значения в интервале [0; 1].

2.1. Перевод единичных критериев в безразмерные величины

Каждый единичный критерий косвенно характеризует вышестоящий комплексный критерий. Как указано в работе [3], он является критерием-заместителем вышестоящего комплексного. Для перевода единичного критерия k_j в безразмерную шкалу интервалов [0; 1] необходимо задать два шкальных значения: эталон 1, соответствующий нулевому значению, и эталон 2 — единичному значению. Для этого по каждому критерию k_j в интервале $[k_{j\min}; k_{j\max}]$ ЛПР должен указать значение k_j^-, соответствующее минимальному предпочтению, и k_j^+ — максимальному предпочтению. Какое из k_j^- и k_j^+ больше, зависит от содержания критерия: так, для заработной платы $k_1^- < k_1^+$, а для времени поездки до места работы $k_2^- > k_2^+$. Значения k_j^- и k_j^+ не обязательно должны принимать граничные значения интервала $[k_{j\min}; k_{j\max}]$.

Следует обратить внимание на то, что значения k_j^- и k_j^+ одного критерия могут зависеть от k_s^- и k_s^+ других критериев. Например, в нашем примере при $k_2^- = 90$ мин допустимым может быть вариант с $k_1^+ = 130$ тыс. руб. Значит, интервал для k_1 следует расширить до [25; 130]. В результате коррекции границ необходимо добиться, чтобы предпочтение объекта $B^- = \{k_1^-, ..., k_m^-\}$ было минимальным, а $B^+ = \{k_1^+, ..., k_m^+\}$ максимальным во всем пространстве допустимых значений.

Для функции предпочтения, измеряемой в шкале интервалов [0; 1], эталонами являются $u(\mathbf{k}^-) = 0$ и $u(\mathbf{k}^+) = 1$.

Определение 2. *Частная нормированная функция предпочтения* (ЧНФП) j-го критерия ($u_j(k_j) \in [0; 1]$) — одномерная функция предпочтения по j-му критерию, определяемая при фиксированном дополнении $\overline{k}_j^-$, т. е. $u_j(k_j) = \dfrac{u(k_j, \overline{k}_j^-)}{u(k_j^+, \overline{k}_j^-)}$.

Следует подчеркнуть, что ЧНФП $u_j(k_j)$ не является условной функцией предпочтения $u(k_j, \overline{k}_j')$ [3], которая определяется при заданном значении дополнения $\overline{k}_j'$. В ЧНФП дополнение четко фикси-

ровано и не меняется, поэтому в функции $u_j(k_j)$ один аргумент.

Чтобы построить $u_j(k_j)$ ЛПР достаточно указать, как изменяются предпочтения на допустимом интервале $[k_{j\min};\ k_{j\max}]$, при этом для определения единой шкалы интервалов для всех критериев следует принять $u_j(\underline{k}_j) = 0$ и $u_j(k_j^+) = 1$.

Для построения ЧНФП следует использовать методы построения шкал интервалов. В теории измерений [7, стр. 23] шкала измерений определяется как гомоморфизм неприводимой эмпирической системы с отношениями (э.с.о.) в числовую систему. Шкала интервалов может быть построена как на основе операций (э.с.о. включает трехместное отношение — операцию) [7, гл. 6], так и на основе расстояний (э.с.о. включает четырехместное отношение — систему расстояний) [7, гл. 9]. В [3, стр. 123 п. 3.7.2] приводится пример построения одномерной функции предпочтений с использованием операции деления пополам.

Функции $u_j(k_j)$ могут быть как убывающие, так и возрастающие в зависимости от содержания критерия. Возможны и немонотонные функции $u_j(k_j)$, например, покупатель в магазине выбирает телевизор, соответственно он определяет наиболее желательный размер экрана k_j^+ из допустимого интервала $[k_{j\min};\ k_{j\max}]$, который включает k_j^+. Какое из $k_{j\min}$ и $k_{j\max}$ или оба значения принять за нулевую безразмерную оценку, определяется покупателем.

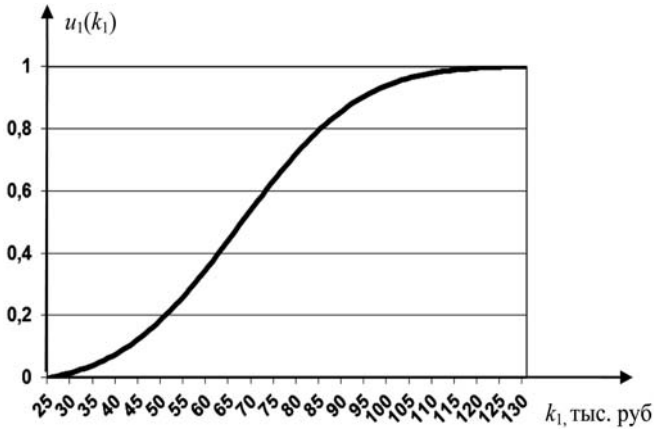


Рис. 1. ЧНФП для заработной платы

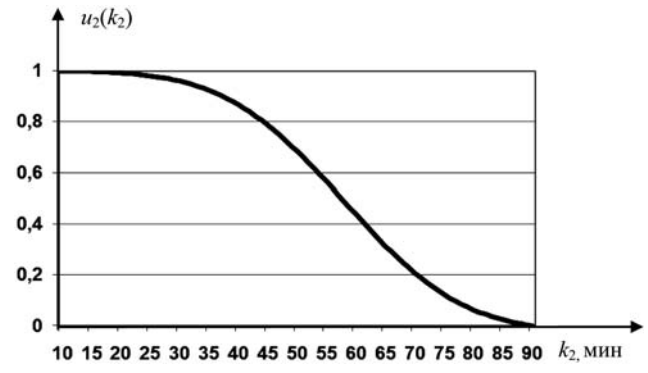


Рис. 2. ЧНФП для времени поездки до места работы

Для текстовых критериев множество возможных значений конечно, например, в нашем примере приняв за k_2 перечень из четырех районов ($q = 1, 2, 3, 4$), определение ЧНФП для k_2 означает задание ЛПР предпочтений для всех районов u_2^q. Причем минимальное по предпочтению значение $\underline{k}_2$ должно иметь нулевое значение $u_2^1 = 0$, а максимальное k_2^+ — единице: $u_2^4 = 1$. Такую оценку текстовых значений удобно проводить, используя процедуры опроса экспертных методов [8, 9], основанных на шкалах интервалов.

Если значения критерия определяются ЛПР или экспертами в баллах, то, естественно, такой критерий измеряется в безразмерных величинах, которые приводятся к интервалу [0; 1]. Например, в рассматриваемом примере оценки места работы можно ввести третий критерий k_3 — ее перспективность, которую можно оценивать в баллах. Надо сказать, что этот критерий измеряется в шкале отношений, так как существует абсолютный нуль, соответствующий полному отсутствию перспектив.

На рис. 1 и 2 представлены ЧНФП для рассматриваемого примера.

После определения ЧНФП получаем функцию предпочтения, в которой переменными являются безразмерные критерии $g(\mathbf{u}) = g(u_1(k_1), ..., u_m(k_m)) = u(k_1, ..., k_m)$, которую будем называть *функцией предпочтения безразмерных критериев* (ФПБК) $g(u_1, ..., u_m) = g(u_j, \overline{u}_j)$. ФПБК при нулевом значении дополнения $\overline{u}_j = 0$ будем обозначать $g(u_j, 0)$.

2.2. Аксиомы предпочтений

Перейдем к формулированию аксиом (свойств) предпочтений. Начнем с общепринятых в теории полезности.

Аксиома 1 единственности нулевого значения ФПБК: $g(\mathbf{u}) = 0$ только при всех $u_j = 0$.

Аксиома 2 единственности единичного значения ФПБК: $g(\mathbf{u}) = 1$ только при всех $u_j = 1$.

Аксиома 3 монотонности предпочтений (ФПБК): $g(u_j^1, \overline{u}_j) > g(u_j^2, \overline{u}_j)$, где $u_j^1 > u_j^2$ для любого дополнения $\overline{u}_j \left(\dfrac{\partial g(u_j, \overline{u}_j)}{\partial u_j} > 0\right)$.

Определим порядок важности частных критериев: $k_1 \succ k_2 \succ ... \succ k_m$, а затем веса критериев $V_1 > V_2 > ... > V_m \left(\sum\limits_{j=1}^{m} V_j = 1\right)$. Вопросы определения весов будут рассмотрены далее.

Зададим произвольное значение $u' \in [0; 1]$ для всех критериев. ЛПР удобнее оценивать предпочтение многокритериальных объектов, заданных в фи-

зических единицах измерения k_j. Поэтому, используя ЧНФП, для всех критериев значение u' переведем в $k'_j = u_j^{-1}(u')$, где $u_j^{-1}(u')$ — обратная функция от $u_j(k_j)$. Лицу, принимающему решение, предлагается оценить предпочтения объектов $u(k'_j, \overline{k}_j^-)$.

Аксиома 4 относительной важности критериев. Для любой пары критериев k_j и k_s отношение предпочтений $\dfrac{u(k'_j, \overline{k}_j^-)}{u(k'_s, \overline{k}_s^-)} = \dfrac{V_j}{V_s}$ неизменно при любом значении $u' \in [0; 1]$.

Рассматривая u' как переменную u, получим для каждого критерия функцию $g(u_j, 0) = g_j(u) = u(u_j^{-1}(u), \overline{k}_j^-)$. Тогда условие относительной важности $\dfrac{g(u_j, 0)}{g(u_s, 0)} = \dfrac{g_j(u)}{g_s(u)} = \dfrac{V_j}{V_s} = \text{const}$ выполняется, если обе функции $g(u_j, 0)$ и $g(u_s, 0)$ линейные, т. е. $g(u_j, 0) = kV_j u$ и $g(u_s, 0) = kV_s u$, где коэффициент $k = \text{const}$, а V_j и V_s — веса критериев.

2.3. Проверка аксиом предпочтения

Рассмотрим процедуру проверки аксиомы относительной важности критериев. Выбрав $u' = u^{(1)}$ и определив для всех критериев соответствующие значения $k_j^{(1)} = u_j^{-1}(u^{(1)})$, необходимо ЛПР предъявить для оценки предпочтений множество объектов, представленных в табл. 1. Шкала интервалов для оценивания предпочтений определяется двумя эталонами: $\mathbf{k}^+$ и $\mathbf{k}^-$. Сравнивать объекты $\{k_j^{(1)}, \overline{k}_j^-\}$ с эталоном $\mathbf{k}^+$ из-за большой разницы между $\overline{k}_j^-$ и $\overline{k}_j^+$ сложно, поэтому для повышения точности оценок $\{k_j^{(1)}, \overline{k}_j^-\}$ введем объект $B^= \equiv \{k_1^{(1)}, ..., k_m^{(1)}\}$. В безразмерных единицах $B^=$ имеет равные значения $u^{(1)}$

Таблица 1

Форма анкеты для оценки объектов $\{k_j^{(1)}, \overline{k}_j^-\}$ для трех критериев

Оцениваемые объекты	Критерии			Оценки ЛПР
	k_1	k_2	k_3	
Объект эталон B^+	k_1^+	k_2^+	k_3^+	1,0
Объект $B^=$	$k_1^{(1)}$	$k_2^{(1)}$	$k_3^{(1)}$	$R^{(1)}$
Объект B^1	$k_1^{(1)}$	k_2^-	k_3^-	$W_1^{(1)}$
Объект B^2	k_1^-	$k_2^{(1)}$	k_3^-	$W_2^{(1)}$
Объект B^3	k_1^-	k_2^-	$k_3^{(1)}$	$W_3^{(1)}$
Объект эталон B^-	k_1^-	k_2^-	k_3^-	0,0

всех критериев. Оценка этого объекта $R^{(1)}$ будет сопоставима с $W_j^{(1)}$ — оценками B^j. ФПБК $g(\mathbf{u})$ при равных значениях критериев образует функцию одной переменной $g(u)$, в работе [5] она названа функцией равных значений переменных. Чтобы подчеркнуть, что переменными являются безразмерные значения критериев, будем называть $g(u)$ *функцией равных значений безразмерных критериев* (ФРЗБК).

Если есть текстовые переменные k_j, то в качестве уровней $u^{(i)}$ следует брать имеющиеся для него значения u_j^q. В случае, когда имеется несколько текстовых критериев k_j и k_s, используются u_j^q и u_s^l, которые, как правило, не совпадают. Поэтому в опросной таблице для u_j^q в качестве объекта $B^=$ используется объект $\{k_1^q, ..., k_j^{(q)}, ..., k_s^l\}$. В этом случае множество оценок $B^=$ не точно образует ФРЗБК. Чтобы приблизить множество к ФРЗБК в $B^=$ следует использовать ближайшее к u_j^q значение u_s^l.

Для оценки объектов табл. 1 удобно использовать экспертные методы [8, 9], позволяющие получить результаты в шкале интервалов.

Перейдя к новому значению $u' = u^{(2)}$ и проведя опрос ЛПР, получим $R^{(2)}$, $W_j^{(2)}$.

Надо отметить, что $R^{(2)} = u(\mathbf{k}^{(2)}) = g(u^{(2)})$, а $W_j^{(2)} = u(k_j^{(2)}, \overline{k}_j^-) = g(u_j^{(2)}, 0)$. Сравнивая с построением ЧНФП, видим, что $W_j^{(2)}$ есть значение ЧНФП, соответствующее $k_j^{(2)}$. Но в шкале интервалов измерения предпочтений для $W_j^{(2)}$ изменен коэффициент сжатия: единичным эталоном при построении ЧНФП был $u(k_j^+, \overline{k}_j^-)$, а стал $u(k_j^+, \overline{k}_j^+)$. Учитывая, что для всех $W_j^{(2)} = u(k_j^{(2)}, \overline{k}_j^-) = g(u_j^{(2)}, 0)$ значение $u_j^{(2)}$ одинаковое, получается что отношение $\dfrac{g(u_j, 0)}{g(u_s, 0)} = \dfrac{u(k_j^+, \overline{k}_j^-)}{u(k_s^+, \overline{k}_s^-)}$ не будет меняться при переходе от $u_j^{(1)}$ к $u_j^{(2)}$.

Значит, если ЧНФП отражают предпочтения ЛПР в шкале интервалов при изменении критериев, автоматически будет выполняться аксиома относительной важности. Тогда $g(u_j^{(i)}, 0)$ должны образовывать линию $g(u_j, 0) = kV_j u_j$.

Если $g(u_j, 0)$ не образует линии, то ошибки следует искать при построении ЧНФП, и их необходимо откорректировать. Можно коррекцию $u_j(k_j)$ провести, используя кусочно-линейную функцию $g_j(u_j) = g(u_j, 0)$, тогда откорректированная ЧНФП будет определяться как $\tilde{u}_j(k_j) = g_j(u_j(k_j))$. При этом надо иметь в виду, что изменение $u_j(k_j)$ влечет изменение $k_j^{(i)}$ в табл. 1 опроса, соответственно меня-

ются оценки $W_j^{(i)}$, и не факт, что в результате коррекции ЧНФП получим линейную $g(u_j, 0)$. Поэтому коррекцию ЧНФП следует проводить в несколько итераций.

Чтобы по оценкам $W_j^{(i)}$ определить веса V_j, которые должны быть в сумме равны единице, $W_j^{(i)}$ надо пронормировать. Для этого введем масштабный коэффициент $k = \frac{1}{u^{(i)}} \sum_{j=1}^{m} W_j^{(i)}$, который при выполнении аксиомы относительной важности не будет зависеть от $u^{(i)}$. Тогда веса критериев равны $V_j = \frac{W_j^{(i)}}{k}$.

Если среди множества критериев есть текстовые критерии, то масштабный коэффициент k и веса V_j следует вычислять по значениям $W_j^{(n)}$, так как все они будут соответствовать одинаковому значению $u^{(n)} = 1$, т. е. $k = \sum_{j=1}^{m} W_j^{(n)}$.

В рассматриваемом примере с двумя критериями в соответствии с ЧНФП, приведенными на рис. 1, 2, определим для пяти значений $u^{(i)}$ величины $k_j^{(i)}$. Результаты представлены в табл. 2, 3.

Проводим опрос ЛПР и при остальных $u^{(i)}$ (i = 2, 3, 4, 5). По результатам опросов строим зависимости $g(u_j, 0)$. Они представлены на рис. 3. Из данных на рисунке получим масштабный коэффициент $k = 0{,}3 + 0{,}15 = 0{,}45$ и веса критериев $V_1 = 0{,}3/0{,}45 = 2/3$ и $V_2 = 0{,}15/0{,}45 = 1/3$.

Таблица 2

Значения обратных ЧНФП для заработной платы ($k_1^{(i)}$) и времени поездки ($k_2^{(i)}$)

$u^{(i)}$	0,2	0,4	0,6	0,8	1,0
$k_1^{(i)} = u_1^{-1}(u^{(i)})$	52	63	73	85	130
$k_2^{(i)} = u_2^{-1}(u^{(i)})$	70	62	53	45	10

Таблица 3

Форма опросной анкеты для $u^{(1)} = 0{,}2$

Объекты оценки	Зарплата, тыс. руб.	Время поездки, мин	Оценки ЛПР
Объект эталон B^+	130	10	1,0
Объект $B^=$	52	70	$R^{(1)}$
Объект B^1	52	90	$W_1^{(1)}$
Объект B^2	25	70	$W_2^{(1)}$
Объект эталон B^-	25	90	0,0

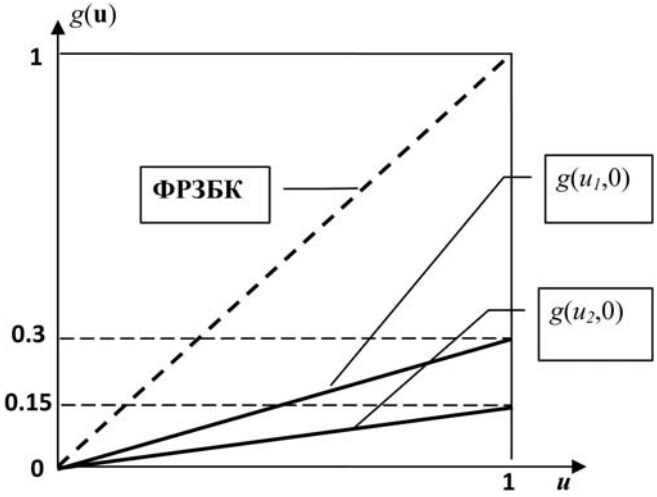


Рис. 3. Результаты проверки аксиомы относительной важности критериев

2.4. Выбор типа оператора агрегирования на нижнем уровне иерархии

В работах [5, 6] рассматриваются вопросы генерирования различных операторов агрегирования с безразмерными критериями, к которым относится ФПБК.

Аксиома относительной важности критериев выполняется для операторов следующего типа:

$$g(\mathbf{u}) = \psi\left(\sum_{j=1}^{m} \psi^{-1}(kV_j u_j)\right) + \varepsilon(u_1, ..., u_m), \qquad (1)$$

где $\psi(x)$ — генерирующая функция оператора ($\psi^{-1}(u)$ — обратная функция); k — масштабный коэффициент, обеспечивающий условие равенства единице $g(\mathbf{u})$ при единичных значениях всех u_j.

Всякое измерение выполняется с ошибками, тем более измерение предпочтений, поэтому в формуле (1) введена функция ошибок $\varepsilon(u_1, ..., u_m)$. В случае выполнения линейности $g(u_j, 0) = kV_j u_j$ функция ошибок $\varepsilon(u_j, 0) = 0$ (j = 1, 2, ..., m).

Чтобы выполнялась аксиома 3 монотонности ФПБК генерирующая функция $\psi(x)$ на интервале [0; 1] должна быть непрерывной, монотонно-возрастающей. Для выполнения аксиом 1 и 2 необходимо, чтобы значения $\psi(x)$ на границах равнялись: $\psi(0) = 0$ и $\psi(1) = 1$. Масштабный коэффициент k определяется из уравнения

$$\sum_{j=1}^{m} \psi^{-1}(kV_j) = 1. \qquad (2)$$

Для формирования генерирующей функции можно взять любую монотонно возрастающую функцию $f(x)$ и пронормировать ее:

$$\psi(x) = \frac{f(x) - f(0)}{f(1) - f(0)}.$$

Использование разных видов $f(x)$ приводит к формированию отдельных *видов генерирующих функций*. Каждый вид генерирующих функций должен иметь

параметр, с помощью которого формируется семейство генерирующих функций одного вида. Например, степенная функция $\varphi(x) = x^p$ имеет параметр p, позволяющий формировать семейство функций. Так как масштабный коэффициент k определяется по результатам опроса ЛПР, то условие (2) используется для идентификации параметра p. Тем самым из семейства функций одного вида определяется функция с конкретным значением параметра.

Если принять, что функция ошибок $\varepsilon(u_1, ..., u_m)$ равна нулю, то функция предпочтения $u(\mathbf{k})$ оператора (1) запишется в следующем виде:

$$u(\mathbf{k}) = \psi\left(\sum_{j=1}^{m} \psi^{-1}(kV_j u_j(k_j))\right). \qquad (3)$$

Для такой функции предпочтения выполняются аксиомы 3 и 4 для любой монотонно-возрастающей $\psi(x)$. При использовании разных видов генерирующих функций будем получать разные функции предпочтения.

Остается вопрос, как по результатам опроса ЛПР идентифицировать вид генерирующей функции $\psi(x)$ и как проверить, что функция ошибок $\varepsilon(\mathbf{u})$ незначительна.

2.5. Идентификация функции предпочтения

Для идентификации генерирующей функции $\psi(x)$ в (3) необходимо получить оценки предпочтений по множеству объектов со значениями критериев, не равных $\bar{k}_j$. Такими объектами являются оценки ФРЗБК, поэтому будем использовать результаты опроса ЛПР по опросной табл. 1.

Для идентификации $\psi(x)$ необходимо для разных видов генерирующих функций в соответствии с условием (2) определить ее параметр, вычислить значения $g(u^{(i)}) = \psi\left(\sum_{j=1}^{m} \psi^{-1}(kV_j u^{(i)})\right)$ для заданного множества u' и сравнить их с оценками $R^i(u^{(i)})$, например, по сумме квадратов разностей

$$S^2 = \sum_{j=1}^{m} (R^i(u^{(i)}) - g(u^{(i)}))^2. \qquad (4)$$

Чтобы иметь возможность подобрать $\psi(x)$, обеспечивающую более адекватное описание предпочтений ЛПР, необходимо иметь значительное число видов генерирующих функций.

Примеры выпуклых видов $\psi(x)$ приведены в Приложении. Из данных Приложения следует, что линейность ФРЗБК выполняется только для степенной генерирующей функции (линейная функция является частным случаем степенной). Поэтому можно сформулировать следующее утверждение.

Утверждение 1. Необходимыми условиями для использования метрики $L^p(\mathbf{u}) = k\left[\sum_{j=1}^{m} (V_j u_j(k_j))^{1/p}\right]^p$ в качестве функции предпочтения являются: а) масштабный коэффициент $k \neq 1$; б) линейная функция равных значений безразмерных критериев.

Если в результате опроса ЛПР получили масштабный коэффициент $k = 1$ и линейную ФРЗБК, то его предпочтения описываются аддитивной функцией.

Для рассматриваемого примера с оценкой мест работы на рис. 3 приведены линейная ФРЗБК и масштабный коэффициент $k = 0{,}45$. В соответствии с утверждением 1 функция предпочтения степенная. Используя условие (2), вычислим параметр p, для весов $V_1 = 2/3$ и $V_2 = 1/3$ он будет равен 2,2. В результате получим функцию предпочтения:

$$u(k_1, k_2) = 0{,}45\left[\left(\frac{2}{3}u_1(k_1)\right)^{1/2{,}2} + \left(\frac{1}{3}u_2(k_2)\right)^{1/2{,}2}\right]^{2{,}2},$$

где $u_1(k_1)$ и $u_2(k_2)$ — ЧНФП, приведенные на рис. 1 и 2. Так как утверждение 1 определяет только необходимые условия, то для проверки адекватности полученной функции можно провести дополнительный опрос ЛПР не только на линии равных значений, но и в других точках пространства критериев.

Отметим, что в одном виде $\psi(x)$ можно ввести два параметра, например, в степенной функции $\psi(x) = (x + b)^p$, тогда один параметр p можем использовать для нормировочного условия (2), а второй b — для минимизации (4).

Из двух функций разного вида, приведенных в Приложении, можем образовать функцию с двумя параметрами: $\psi(x) = f_1(f_2(x))$, обратная функция $\psi^{-1}(u) = f_2^{-1}(f_1^{-1}(u))$. Например, $\psi(x) = \left(\frac{\mathrm{tg}(ax)}{\mathrm{tg}(a)}\right)^p$ образована из степенной и тангенциальной функций. Параметр a в этой функции можем использовать для нормировочного условия (2), а p — для минимизации (4), или наоборот. При определенных значениях a ($f_2(x)$ — выпуклая вниз) и $p < 1$ ($f_1(x)$ — выпуклая вверх) сложная генерирующая функция $\psi(x)$ может образовывать S-образную функцию с точкой перегиба (вторая производная меняет знак).

Перейдем к вопросу об оценке функции ошибок $\varepsilon(\mathbf{u})$. Как отмечалось ранее, линейность $g(u_j, 0)$ обеспечивает $\varepsilon(u_j, 0) = 0$ ($j = 1, 2, ..., m$). Кроме того, при достижении разности (4), близкой к нулю, обеспечиваются небольшие значения $\varepsilon(\mathbf{u})$ по линии ФРЗБК.

Для проверки близости к нулю $\varepsilon(\mathbf{u})$ во всей области определения критериев можно провести опрос ЛПР по схеме полного факторного эксперимента (см. [5]). Но лучше определить функцию, для наглядности желательно одного аргумента, которая будет определяться функцией предпочтения $g(\mathbf{u})$. ФРЗБК является именно такой функцией.

Еще одна такая функция рассмотрена в работе [6], она получена использованием в формуле (3) в качестве переменной $t = V_j u_j$. Тогда (3) запишется в виде:

$$g(t) = \psi\left(\sum_{j=1}^{m} \psi^{-1}(kt) \right) = \psi(m\psi^{-1}(kt)). \qquad (5)$$

Так как критерии u_j изменяются в интервале [0; 1], то переменная $t \in [0; V_1]$, где V_1 — максимальный весовой коэффициент. Напомним, в п. 2.2 мы упорядочили по важности критерии, поэтому максимальным весом является V_1. Выражение (5) справедливо для интервала $t \in [0; V_m]$, где V_m — минимальный вес, соответствующий наименее важному критерию. Как только t достигает V_m, критерий u_m становится равным 1. Поэтому на следующем интервале $t \in [V_m; V_{m-1}]$ вместо (5) получаем следующую функцию:

$$g(t) = \psi(\psi^{-1}(kV_m) + (m-1)\psi^{-1}(kt)). \qquad (6)$$

На последнем интервале $t \in [V_2; V_1]$ получим функцию:

$$g(t) = \psi\left(\sum_{j=2}^{m} \psi^{-1}(kV_j) + \psi^{-1}(kt) \right).$$

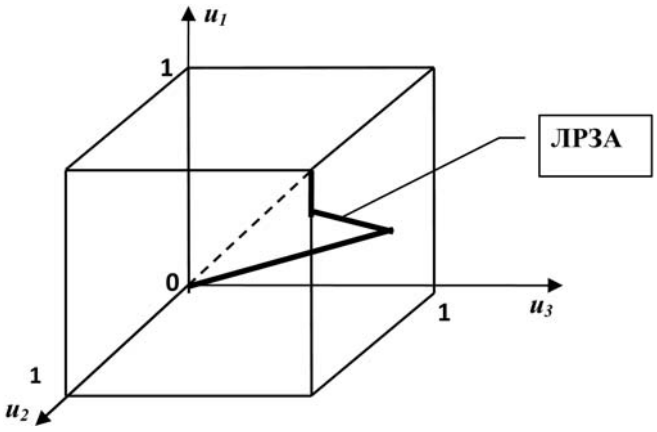


Рис. 4. Линия равных значений аргумента для трех критериев

Таблица 4

Построение анкеты для оценки ФЛРЗА для трех критериев

Оцениваемые объекты	Веса критериев			Оценки ЛПР
	u_1	u_2	u_3	
Объект эталон B^-	k_1^-	k_2^-	k_3^-	0,0
На 1-м интервале $t \in [0; V_3]$	$u_1^{-1}(t/V_1)$	$u_2^{-1}(t/V_2)$	$u_3^{-1}(t/V_3)$	$Z^{(1)}$
На 2-м интервале $t \in [V_3; V_2]$	$u_1^{-1}(t/V_1)$	$u_2^{-1}(t/V_2)$	k_3^+	$Z^{(2)}$
На 3-м интервале $t \in [V_2; V_1]$	$u_1^{-1}(t/V_1)$	k_2^+	k_3^+	$Z^{(3)}$
Объект эталон B^+	k_1^+	k_2^+	k_3^+	1,0

При изменении переменной t в интервале $[0; V_1]$ мы двигаемся по линии, представленной на рис. 4. Будем называть ее *линией равных значений аргумента* (ЛРЗА), а функцию $g(t)$ по этой линии — *функцией по линии равных значений аргумента* (ФЛРЗА). Отметим, что ФЛРЗА непрерывна и определяется генерирующей функцией $\psi(x)$. В [6] показано, что для выпуклой вниз $\psi(x)$ по такой линии оператор агрегирования (3) принимает максимальное значение при заданном t.

Для оценки ФЛРЗА лицом, принимающим решения, необходимо построить множество объектов по ЛРЗА. Ниже приводится план опроса для случая трех критериев (табл. 4).

На последнем интервале $[V_2; V_1]$ ФЛРЗА линейная у мультипликативной функции: $u(\mathbf{k}) = \frac{1}{C} = \left\{ \left[\prod_{j=1}^{m} (1 + CkV_j u_j(k_j)) \right] - 1 \right\}$, образованной на основе показательной генерирующей функции $\psi(x) = \frac{\mathbf{e}^{ax} - 1}{\mathbf{e}^a - 1}$. Параметр a функции $\psi(x)$ определяется из уравнения (2), а коэффициент $C = \mathbf{e}^a - 1$ (см. Приложение). Теория мультипликативной функции приводится в [3]. Там же сформулированы условия ее существования [3, стр. 279, теорема 6.1]: все критерии должны быть взаимонезависимы по полезности.

Используя данные по ФЛРЗА, приведенные в приложении, сформулируем следующее утверждение.

Утверждение 2. Необходимыми условиями для использования мультипликативной функции в качестве функции предпочтения являются: а) масштабный коэффициент $k \neq 1$; б) функция по линии равных значений аргумента на последнем интервале $[V_2; V_1]$ линейная.

Для проверки взаимонезависимости по полезности исследуется вся область определения критериев. Используя ЛРЗА и проведя опрос на последнем ее интервале, можно решить вопрос о применении мультипликативной функции: если ФЛРЗА на этом участке не линейная, то ФП не мультипликативная.

На основании опроса ЛПР по точкам ФЛРЗА можно делать выводы о функции предпочтения. В связи с этим сформулируем следующие утверждения.

Утверждение 3. Необходимые условия для использования L^p-метрики в качестве функции предпочтения: а) масштабный коэффициент $k \neq 1$; б) ФЛРЗА — линейная на первом интервале $[0; V_m]$.

Утверждение 4. Необходимые условия для использования аддитивной функции предпочтения: а) масштабный коэффициент $k = 1$; б) ФЛРЗА — кусочно-линейная на всех интервалах.

Завершая вопрос об идентификации функции предпочтения, следует сказать, что при определении

вида генерирующей функции и ее параметров надо использовать оценки ЛПР по всей области пространства критериев. В качестве критерия близости использовать меру расхождения типа (4) по всем оценкам ЛПР, включая ФРЗБК, ФЛРЗА.

3. Идентификация функций предпочтения для комплексных критериев иерархии

Решив задачу формирования ЧНФП и выбора функции предпочтения на нижних уровнях дерева, можно переходить к идентификации операторов агрегирования на всех вышестоящих уровнях. Определение оператора для каждой ветви следует рассматривать как отдельную задачу.

В каждой такой задаче агрегируемыми критериями являются комплексные, т. е. они безразмерные $k_j \in [0; 1]$. Эти критерии определяют вышестоящий по иерархии критерий в шкале порядков. Для описания предпочтений в таких задачах можно использовать операторы агрегирования, основанные на двух генерирующих функциях. В работах [5, 6] сформулированы аксиомы, при выполнении которых они используются. Это аксиомы монотонности, единственности граничных значений и аксиома порядковой важности.

Аксиома 5 порядковой важности критериев. Для любых двух критериев k_j и k_s, из которых k_j важнее (предпочтительней) k_s, условная функция предпочтения $u(k_j, k_s^-, \overline{k}'_{j,s})$ больше $u(k_j^-, k_s, \overline{k}'_{j,s})$ при любом фиксированном дополнении $\overline{k}'_{j,s}$.

В работах [5, 6] изложены вопросы проверки аксиом и идентификации операторов агрегирования на основе двух генерирующих функций. Если в результате аппроксимации предпочтений получаем приемлемые ошибки, то идентификация оператора прошла успешно.

Если окажется, что аксиома порядковой важности критериев не выполняется или аппроксимация операторами не дает результатов, то необходимо построить множество ЧНФП агрегируемых критериев. Ввиду того, что комплексные критерии изменяются в одинаковом интервале [0; 1], удобно одновременно строить все ЧНФП.

Напомним, что ЧНФП k_j измеряет вышестоящий критерий в шкале интервалов при дополнении $k_j^- = 0$.

Зафиксируем значение критериев на уровне $k^{(1)}$ и проведем опрос ЛПР по предпочтению объектов в шкале интервалов с эталонами $B^+ = \{1, ..., 1\}$ и $B^- = \{0, ..., 0\}$, так же как в п. 2.3 (табл. 5).

Такой опрос проводится для нескольких значений $k^{(i)}$ $(i = 1, ..., n)$. Последним значением должно быть $k^{(n)} = 1$. Оценки $W_j^{(i)} = u(k_j^{(i)}, 0)$ должны быть монотонно возрастающими, так как нижестоящие комплексные критерии в иерархии описывают вышестоящий критерий в шкале порядков.

Таблица 5

Опросная анкета для построения ЧНФП комплексных критериев для одного значения $k^{(1)}$

Оцениваемые объекты	Критерии			Оценки ЛПР
	k_1	k_2	k_3	
Объект эталон B^+	1	1	1	1,0
Объект $B^=$	$k^{(1)}$	$k^{(1)}$	$k^{(1)}$	$R^{(1)}$
Объект B^1	$k^{(1)}$	0	0	$W_1^{(1)}$
Объект B^2	0	$k^{(1)}$	0	$W_2^{(1)}$
Объект B^3	0	0	$k^{(1)}$	$W_3^{(1)}$
Объект эталон B^-	0	0	0	0,0

Кусочно-линейную ЧНФП для k_j образуют множество точек $\{k^{(i)};\ W_j^{(i)} / W_j^{(n)}\}$ $(i = 1, ..., n)$. Действительно,

$$u_j(k_j^{(i)}) = \frac{u(k_j^{(i)}, \overline{k}_j^-)}{u(k_j^+, \overline{k}_j^-)} = \frac{u(k_j^{(i)}, 0)}{W_j^{(n)}} = \frac{W_j^{(i)}}{W_j^{(n)}}.$$

Масштабный коэффициент $k = \sum_{j=1}^{n} W_j^{(n)}$, а веса $V_j = \frac{W_j^{(n)}}{k}$.

Для проверки правильности построенных ЧНФП можно дополнительно провести опрос с одинаковыми $u_j(k_j) = u'$, как это описано в п. 2.3. Если ЧНФП описывают вышестоящий критерий в шкале интервалов, должны получиться линейные функции $g(u_j, 0)$.

Если среди агрегируемых критериев есть единичные и комплексные критерии, то сначала следует построить ЧНФП для комплексных критериев $u_j(k_j^{(i)}) = \frac{W_j^{(i)}}{W_j^{(n)}}$. Затем уже проверить правильность построения ЧНФП через линейность $g(u_j, 0)$ для всех критериев вместе. После этого определяются масштабный коэффициент k и веса критериев V_j.

После определения ЧНФП, k и V_j в качестве оператора можем использовать функцию предпочтения (3). Для идентификации вида генерирующей функции $\psi(x)$ и ее параметров можно использовать полученные в результате опроса ЛПР оценки $R^{(i)}$ или провести дополнительный опрос, как описано в п. 2.5.

Заключение

Описание предпочтений рассматривается как задача измерения свойства "предпочтение". Описание предпочтений в МКЗ с иерархической системой критериев разбивается на множества задач описания комплексных критериев множеством критериев, нижестоящих по уровню.

На нижних уровнях проблема перехода от физических единиц измерения к безразмерным решается построением шкал интервалов измерения единичными критериями вышестоящего комплексного (частных нормированных функции предпочтения (ЧНФП)). Использование ЧНФП определяет тип функции предпочтения для описания вышестоящего комплексного критерия (см. выражение (3)).

Для решения МКЗ с одним уровнем иерархии достаточно использовать рассмотренные в п. 2 процедуры идентификации функции предпочтения.

На последующих уровнях иерархии для описания предпочтений используются либо операторы агрегирования с двумя генерирующими функциями, изложенные в работах [5, 6], при выполнении аксиомы порядковой важности критериев, либо строятся ЧНФП для агрегируемых критериев.

Предлагаемый подход отличает технологичность. Опрос ЛПР для выявления его предпочтений проводится по единому плану для всех уровней иерархии. По результатам опроса определяется функция предпочтения и все ее параметры.

В качестве полученных в статье дополнительных результатов следует отметить формулирование с использованием введенных понятий ФРЗБК и ФЛРЗА необходимых условий применения L^p-метрики, аддитивной и мультипликативной функций для агрегирования критериев.

Список литературы

1. **Саати Т.** Принятие решений — Метод анализа иерархий. М.: Радио и связь, 1993. 320 с.
2. **Саати Т. Л.** Принятие решений при зависимостях и обратных связях. М.: Изд-во ЛКИ, 2008. 360 с.
3. **Кини Р. Л., Райфа Х.** Принятие решений при многих критериях: предпочтения и замещения: Пер. с анг. М.: Радио и связь, 1981. 560 с.
4. **Азгальдов Г. Г.** Теория и практика оценки качества товаров. (Основы квалиметрии). М.: Экономика, 1982. 256 с.
5. **Елтаренко Е. А.** Аппроксимация предпочтений в многокритериальных задачах // Информационные технологии. 2011. № 6. С. 22—32.
6. **Елтаренко Е. А.** Операторы мягкой логики в многокритериальных задачах // Информационные технологии. 2011. № 10. С. 8—18.
7. **Пфанцагль И.** Теория измерений. М.: Мир, 1976. 248 с.
8. **Бешелев С. Д., Гурвич Ф. Г.** Математико-статистические методы экспертных оценок. М.: Статистика, 1980. 263 с.
9. **Литвак Б. Г.** Экспертная информация. Методы получения и анализа. М.: Радио и связь, 1982. 184 с.

Приложение. Примеры функций предпочтений, построенных на основе выпуклых генерирующих функций $\psi(x)$

Название генерирующей функции	$\psi(x)$; $\psi^{-1}(u)$	Уравнение (2) для определения параметра $\psi(x)$	ФРЗБК ($g(u)$)	ФЛРЗА ($g(t)$)	Функция предпочтения (ФП) $u(u_1, ..., u_m)$
Линейная	x; u		u	$mt\ t \in [0;\ V_m]$ $V_m + (m-1)t\ t \in [V_m;\ V_{m-1}]$ $\sum_{s=1}^{m-1} V_s + t$ (линейная) $t \in [V_2; V_1]$	$\sum_{j=1}^{m} V_j u_j(k_j)$ (аддитивная)
Степенная $p > 1$ — выпуклая вниз $p < 1$ — выпуклая вверх	x^p; $u^{1/p}$	$\sum_{j=1}^{m} (kv_j)^{1/p} = 1$	u	$m^p kt\ t \in [0;\ V_m]$ (линейная) $(m-1)^p kt\ t \in [v_m;\ v_{m-1}]$ $\left(\sum_{s=1}^{m-1} (kv_s)^{1/p} + (kt)^{1/p}\right)^p$ $t \in [V_2;\ V_1]$	$k\left[\sum_{j=1}^{m} (V_j u_j(k_j))^{1/p}\right]^p$ (степенная, L^p-метрика)
Показательная выпуклая вниз	$\frac{\mathbf{e}^{ax}-1}{\mathbf{e}^a-1}$; $\frac{1}{a}\ln(1+Cu)$, где $C = \mathbf{e}^a - 1$	$\prod_{j=1}^{m} (1 + (\mathbf{e}^a - 1)kV_j) = \mathbf{e}^a$	$\frac{1}{C}\left\{\left[\prod_{j=1}^{m} (1+CkV_j u)\right]-1\right\}$	$\frac{1}{C}[(1+Ckt)^m - 1]$ $t \in [0;\ V_m]$ $\frac{1}{C}\left\{(1+Ckt)\prod_{s=2}^{m}(1+CkV_s)-1\right\}$ (линейная) $t \in [V_2;\ V_1]$	$\frac{1}{C}\left\{\left[\prod_{j=1}^{m} (1+CkV_j u_j(k_j))\right]-1\right\}$ (мультипликативная)
Показательная выпуклая вверх	$\frac{1-\mathbf{e}^{-bx}}{1-\mathbf{e}^{-b}}$; $-\frac{1}{b}\ln(1-Cu)$, где $C = 1 - \mathbf{e}^{-b}$	$\prod_{j=1}^{m} (1 - (1-\mathbf{e}^{-b})kV_j) = \mathbf{e}^{-b}$	$\frac{1}{C}\left(1 - \prod_{j=1}^{m} (1 - CkV_j u)\right)$	$\frac{1}{C}[1 - (1-Ckt)^m]$ $t \in [0;\ V_m]$ $\frac{1}{C}\left\{1-(1-Ckt)\prod_{s=2}^{m}(1+CkV_s)\right\}$ (линейная) $t \in [V_2;\ V_1]$	$\frac{1}{C}\left(1-\prod_{j=1}^{m} (1-CkV_j u_j(k_j))\right)$ (мультипликативная)
Тангенциальная выпуклая вниз	$\frac{\mathrm{tg}(ax)}{\mathrm{tg}(a)}$; $a \in (0;\ \pi/2)$; $\frac{1}{a}\mathrm{arctg}(Cu)$, где $C = \mathrm{tg}(a)$	$\sum_{j=1}^{m} \mathrm{arctg}(\mathrm{tg}(a)kV_j) = a$	$\frac{1}{C}\mathrm{tg}\left(\sum_{j=1}^{m} \mathrm{arctg}(CkV_j u)\right)$	$\frac{1}{C}\mathrm{tg}(m \cdot \mathrm{arctg}(Ckt))$ $t \in [0;\ V_m]$ $\frac{1}{C}\mathrm{tg}\left(\sum_{s=2}^{m}\mathrm{arctg}(CkV_j)+\mathrm{arctg}(Ckt)\right)$ $t \in [V_2;\ V_1]$	$\frac{1}{C}\mathrm{tg}\left(\sum_{j=1}^{m} \mathrm{arctg}(CkV_j u_j(k_j))\right)$ (тангенциальная)

УДК 519.584

А. А. Зайцев, студент,
Московский физико-технический институт,
стажер-исследователь, Datadvance,
e-mail: likzet@gmail.ru,
В. В. Стрижов, канд. физ.-мат. наук, науч. сотр.,
e-mail: strijov@ccas.ru,
Вычислительный центр РАН,
А. А. Токмакова, студент,
Московский физико-технический институт,
e-mail: aleksandrova-tok@yandex.ru

Оценка гиперпараметров регрессионных моделей методом максимального правдоподобия[1]

Рассматривается задача выбора регрессионной модели. Предполагается, что вектор параметров модели — многомерная случайная величина с независимо распределенными компонентами. Предложен способ оптимизации параметров и гиперпараметров. Приведены явные оценки гиперпараметров для случая линейных и нелинейных моделей. Показано, как полученные оценки используются для отбора признаков. Предложенный подход сравнивается с подходом, использующим для оценки гиперпараметров аппроксимацию Лапласа.

Ключевые слова: *регрессия, выбор признаков, распределение параметров, оценка гиперпараметров, байесовский вывод*

Введение

В данной работе рассматривается задача выбора регрессионной модели [1] из заданного параметрического семейства регрессионных моделей. Один из возможных подходов — введение предположения о распределении параметров модели [2]. В этом случае предполагается, что функция регрессии задана оценкой вектора параметров, который считается нормально распределенной многомерной случайной величиной. Параметры распределения заданы вектором, в дальнейшем называемым вектором гиперпараметров модели.

Впервые этот подход к выбору признаков методом анализа распределения параметров был предложен в работе [3]. Более общий подход был предложен Маккаем в работе [2], где было введено понятие гиперпараметров. Бишоп предложил ряд других способов оценки гиперпараметров, таких как Марковские цепи Монте-Карло и аппроксимация Лапласа [4, 5]. Подход, использующий аппроксимацию Лапласа, был развит в работах [6, 7].

Предлагается для линейной регрессионной модели записать явное выражение функции правдоподобия с учетом введенных вероятностных предположений. Максимизируя функцию правдоподобия, получаем оценки наиболее правдоподобных значений гиперпараметров модели. Такой подход позволяет получать оценки гиперпараметров регрессионных моделей. Для полученных оценок гиперпараметров явно выписываются оценки параметров модели. Они используются для отбора признаков. Предложенный подход сравнивается с подходом, использующим аппроксимацию Лапласа распределения параметров модели [6].

1. Постановка задачи

Задана выборка $D = (\mathbf{X},\ \mathbf{y}) = \{(\mathbf{x}_i,\ y_i)\}_{i=1}^{m}$, где $\mathbf{x}_i \in \mathsf{R}^n$, $y_i \in \mathsf{R}$. Выборка D содержит m элементов. Вектор $\mathbf{x}$ состоит из n независимых переменных. Рассматривается класс регрессионных моделей вида

$$\mathbf{y} = \mathbf{f}(\mathbf{X},\ \mathbf{w}) + \varepsilon. \tag{1}$$

Здесь $\mathbf{X}$ — матрица плана, а $\mathbf{w}$ — вектор параметров модели $\mathbf{f}$. Предполагается, что шум ε — многомерная нормальная случайная величина с нулевым математическим ожиданием и матрицей ковариации $\mathbf{B}^{-1}$:

$$\varepsilon : N(0,\ \mathbf{B}^{-1}), \tag{2}$$

вектор параметров модели $\mathbf{w}$ — многомерная нормальная случайная величина с нулевым математическим ожиданием и матрицей ковариации $\mathbf{A}^{-1}$:

$$\mathbf{w} : N(0,\ \mathbf{A}^{-1}). \tag{3}$$

Требуется получить оценки матриц $\mathbf{A}$, $\mathbf{B}$ согласно гипотезам порождения данных (2) и (3).

2. Функция правдоподобия линейной модели

Рассмотрим линейную регрессионную модель. Тогда выражение (1) имеет вид

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \varepsilon.$$

Плотность распределения параметров $\mathbf{w}$ согласно теореме Байеса имеет вид

$$p(\mathbf{w}|\mathbf{A},\ \mathbf{B},\ D) = \frac{p(D|\mathbf{w},\ \mathbf{B})p(\mathbf{w}|\mathbf{A})}{p(D|\mathbf{A},\ \mathbf{B})}, \tag{4}$$

в котором $p(D|\mathbf{w},\ \mathbf{B})$, $p(\mathbf{w}|\mathbf{A})$ — плотности многомерных нормальных случайных величин вида

$$p(D|\mathbf{w},\ \mathbf{B}) = \frac{1}{(2\pi)^{\frac{m}{2}}|\mathbf{B}|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^{\mathrm{T}}\mathbf{B}(\mathbf{y} - \mathbf{X}\mathbf{w})\Big), \tag{4'}$$

согласно предположению о нормальности распределения шумов (2), и

$$p(\mathbf{w}|\mathbf{A}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{A}|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w}\Big), \tag{4''}$$

[1] Работа выполнена при поддержке РФФИ, грант 10-07-00422.

согласно предположению о распределении вектора параметров модели (3). Функция $p(D|\mathbf{A}, \mathbf{B})$ правдоподобия модели $\mathbf{f}$ имеет вид

$$p(D|\mathbf{A}, \mathbf{B}) = \int_{R^n} p(D|\mathbf{w}, \mathbf{B})p(\mathbf{w}|\mathbf{A})d\mathbf{w}. \quad (5)$$

Для линейных моделей явно выпишем оценки гиперпараметров модели $p(D|\mathbf{A}, \mathbf{B})$. Отметим, что в работе [6] был предложен подход, в котором эти оценки получены с использованием аппроксимации Лапласа. Верна следующая теорема.

Теорема. Функция правдоподобия в предположениях о распределении шума ε (2) и параметров модели $\mathbf{w}$ (3) определяется выражением

$$p(D|\mathbf{A}, \mathbf{B}) = \frac{|\mathbf{B}|^{\frac{1}{2}}|\mathbf{A}|^{\frac{1}{2}}}{(2\pi)^{\frac{m}{2}}|\mathbf{K}|^{\frac{1}{2}}} \exp\left(\frac{1}{2}\mathbf{y}^{\mathrm{T}}(\mathbf{C}^{\mathrm{T}}\mathbf{K}\mathbf{C} - \mathbf{B})\mathbf{y}\right), \quad (6)$$

а его логарифм имеет вид

$$\ln p(D|\mathbf{A}, \mathbf{B}) = -\frac{1}{2}(\ln|\mathbf{K}| + m\ln 2\pi - \ln|\mathbf{B}| - \ln|\mathbf{A}| - \mathbf{y}^{\mathrm{T}}(\mathbf{C}^{\mathrm{T}}\mathbf{K}\mathbf{C} - \mathbf{B})\mathbf{y}), \quad (7)$$

где

$$\mathbf{K} = \mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{X} + \mathbf{A},\ \mathbf{C} = \mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{B}.$$

Доказательство. Подставляя (4′) и (4″) в (5), получим следующее выражение:

$$p(D|\mathbf{A}, \mathbf{B}) = \int_{R^n} \frac{1}{(2\pi)^{\frac{m}{2}}|\mathbf{B}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^{\mathrm{T}}\mathbf{B}(\mathbf{y}-\mathbf{X}\mathbf{w})\right) \times \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{A}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w}\right)d\mathbf{w}.$$

Перепишем произведение двух экспонент как экспоненту от их суммы:

$$p(D|\mathbf{A}, \mathbf{B}) = \int_{R^n} \frac{|\mathbf{B}|^{\frac{1}{2}}|\mathbf{A}|^{\frac{1}{2}}}{(2\pi)^{\frac{n+m}{2}}} \exp\left(-\frac{1}{2}((\mathbf{y}-\mathbf{X}\mathbf{w})^{\mathrm{T}}\mathbf{B}(\mathbf{y}-\mathbf{X}\mathbf{w}) + \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w})\right)d\mathbf{w}.$$

Раскрывая скобки, получаем:

$$p(D|\mathbf{A}, \mathbf{B}) = \int_{R^n} \frac{|\mathbf{B}|^{\frac{1}{2}}|\mathbf{A}|^{\frac{1}{2}}}{(2\pi)^{\frac{n+m}{2}}} \exp\left(-\frac{1}{2}(\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{X}\mathbf{w} - 2\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{y} + \mathbf{y}^{\mathrm{T}}\mathbf{B}\mathbf{y} + \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w})\right)d\mathbf{w}.$$

Введем обозначения $\mathbf{K} = \mathbf{A} + \mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{X}$, $\mathbf{C} = \mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}}$ и, выделяя полный квадрат по выражению $(\mathbf{w} - \mathbf{C}\mathbf{y})$, получим:

$$p(D|\mathbf{A}, \mathbf{B}) = \int_{R^n} \frac{|\mathbf{B}|^{\frac{1}{2}}|\mathbf{A}|^{\frac{1}{2}}}{(2\pi)^{\frac{n+m}{2}}} \exp\left(-\frac{1}{2}((\mathbf{w} - \mathbf{C}\mathbf{y})^{\mathrm{T}}\mathbf{K}(\mathbf{w} - \mathbf{C}\mathbf{y}) - \mathbf{y}^{\mathrm{T}}(\mathbf{C}^{\mathrm{T}}\mathbf{K}\mathbf{C} - \mathbf{B})\mathbf{y})\right)d\mathbf{w}.$$

Так как интеграл плотности многомерного нормального распределения по вектору параметров равен единице, то

$$p(D|\mathbf{A}, \mathbf{B}) = \frac{|\mathbf{B}|^{\frac{1}{2}}|\mathbf{A}|^{\frac{1}{2}}}{(2\pi)^{\frac{m}{2}}|\mathbf{K}|^{\frac{1}{2}}} \exp\left(\frac{1}{2}(\mathbf{y}^{\mathrm{T}}(\mathbf{C}^{\mathrm{T}}\mathbf{K}\mathbf{C} - \mathbf{B})\mathbf{y})\right).$$

Следовательно, искомая функция правдоподобия модели $p(D|\mathbf{A}, \mathbf{B})$ имеет вид (6), а его логарифм — вид (7).

Рассмотрим теперь случай, когда матрица $\mathbf{A}$ — диагональная, а матрица $\mathbf{B} = \beta\mathbf{I}$, где $\mathbf{I}$ — единичная матрица размерности $m \times m$. Логарифм функции правдоподобия $\ln p(D|\mathbf{A}, \mathbf{B})$ имеет вид

$$\ln p(D|\mathbf{A}, \beta) = -\frac{1}{2}(\ln|\mathbf{K}| + m\ln 2\pi - m\ln\beta - \ln|\mathbf{A}| - \beta\mathbf{y}^{\mathrm{T}}(\beta\mathbf{X}\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}} - \mathbf{I})\mathbf{y}),$$

где $\mathbf{K} = \mathbf{A} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}$.

2.1. Вычисление производных функции правдоподобия модели $\ln p(D|\mathbf{A}, \mathbf{B})$ по гиперпараметрам $\mathbf{A}$, $\mathbf{B}$

Для поиска максимума функции правдоподобия будем пользоваться градиентными методами оптимизации [8], поэтому нам понадобятся выражения для производных $\ln p(D|\mathbf{A}, \mathbf{B})$ по гиперпараметрам $\mathbf{A}$, $\mathbf{B}$.

Пусть матрица $\mathbf{A}$ имеет вид $\mathbf{A} = \{\alpha_{ij}\}$, $i, j = \overline{1, n}$, а матрица $\mathbf{B}$ имеет вид $\mathbf{B} = \{\beta_{ij}\}$, $i, j = \overline{1, m}$. Обе матрицы являются симметричными и неотрицательно определенными, так как являются матрицами ковариации.

Верны следующие два свойства производных матриц [9]. Для симметричной матрицы $\mathbf{M}$ верно, что

$$\frac{\partial \ln|\mathbf{M}|}{\partial t} = \mathrm{tr}\left(\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial t}\right),$$

где t — некоторый параметр, $\mathbf{M} = \mathbf{M}(t)$ и tr — след матрицы. Также верно, что

$$\frac{\partial \mathbf{M}^{-1}}{\partial t} = -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial t}\mathbf{M}^{-1}.$$

Введем обозначение $\mathbf{S}^{ij}$ — такая матрица, что для двух индексов k, l верно

$$\mathbf{S}^{ij}_{kl} = \begin{cases} 1, & k = i,\ l = j \text{ или } k = j,\ l = i, \\ 0 & \text{иначе.} \end{cases}$$

Запишем производную $\ln p(D|\mathbf{A}, \beta)$ по β_{ij}:

$$\frac{\partial \ln p(D|\mathbf{A}, \mathbf{B})}{\partial \beta_{ij}} = -\frac{1}{2}(\mathrm{tr}(\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{S}^{ij}\mathbf{X}) - \mathrm{tr}(\mathbf{B}^{-1}\mathbf{S}^{ij}) - \mathbf{y}^{\mathrm{T}}(\mathbf{S}^{ij}\mathbf{X}\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{B} + \mathbf{B}^{\mathrm{T}}\mathbf{X}\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{S}^{ij} - \mathbf{B}^{\mathrm{T}}\mathbf{X}\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{S}^{ij}\mathbf{X}\mathbf{K}^{-\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B} - \mathbf{S}^{ij})\mathbf{y}).$$

Аналогично запишем производную $\ln p(D|\mathbf{A}, \beta)$ по α_{ij}:

$$\frac{\partial \ln p(D|\mathbf{A}, \mathbf{B})}{\partial \alpha_{ij}} = -\frac{1}{2}(\mathrm{tr}(\mathbf{K}^{-1}\mathbf{S}^{ij}) - \mathrm{tr}(\mathbf{A}^{-1}\mathbf{S}^{ij}) + \\ + \mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{X}\mathbf{K}^{-1}\mathbf{S}^{ij}\mathbf{K}^{-\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{y}).$$

Запишем производные функции правдоподобия по гиперпараметрам $\mathbf{A} = \mathrm{diag}(\alpha_1, \alpha_2, ..., \alpha_n)$, $\mathbf{B} = \beta\mathbf{I}$:

$$\frac{\partial \ln p(D|\mathbf{A}, \beta)}{\partial \beta} = -\frac{1}{2}\left((\mathrm{tr}(\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}) - \frac{m}{\beta} + \right.$$

$$\left. + \mathbf{y}^{\mathrm{T}}(2\beta\mathbf{X}\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}} - \mathbf{I} - \beta^2\mathbf{X}\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}})\mathbf{y}\right);$$

$$\frac{\partial \ln p(D|\mathbf{A}, \beta)}{\partial \alpha_i} =$$

$$= -\frac{1}{2}\left(\mathrm{tr}(\mathbf{K}^{-1}\mathbf{I}^{ii}) - \frac{1}{\alpha_i} - \beta^2\mathbf{y}^{\mathrm{T}}\mathbf{X}\mathbf{K}^{-1}\mathbf{I}^{ii}\mathbf{K}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right).$$

Так как получены значения производных функции правдоподобия модели $\ln p(D|\mathbf{A}, \mathbf{B})$ по гиперпараметрам $\mathbf{A}$, $\mathbf{B}$, можно использовать любой градиентный метод оптимизации для оценки гиперпараметров $\mathbf{A}$, $\mathbf{B}$, максимизирующих функцию правдоподобия.

3. Функция правдоподобия нелинейных регрессионных моделей

Функция правдоподобия и ее производные по гиперпараметрам могут не выписываться явно для нелинейных регрессионных моделей в силу того, что интеграл (5) может не браться аналитически. В этом случае возникает необходимость применения приближенных методов оценки гиперпараметров, таких как, например, аппроксимация Лапласа.

Оценим значение интеграла (5) с помощью функции максимального правдоподобия вектора параметров $\mathbf{w}$. Пусть задан вектор параметров $\mathbf{w}_0$, максимизирующих функцию правдоподобия (4), которая с точностью до коэффициента нормализации может быть выписана явно. И пусть вектору $\mathbf{w}_0$ соответствует максимум плотности распределения $p(\mathbf{w}|D, \mathbf{A})$. Тогда

$$p(D|\mathbf{A}, \mathbf{B}) \approx p_L(D|\mathbf{A}, \mathbf{B}) = p(D|\mathbf{w}_0, \mathbf{A}, \mathbf{B})\sqrt{\frac{(2\pi)^n}{|\mathbf{H}|}}, \quad (8)$$

где $\mathbf{H}$ — гессиан, т. е. матрица, элементы которой

$$\mathbf{H}_{ij} = -\frac{\partial^2}{\partial w_i \partial w_j}\ln(p(D|\mathbf{w}, \mathbf{B})p(\mathbf{w}|\mathbf{A})).$$

Подставляя выражения для $p(D|\mathbf{w}_0, \mathbf{A}, \mathbf{B})$ в (8), получаем:

$$\ln p_L(D|\mathbf{A}, \mathbf{B}) = -\frac{1}{2}((\mathbf{y} - f(\mathbf{w}_0, \mathbf{X}))^{\mathrm{T}}\mathbf{B}(\mathbf{y} - f(\mathbf{w}_0, \mathbf{X})) - \\ - m\ln 2\pi + \ln|\mathbf{B}|) - \frac{1}{2}(\mathbf{w}_0^{\mathrm{T}}\mathbf{A}\mathbf{w}_0 + \ln|\mathbf{A}| + \ln|\mathbf{H}|).$$

Отметим, что гессиан $\mathbf{H}$ зависит от гиперпараметров $\mathbf{A}$, $\mathbf{B}$. Так же как и в предыдущем разделе, явно выписываются производные логарифма функции правдоподобия по гиперпараметрам и решается задача максимизации функции правдоподобия $\ln p_L(D|\mathbf{A}, \mathbf{B})$.

4. Использование алгоритма Левенберга—Марквардта для оценки оптимального значения параметров

Если известны значения гиперпараметров $\mathbf{A}$, $\mathbf{B}$ для нелинейной регрессионной модели, то можно использовать алгоритм Левенберга—Марквардта для оценки вектора параметров $\mathbf{w}$. Пусть задано некоторое приближение для значений параметров $\mathbf{w}$. Тогда функция ошибки имеет вид

$$S = \frac{1}{2}(\mathbf{w} + \Delta\mathbf{w})^{\mathrm{T}}\mathbf{A}(\mathbf{w} + \Delta\mathbf{w}) + \\ + \frac{1}{2}(\mathbf{X}(\mathbf{w} + \Delta\mathbf{w}) - \mathbf{y})^{\mathrm{T}}\mathbf{B}(\mathbf{X}(\mathbf{w} + \Delta\mathbf{w}) - \mathbf{y}). \quad (9)$$

Для минимизации функции ошибки воспользуемся алгоритмом Левенберга—Марквардта, который предназначен для оптимизации параметров нелинейных регрессионных моделей. Алгоритм заключается в последовательном приближении заданных начальных значений параметров к искомому локальному оптимуму и является обобщением метода сопряженных градиентов и алгоритма Ньютона—Гаусса.

На нулевой итерации алгоритма задается начальное приближение вектора $\mathbf{w}$. Приращение $\Delta\mathbf{w}$ в точке оптимума для функции ошибки (9) равно нулю. Поэтому для нахождения экстремума приравняем вектор частных производных S по $\mathbf{w}$ к нулю. Для этого представим S в виде двух слагаемых:

$$S_1 = \frac{1}{2}(\mathbf{w} + \Delta\mathbf{w})^{\mathrm{T}}\mathbf{A}(\mathbf{w} + \Delta\mathbf{w}); \quad (10)$$

$$S_2 = \frac{1}{2}(\mathbf{X}(\mathbf{w} + \Delta\mathbf{w}) - \mathbf{y})^{\mathrm{T}}\mathbf{B}(X(\mathbf{w} + \Delta\mathbf{w}) - \mathbf{y}). \quad (11)$$

После дифференцирования получим следующие выражения:

$$\frac{\partial S_1}{\partial \mathbf{w}} = \frac{1}{2}(\mathbf{w} + \Delta\mathbf{w})^{\mathrm{T}}(\mathbf{A} + \mathbf{A}^{\mathrm{T}});$$

$$\frac{\partial S_2}{\partial \mathbf{w}} = \frac{1}{2}[(\mathbf{X}(\mathbf{w} + \Delta\mathbf{w}) - \mathbf{y})^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{X} + (\mathbf{X}(\mathbf{w} + \Delta\mathbf{w}) - \mathbf{y})^{\mathrm{T}}\mathbf{B}\mathbf{X}].$$

Таким образом, чтобы найти приращение $\Delta\mathbf{w}$, необходимо решить систему линейных уравнений:

$$\nabla S = \frac{1}{2}(\mathbf{w} + \Delta\mathbf{w})^{\mathrm{T}}(\mathbf{A} + \mathbf{A}^{\mathrm{T}}) + \frac{1}{2}[(\mathbf{X}(\mathbf{w} + \Delta\mathbf{w}) - \mathbf{y})^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{X} + \\ + (\mathbf{X}(\mathbf{w} + \Delta\mathbf{w}) - \mathbf{y})^{\mathrm{T}}\mathbf{B}\mathbf{X} = 0.$$

Раскроем скобки и приведем подобные слагаемые:

$$\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{X} + \Delta\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{X} - \mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{X} + \mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{X} + \Delta\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{X} - \\ - \mathbf{y}^{\mathrm{T}}\mathbf{B}\mathbf{X} + \mathbf{w}^{\mathrm{T}}\mathbf{A} + \Delta\mathbf{w}^{\mathrm{T}}\mathbf{A} + \mathbf{w}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} + \Delta\mathbf{w}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} = 0.$$

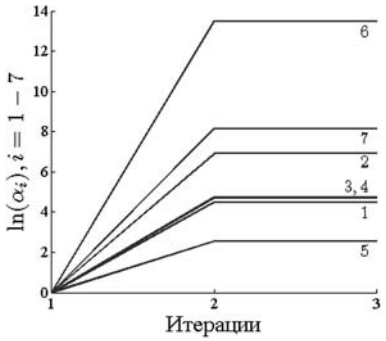


Рис. 1. Зависимость логарифмов значений диагональных элементов матрицы A от номера итерации

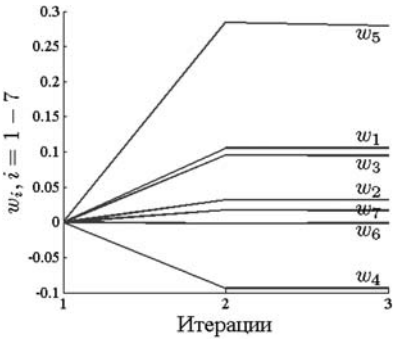


Рис. 2. Зависимость значений параметров w от номера итерации

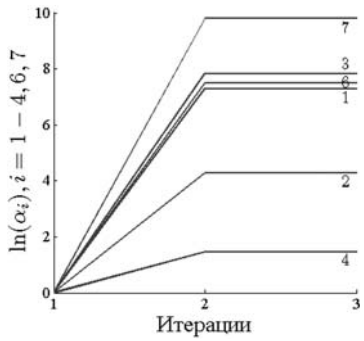


Рис. 3. Зависимость логарифмов значений диагональных элементов матрицы A от номера итерации

Сгруппируем и перенесем в одну сторону члены, содержащие приращение параметров $\Delta\mathbf{w}$:

$$\Delta\mathbf{w}^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{X} + \mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{X} + \mathbf{A} + \mathbf{A}^{\mathrm{T}}) = -\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{X} + \mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{X} - \\ - \mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{X} + \mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{X} - \mathbf{w}^{\mathrm{T}}\mathbf{A} - \mathbf{w}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}.$$

Выразив приращение $\Delta\mathbf{w}$, получим следующую рекуррентную формулу:

$$\Delta\mathbf{w} = [(\mathbf{A} + \mathbf{A}^{\mathrm{T}} + \mathbf{X}^{\mathrm{T}}(\mathbf{B}^{\mathrm{T}} + \mathbf{B})\mathbf{X})^{-1}]^{\mathrm{T}}(-\mathbf{w}^{\mathrm{T}}(\mathbf{A} + \mathbf{A}^{\mathrm{T}}) + \\ + (\mathbf{y} - \mathbf{X}\mathbf{w})^{\mathrm{T}}(\mathbf{B}^{\mathrm{T}} + \mathbf{B})\mathbf{X})^{\mathrm{T}}.$$

Так как матрицы **A**, **B** — симметричные, положительно определенные матрицы ковариации, то приращение вектора $\Delta\mathbf{w}$ определяется выражением

$$\Delta\mathbf{w} = [(\mathbf{A} + \mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{X})^{-1}]^{\mathrm{T}}(-\mathbf{w}^{\mathrm{T}}\mathbf{A} + (\mathbf{y} - \mathbf{X}\mathbf{w})^{\mathrm{T}}\mathbf{B}\mathbf{X})^{\mathrm{T}},$$

т. е. $\Delta\mathbf{w} = (\mathbf{X}^{\mathrm{T}}\mathbf{B}\mathbf{X} + \mathbf{A})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{y} - \mathbf{w}$.

Алгоритм останавливается в том случае, если приращение $\Delta\mathbf{w}$ в последующей итерации меньше заданного значения, либо если параметры **w** доставляют ошибку *S*, меньшую заданного значения. Значение вектора **w** на последней итерации считается искомым.

5. Алгоритм отбора признаков

Полученные значения гиперпараметров α_i, $i = 1, \ldots, n$, для диагональной матрицы **A** могут быть использованы для отбора признаков и выбора модели линейной регрессии. Параметры w_i модели **f** сравниваются по оценкам значений гиперпараметров α_i. Большие значения гиперпараметра α_i означают больший штраф на значение параметра и, следовательно, меньшую значимость данных параметров для качества модели. Малые значения α_i показывают бóльшую значимость данного компонента модели для ее качества.

6. Вычислительный эксперимент

Результатом вычислительного эксперимента является фильтрация шумовых и коррелирующих признаков. Тестирование алгоритма проводится на временном ряде, содержащем информацию о семи компонентах, входящих в состав бетона. Исследуется два отклика: предел прочности при сжатии и морозостойкость. Ряд содержит 103 записи. Необходимо построить регрессионную модель и оценить ее параметры.

При исследовании предела прочности при сжатии алгоритм приводит к следующим результатам.

На рис. 1 представлены логарифмы диагональных элементов матрицы **A**. Шестой элемент почти в два раза больше всех остальных, поэтому соответствующий ему параметр модели w_6 мал, как мы видим из рис. 2 и табл. 1. Однако α_6 не настолько велик, чтобы мы могли убрать соответствующий столбец матрицы плана, так как при этом произойдет увеличение функции ошибки на 20 %.

При исследовании морозостойкости наблюдается вырождение матрицы **A**. Так, на рис. 3 приведен итерационный процесс для всех диагональных элементов α, кроме пятого, так как на третьей итерации $\log(\alpha_5)$ достигает значения 66, что в шесть раз превышает все остальные логарифмы элементов матрицы **A**. Рассматривая рис. 4, 5 и табл. 2, получим, что пятый признак является неинформативным и может быть исключен из матрицы плана. Функция ошибки увеличится менее чем на 1 %.

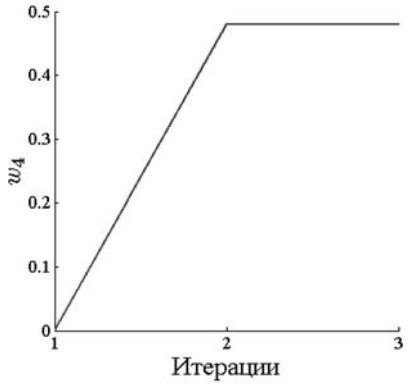


Рис. 4. Зависимость значений параметра w_4 от номера итерации

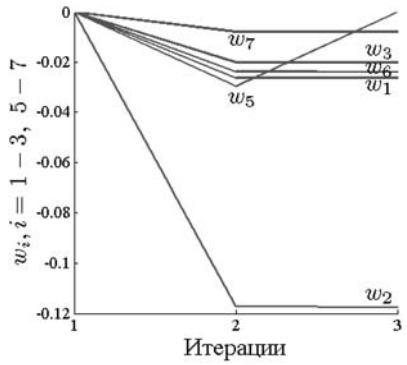


Рис. 5. Зависимость значений параметров w от номера итерации

Таблица 1

Числовые значения параметров модели

w_1	w_2	w_3	w_4	w_5	w_6	w_7
0,1054	0,0317	0,0951	−0,0937	0,2790	−0,0013	0,0168

Таблица 2

Числовые значения параметров модели

w_1	w_2	w_3	w_4	w_5	w_6	w_7
−0,0262	−0,1176	−0,0201	0,4801	0	−0,0238	−0,0079

В обоих случаях использование аппроксимации Лапласа для оценки гиперпараметров приводит к увеличению функции ошибки менее чем на 1 %.

Заключение

Получено точное выражение для функции правдоподобия $\ln p(D|\mathbf{A}, \mathbf{B})$ и предложен подход к его оптимизации. Также проведено сравнение предложенного подхода с аппроксимацией Лапласа искомого правдоподобия. Использование точного выражения для вычисления правдоподобия позволяет получить наиболее точные оценки гиперпараметров.

Список литературы

1. **Burnham K. P., Anderson D. R.** Model selection and multimodel inference: a practical information-theoretic approach. Berlin: Springer. 2002.
2. **MacKay D.** Choice of basis for laplace approximation // Technical report, machine learning. Oxford: Oxford University. 1998.
3. **LeCun Y., Denker J., Solla S., Howard R. E., Jackel L. D.** Optimal brain damage // Advances in neural information processing systems II. San Mateo: Morgan Kauffman. 1990.
4. **Bishop C. M., Tipping M. E.** Bayesian regression and classification // Advances in learning theory: methods, models and applications. Washington: IOS Press. 2000. P. 267—285.
5. **Bishop C. M.** Pattern recognition and machine learning. Berlin: Springer. 2006.
6. **Стрижов В. В., Сологуб Р. А.** Индуктивное порождение регрессионных моделей предполагаемой волатильности для опционных торгов // Вычислительные технологии. 2009. Т. 14, № 5. С. 102—113.
7. **Стрижов В. В.** Поиск параметрической регрессионной модели в индуктивно заданном множестве // Вычислительные технологии. 2007. № 1. С. 93—102.
9. **Нестеров Ю. Е.** Введение в выпуклую оптимизацию. М.: МЦНМО. 2010.
10. **Rasmussen C. E.** Gaussian processes in machine learning // Advanced lectures on machine learning. 2004. Vol. 1. P. 63—71.

УДК 519.612.631

К. Ф. Иванова, канд. техн. наук,
Санкт-Петербургский
государственный университет,
e-mail: Klara.I2010@yandex.ru

Интервальная модель задачи теплопроводности в почве

Предлагается новый подход к оценке интервального решения одномерного дифференциального уравнения теплопроводности в почве. Эмпирические коэффициенты модели уравнения, как правило, не отражают неоднородности теплофизических пространственно-временных характеристик почвы, вызывая существенные ошибки решения. Задание интервальных коэффициентов уравнения позволяет определить границы интервальной температуры в почвенных слоях. Численная аппроксимация дифференциального оператора конечно-разностным аналогом позволяет получить систему линейных алгебраических уравнений с матрицей, элементы которой имеют интервальные границы. Благодаря новому алгебраическому подходу к оценке решения определены диапазон интервального решения и чувствительность задачи, вызванные неточностью задаваемых коэффициентов.

Ключевые слова: *уравнение теплопроводности, пространственная неоднородность, численная аппроксимация, интервальные коэффициенты, оценка, чувствительность*

Введение

В данном исследовании предлагается новый подход к оценке чувствительности решения уравнения теплопроводности, зависящей от интервального задания коэффициентов. Прогнозирование распределения температуры в почвенном слое в общем случае является крайне трудной задачей, так как зависит от многих параметров, таких как состояние атмосферных условий и пространственно-временная неоднородность почвенных параметров [1]. Вариабельность значений теплофизических характеристик почвы зависит от воздействия радиационных потоков Солнца, выпадения осадков, испарения, нарушения плотности поверхностных слоев почвы при проведении сельскохозяйственных мероприятий и других факторов. Даже за сравнительно короткий интервал времени (1—2 суток) мы имеем дело с большим набором неопределенностей. Уточнение температуры ориентировано на все большее включение в расчет взаимосвязанных параметров и функций воздействия на процесс теплопереноса. Однако недостаточная точность входных параметров приводит к неопределенности решения математических моделей и нестабильности прогноза температурного поля независимо от сложности модели. Статистические оценки, такие как числа обусловленности, характеризуют апостериорную усредненную погрешность по всему пространственно-

временному температурному ансамблю и не могут гарантировать реальное значение ошибок прогноза.

Для определения погрешности распределения температуры используется новая методика, связанная с интервальной оценкой погрешности теплофизических коэффициентов, входящих в уравнение. Это становится возможным, если дифференциальный оператор исходного уравнения представить его конечно-разностным аналогом и рассматривать систему линейных алгебраических уравнений (СЛАУ) с интервальными коэффициентами, определяющими решение задачи в узловых точках дискретизации. Для оценки погрешности решения интервальной системы используется "знаковый" подход, предложенный для решения СЛАУ [2] и развитый для оценки чувствительности решений уравнений в частных производных — уравнений Лапласа и Пуассона в [3]. "Знаковый" подход для решения интервальных линейных систем уравнений (ИСЛАУ) впервые был реализован в работе [4], где показана идентичность решения "внешней" интервальной задачи [5] и оценка чувствительности ИСЛАУ предлагаемым методом. В данной работе развитие интервального подхода оценки чувствительности решения уравнений в частных производных проводится в приложении к нестационарному уравнению теплопроводности в почвенном слое.

Основными параметрами, влияющими на теплофизические коэффициенты и определяющими процесс теплопереноса, являются влажность W и плотность почвы ρ. Получение экспериментальных зависимостей теплофизических коэффициентов: теплопроводности $\lambda(W, \rho)$, объемной теплоемкости $c(W, \rho)$ и температуропроводности $a(W, \rho)$ является очень трудоемкой работой. Она сопряжена с большими затратами времени и ресурсов, и построение эмпирических зависимостей зачастую имеет разброс, достигающий 100 %. Поэтому такие измерения в чистом виде можно отнести к неточным или неопределенным с математической точки зрения.

Модель температурного режима почвы позволяет рассматривать почвенный массив как сплошную среду, тепловые свойства которой учитываются эффективными теплофизическими коэффициентами λ, c и a, среди которых наиболее часто используется эффективная температуропроводность (коэффициент температуропроводности) $a(z)$ как функция координаты глубины почвенного слоя. Однако выбор зависимости коэффициента температуропроводности от координаты не может обеспечить достаточно точный прогноз распределения температуры в случае неожиданных или намеренных изменений почвенных характеристик, измерение которых сопряжено с бóльшими трудностями, чем измерение самой температуры. Поэтому переход от эмпирической зависимости коэффициентов к их интервальному представлению является необходимой и закономерной процедурой решения задачи теплопереноса.

Примем в качестве модели распределения температуры T в почвенном слое одномерное нестационарное уравнение теплопроводности с переменными коэффициентами. Допущение одномерности определяется значительно меньшими изменениями градиентов температуры в горизонтальных направлениях по сравнению с вертикальным (по глубине почвы). Направим координатную ось z вниз по вертикали, а время t — по горизонтали и запишем уравнение теплопроводности в общем случае:

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(a(z)\frac{\partial T}{\partial z}\right) + f(z, t),\ 0 < z < h,\ t > 0, \quad (1)$$

где $a(z)$ — коэффициент температуропроводности, характеризующий изменение температуры на единице площади в единицу времени и равный частному от деления коэффициента теплопроводности на объемную теплоемкость почвенной фракции — $\lambda(z)/c(z)$. Источник, выраженный слагаемым $f(z, t)$, принят равным 0.

Задание граничных условий при $z = 0$ отражает периодическое изменение температуры поверхности почвы (в частности, в течение суток), когда температура поверхности тела меняется согласно заданному закону как функция времени:

$$T|_{z=0,t} = T_c + A\cos(wt), \quad (2)$$

T_c — среднее значение температуры по глубине почвы; амплитуда A — отклонение температуры на верхней границе от T_c. На нижней границе почвенного массива температуру почвы T_h можно считать условно постоянной величиной, учитывая ее слабую зависимость от времени:

$$T|_{z = z_n,\, t} = T_n. \quad (3)$$

Начальное распределение температуры по глубине почвы при $t = 0$ зависит от координаты и может быть экспериментально получено или смоделировано на основе глубины затухания температурной волны:

$$T|_{z,\, t=0} = G(z, 0). \quad (4)$$

Представление дифференциального уравнения и граничных условий конечными разностями приводит к системе линейных алгебраических уравнений размерности $n \times s$. На отрезке $[z_{\min}, z_{\max}]$ задается равномерная сетка по z с шагом Δz, $z = \{z_i | i = 1, 2, ..., n\}$; на отрезке $[t_{\min}, t_{\max}]$ — равномерная сетка по t с шагом Δt, $t = \{t_l | l = 1, 2, ..., s\}$. Пространственно-временное разбиение области по z и t определяет сетку, на которой дискретизируется диф-

ференциальный оператор уравнения со своими граничными условиями:

$$\frac{T_i^{l+1} - T_i^l}{\Delta t} = \frac{a_i^{l+1}(T_{i+1}^{l+1} - T_i^{l+1}) - a_{i-1}^{l+1}(T_i^{l+1} - T_{i-1}^{l+1})}{\Delta z^2},$$

$$i = \overline{1, n}\ ,\ l = \overline{1, s}\ , \tag{5}$$

$$g1(z_1, t_l) = T_c + A\cos(wt); \tag{6}$$

$$g2(z_n, t_l) = T_h; \tag{7}$$

$$g3(z_i, t_1) = G(z, 0), \tag{8}$$

где функция (6) соответствует периодическому колебанию температуры на поверхности почвы, а функции (7)—(8) — заданным граничным условиям 1-го рода.

Решение системы уравнений (5)—(8) ищется в узлах сеточной области. Значение температуры в узлах пространственно-временной сетки имеет сквозную нумерацию по столбцам сетки. Коэффициенты разностной системы (5) с ее граничными условиями (6)—(8) являются элементами четырехдиагональной матрицы **U** размерности $n \times s$.

В том случае, когда входные данные системы заданы с ошибками, вызванными погрешностями измерений, исходная система (5)—(8) с эмпирическими матрицами и правыми частями может быть записана в виде матричного уравнения

$$\breve{\mathrm{U}}\,\mathrm{x} = \breve{\mathrm{b}}\ (\breve{\mathrm{U}} \in \mathbf{R}^{n \times s},\ \breve{\mathrm{b}} \in \mathbf{R}^n). \tag{9}$$

Полагая, что для заданных погрешностей известны границы допустимых отклонений, систему уравнений (9) можно представить интервальным матричным уравнением

$$\mathbf{UT} = \mathbf{b}$$

с интервальной матрицей $\mathbf{U} \in \mathbf{IR}^{n \times s}$ и интервальным вектором $\mathbf{b} \in \mathbf{IR}^n$. Интервальную матрицу **U** и вектор **b** можно отождествить с матрицей и вектором, составленным из интервалов $\mathbf{U} = [\underline{\mathrm{U}}, \overline{\mathrm{U}}]$, и $\mathbf{b} = [\underline{\mathrm{b}}, \overline{\mathrm{b}}]$, границы которых являются симметричными отклонениями слева и справа относительно измеренных значений $\breve{\mathrm{U}}\,(u_{il})$ и $\breve{\mathrm{b}}\,(b_i)$:

$$\mathbf{U} = [\underline{\mathrm{U}}, \overline{\mathrm{U}}] = \{\breve{\mathrm{U}} \in \mathbf{R}^{n \times s} | \underline{\mathrm{U}} \leqslant \breve{\mathrm{U}} \leqslant \overline{\mathrm{U}}\},$$
$$\underline{\mathrm{U}} = \breve{\mathrm{U}} - \Delta,\ \overline{\mathrm{U}} = \breve{\mathrm{U}} + \Delta; \tag{10}$$

$$\mathbf{b} = [\underline{\mathrm{b}}, \overline{\mathrm{b}}] = \{\breve{\mathrm{b}} \in \mathbf{R}^{n \times s} | \underline{\mathrm{b}} \leqslant \breve{\mathrm{b}} \leqslant \overline{\mathrm{b}}\},$$
$$\underline{\mathrm{b}} = \breve{\mathrm{b}} - \mathrm{d},\ \overline{\mathrm{b}} = \breve{\mathrm{b}} + \mathrm{d}$$

или в компонентах

$$\boldsymbol{u}_{il} = [\breve{u}_{il} - \Delta_{il},\ \breve{u}_{il} + \Delta_{il}],\ i = \overline{2, n-1}\ ,\ l = \overline{2, s}\ ,$$
$$\boldsymbol{b}_i = [b_i - d_i,\ b_i + d_i],\ i = \overline{2, n-1}\ . \tag{11}$$

В развернутом виде матричное уравнение записывается как

$$\begin{pmatrix}
1\ 0\ 0\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 0\\
0\ 1\ 0\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 0\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 0\\
0\ \ldots\ldots 0\ 1\ 0\ \ldots\ldots\ldots\ldots\ldots\ldots 0\\
0\ \ldots\times\ldots\ldots 0\ \times\times 0\ \ldots 0\ \times 0\ \ldots\ldots\ldots\ldots 0\\
0\ \ldots\ldots\times\ldots\ldots 0\ \times\times 0\ \ldots\times\ \ldots\ldots\ldots\ldots\ldots\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
0\ \ldots\ldots\ldots\ldots 0\times\ldots\ldots 0\times\times 0\ 0\ \ldots\times\ldots\ldots 0\\
0\ \ldots\ldots\ldots\ldots\ldots\ldots 0\ 0\ 1\ 0\ \ldots\ldots\ldots 0\\
0\ \ldots\ldots\ldots\ldots\ldots 0\times 0\ \ldots 0\ \times\times 0\ \ldots\ldots\ldots\times\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
0\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 1\ \ldots\ldots\ldots 0\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 1\ 0\\
0\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 1
\end{pmatrix} =$$

$$= \begin{pmatrix} T_1\\ T_2\\ \cdot\\ T_{i-2}\\ T_{i-1}\\ T_i\\ \cdot\\ \cdot\ T_{m-1}\\ T_m\\ T_{m+1}\\ \cdot\\ T_p\\ \cdot\\ \cdot\\ T_n \end{pmatrix} = \begin{pmatrix} b_1\\ b_2\\ \cdot\\ b_{i-2}\\ b_{i-1}\\ b_i\\ \cdot\\ b_{m-1}\\ b_m\\ b_{m+1}\\ \cdot\\ b_p\\ \cdot\\ \cdot\\ b_n \end{pmatrix}, \begin{pmatrix} b_1\\ b_2\\ \cdot\\ b_{i-2}\\ b_{i-1}\\ b_i\\ \cdot\\ b_{m-1}\\ b_m\\ b_{m+1}\\ \cdot\\ b_p\\ \cdot\\ \cdot\\ b_n \end{pmatrix} = \begin{pmatrix} g1\\ g1\\ \cdot\\ g3\\ f_{i-1}\\ f_i\\ f_{i+1}\\ \cdot\\ f_{m-1}\\ g3\\ \cdot\\ \cdot\ g3\\ \cdot\\ g2\\ g2 \end{pmatrix}. \tag{12}$$

Порядковый номер узловой функции $\boldsymbol{T}_j$, $j = \overline{1, n \times s}$ определяется произведением $n(i-1) + l$, $l = \overline{1, s}$; $i = \overline{1, n}$ (i — индекс строки, l — столбца) [6, 7]. Такая индексация позволяет программно очень гибко менять порядок алгебраических систем и вводить соответствующие входные параметры. Внутренние элементы матрицы, обозначенные значками "×", имеют следующее индексное представление:

$$u(n(i-1) + l,\ n(i-1) + l) =$$
$$= 1/\Delta t + a(z(i))/\Delta z^2 + a(z(i-1))/\Delta z^2;$$

$$u(n(i-1) + l,\ n(i-2) + l) = -a(z(i-1))/\Delta z^2;$$

$$u(n(i-1) + l,\ ni + l) = -a(z(i))/\Delta z^2; \tag{13}$$

$$u(n(i-1) + l,\ n(i-1) + l - 1) = -1/\Delta t,$$
$$i = \overline{2, n-1}\ ,\ l = \overline{2, s}\ .$$

Интервальный вектор правых частей **b** состоит из компонент, которые являются значениями функции источника f во внутренних точках области и значениями функций $\boldsymbol{g1}$, $\boldsymbol{g2}$, $\boldsymbol{g3}$, заданными на границах области

$$\mathbf{b}^{\mathrm{T}} = (b_1\ b_2\ \ldots\ b_{i-2} b_{i-1}\ b_i\ \ldots\ b_{m-1}\ b_m\ b_{m+1}\ \cdots$$
$$\ldots\ b_p\ \ldots\ b_{n-1}\ b_n)^{\mathrm{T}} = (g1\ g2\ \ldots\ g3\ f_{i-1} f_i\ \ldots$$
$$\ldots f_{m-2} f_{m-1}\ g3\ \ldots\ g3\ \ldots\ g2g2)^{\mathrm{T}}.$$

Разреженная матрица системы (12), как матрицы исходной системы (9), имеет размерность $n \times s$. Так как решение интервальной системы (12) построено на методике, допускающей существование обратных матриц, полагаем $s = n$. Ограничить размерность матрицы можно выбором шагов дискретизации по z и t с сохранением расчета температуры адекватности физике процесса. В данной работе нас не интересует точность, с которой будет считаться конечно-разностная система в зависимости от точности аппроксимации. Достаточно, чтобы разностная схема (5) была согласованной. Задачей исследования является определение максимально возможных отклонений температуры в почвенном слое при решении системы (12) с интервальными коэффициентами при одних и тех же шагах дискретизации по координате и по времени, что и для исходной системы (9).

Способы решения интервальных систем большого размера известны в литературе и представляют собой значительные трудности как в математическом плане, так и в программной реализации [8]. В частности, интервальный вектор **x** может быть вычислен с помощью метода Гаусса, если **U** является M-матрицей [9]. Напомним, что матрица $U \in \mathbf{R}^{n \times n}$ называется M-матрицей, если $u_{il} \leqslant 0$ для $i \neq l$ и существует обратная неотрицательная матрица U^{-1}, т. е. $\mathrm{U}^{-1} \geqslant 0$. Если каждая вещественная матрица U, принадлежащая заданной интервальной матрице **U**, является M-матрицей, то **U** также является M-матрицей.

Предлагаемый подход к решению интервальной системы (12) основывается на определении максимальных отклонений компонент вектора неизвестных исходя из наибольшего линейного приращения определителя исходной матрицы и вектора правой части. Эта методика приводит к нахождению таких точечных матриц U^- и U^+ и векторов b^- и b^+, которые способствуют максимальному изменению температурных профилей по сравнению с их эмпирическими значениями. Для заданных интервальной матрицы $\mathbf{U} \in \mathbf{IR}^{n \times n}$ и интервального вектора $\mathbf{b} \in \mathbf{IR}^n$ рассматривается задача оценивания множества решений $\Xi = \{\mathbf{x} = \mathrm{U}^{-1}\mathrm{b} | \mathrm{U} \in \mathbf{U}, \mathrm{b} \in \mathbf{b}\}$, где в качестве матриц U берутся точечные матрицы U^- и U^+, а векторов $\mathrm{b} - \mathrm{b}^-$ и b^+.

С этой целью проводится преобразование интервальной матрицы **U** в две точечные $\mathrm{U}^- \in \mathbf{U}$ и $\mathrm{U}^+ \in \mathbf{U}$ и формирование на этом основании двух алгебраических систем уравнений

$$\mathrm{U}^-\mathrm{x} = \mathrm{b}^-(\mathrm{U}^- \in \mathbf{R}^{n \times n},\ \mathrm{b}^- \in \mathbf{R}^n) \text{ и}$$
$$\mathrm{U}^+\mathrm{x} = \mathrm{b}^+\ (\mathrm{U}^+ \in \mathbf{R}^{n \times n},\ \mathrm{b}^+ \in \mathbf{R}^n). \qquad (14)$$

Матрицы U^- и U^+ конструируются из элементов граничных матриц $\underline{\mathrm{U}}$ и $\overline{\mathrm{U}}$ исходя из величины максимальных линейных приращений определителя матрицы измерений $\breve{\mathrm{U}}$ в положительном и в отрицательном направлениях. Этот результат достигается выбором определенной границы (левой или правой) каждого элемента интервальной матрицы исходя из знака произведения измеренного значения элемента $\breve{u}_{il}$ на соответствующее ему алгебраическое дополнение $\breve{\mathrm{U}}_{il}$. Формально матрицы U^- и U^+ уже не являются границами матрицы **U**, подобно начальным $\underline{\mathrm{U}}$ и $\overline{\mathrm{U}}$.

Соотношения (11) можно переписать с учетом относительных погрешностей, где абсолютные отклонения $\Delta_{il} = \varepsilon_{il}\breve{u}_{ij}$ и $d_i = \delta_i \breve{b}_i$ выражаются через произведение относительных погрешностей ε_{il} и δ_i на измеренные значения $\breve{u}_{il}$ и $\breve{b}_i$:

$$\begin{cases} \breve{u}_{il}^{\pm} = [\breve{u}_{il} \pm \breve{u}_{il}\varepsilon_{il}\mathrm{sgn}(\breve{u}_{il}\breve{\mathrm{U}}_{il})], \\ (+), \text{ если } \varepsilon_{il} \text{ и } \mathrm{sgn}(\breve{u}_{il}\breve{\mathrm{U}}_{il}) \text{ одного знака}, \\ (-), \text{ если } \varepsilon_{il} \text{ и } \mathrm{sgn}(\breve{u}_{il}\breve{\mathrm{U}}_{il}) \text{ разного знака}, \end{cases}$$
$$i = \overline{2, n-1},\ l = \overline{2, n}; \qquad (15)$$

$$\begin{cases} \breve{b}_i^{\pm} = [\breve{b}_i \pm \breve{b}_i\delta_i\mathrm{sgn}(\breve{b}_i\breve{\mathrm{U}}_{il})], \\ (+), \text{ если } \delta_i \text{ и } \mathrm{sgn}(\breve{b}_i\breve{\mathrm{U}}_{il}) \text{ одного знака}, \\ (-), \text{ если } \delta_i \text{ и } \mathrm{sgn}(\breve{b}_i\breve{\mathrm{U}}_{il}) \text{ разного знака}, \end{cases}$$
$$i = \overline{2, n-1},\ l = \overline{2, n}. \qquad (16)$$

Важным моментом при оценке погрешности элемента является матрица алгебраических дополнений $\breve{\mathrm{U}}_{il}$, которая в соответствии с определением M-матрицы, в нашем случае содержит лишь положительные компоненты, и конструирование точечных матрицы U^- и U^+ и векторов b^- и b^+ определяется только знаками элементов $\breve{u}_{il}$ и компонент $\breve{b}_i$. В этом случае компоненты b^- и b^+ совпадают с прежними граничными значениями $\underline{\mathrm{b}}$ и $\overline{\mathrm{b}}$ вектора $\mathbf{b} = [\underline{\mathrm{b}}, \overline{\mathrm{b}}]$. Из условий (15) следует, что положительные элементы $\breve{u}_{il}$ обеспечивают максимальное приращение определителя в положительном направлении с положительными значениями относительных погрешностей ε_{ij}, а отрицательные элементы — с отрицательными. Для изменения направления приращения детерминанта знаки относительных погрешностей ε_{ij} для всех коэффициентов должны измениться на противоположные.

Интерес представляет погрешность коэффициента температуропроводности $a(z)$, являющегося

только частью, а не всем элементом $\breve{u}_{il}$, что не изменяет в этой модели общей идеологии оптимизации определителя. Выбор приращений коэффициентов, имеющих разные знаки на соседних диагоналях матрицы, соответствует физически реальному изменению коэффициента температуропроводности при послойной дискретизации дифференциального оператора уравнения. В этом состоит принципиальная разница между коэффициентными матрицами конечно-разностных уравнений и заполненными матрицами СЛАУ, коэффициенты которых, как правило, не содержат зависимых по знаку компонент.

Одновременные отклонения всех компонент матрицы, определяющие наибольшую погрешность решения, хотя и маловероятны, но в случае ленточных матриц вполне реальны. Максимальные изменения температуры по сравнению с их эмпирическими значениями представляют собой угловые значения компонент вектора неизвестных и являются решением так называемой "внешней" задачи для системы линейных интервальных алгебраических уравнений.

Уменьшение шагов дискретизации системы (12) ограничивается ростом размерности системы, так как может привести к плохой обусловленности и вырожденности систем (14). Это происходит как при увеличении относительной погрешности ε, так и в случае распространения интервального задания коэффициентов на все более глубокие почвенные слои.

Во многих случаях нас интересует решение, вызванное ограниченным числом интервальных коэффициентов, а не всех элементов матрицы, что легко объясняется разной степенью неопределенности теплофизических параметров по глубине слоя. В естественных условиях наибольшие изменения влажностного состояния и механического состава претерпевают именно верхние слои почвы, и заданием однозначной эмпирической зависимости теплофизических коэффициентов нельзя учесть вариации температурного поля и его градиентов, возникающие под влиянием антропогенных воздействий и погодных изменений. В математической модели этого можно достигнуть послойным интервальным представлением коэффициентов только одной или нескольких верхних строк и выяснением их влияния на погрешность решения.

Рассмотрим пример решения задачи при задании следующих параметров, входящих в уравнение, граничные и начальное условия. Глубина почвенного слоя $h = 50$ см, время расчета $t = 24$ ч, среднее значение температурного профиля $T_c = 22$ °C, амплитуда $A = 12{,}4$ °C, температура $T_h = 16{,}5$ °C на нижней расчетной глубине почвенного массива. Зависимость коэффициента температуропроводности от координаты аппроксимируется полиномом 4-й степени:

$$a(z) = 0{,}0141z - 0{,}0807z^2 + 0{,}2474z^3 - 0{,}24z^4,$$

который может быть получен из решения задачи восстановления. Начальный температурный профиль $T(z, 0)$ записывается функцией

$$g3(z) = T_c + A\exp(-z/d)\cos(-\log(A_z/A),$$

где A_z — изменение амплитуды температуры по глубине почвенного слоя; d — глубина затухания максимальной температуры в **e** раз.

Для более удобного анализа решения уравнения теплопроводности введены безразмерные величины для координаты и времени:

$$\tilde{z} = z/h;\ \mathrm{Fo} = ta_1/h^2,$$

где Fo — число Фурье; a_1 (м^2/ч) — первый коэффициент в функции температуропроводности, вынесенный за скобки в полиномиальной зависимости $a(z)$. Интервалы изменения безразмерных величин $0 \leqslant \tilde{z} \leqslant 1$; $0 \leqslant \mathrm{Fo} \leqslant 0{,}672$, где $\tilde{z} = 1$ соответствует 50 см, Fo = 0,612 — суткам или 24 ч. Для матрицы размерности 16 × 16 в реальном масштабе шагу по координате соответствует $\Delta z = 3{,}33$ см, шагу по времени — $\Delta t = 1{,}67$ ч, которые гарантируют сохранение физического смысла задачи. С увеличением числа Фурье Fo возрастает время счета, но одновременно увеличивается шаг дискретизации по t, что не всегда удобно для прогноза температуры.

Рассмотрим результаты проведенных вычислений. На графиках рис. 1, *а* (см. третью сторону обложки) изображены температурные профили, полученные по эмпирическим зависимостям и вычисленные при правых границах элементов матрицы u_{il}^{+} при возрастании коэффициентов температуропроводности и левых границах u_{il}^{-} при уменьшении $a(z)$. Из графиков видно, что ширина отклонений профилей меняется по глубине почвы, и ее максимальное значение определяется шириной l заданного промежутка интервальности коэффициентов. Так, при $l = 20$ см максимальные отклонения температуры, близкие к 7 °C, уменьшаются с глубиной, приближаясь к нулевому значению на нижней границе почвы. По-разному ведут себя градиенты температуры интервального решения задачи. Для верхнего почвенного горизонта (~10 см) характерно возрастание градиента температуры при увеличении коэффициента температуропроводности и его уменьшение для левой границы $a(z)$. Можно считать, что использование интервальных коэффициентов температуропроводности в модели как бы предусматривает зависимость температуры от изменения плотности и влажности деятельного слоя почвы в реальных условиях. Графически отражено, в каком направлении в этом случае можно прогнозировать характер отклонения температуры от тех ее значений, которые рассчитываются при эмпирических коэффициентах уравнения. С увеличением глубины почвы градиентный характер изменения температуры приобретает более сложный

характер и требует дополнительного детального исследования. Общий характер отклонения эмпирического решения с интервальными коэффициентами матрицы показан на графиках рис. 1, *а* и 1, *б* (см. третью сторону обложки).

На рис. 2 показан характер изменения относительных погрешностей эмпирического решения при предыдущем интервальном задании коэффициентов температуропроводности с $\varepsilon = 5$ % на глубине 20 см. Из графиков видно, что максимальная относительная погрешность отклонения температуры составляет ~30 % для положительных отклонений и ~20 % — для отрицательных. Однако такие значительные погрешности возникают только в очень узком слое (примерно 20 см). Если же рассчитывать среднее значение по всему ансамблю относительных отклонений решения, то оно не превосходит 7 %. Этот факт свидетельствует о том, что предложенная методика интервальной оценки решения задачи направлена на выявление именно определенной области, подозрительной на возможные максимальные отклонения температуры.

На графиках рис. 3 показаны послойные распределения температуры в течение суток при относительной погрешности $\varepsilon = 5$ % интервальных коэффициентов температуропроводности в слое $l = 20$ см (что соответствует предыдущим данным) и с относительной погрешностью начального распределения температуры по глубине, равной $\delta = 5$ %. Из графиков видно, в каких пределах в зависимости от времени меняется диапазон интервальных отклонений температуры, возрастая на первых шагах и убывая к концу периода счета при интервальном задании коэффициентов матрицы. При интервальном задании граничных условий относительная погрешность температуры остается постоянной.

Присутствие погрешности в граничных условиях линейно влияет на погрешность решения, и совместное действие на отклонение температуры интервальности коэффициентов матрицы и граничных условий суммируется.

На графиках рис. 4 показан характер изменения погрешности температурных профилей в зависимости от задания глубины слоя ($l = 20$ см и $l = 40$ см) коэффициентов, представленных в интервальном виде. Диапазон погрешности температуры при $l = 20$ см составляет через первые два часа ~7 °С, через 12 ч уменьшается до ~5,5 °С, через 24 ч диапазон погрешности сокращается до ~2,2 °С. При $l = 40$ см этот диапазон составляет соответственно ~15, ~10 и ~5,5 °С. В обоих вариантах расчета интервальная погрешность температуры, возникающая в начальный момент времени, к 24 ч счета значительно понижается, что естественно для процесса стабилизации температуры в почвенном слое.

С точки зрения оценки чувствительности решения интересно посмотреть влияние интервального задания коэффициентов уравнения и граничных условий, действующих в "противофазе", т. е. когда их приращения инициируют погрешность решения в противоположных направлениях. При относительных погрешностях коэффициентов с $\varepsilon = 10$ % на

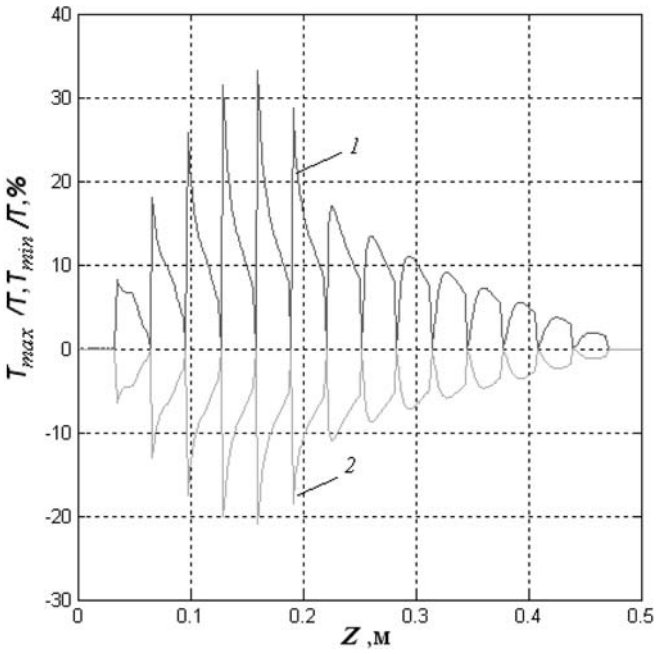


Рис. 2. Относительные погрешности температуры, полученные при сравнении решений с эмпирическими и интервальными коэффициентами, заданными в слое $l = 20$ см, с $\varepsilon = 5$ %:
1 — для правой, *2* — для левой интервальных границ коэффициента температуропроводности

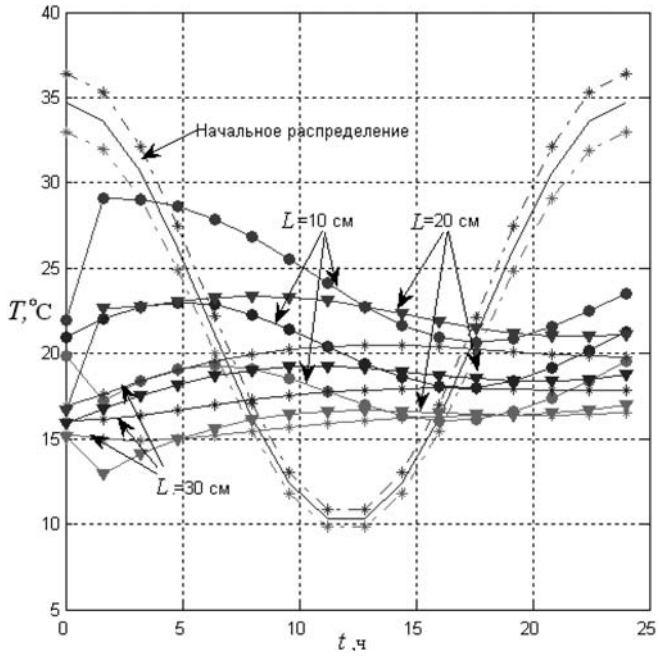


Рис. 3. Температурные распределения в течение 24 ч в слоях разной глубины (10, 20 и 30 см) с интервальными коэффициентами температуропроводности в слое $l = 20$ см, с относительной погрешностью $\varepsilon = 5$ % и начальным интервальным распределением температуры с относительной погрешностью $\delta = 5$ %

глубине $l = 15$ см и граничных условий, равных $\delta = 0$, 10 и 20 %, погрешность решения может значительно компенсироваться, что заметно по всей глубине почвенного массива. Этот результат отражен на графиках рис. 5 (см. четвертую сторону обложки).

Пространственное представление изменения профилей температуры в координатах $z - t - T$ показаны на рис. 6 (см. четвертую сторону обложки). Из их сравнения виден характер изменения температуры с интервальной погрешностью и без погрешности коэффициентов температуропроводности. На рис. 6, *б* наблюдается заметная вогнутость поверхности температуры, а на рис. 6, *в* — выпуклость, что лишний раз дополняет картину интервальных профилей температуры на плоскости.

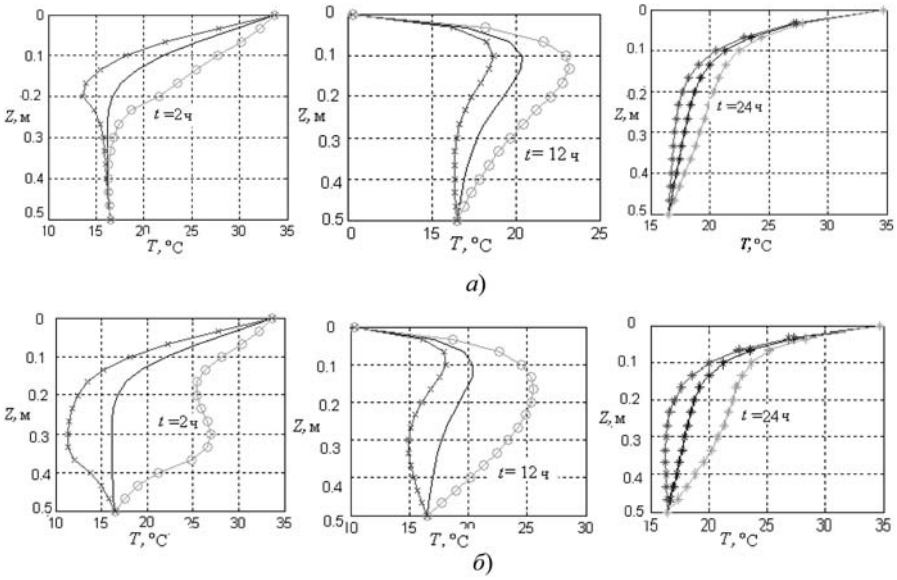


Рис. 4. Интервальные профили температуры, полученные через 2, 12 и 24 ч после задания интервальных коэффициентов в слоях:
а — $l = 20$ см; *б* — $l = 40$ см ($\varepsilon = 5$ %)

Выводы

- Заявлен новый алгебраический подход к оценке решения классической "внешней" задачи уравнения в частных производных для интервальной линейной системы, полученной конечно-разностной аппроксимацией дифференциального оператора уравнения.
- Впервые предложен интервальный метод численного решения уравнения теплопроводности в почвенном слое с представлением коэффициента температуропроводности в интервальном виде.
- Проведена оценка диапазона интервальных отклонений решения при относительных погрешностях коэффициентов уравнения, заданных в пределах 5...20 %.
- Получены максимальные интервалы температурного распределения при послойно задаваемых коэффициентах температуропроводности в почве.
- Вычислены относительные погрешности максимальных отклонений температуры в расчетных точках области.
- Установлена динамика погрешности температуры в течение суток при задании интервальных коэффициентов матрицы в слое на первом шаге счета по времени.
- Определено влияние на погрешность температурного поля задания начальных и граничных условий в интервальном виде.
- Показана целесообразность интервальных оценок температурных профилей в почвенном массиве в целях прогноза динамики температуры при изменении теплофизических характеристик почвы.

Разработан алгоритм и составлена программа расчета интервального решения задачи теплопроводности в почвенном слое, формализованная *m*-файлом в системе MATLAB с привлечением графического редактора GUIDE, позволяющего во входном листинге в режиме on-line задавать входные параметры рассматриваемой задачи с дальнейшим графическим представлением решения и результатов проводимого исследования [11].

Список литературы

1. **Архангельская Т. А.** Закономерности пространственного распределения температуры почв в комплексном почвенном покрове: Автореф. дисс. ... д-ра биол. наук. МГУ, кафедра физики и мелиорации почв. 2008. 50 с.
2. **Петров Ю. П.** Как обеспечить надежность решения систем уравнений. Л.: БХВ-СПб. 2009. 172 с.
3. **Иванова К. Ф.** Оценка погрешности численного решения уравнений Пуассона под воздействием флуктуации входных параметров в среде Matlab. Санкт-Петербург: ПИЯФ РАН. 2010. 34 с.
4. **Иванова К. Ф.** Знаковый подход к оценке решения интервальных линейных систем // Информационные технологии. 2012. № 9. С. 46—53.
5. **Шарый С. П.** Конечномерный интервальный анализ. Новосибирск: "XYZ". 2007. 700 с.
6. **Петрусев А. С.** Разностные схемы и их анализ: учеб. пособие. М.: МФТИ. 2004. 200 с.
7. **Ортега Дж., Пул У.** Введение в численные методы решения дифференциальных уравнений. М.: Наука, 1986.
8. **Alefeld G., Mayer G.** Interval analysis: theory and applications // Journal of Computational Applied Mathematics. 2000. Vol. 121. P. 421—464.
9. **Шарый С. П.** Алгебраический подход во "внешней задаче" для интервальных линейных систем // Фундаментальная и прикладная математика. 2002. Т. 8, № 2. С. 567—610.
10. **Gulsor C. and Ekberli I.** A Comparison of Estimated and Measured Diurnal Soil Temperature Through a Clay Soil Depth // J. of Applied Sciences. 2004.№ 4 (3). P. 418—423.
11. **Иванова К. Ф.** Свидетельство РФ о государственной регистрации программы для ЭВМ № 2011611641 "Программный комплекс оценки экстремальных значений погрешности выходных характеристик решения стационарных задач строительной механики, тепловых и электромагнитных процессов (ПКОПП)". 2011.

УДК 004.272.2

А. Л. Стемпковский,
акад. РАН, д-р техн. наук, директор,
В. М. Амербаев, д-р техн. наук, науч. сотр.,
Р. А. Соловьев, канд. техн. наук, зав. отделом,
e-mails: ippm@ippm.ru, turbo@ippm.ru,
Институт проблем проектирования
в микроэлектронике РАН, г. Москва

Принципы рекурсивных модулярных вычислений

Предложен новый метод, который базируется на идее выразить систему модулей традиционной модулярной арифметики через систему субмодулей, имеющую меньшую размерность. Новое рекурсивное представление данных позволяет устранить часть известных недостатков модулярной арифметики. Несмотря на ограничения, которые накладываются на систему модулей, предложенный метод, как показывают эксперименты, обеспечивает выигрыш по скорости и может быть применен в параллельных высокоскоростных вычислительных устройствах.

Ключевые слова: *модулярная арифметика, параллельные вычисления, система остаточных классов*

1. Введение в рекурсивную модулярную арифметику

В настоящей статье рассматривается развитие некоторых конструктивных идей, которые были изложены одним из соавторов в докладе "Рекурсивная модулярная арифметика" в сентябре 2010 г. на Ученом совете ИППМ РАН и изложены в обобщенной схеме в патенте на полезную модель [1]. В истории вычислительной техники известны случаи, когда при проектировании специализированных устройств не удавалось эффективно обеспечить нужные быстродействие и надежность, используя обычную позиционную (двоичную) арифметику. В то же время использование модулярных вычислений позволяло решить проблему. Модулярная арифметика не является универсальным способом построения вычислителей, но в некоторых специализированных применениях она незаменима. В связи с этим интерес к ней не угасает вот уже многие десятилетия. Результаты исследований, которые позволили бы преодолеть ее недостатки и расширить область ее применения, постоянно публикуются, практически существует целое научное направление, которое занимается этим вопросом [2—5].

Известны преимущества, которые дает использование модулярной арифметики при проектировании вычислителей:

- естественный параллелизм вычислений;
- возможность самоконтроля и исправления неисправностей.

Известны также и ее недостатки:

- большие накладные расходы (наличие преобразователей из позиционного кода в модулярный и обратно);
- представление модульных операций через операции позиционной двоичной арифметики, что приводит к избыточности оборудования при их реализации;
- неравномерность (неоднородность) модульных вычислителей по сложности и времени выполнения операций;
- отсутствие должной поддержки проектирования модулярных вычислителей устройств со стороны САПР (средств структурного синтеза).

Первый недостаток может быть нивелирован, если проектируются достаточно сложные вычислители. Поскольку аппаратные затраты преобразователей ограничены проектными нормами, то увеличивая сложность устройства в целом, можно снизить долю накладных расходов. То же самое можно сказать и о временны́х затратах.

Четвертый недостаток может быть преодолен использованием так называемых IP-генераторов — программных модулей, производящих поведенческое синтезируемое описание на уровне RTL-устройств, выполняющих те или иные модулярные (возможно и не модулярные) процедуры.

Каких-либо существенных способов по преодолению второго и третьего недостатков не известно. На преодоление именно этих недостатков направлен новый подход к проектированию модулярных вычислителей, который назван *рекурсивная модулярная арифметика*.

Идеи, предложенные в настоящей статье, основаны на принципе глубокого распараллеливания модульных операций модулярной арифметики с основаниями p_1, p_2, ..., p_n посредством редуцирования модульных операций по каждому рабочему основанию p_i ($i \leqslant j \leqslant n$) к модульным вычислениям по предшествующим рабочим основаниям p_1, p_2, ..., p_{i-1}, имеющим то или иное технологическое преимущество (например, малобитным), которые

будем называть базисными основаниями. При этом упомянутая редукция допустима только при выполнении так называемого условия согласования вычислительных диапазонов по каждому рабочему модулю p_i с вычислительными диапазонами по соответствующим комплексам базисных оснований. Принцип согласования гарантирует, во-первых, изоморфизм кольцевых операций по соответствующим им комплексам базисных оснований и, во-вторых, выполняемость обращения каждого шага рекурсии посредством перевода соответствующих модулярных кодов по базисным основаниям в позиционный код (например, на основе китайской теоремы об остатках или переводом их в полиадический код).

2. Идея рекурсивной модулярной арифметики

Поясним процедуру рекурсивных преобразований на простом примере. Возьмем в качестве базисных модулей двухбитные простые числа $p_1 = 2$, $p_2 = 3$. Очевидно, что вычетами по модулям 2 и 3 можно однозначно представить любой вычет по модулю 5. В то же время вычетами по модулям 2, 3 и 5, где вычеты по модулю 5 представимы по модулям 2 и 3, можно однозначно представить любой вычет по модулю 29. Вычетами по модулям 2, 3, 5 и 29 можно однозначно представить любой вычет по модулю 863. И т. д., пока не получим нужный набор рабочих оснований: 2, 3, 5, 29, 863, ... Данный пример наглядно иллюстрирует четыре факта:

- аппаратные и временны́е затраты на представление чисел по базовым модулям 2 и 3 примерно одинаковы (оба базисных модуля двухбитные);
- наблюдается более высокая степень распараллеливаемости;
- появляется регулярность (все вычисления по модулям 2 и 3);
- столь малая разрядность базовых модулей позволяет эффективно реализовать модульные операции по базисным модулям в комбинационных схемах.

На самом деле не все так красиво. Реально существует ряд ограничений, которые надо выполнять и которые приводят к усложнению устройств, выполненных по предлагаемой методологии. Рассмотрим эти ограничения.

Пусть имеем систему базисных модулей (p_1, p_2, ..., p_m) и необходимо представить вычеты по модулю p_{m+1} через вычеты по упомянутой системе базовых модулей. Очевидно, что максимальный вычет по модулю p_{m+1} равен $\max = p_{m+1} - 1$. Зная это значение и последовательность выполняемых операций, можно рассчитать максимальное значение MAX результата арифметической операции. Очевидно, что для однозначного представления результата арифметических операций необходимо, чтобы $\mathrm{MAX} < Q$, где $Q = p_1 \cdot p_2 \cdot \ldots \cdot p_m$. Для остальных модулей расчет выполняется аналогично. Вернемся к нашему примеру с базовыми модулями 2 и 3. В этом случае $Q = 2 \cdot 3 = 6$. Наименьшее простое число (после 2 и 3, конечно) есть 5 ($\max = 4$). Мы не можем выполнить ни операцию сложения, так как $2 \cdot \max > Q$ ($8 > 6$), ни, тем более, операцию умножения, поскольку $\max^2 > Q$ ($16 > 6$). Чтобы выполнять любую из арифметических операций (сложение или умножение), нам необходимо увеличить число Q (увеличить значения базисных модулей и/или их число). Возьмем в качестве базисных модулей все взаимно простые трехбитные числа: 4, 5 и 7. В этом случае $Q = 4 \cdot 5 \cdot 7 = 140$. Чтобы имело место $\mathrm{MAX} < Q$ для операции умножения, необходимо выполнение условия $\max^2 < Q$, или $p_{i-1} < \sqrt{Q}$ ($p_{i-1} < 11{,}8$). Таким образом, выбираем $p_i = 11$ (ближайшее к 7 простое число). Далее совокупность рабочих модулей строится без каких-либо проблем с помощью аналогичного расчета, пока не будет достигнут требуемый вычислительный диапазон.

Наконец, рассмотрим реальный случай. Пусть нам нужно реализовать преобразователь Фурье для 24-битных аргументов при числе точек 1024. Для этого потребуется вычислять сумму 1024 произведений, т. е. обеспечить $1024 \cdot \max^2 < Q$. Здесь уже не обойтись системой только трехбитных базисных модулей. Добавим к ним четырехбитные: 5, 7, 8, 9, 11 и 13 ($Q = 360\,360$). Для выбора ближайшего рабочего модуля необходимо $p_i - 1 < \sqrt{\frac{Q}{1024}}$. Получаем $p_i - 1 < 32$. Выбираем $p_i = 31$. Аналогичным расчетом реализуем рекурсивное дерево рабочих оснований целиком. Чтобы сделать такое устройство, нужно спроектировать блок из шести вычислителей (для каждого базисного модуля), и таких блоков будет 16. Вот где работает регулярность. Заметим, что все вычислители будут иметь высокую скорость модульных операций (суперраспараллеливание) и небольшие аппаратные затраты в силу малости базисных модулей или их близости к степеням двойки.

Таким образом, предложенный аппарат рекурсивных модулярных вычислений дает следующие преимущества:

- устранение дисбаланса в операциях с малыми и большими модулями (аппаратные и временные затраты примерно одинаковы, так как в идеале все базисные модули имеют одинаковое число битов);
- существенно более высокая степень распараллеливаемости, а значит, и более высокое быстродействие;
- появляется регулярность (большое число одинаковых базисных модулей);
- малая разрядность базисных модулей позволяет реализовать модульные операции на комбинационных схемах, оптимизированных в базисе булевых функций.

3. Представление данных и основные операции в рекурсивной модулярной арифметике

Идея, на которой базируется рекурсивная модулярная арифметика: выразить систему модулей через систему субмодулей, имеющую меньшую размерность.

Пусть задана система модулей $(p_1, p_2, ..., p_i, ..., p_n)$ и задан некоторый вектор $A = (a_1, a_2, ..., a_n)$. Выразим a_i через систему субмодулей $(p_{i,1}, p_{i,2}, ..., p_{i,k})$, где $P_i = p_{i,1} \cdot p_{i,2} \cdot ... \cdot p_{i,k}$ и $a_i < P_i$: $a_i = (a_{i,1}, a_{i,2}, ..., a_{i,k})$. В этом случае вектор A можно представить в следующем виде: $(a_1, a_2, ..., a_{i-1}, (a_{i,1}, a_{i,2}, ..., a_{i,k}), a_{i+1}, ..., a_n)$ (рис. 1).

Назовем систему из m младших модулей системой базовых модулей, а их произведение обозначим $Q = (p_1 \cdot p_2 \cdot ... \cdot p_m)$. Пусть $p_{i,1} = p_i, p_{i,2} = p_2, ..., p_{i,i-1} = p_{i-1}$, а $k = i - 1$, тогда при $i = m + 1, ..., n$ имеет место рекурсия $a_i = (a_1, a_2, ..., a_m, a_{m+1}, ..., a_{i-1})$ по системе модулей $p_1, p_2, ..., p_m, p_{m+1}, ..., p_{i-1}$ или, раскрывая рекурсию, $a_i = (a_1, a_2, ..., a_m, (a_{m+1,1}, a_{m+1,2}, ..., a_{m+1,m}), ...)$. На рис. 2 приведен частный случай разложения элемента a_i для случая когда $n = 6$ и $m = 3$.

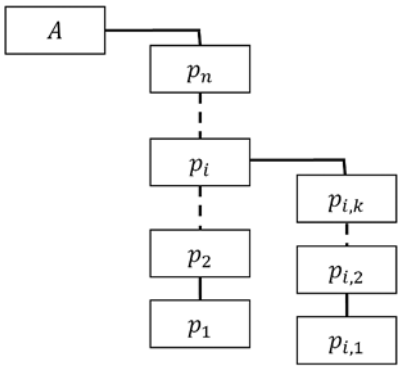


Рис. 1. Иерархия в модулярной арифметике

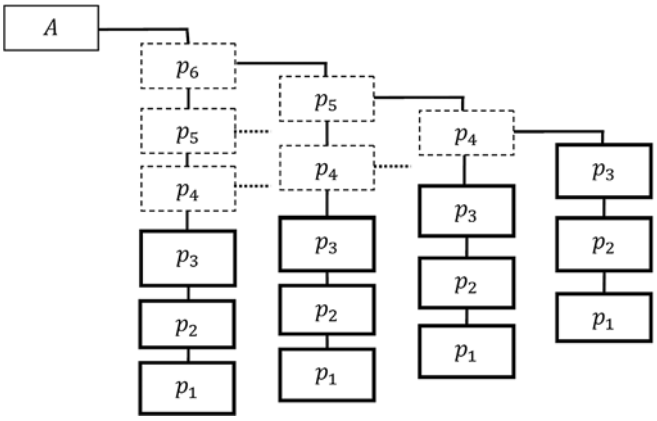


Рис. 2. Рекурсивное разложение элемента p_6 через систему субмодулей (p_1, p_2, p_3)

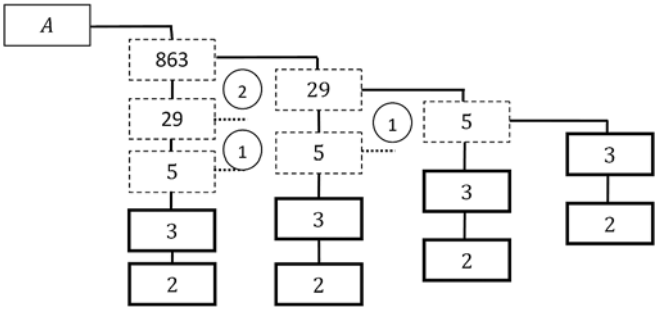


Рис. 3. Разложение числа в системе (2, 3, 5, 29, 863) через систему базовых модулей (2, 3)

3.1. Прямое преобразование числа в позиционной системе счисления в представление в рекурсивной модулярной арифметике

Если в традиционной модулярной арифметике число элементов вектора равно числу элементов в системе модулей, то в рекурсивной модулярной арифметике число элементов вектора увеличивается в зависимости от заданных n и m.

Очевидно, что каждый из первых m элементов представляется в виде одного числа. $(m + 1)$-й элемент содержит m элементов, поскольку выражается через m элементов системы субмодулей: $a_{m+1} = (a_{m+1,1}, a_{m+1,2}, ..., a_{m+1,m})$; $(m + 2)$-й элемент содержит $2m$ элементов, поскольку выражается через m элементов системы субмодулей и m элементов вектора a_{m+1}. Таким образом, продолжая рассуждения, можно заключить, что число элементов L_i для вектора a_i может быть выражено следующей формулой:

$$L_i = \begin{cases} 1, & i \leqslant m; \\ 2^{i-m-1} \cdot m, & m < i \leqslant n. \end{cases} \quad (1)$$

Общее же число элементов L вектора A можно рассчитать, воспользовавшись формулой суммы геометрической прогрессии [6]:

$$L = \sum_{i=1}^{i \leqslant n} L_i = m + m(2^0 + 2^1 + ... + 2^{n-m-1}) =$$

$$= m\left(1 + \frac{2^{n-m} - 1}{2 - 1}\right) = 2^{n-m}m. \quad (2)$$

Рассмотрим численный пример. Пусть задана система модулей: $(p_1, p_2, p_3, p_4, p_5) = (2, 3, 5, 29, 863)$. $P = 2 \cdot 3 \cdot 5 \cdot 29 \cdot 863 = 750\,810$. Также необходимо убедиться, что: $2 \cdot 3 > 5$, $2 \cdot 3 \cdot 5 > 29$, $2 \cdot 3 \cdot 5 \cdot 29 > 863$. Выберем систему базовых модулей (p_1, p_2). В этом случае $Q = p_1 \cdot p_2 = 6$, $n = 5$, $m = 2$. Число элементов вектора $L = 2^3 \cdot 2 = 16$.

Разложим число $A = 865$, заданное в позиционной системе счисления, в вектор по обычному базису:

$$A = (|865|_2, |865|_3, |865|_5, |865|_{29}, |865|_{863}) =$$
$$= (1, 1, 0, 24, 2).$$

Теперь разложим число A по рекурсивному базису:

$$A = (|865|_2, |865|_3, (\|865|_5|_2, \|865|_5|_3),$$
$$(\|865|_{29}|_2, \|865|_{29}|_3, (\|\|865|_{29}|_5|_2, \|\|865|_{29}|_5|_3)),$$
$$(\|865|_{863}|_2, \|865|_{863}|_3, (\|\|865|_{863}|_5|_2, \|\|865|_{863}|_5|_3),$$
$$(\|\|865|_{863}|_{29}|_2, \|\|865|_{863}|_{29}|_3, (\|\|\|865|_{863}|_{29}|_5|_2,$$
$$\|\|\|865|_{863}|_{29}|_5|_3)))) = (1, 1, (0, 0), (0, 0, (0, 1)),$$
$$(0, 2, (0, 2), (0, 2, (0, 2)))).$$

Поскольку все числа в этом векторе не превышают 3, то для хранения вектора потребуется $16 \cdot 2 = 32$ бит вместо 20 бит для хранения числа в позиционной системе счисления. Степень избыточности составит 1.6. Иллюстрацию к примеру см. на рис. 3.

3.2. Ограничения на выбор базиса

Максимальное значение, которое можно представить с помощью p_{m+1}, равно $\max = p_{m+1} - 1$. Чтобы была возможность выполнять арифметические операции над числами, требуется, чтобы результат операции для (p_{m+1})-го элемента был меньше $Q = (p_1 \cdot p_2 \cdot \ldots \cdot p_m)$. Для сложения это будет $2 \cdot \max$, а для умножения — $\max^2$.

Рассмотрим следующий пример. Пусть задан базис (2, 3, 5). Выберем систему базовых модулей как (2, 3). В этом случае $Q = 2 \cdot 3 = 6$, $\max = 4$. Поскольку для сложения потребуется максимально представимое число 8, что больше 6, а для умножения 16, что тоже больше 6, то в таком базисе можно выполнять взаимно однозначное разложение чисел, но для базовых арифметических операций он не подходит. Для реальных задач требуется увеличение числа Q.

Как именно выбирать элементы базиса? Рассмотрим следующий пример. Пусть задана система базовых модулей (4, 5, 7). Требуется определить p_4, чтобы в рамках такого рекурсивного базиса можно было использовать операцию умножения. $Q > \text{MAX} \rightarrow$ $\rightarrow Q > \max^2 \rightarrow Q > (p_4 - 1)^2 \rightarrow p_4 < \sqrt{Q} + 1 \rightarrow$ $\rightarrow p_4 < \sqrt{140} + 1 \rightarrow p_4 < 12{,}8$. Следовательно, для реализации рекурсивного базиса мы можем выбрать $p_4 = 11$.

3.3. Обратное преобразование числа из рекурсивного представления в позиционное

Для обратного представления также требуется рекурсивная реализация на базе того же метода, который используется для преобразования из вектора традиционной модулярной арифметики в позиционную систему счисления.

Пусть задан некоторый вектор $A = (a_1, a_2, \ldots, a_n)$. Из свойств систем остаточных классов известно, что $A = (a_1, a_2, \ldots, a_n) = (a_1, 0, \ldots, 0) + (0, a_2, \ldots, 0) +$ $+ \ldots + (0, 0, \ldots, a_n) = \left| \sum_{i=1}^{n} a_i \cdot B_i \right|_P$, где $B_0 = (1, 0, \ldots, 0)$; $B_1 = (0, 1, \ldots, 0), \ldots$; $B_n = (0, 0, \ldots, 1)$ — система ортогональных базисов [7].

В рекурсивной модулярной арифметике требуется найти набор ортогональных базисов для следующих систем остаточных классов: $(p_1, p_2, \ldots, p_m)$, $(p_1, p_2, \ldots, p_{m+1}), \ldots, (p_1, p_2, \ldots, p_n)$.

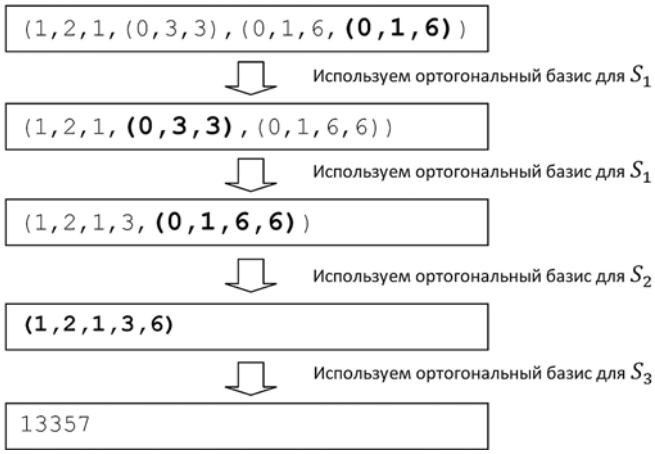


Рис. 4. Процесс обратного преобразования вектора

Рассмотрим пример. Пусть задан базис (3, 5, 7, 11, 13) с системой базовых модулей (3, 5, 7). И пусть требуется преобразовать вектор (1, 2, 1, (0, 3, 3), (0, 1, 6, (0, 1, 6))) в позиционную систему счисления. Для обратного преобразования требуется найти ортогональные базисы для каждой из следующих систем остаточных классов:

$$S_1 = (3, 5, 7) \rightarrow B_{1,1} = (1, 0, 0) \equiv 70;$$
$$B_{1,2} = (0, 1, 0) \equiv 21;\ B_{1,3} = (0, 0, 1) \equiv 15;$$
$$S_2 = (3, 5, 7, 11) \rightarrow B_{2,1} \equiv 385;\ B_{2,2} \equiv 231;$$
$$B_{2,3} \equiv 330;\ B_{2,4} \equiv 210;$$
$$S_3 = (3, 5, 7, 11, 13) \rightarrow B_{3,1} \equiv 5005;\ B_{3,2} \equiv 6006;$$
$$B_{3,3} \equiv 10\ 725;\ B_{3,4} \equiv 1365;\ B_{3,5} \equiv 6930;$$

Процесс обратного преобразования приведен на рис. 4. В позиционной системе счисления искомый вектор равен 13 357.

3.4. Сложение и умножение в рекурсивной модулярной арифметике

Соответственно, если выполнены ограничения, наложенные на базис (см. раздел 3.2), то сложение и умножение чисел выполняется так же, как и в традиционной модулярной арифметике. Чтобы сложить (умножить) два числа, требуется сложить (умножить) соответствующие элементы вектора по модулю p_i. А поскольку все элементы вектора имеют малую разрядность, параллельное сложение (умножение) выполняется очень быстро.

4. Экспериментальные результаты

В рамках эксперимента сравнивалась скорость выполнения скалярного умножения векторов тремя способами: в позиционной системе счисления, в рамках обычного модулярного базиса и в рамках рекурсивного модулярного базиса.

Для построения модели устройств выбран маршрут проектирования цифровых ИС на основе

библиотек стандартных ячеек. В маршруте используется:

- поведенческое описание устройства на языке Verilog HDL;
- средства логического синтеза Synopsys Design Compiler;
- средства статического временного анализа Synopsys Prime Time;
- библиотека стандартных ячеек Nangate Open Cell Library с проектными нормами 45 нм.

В разработанном устройстве для рекурсивной модулярной арифметики прямое преобразование выполняется конвейерным образом, и задержка на прямое преобразование для заданных параметров всегда меньше, чем основное тело скалярного умножения. Обратное преобразование выполняется довольно долго, но из-за того, что на обратное преобразование выделяется число циклов, равное числу элементов вектора, даже в самых сложных случаях обратное преобразование успевает завершиться намного раньше, чем на вход обратному преобразователю поступает новая порция данных.

Таким образом, тактовая частота устройства определяется блоком, имеющим максимальную задержку, а именно — основным телом скалярного умножения.

Пусть векторы состоят из 1024 элементов, а аргументы у векторов 20-битные. Потребуется вычислять сумму 1024 произведений, т. е. обеспечить $1024\max^2 < Q$. Здесь не обойтись системой только трехбитных базовых модулей. Добавим к ним четырехбитные: 5, 7, 8, 9, 11 и 13 (Q = 360 360). Для выбора ближайшего модуля воспользуемся формулами из раздела 4. Получаем $p_7 < 18$. Выбираем $p_7 = 17$. Аналогичным расчетом строим все дерево целиком: (5, 7, 8, 9, 11, 13, 17, 73, 659, 16 963). Чтобы сделать такое устройство, нужно спроектировать блок из шести вычислителей (для каждого базового модуля), и таких блоков нужно 16. Вот где работает регулярность. Заметим, что все вычислители имеют крайне высокое быстродействие (суперраспараллеливание) и небольшие аппаратные затраты в силу малости значений базовых модулей.

Рассмотрим комбинационный участок синхронной схемы скалярного умножения (рис. 5).

Тактовая частота устройства напрямую зависит от длины критического пути на этом участке, которое содержит три операции: умножение (×), сложение (+) и взятие остатка по модулю (%).

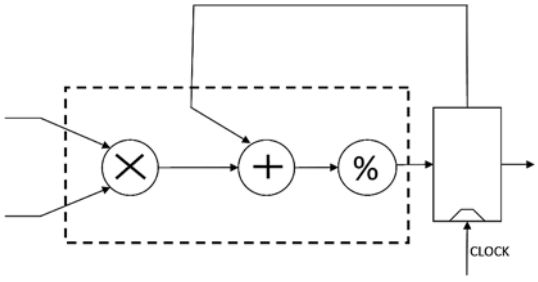


Рис. 5. Комбинационный участок скалярного умножения векторов

Было разработано несколько потоковых устройств для скалярного умножения векторов. Каждое модулярное устройство состоит из трех участков: прямое преобразование из позиционной системы счисления, основной блок скалярного умножения и обратное преобразование из рекурсивного модулярного представления в позиционное. Тактовая частота определяется самым медленным участком схемы. В нашем случае этим участком был блок скалярного умножения.

Таблица 1

Наборы модулей для скалярного умножения в традиционной и рекурсивной модулярной арифметике

Длина вектора	Разрядность данных, бит	Набор модулей	
		Модулярная арифметика	Рекурсивная модулярная арифметика
512	16	7, 11, 13, 16, 17, 19, 23, 27, 29, 31	Базовые: 5, 7, 8, 9, 13, 17 Дополнительные: 31, 181, 709
1024	20	37, 41, 43, 47, 53, 59, 61, 63, 64	Базовые: 5, 7, 8, 9, 11, 13 Дополнительные: 17, 73, 659, 16 963
2048	24	5, 7, 17, 31, 61, 67, 71, 73, 113, 127, 128	Базовые: 13, 17, 19, 23, 29, 31 Дополнительные: 199, 2903, 11 497

Таблица 2

Тактовая частота разработанного блока для разных методов реализации

Длина вектора	Разрядность данных, бит	Тактовая частота, МГц		
		Позиционная система счисления	Модулярная арифметика	Рекурсивная модулярная арифметика
512	16	409	726	855
1024	20	346	581	986
2048	24	294	537	794

Таблица 3

Площадь комбинационной части разработанного блока для разных методов реализации

Длина вектора	Разрядность данных, бит	Площадь комбинационной части блока скалярного умножения		
		Позиционная система счисления	Модулярная арифметика	Рекурсивная модулярная арифметика
512	16	3119	3928	11 443
1024	20	5040	5857	22 745
2048	24	6915	6742	26 038

Оценивались: максимальная тактовая частота и общая площадь устройства для трех методов расчета: обычная реализация в позиционной системе счисления, реализация в традиционной модулярной арифметике и реализация в рекурсивной модулярной арифметике. Результаты расчетов приведены в табл. 1, 2 и 3.

5. Недостатки рекурсивной модулярной арифметики и возможности для их нейтрализации

Основные недостатки предложенного метода заключаются в следующем:

- аппаратная избыточность;
- усложнение прямого и обратного преобразователей из позиционной системы счисления (усложняются также и другие немодульные операции);
- ограничения на выбор базисов;
- ограничение на число последовательных операции без обращения к рекурсии.

Некоторые недостатки рекурсивной модулярной арифметики напрямую следуют из того факта, что маршрут включает в себя операцию нахождения вычета довольно больших чисел. В традиционной модулярной арифметике обычно используются только набор из малых модулей или модулей специального вида. При дальнейшем развитии рекурсивной модулярной арифметики можно использовать в качестве модулей числа Мерсена [8] и/или числа вида $2^n \pm k$, для которых операция взятия остатка от деления на аппаратном уровне потребляет мало аппаратных ресурсов.

Список литературы

1. **Устройство** для вычисления по модулю // Патент на полезную модель № 103010 Российская Федерация, МПК G06F7/72. Заявитель ИППМ РАН. № 2010148522; заявл. 29.11.2010; зарегистрировано 20.03.2011.
2. **Акушский И. Я., Юдицкий Д. И.** Машинная арифметика в остаточных классах. М.: Сов. Радио, 1968. 440 с.
3. **Szabo N. S. and Tanaka R. I.** Residue Number System and its applications to Computer Technology. New York: McGraw-Hill, 1967.
4. **Soderstrand M. A.** et al. (Eds). Residue Number System Arithmetic: Modern Applications in Digital Signal Proccessing // IEEE Press. 1986.
5. **Omondi M. A., Premkumar B.** Residue Number Systems: Theory and Implementation. Imperial College Press. 2007. 296 p.
6. **Выгодский М. Я.** Справочник по элементарной математике. М.: АСТ Астрель, 2006.
7. **Tseng B., Jullien G. A., Miller W. C.** Implementation of FFT structures using the residue number systems // IEEE Transactions on Computers. 1992. 28 (11).
8. URL: http://ru.wikipedia.org/wiki/Числа_Мерсенна

УДК 004.052.3

В. А. Богатырев, д-р техн. наук, проф.,
С. В. Богатырев, аспирант,
А. В. Богатырев, студент,
Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики,
e-mail: vladimir.bogatyrev@gmail.com

Надежность кластерных вычислительных систем с дублированными связями серверов и устройств хранения

Предложена оценка надежности кластеров с прямым подключением устройств хранения к дублированным серверам, в которых каждый сервер имеет два порта для подключения двухвходовых устройств хранения. Показана существенная зависимость надежности и отказоустойчивости рассматриваемых кластеров от порядка подключения устройств хранения к серверам.

Ключевые слова: *отказоустойчивость, кластер, надежность, резервирование, устройство хранения, сервер*

Введение

Высокая надежность, отказоустойчивость и производительность центров обработки и хранения данных достигается при объединении их узлов в кластеры. В кластерных системах консолидация устройств хранения достигается на основе технологии сетей хранения данных SAN (*Storage Area Network*) [1, 2]. Коммутационные узлы сетей хранения, обеспечивающие взаимосвязь узлов, характеризуются высокой стоимостью и сложностью, что может отрицательно влиять на эффективность и надежность кластерной системы в целом. В связи с этим при построении недорогих кластерных систем в ряде случаев целесообразно непосредственное (прямое) подключение устройств хранения к серверным (вычислительным) узлам на основе архитектуры DAS (*Directly Attached Storage*) [1].

Отказоустойчивость кластеров с непосредственной связанностью узлов достигается при резервировании узлов и соединений кластера, причем для гарантированной устойчивости системы, хотя бы к однократным отказам, все узлы и связи как минимум должны дублироваться. Учитывая влияние вариантности объединения узлов в кластер на его надежность и производительность, представляет интерес исследование вариантов дублированного подключе-

ния серверов к устройствам хранения. Цель таких исследований — обоснование выбора вариантов, обеспечивающих при одинаковых затратах на реализацию системы ее большую отказоустойчивость, надежность и доступность.

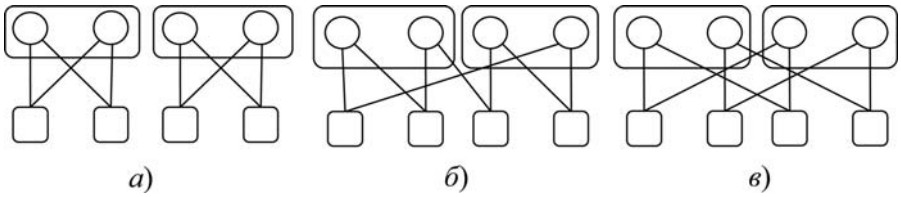


Рис. 1. Варианты кластеров с непосредственным подключением серверов и устройств хранения:
*а — S*1; *б — S*2; *в — S*3

Сложность предполагаемых исследований связана с тем, что в общем случае модели надежности рассматриваемых конфигураций дублированного соединения серверов с устройствами хранения не сводятся к параллельно-последовательным моделям [3, 4] и требуют учета комбинаторного влияния расположения отказавших узлов и их связей [5—9].

Варианты кластеров с непосредственным подключением серверов и устройств хранения

Рассмотрим конфигурации кластерных систем с непосредственным подключением устройств хранения к серверам приложений m типов. Будем считать, что серверы каждого типа дублированы, в этом случае общее число серверов — $2m$, общее число устройств хранения будем также считать равным $2m$. Рассмотрим структуры, в которых серверы объединяются в пары (дублируются по функциональной принадлежности приложений или по объединению обслуживаемого ими потока запросов). Будем считать, что каждый сервер имеет два порта (входа) для подключения устройств хранения, а каждое устройство хранения — два входа для подключения серверов, что дает возможность строить резервированные системы с отсутствием единой точки отказа без использования коммутаторов, которые в ряде случаев могут вносить дополнительную ненадежность в систему.

Предположим, что условие работоспособности кластера заключается в исправности, хотя бы одного сервера каждой из m пар, при доступности для серверов каждой пары (каждого типа приложений), хотя бы одного устройства хранения.

Типовой вариант $S1$ кластера с прямым подключением устройств хранения к дублированным серверам [1] представлен на рис. 1, *а*.

Для рассматриваемых вариантов структур кластеров каждый i-й сервер по первой из двух связей соединен с i-м устройством хранения.

Структуру кластера с учетом второй связи каждого сервера охарактеризуем вектором (l, s, r, m), в котором m — число типов серверов приложений, r — кратность их резервирования (при дублировании $r = 2$), l — число типов серверов приложения, объединяемых в группы, s — смещение при связи сервера и устройства хранения в группе. Узлы относятся к разным группам, если между ними нет ни одной связи. Число групп определяется как m/l.

Рассматриваемые варианты кластеров формируются по следующему правилу: в каждой группе первый сервер подключен к первому входу первого и ко второму входу s-го устройства хранения, i-й сервер подключен к первому входу i-го и ко второму входу $(i + s)$-го устройства хранения; $(l - s + 1)$-й сервер подсоединен к первому входу $(l - s + 1)$-го устройства хранения и ко второму входу первого устройства хранения, второй вход последнего l-го сервера группы подключен ко второму выходу s-го устройства хранения. Конфигурации $S1$, $S2$, $S3$ на рис. 1, *а—в* соответствуют $(l = 1, s = 1, r = 2, m = 2)$, $(l = 2, s = 1, r = 2, m = 2)$ и $(l = 2, s = 2, r = 2, m = 2)$ [10].

Подчеркнем, что все представленные варианты кластеров характеризуются одинаковыми затратами на их построение, но позволяют достичь различной надежности и отказоустойчивости, что и обусловливает актуальность исследования вариантов объединения узлов в кластеры.

Соединения серверов и устройств хранения по второй (резервной) связи отобразим в виде подстановки [11], первой строке (операнду) которой соответствуют серверы (прономерованные от 1 до $2m$), а второй строке — связанные с ними по резервной (второй) связи устройства хранения. Подстановки, соответствующие при $m = 2$ структурам $S1$, $S2$, $S3$, имеют вид:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}.$$

Структуру кластера представим матрицей $\mathbf{S}$, элемент которой $s_{ij} = 1$, если i-й сервер соединен с j-м устройством хранения, иначе $s_{ij} = 0$ [12].

Матрицы $\mathbf{S}$, соответствующие структурам $S1$, $S2$, $S3$, имеют вид:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

При $m = 4$ на основе структур $S1$, $S2$, $S3$ возможно формирование конфигураций, представляемых при $(l = 1, s = 1, r = 2, m = 4)$, $(l = 2, s = 1, r = 2, m = 4)$, $(l = 2, s = 2, r = 2, m = 4)$ подстановками

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 1 & 4 & 3 & 6 & 5 & 8 & 7 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 1 & 6 & 7 & 8 & 5 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 1 & 2 & 7 & 8 & 5 & 6 \end{pmatrix},$$

которым соответствуют матрицы связанности:

$$\begin{bmatrix}1&1&0&0&0&0&0&0\\1&1&0&0&0&0&0&0\\0&0&1&1&0&0&0&0\\0&0&1&1&0&0&0&0\\0&0&0&0&1&1&0&0\\0&0&0&0&1&1&0&0\\0&0&0&0&0&0&1&1\\0&0&0&0&0&0&1&1\end{bmatrix}, \begin{bmatrix}1&1&0&0&0&0&0&0\\0&1&1&0&0&0&0&0\\0&0&1&1&0&0&0&0\\1&0&0&1&0&0&0&0\\0&0&0&0&1&1&0&0\\0&0&0&0&0&1&1&0\\0&0&0&0&0&0&1&1\\0&0&0&0&1&0&1&1\end{bmatrix}, \begin{bmatrix}1&0&1&0&0&0&0&0\\0&1&0&1&0&0&0&0\\1&0&1&0&0&0&0&0\\0&1&0&1&0&0&0&0\\0&0&0&0&1&0&1&0\\0&0&0&0&0&1&0&1\\0&0&0&0&1&0&1&0\\0&0&0&0&0&1&0&1\end{bmatrix}.$$

Структуры ($l = 4$, $s = 1$, $r = 2$, $m = 4$), ($l = 4$, $s = 2$, $r = 2$, $m = 4$) представимы следующими подстановками и соответствующими им матрицами связанности:

$$\begin{pmatrix}1&2&3&4&5&6&7&8\\2&3&4&5&6&7&8&1\end{pmatrix}, \begin{pmatrix}1&2&3&4&5&6&7&8\\3&4&5&6&7&8&1&2\end{pmatrix},$$

$$\begin{bmatrix}1&1&0&0&0&0&0&0\\0&1&1&0&0&0&0&0\\0&0&1&1&0&0&0&0\\1&0&0&1&0&0&0&0\\0&0&0&0&1&1&0&0\\0&0&0&0&0&1&1&0\\0&0&0&0&0&0&1&1\\0&0&0&0&1&0&0&1\end{bmatrix}, \begin{bmatrix}1&0&1&0&0&0&0&0\\0&1&0&1&0&0&0&0\\0&0&1&0&1&0&0&0\\0&0&0&1&0&1&0&0\\0&0&0&0&1&0&1&0\\0&0&0&0&0&1&0&1\\1&0&0&0&0&0&1&0\\0&1&0&0&0&0&0&1\end{bmatrix}.$$

Пусть для функционирования системы требуется исправность хотя бы одного сервера каждой пары (выделенной под соответствующее приложение) при условии доступности для него хотя бы одного устройства хранения.

Работоспособность конфигураций $S1$, $S2$ и $S3$ теряется, если отказывает пара дублированных серверов хотя бы одного типа. Помимо этого система отказывает при отказе для $S1$ двух, для $S2$ трех, а для $S3$ четырех связанных с каждой парой серверов (однотипных приложений) устройств хранения. Таким образом, число отказов устройств хранения, гарантированно выдерживаемых конфигурациями $S1$, $S2$ и $S3$, равно соответственно двум, трем и четырем. Следовательно, по отказоустойчивости доминирует конфигурация $S3$, а наихудшей является конфигурация $S1$.

Надежность кластерных конфигураций $S1$ без учета отказов связей. Оценим надежность рассматриваемых вариантов построения кластерных систем с непосредственным подключением серверов и устройств хранения.

Для конфигураций $S1$ (см. рис. 1, *а*) при требовании исправности хотя бы одного сервера и одного устройства хранения в каждой из m дублированных групп вероятность безотказной работы без учета надежности связей оценивается как

$$P(t) = \{[1 - (1 - p_1(t)^2][1 - (1 - p_2(t))^2]\}^m,$$
$$p_1(t) = \exp(-\lambda_1 t),\ p_1(t) = \exp(-\lambda_1 t),$$

где λ_1, λ_2 — интенсивность отказов сервера и устройства хранения.

Надежность кластерных конфигураций $S1$ с учетом отказов связей. С учетом ненадежности соединений имеем

$$P(t) = \{p_1(t)^2 p_2(t)^2[1 - (1 - p_3(t))^4] +$$
$$+ [2p_1(t)(1 - p_1(t))p_2(t)^2 + 2p_2(t)(1 - p_2(t))p_1(t)^2] \times$$
$$\times [1 - (1 - p_3(t))^2] + 4p_1(t)(1 - p_1(t)) \times$$
$$\times p_2(t)(1 - p_2(t))p_3(t)\}^m,$$

где $p_3(t) = \exp(-\lambda_3 t)$, λ_3 — интенсивность отказов связей.

Надежность кластерных конфигураций $S3$ без учета надежности отказов связей. Для кластеров на основе конфигурации $S3$ (см. рис. 1 *в*), вероятность безотказной работы группы определим как

$$P(t) = \sum_{i=2}^{4} P_i(t),$$

где $P_i(t)$ — вероятность работоспособных состояний с исправностью i из четырех серверов в группе ($i = 2, 3, 4$), при $i = 1$ состояние неработоспособно, так как для каждой пары должен быть исправен хотя бы один сервер.

При исправности четырех серверов кластер работоспособен при исправности хотя бы одного из четырех устройств хранения, поэтому

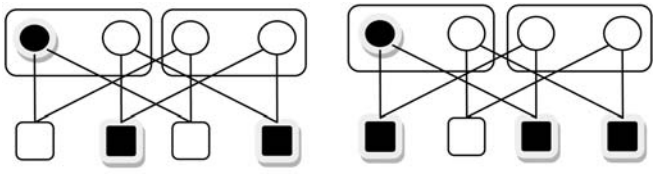
$$P_4(t) = p_1(t)^4[1 - (1 - p_2(t))^4].$$

При исправности трех серверов группа работоспособна в случае исправности:

— четырех или любых трех устройств хранения;

— двух устройства хранения, кроме случая, представленного на рис. 2, *а*, когда оба исправных устройства хранения подключены к отказавшему серверу (число таких работоспособных комбинаций — пять);

— одного устройства хранения, если оно не подключено к отказавшему серверу (рис. 2, *б*), таких комбинаций — две.



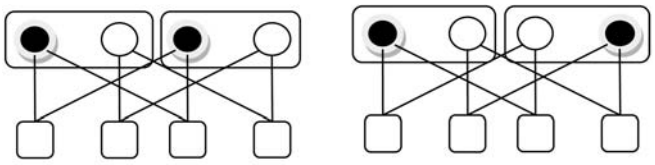
а) *б)*

Рис. 2. Варианты отказа трех серверов (отказавшие серверы затемнены)

Таким образом, при исправности трех из четырех серверов вероятность работоспособности системы будет

$$P_3(t) = 4p_1(t)^3(1 - p_1(t))[p_2(t)^4 + 4p_2(t)^3(1 - p_2(t)) + 5p_2(t)^2(1 - p_2(t))^2 + 2p_2(t)(1 - p_2(t))^3].$$

При исправности двух серверов в группе необходимым условием ее работоспособности является исправность двух серверов разной функциональности (принадлежащих разным парам серверов). Если исправны два сервера разной функциональности и они по всем связям подключены только к двум устройствам хранения (рис. 3, *а*), то кластер работоспособен при исправности хотя бы одного из этих устройств хранения.



а) *б*)

Рис. 3. Варианты отказа двух серверов (отказавшие серверы затемнены)

Если исправны два функционально разных сервера, вместе подключенные ко всем устройствам хранения (рис. 3, *б*), то кластер работоспособен при исправности хотя бы одного устройства хранения, подключенного к каждому первому исправному серверу.

Таким образом, при исправности двух из четырех серверов вероятность работоспособности системы без учета надежности связей определяется как

$$P_2(t) = 2p_1(t)^2(1 - p_1(t))^2[1 - (1 - p_2(t))^2] + 2p_1(t)^2(1 - p_1(t))^2[1 - (1 - p_2(t))^2]^2 = 2p_1(t)^2(1 - p_1(t))^2[1 - (1 - p_2(t))^2] \times (1 + [1 - (1 - p_2(t))^2]).$$

Надежность кластерных конфигураций *S*3 с учетом отказов связей. Определим вероятность безотказной работы конфигурации *S*3 по рис. 1, *в* с учетом надежности соединений. Расчет проведем, выделяя состояния с исправностью i из четырех серверов ($i = 4, 3, 2$) с учетом того, что для каждой пары серверов, хотя бы один исправный сервер должен быть подключен хотя бы к одному устройству хранения (при $i = 1$ кластер не работоспособен).

При исправности всех четырех серверов в случае отказа одного устройства хранения теряется одна из связей с серверами каждой пары одновременно. Таким образом, при исправности $i = 4, 3, 2, 1$ устройств хранения кластер работоспособен при исправности хотя бы одной из i связей первой и второй пары серверов, т. е.

$$P_4(t) = p_1(t)^4[p_2(t)^4[1 - (1 - p_3(t))^4]^2 + 4p_2(t)^3 \times (1 - p_2(t))[1 - (1 - p_3(t))^3]^2 + 6p_2(t)^2(1 - p_2(t))^2 \times [1 - (1 - p_3(t))^2]^2 + 4p_2(t)(1 - p_2(t))^3p_3(t)^2].$$

При исправности трех из четырех серверов вероятность безотказной работы конфигурации *S*3

$$P_3(t) = 4p_1(t)^3(1 - p_1(t)) \sum_{i=1}^{4} P_{3i}(t),$$

где $P_{3i}(t)$ — вероятность работоспособности кластера при исправности трех серверов и i устройств хранения ($i = 4, 3, ..., 1$).

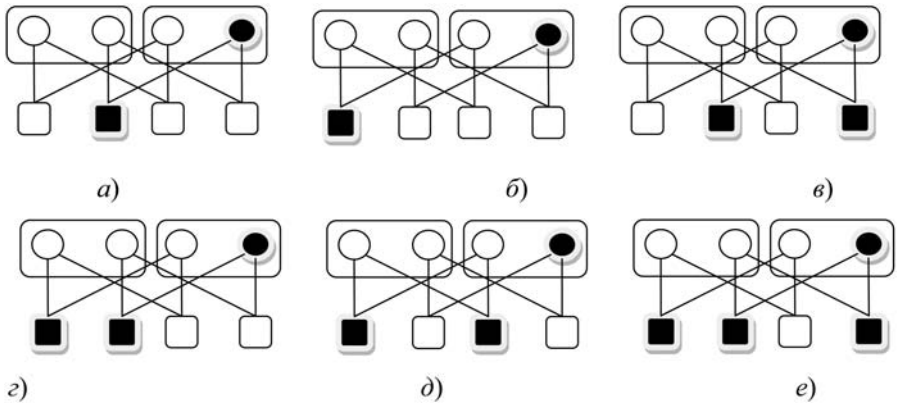
Учитывая инвариантность надежности к месту расположения единственного отказавшего сервера, будем считать, что он принадлежит второй паре.

При исправности всех четырех устройств хранения и трех из четырех серверов связанность хотя бы одного из двух исправных серверов первой пары с устройствами хранения обеспечивается при исправности хотя бы одной из четырех связей, а для исправного сервера второй пары связанность поддерживается при исправности хотя бы одной из двух его связей.

Таким образом, при исправности трех серверов вероятность работоспособности кластера равна

$$P_{34}(t) = [p_2(t)^4[1 - (1 - p_3(t))^4][1 - (1 - p_3(t))^2].$$

При исправности трех из четырех устройств хранения и трех из четырех серверов выделим случаи, когда отказывает устройство хранения либо связанное (рис. 4, *а*), либо не связанное (рис. 4, *б*) с отказавшим сервером. В обоих случаях (рис 4, *а*, *б*) связанность хотя бы одного сервера первой пары хотя бы с одним устройством хранения обеспечивается при исправности хотя бы одной из трех связей.



а) *б*) *в*)

г) *д*) *е*)

Рис. 4. Состояния кластера при исправности трех серверов в случае работоспособности одного (*а*, *б*), двух (*в*, *г*, *д*) и трех устройств хранения (*е*)

Связанность единственного исправного сервера второй пары хотя бы с одним устройством хранения для структуры, показанной на рис. 4, *а* и *б*, реализуема, соответственно, при сохранении:

— хотя бы одной из двух его связей с исправными устройствами хранения;

— единственной его связи с работоспособным устройством хранения.

Таким образом, вероятность безотказности структуры $S3$ при исправности трех серверов и трех из четырех устройств хранения будет

$$P_{33}(t) = 2p_2(t)^3(1 - p_2(t))[1 - (1 - p_3(t))^3] \times$$
$$\times [1 - (1 - p_3(t))^2] + 2p_2(t)^3(1 - p_2(t)) \times$$
$$\times [1 - (1 - p_3(t))^3]p_3(t) = 2p_2(t)^3(1 - p_2(t)) \times$$
$$\times [1 - (1 - p_3(t))^3][[1 - (1 - p_3(t))^2] + p_3(t)].$$

При исправности двух из четырех устройств хранения и условии исправности трех из четырех серверов выделим работоспособные состояния, при которых к отказавшему серверу подключены:

— оба отказавших устройства хранения (рис. 4, *в*);

— одно из двух отказавших устройств хранения (рис. 4, *г*).

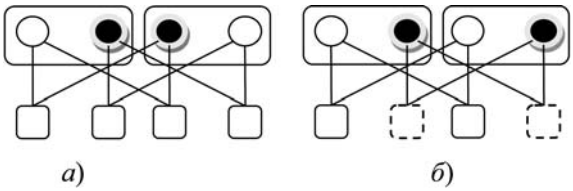
Состояние, для которого (рис. 4, *д*) к единственному исправному серверу второй пары подключена пара неисправных устройств хранения, неработоспособно. Из $C_4^2 = 6$ возможных комбинаций отказа двух из четырех устройств хранения, одна соответствует рис. 4, *в*, а четыре — рис. 4, *г*. Одна комбинация с отказом двух устройств хранения, соответствующая рис. 2, *а*, неработоспособна.

Для структуры, показанной на рис. 4, *в*, имеется по одному исправному серверу каждой пары, подключенному к общей паре исправных устройств хранения. Связанность каждого из двух работоспособных серверов обеспечивается при исправности хотя бы одной из двух его связей с функционирующими устройствами хранения, а вероятность связанности обоих работоспособных серверов равна $[1 - (1 - p_3(t))^2]^2$.

Для структуры, показанной на рис. 4, *г*, для первой пары серверов связанность хотя бы с одним из исправных устройств хранения сохраняется при работоспособности хотя бы одной из двух связей, а для исправного сервера второй пары — при исправности единственной связи с исправным устройством хранения. Таким образом, вероятность связанности обеих пар серверов равна $4[1 - (1 - p_3(t))^2]p_3(t)$.

При исправности трех серверов и трех устройств хранения вероятность безотказной работы кластера вычисляется как

$$P_{32}(t) = p_2(t)^2(1 - p_2(t))^2[[1 - (1 - p_3(t))^2]^2 +$$
$$+ 4[1 - (1 - p_3(t))^2]p_3(t)] = p_2(t)^2(1 - p_2(t))^2 \times$$
$$\times [1 - (1 - p_3(t))^2]\{[1 - (1 - p_3(t))^2] + 4p_3(t)\}.$$



а) *б)*

Рис. 5. Работоспособные варианты при исправности двух из четырех серверов кластерной группы

При исправности одного из четырех устройств хранения и трех из четырех серверов кластер работоспособен, когда исправное устройство хранения подключено к паре исправных серверов (рис. 4, *е*), таким образом $P_{31}(t) = 2p_2(t)(1 - p_2(t))^3p_3(t)^2$.

Исправность двух из четырех серверов. Для работоспособных состояний в каждой паре резервированных серверов должен быть один исправный сервер. Из общего числа $C_4^2 = 6$ комбинаций с отказом двух серверов имеются по две работоспособные комбинации, представляемые рис. 5, *а* и *б*. Для двух работоспособных состояний (рис 5, *а*) требуется, чтобы каждый сервер был подключен по исправной связи хотя бы к одному не отказавшему устройству хранения. Для двух работоспособных состояний (рис. 5, *б*) два устройства хранения, подключенные к паре неработоспособных серверов, не доступны для вычислительного процесса (отмечены штриховой линией). При исправности двух устройств хранения, подключенных к исправным серверам, для каждого из них (рис. 5, *б*) требуется исправность хотя бы одной связи с устройствами хранения. При исправности одного из устройства хранения необходима его связанность с обоими исправными серверами. Таким образом, при отказе двух серверов кластер работоспособен с вероятностью

$$P_2(t) = 2p_1(t)^2(1 - p_1(t))^2\{[1 - (1 - p_2(t)p_3(t))^2] +$$
$$+ p_2(t)^2[1 - (1 - p_3(t))^2]^2 + 2p_2(t)(1 - p_2(t))p_3(t)^2\}.$$

Примеры оценки надежности

Результаты расчета вероятностей безотказной работы $P(t)$ конфигураций $S1$ и $S3$ представлены на рис. 6, *а*, *б* для времени работы системы соответственно до 500 ч и до 100 ч. Расчеты проведены при $p_1(t) = \exp(-\lambda_1 t)$, $p_2(t) = \exp(-\lambda_2 t)$, $p_3(t) = \exp(-\lambda_3 t)$ и $\lambda_1 = 10^{-3}$ ч$^{-1}$, $\lambda_2 = 2 \cdot 10^{-3}$ ч$^{-1}$, $\lambda_1 = 0{,}5 \cdot 10^{-3}$ ч$^{-1}$.

На рис. 6 кривые *1* и *2* соответствуют оценке $P(k)$ конфигураций $S3$ без учета и с учетом ненадежности связей, кривые *3* и *4* представляют указанные зависимости для конфигурации $S1$. Кривые *5* представляют разницу надежности конфигураций $S3$ и $S1$ при ее расчете с учетом надежности связей. Кривые *6*, *7* отражают разницу вероятностей безотказной работы, рассчитываемых с учетом и без учета ненадежности связей соответственно для конфигураций $S3$ и $S1$.

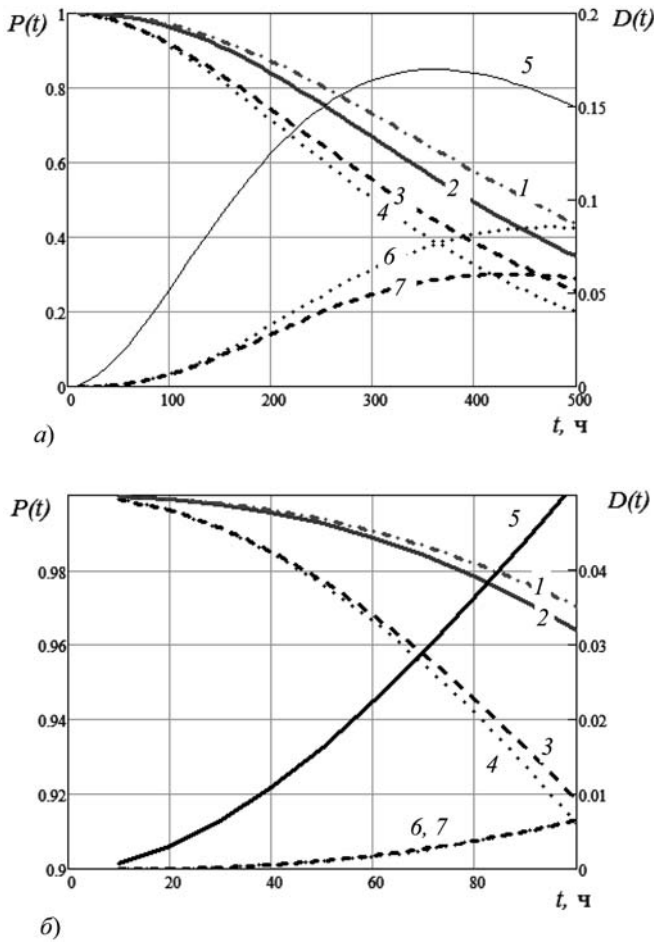


Рис. 6. Вероятность безотказной работы конфигурации *S*1 и *S*3

Проведенные расчеты подтверждают существенную зависимость надежности кластеров с прямым подключением устройств хранения к серверам приложений от порядка этого подключения при явном преимуществе конфигураций *S*3. Расчеты показывают необходимость учета влияния связей на надежность кластера.

Заключение

Предложена оценка надежности кластеров с прямым подключением устройств хранения к дублированным серверам, имеющих два порта (входа) для подключения двухвходовых устройств хранения.

Предложенная оценка надежности учитывает комбинаторное влияние расположения отказавших узлов и их связей на надежность кластера.

Показана существенная зависимость надежности и отказоустойчивости рассматриваемых кластеров от вариантов подключения устройств хранения к серверам, для которых при одинаковых затратах на построение кластера удается достичь различного уровня его надежности.

Показано преимущество по надежности конфигураций с максимальным числом устройств хранения, подключаемых к каждой паре дублированных серверов.

Полученные результаты могут быть использованы при выборе высоконадежных отказоустойчивых конфигураций кластеров с прямым подключением устройств хранения к резервированным серверам.

Список литературы

1. **Juud J.** Principles of SAN Design. San Jose: Brocade Bookshelf, 2008. 589 p.
2. **Clark T.** The New Data Center. New technologies are radically reshaping the data center. San Jose: Brocade Bookshelf. 2010. 156 p.
3. **Shooman M.** Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design. John Wiley & Sons, Inc. 2002. 527 c.
4. **Половко А. М., Гуров С. В.** Основы теории надежности. СПб.: БХВ, 2006. 704 с.
5. **Богатырев В. А.** Мультипроцессорные системы с динамическим перераспределением запросов через общую магистраль // Известия ВУЗов. Приборостроение. 1985. № 3. С. 33—37.
6. **Богатырев В. А.** Надежность вариантов размещения функциональных ресурсов в однородных вычислительных сетях // Электронное моделирование. 1997. № 3. С. 21—25.
7. **Богатырев В. А.** Оптимальное резервирование системы разнородных серверов // Приборы и системы. Управление, контроль, диагностика. 2007. № 12. С. 30—36.
8. **Богатырев В. А.** К анализу сохранения эффективности вычислительных систем с функциональной деградацией модулей // Приборы и системы. Управление, контроль, диагностика. 2000. № 12. С. 68—70.
9. **Богатырев В. А.** Отказоустойчивость вычислительных систем с функциональной реконфигурацией // Приборы и системы. Управление, контроль, диагностика. 2001. № 11. С. 51—53.
10. **Bogatyrev V. A.** Fault Tolerance of Clusters Configurations with Direct Connection of Storage Devices // Automatic Control and Computer Sciences. 2011. Vol. 45, No 6. P. 330—337.
11. **Кофман А.** Введение в прикладную комбинаторику. М.: Наука, 1975. 479 с.
12. **Богатырев В. А.** К размещению резервированных функциональных ресурсов в системах с функциональной реконфигурацией // Управляющие системы и машины. 2003. № 3. С. 42—45.

УДК 004.421.3; 004.272

Р. И. Морылев, аспирант,
e-mail: frg10@yandex.ru,
В. Н. Шаповалов, аспирант,
e-mail: Vasiliy.Shapovalov@gmail.com,
Б. Я. Штейнберг, д-р техн. наук, зав. каф.,
e-mail: borsteinb@mail.ru,
Южный федеральный университет,
г. Ростов-на-Дону

Символьный анализ в диалоговом распараллеливании программ

Описывается диалоговый режим оптимизации и распараллеливания программ в распараллеливающей системе. Вопросы пользователю направлены на уточнение информационных зависимостей, которые определяют возможность применения оптимизирующих или распараллеливающихся программ. Для формирования вопросов пользователю используется символьный анализ. Обсуждаются некоторые границы возможностей автоматического распараллеливания программ, которые преодолеваются с помощью диалога.

Ключевые слова: *диалоговое распараллеливание, символьный анализ, зависимости по данным*

Введение

Актуальность данной работы вызвана растущим многообразием вычислительных архитектур, которое ведет к обострению проблемы переносимости эффективного программного обеспечения. В частности, все важнее становится вопрос распараллеливания — переноса последовательной программы на параллельную архитектуру. Этот частный случай является ключом к созданию переносимых программ (на уровне исходного кода): переносимая программа должна допускать автоматический или полуавтоматический анализ и преобразования компилирующей системой. Вместе с тем компилирующие системы должны стремиться к расширению класса оптимизируемых программ.

До недавнего времени, когда процессоры были последовательными, проблема переносимости решалась поддержкой системы команд x86, в которую отображались высокоуровневые программы, и из которой генерировался код на архитектуры процессоров. Но x86 сложно переносить на различные виды параллелизма и коммуникаций [1, 2]. То же самое относится и к внутренним представлениям, близким к этой системе команд. Усложнение архитектур, в частности бурное развитие параллельных архитектур, приводит к необходимости поднять уровень промежуточного представления компиляторов. Поднятие уровня внутреннего представления, необходимое для переносимости, влечет увеличение нагрузки на блок оптимизации компиляторов.

Возможности автоматического распараллеливания имеют некоторые теоретически непреодолимые границы. Главное средство автоматизации преобразования программ — автоматический анализ зависимостей по данным. В классических языках (Фортран, Си и т. п.) автоматическому анализу может мешать отсутствие информации о диапазонах данных. В статье приводится пример кода, который программист может распараллелить вручную, поскольку знает, что элементы его матрицы неотрицательны, а компилятор не может распараллелить, поскольку об элементах матрицы знает только то, что они целого типа. От вставки автоматической динамической проверки диапазонов входных данных программа может потерять эффективность. Если ввести в язык обязательное описание диапазонов данных, то программисту придется выполнять много ненужной работы — ведь далеко не всегда знание диапазона данных переменной может помочь оптимизации. Нам видится разрешение проблемы в диалоговой компиляции: система просит уточнить диапазоны только для тех переменных, зависимость по которым влияет на оптимизацию программы. Вопросы формируются на основании символьного анализа.

Предлагаемый в данной работе подход обещает расширить множество распараллеливаемых программ. Но, разумеется, полностью проблема эффективной переносимости программ решена не будет, поскольку во многих случаях при переходе к новой архитектуре необходима непосильная компиляторам смена алгоритма решения задачи. Следует отметить, что подобная ситуация наблюдалась и ранее: например, появление кэш-памяти привело к появлению блочных алгоритмов [3].

В данной статье описывается диалоговое уточнение информационных зависимостей между вхождениями переменных программы в оптимизирующей распараллеливающей системе [4]. Ранее такое уточнение выполнялось для вхождений, лежащих в одном линейном участке программы [5]. В данной работе рассматриваемые вхождения могут принадлежать разным веткам условных операторов. Для каждого вхождения при символьном анализе вычисляется предикат, который определяет

условие обращения к этому предикату программы. На основе этих предикатов формируется вопрос пользователю системы.

Символьный анализ программ в компиляторе

Значения большей части выражений в тексте программы на этапе компиляции неизвестны. Для статического анализа иногда используется построение соответствия между значениями выражений в тексте программы и абстрактными выражениями специальной математической модели. Такой анализ называют символьным [6].

Пусть компилятор в целях уточнения наличия информационной зависимости должен установить отношения между выражениями языка *C* "`x*x + y*y`" и "`(x + y)*(x – y)`", где `x`, `y` — целочисленные переменные. Символьный анализ ставит этим выражениям в соответствие пару абстрактных выражений "`xa^2 + ya^2`" и "`(xb + yb)(xb – yb)`", где `x`, `y` принадлежат *Z*, и `a`, `b` — контекст употребления переменной. Если окажется, что контексты `a` и `b` совпадают (переменные `x` и `y` в первом выражении имеют то же самое значение, что и во втором), то с помощью анализа будут перемножены суммы в скобках, установлено равенство двух абстрактных выражений. С учетом определенных допущений это будет означать равенство исходных выражений языка *C*.

С помощью символьного анализа, предлагаемого в работе [6], предназначенного для распараллеливания и можно решать следующие задачи: символьное протягивание констант; обобщенная подстановка индуктивных переменных; подстановка вперед (в том числе и межпроцедурная); определение инвариантов цикла и статическая профилировка.

В работе [7] вводится алгебра обращений к элементам массивов и соответствующий символьный анализ. В дальнейшем в работе [8] был использован этот анализ для построения SSA-формы для массивов и последующего распараллеливания.

Группа, работающая над прототипом компилятора PROMIS [9], использовала символьный анализ для удаления мертвого кода с целью оптимизировать трансляцию байт-кода JAVA в машинный.

Символьный анализ, представленный в данной работе, предназначен для формирования вопросов пользователю в процессе работы диалогового распараллеливателя.

Применение символьного анализа для анализа информационных зависимостей

Говорят, что между вхождениями переменной в программе существует информационная зависимость, если выполняются следующие условия [10]:

1) оба вхождения обращаются к одной и той же ячейке памяти;

2) существует исполнимый путь на графе потока управления [10] между данными вхождениями.

В нашей работе символьный анализ применен для того, чтобы показать, что между вхождениями невозможна передача управления, и следовательно, информационная зависимость между ними невозможна.

Назовем предикатом путей из вершины *A* в вершину *B* на графе потока управления набор условий, которые должны быть выполнены, чтобы управление перешло из *A* в *B*. Для того чтобы существовал исполнимый путь на графе потока управления между *A* и *B*, необходимо, чтобы поток управления прошел сначала через *A* и после этого попал в *B*. То есть необходимо, чтобы выполнился предикат путей от истока графа потока управления (т. е. от начала программы или подпрограммы) до *A* и предикат путей из *A* в *B*. Символьный анализ позволяет для пары вхождений (*A*, *B*) найти такие предикаты и составить условие, при котором возможна передача управления от *A* к *B*. Кроме того, символьный анализ часто позволяет упростить полученное выражение и определить, является ли оно тождеством или нет.

Пример 1.

```
for(i = 0; i < 10; i ++)
    X[i] = X[i – 1];
```

Здесь условие существования зависимости между вхождениями массива X является тождественно истинным.

Конец примера.

Пример 2.

```
for(i = 15; i < 20; i++)
   X[i] = X[i – N];
```

Здесь зависимость между вхождениями массива X является условной, так как существует, только если N принимает значения из интервала (−5, 5).

Конец примера.

Пример 3.

```
for(i = 0; i < 10; i++)
   if (x*x*x + y*y*y == z*z*z)
      X[i] = X[i – 1];
```

где `x`, `y`, `z` — целые числа. Согласно теореме Ферма [11], предикат в условном операторе является тождественно ложным. Однако доказательство теоремы Ферма выходит далеко за пределы возможностей любого современного автоматического анализа. Поэтому на сегодняшний день автоматический анализ не сможет определить тождественную ложность данного условия.

Конец примера.

Пример 4.

```
for(i = 1; i < n; i ++)
   for(j = 0; j < n; j++)
   {
      if(A[i][j] = = 0)
         A[i] [j] = A[i – 1] [j];
         }
```

Пусть требуется распараллелить цикл по `i`. Для этого необходимо доказать, что итерации этого цикла можно выполнять независимо. То есть, что циклически порожденной зависимости между вхождениями массива *A* на входных данных не существует [10]. Это верно, когда во входном массиве *A* нет нулевых элементов.

Конец примера.

Символьный анализ в диалоге

Математический аппарат существующих алгоритмов символьного анализа способен установить лишь очень простые свойства программы. Кроме того, значения многих переменных неизвестны на этапе компиляции, так как они считываются из файлов. Поэтому в большинстве случаев условия существования зависимости классифицируются как не являющиеся тождественными. В диалоговом режиме, однако, есть возможность задать вопрос программисту, что позволяет иногда уточнить результаты автоматического анализа и избавиться от некоторых зависимостей.

Например, для уточнения зависимостей фрагмента из примера 2 можно задать пользователю вопрос: "Правда ли что в данном фрагменте всегда выполняется $-5 < N < 5$?". Для примера 3 вопрос будет звучать так: "Правда ли что в данном фрагменте всегда выполняется `x*x*x + y*y*y == z*z*z` для всех допустимых значений переменных?".

Диалоговый анализ зависимостей должен задавать пользователю как можно меньше вопросов, и задавать их в насколько возможно удобной для пользователя форме. Из этого следует несколько важных принципов работы диалогового распараллеливателя. Во-первых, перед тем как спрашивать пользователя о чем-то следует сначала попробовать провести автоматический анализ. Во-вторых, задавать вопрос о существовании зависимости нужно только в том случае, когда ответ может способствовать оптимизации. Может быть так, что в цикле кроме неясной зависимости есть и другие препятствующие распараллеливанию зависимости, которые символьный анализ удалить не может. Тогда удаление условной зависимости никак не поможет распараллелить цикл, а значит, и нет смысла задавать вопрос о ней. В-третьих, вопрос задавать надо так, чтобы пользователь мог понять и ответить.

Реализация символьного анализа в ДВОР

Описанный анализ реализован в рамках проекта "Диалоговый высокоуровневый оптимизирующий распараллеливатель" (ДВОР). Рассмотрим работу символьного анализатора ДВОР на примере алгоритма Флойда—Уоршелла, ядро которого приведено в следующем примере.

Пример 5. Фрагмент реализации алгоритма Флойда—Уоршелла:

```
for (int k = 0; k < n; k++)
  for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
      if (d[i][j] > d[i][k] + d[k][j])
        d[i][j] = d[i][k] + d[k][j];
```

Конец примера.

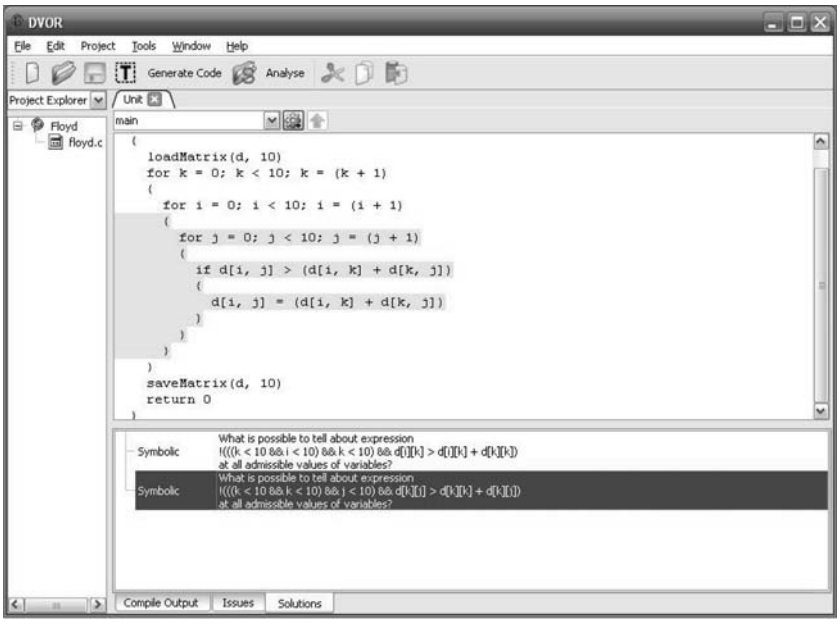
Возможности параллельного выполнения этого алгоритма описаны в работах [12], [13]. Эти параллельные реализации корректны только при неотрицательной входной матрице смежностей (это будет показано дальнейшим анализом).

Система ДВОР во время анализа автоматически определит, что для распараллеливания одного из циклов данного гнезда необходимо предоставление пользователем дополнительной информации.

Для выполнения анализа необходимо разобрать программу во внутреннее представление ДВОР, а затем нажать кнопку "Analyse". При первоначальном анализе на вкладке "Issues" можно увидеть, что ни один из циклов данного гнезда не распараллеливается вследствие наличия зависимостей.

Открыв окно "Solutions" (см. рисунок), можно увидеть, что в результате анализа были сформированы два вопроса к пользователю. Рассмотрим для примера первый из них:

"What is possible to tell about expression!
`(((k < 10 && k < 10) && j < 10) &&`
`d[k] [j] > d[k] [k] + d[k] [j])`
at all admissible values of variables?".



Экранная форма ДВОР со списком вопросов пользователю, сгенерированных при анализе программы, реализующей алгоритм Флойда

Как видно, вопрос содержит предикат, состоящий из двух частей. Первая часть — условие для счетчиков циклов, вторая — ограничения на элементы массива d. Из заголовков циклов видно, что условие для счетчиков циклов всегда истинно внутри гнезда. Приведя подобные слагаемые в ограничениях на элементы массива, получаем

`"0 > d[k] [k]".`

Если массив d — это матрица весов графа, о которых известно, что они неотрицательны, то данный предикат всегда ложен. Отметим, что данная информация известна программисту, но в коде программы в явном виде не содержится, и поэтому без участия пользователя компилятор не может определить его истинность. Пользователь вводит отрицательный ответ на вопрос системы.

Необходимо отметить, что предикат, содержащийся в вопросе, вычисляется автоматически, поэтому он не всегда имеет удобочитаемый вид. Но реализовав серию упрощающих преобразований, можно привести его к виду `"d[k][k]>= 0"`.

После ответа на первый вопрос необходимо повторно проанализировать программу, чтобы новое знание было учтено. Нажав кнопку "Analyse", в окне "Solutions", получим новый вопрос: "The loop can be executed in parallel. Do you want to parallelize it with OpenMP?". Это значит, что анализатор распараллеливаемости циклов на основе введенной информации определил, что итерации цикла по `i` можно выполнять параллельно. В случае ответа "Yes" при генерации кода этот цикл будет помечен прагмой OpenMP и будет выполняться параллельно.

Алгоритм символьного анализа для диалогового распараллеливания

На вход алгоритму символьного анализа подается зависимость в виде пары вхождений (*A*, *B*), а также дополнительные характеристики этой зависимости (циклическая порожденность, антизависимость, ...).

Анализ строит все возможные пути из *B* в *A* на инвертированном графе потока управления. Чтобы поток управления прошел от *A* к *B*, ему необходимо пройти по одному из путей и на этом пути должны быть выполнены все условия ветвления. Для каждого пути строится предикат, собранный из всех условий ветвления на этом пути, объединенных оператором конъюнкции. После этого все предикаты путей объединяют с помощью оператора дизъюнкции.

В примере 4 всего один путь в графе потока управления от вхождения `a[i][j]` к вхождению `a[i – 1][j])`. Этот путь проходит через заголовки двух циклов и условный оператор. На этом пути три условия, из которых конструируется предикат `((i < 10 & & j < 10) & & j < 10) && A [i] [j] == 0.`

После составления предиката, он анализируется и упрощается. В примере 5 при анализе циклически порожденной зависимости между вхождениями `d[i][j]` и `d[k][j]` вычисляется условие существования такой зависимости: `"i = k"`. Выполнив подстановку `i = k` в выражение `"d[i][j] > d[i][k] + d[k][j]"`, получим `"d[k][j] > d[k][k] + d[k][j])"`. После упрощения получим `"d[k] [k] < 0"`. Полученное выражение означает условие, при котором зависимость существует. В диалог пользователю направляется вопрос: верно ли отрицание этого выражения? В случае положительного ответа анализируемую зависимость можно считать несуществующей.

Разработка диалога для уточнения информационных зависимостей выполнена Р. И. Морылевым, символьный анализ разработан В. Н. Шаповаловым, постановка задачи Б. Я. Штейнберга.

Работа поддержана ФЦП "Научные и научно-педагогические кадры инновационной России", ГК № 14.740.11.0006 от 1 сентября 2010.

Список литературы

1. **Microft A.** Programming Language Design and Analysis Motivated by Hardware Evolution / Cambridge University. Computer Laboratory, Faculty of Computer Science and Technology. 2007. URL: http://www.cl.cam.ac.uk/~am21/papers/sas07slides.pdf (дата обращения: 10.04.2012).

2. **Галушкин А. И.** Об архитектуре экзафлопных вычислительных систем (варианты исследований с использованием нейросетевых технологий) // Информационные технологии. 2012. № 2. Приложение. 24 с.

3. **Lam M., Rothberg E., Wolf M. E.** The cache performance and optimizations of blocked algorithms // Proc. of the Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IV). 1991. С. 63—74.

4. **Оптимизирующая** распараллеливающая система. URL: http: www.ops.rsu.ru (дата обращения: 10.04.2012).

5. **Штейнберг Б. Я., Алымова Е. В., Баглий А. П., Гуда С. А., Кравченко Е. Н., Морылев Р. И., Нис З. Я., Петренко В. В., Скиба И. С., Шаповалов В. Н., Штейнберг О. Б.** Особенности реализации распараллеливающих преобразований программ в ДВОР // Труды международной конференции "Параллельные вычисления и задачи управления". PACO'2010. 26—28 октября 2010 г. Москва. М.: Изд. ИПУ РАН. С. 787—854.

6. **Mohammad R.** Haghighat. Symbolic analysis for parallelizing compilers. Norwell: Kluwer Academic Publishers, 1995.

7. **Paek Y., Hoeflinger J., Padua D.** Efficient and Precise Array Access Analysis // ACM Transactions on Programming Languages and Systems. January 2002. Vol. 24, No 1. С. 65—109.

8. **Rus S., He G., Alias C., Rauchwerger L.** Region Array SSA // Proc. of the 15th International conference on "Parallel architectures and compilation techniques". 2006. С. 43—52.

9. **Stavrakos N., Carroll S., Saito H., Polychronopoulos C., Nicolau A.** Symbolic Analysis in the PROMIS Compiler // Lecture Notes in Computer Science. 2000. Vol. 1863/2000. С. 468—471.

10. **Allen R., Kennedy K.** Optimizing Compilers for Modern Architectures: A Dependence-Based Approach. San-Francisco: Morgan Kaufmann, 2002.

11. **Wiles A.** Modular elliptic curves and Fermat's Last Theorem // Annals of Mathematics. 1995. Vol. 141 (3). С. 443—551.

12. **Gergel V. P.** Teaching Course: CS338. Introduction to Parallel Programming. URL: http://www.software.unn.ru/ccam/mskurs/ENG/HTML/cs338_pp_labs.htm (дата обращения: 24.03.2010).

13. **Штейнберг Б. Я.** Блочно-рекуррентное размещение матрицы для параллельного выполнения алгоритма Флойда // Известия ВУЗов. Северокавказский регион. Естественные науки. 2010. № 5. С. 31—33.

УДК 004.514, 004.514.6

А. С. Зуев, канд. техн. наук, доц.,
e-mail: zuev_andrey@mail.ru
Московский государственный университет
приборостроения и информатики

О развитии среды виртуального рабочего стола

Представлено описание программной модели, реализующей оригинальную интерактивную среду виртуального рабочего стола, развивающую актуальные решения в данной сфере человеко-компьютерного взаимодействия.

Ключевые слова: *графический интерфейс, эргономика программного обеспечения, человеко-компьютерное взаимодействие, рабочий стол*

Введение

Компьютеры (ПК и ЭВМ) являются сегодня неотъемлемым элементом быта, а также рабочих мест в массовых видах профессий. Поэтому удобства и простота работы с их программным обеспечением, используемым как в профессиональной деятельности, так и в повседневной жизни, очень важны для пользователей. Коммерческие успехи программного обеспечения (ПО) во многом определяются удобством и простотой его эксплуатации, что вынуждает производителей ПО совершенствовать принципы и средства организации человеко-компьютерного взаимодействия (human-computer interaction, HCI [1]). Необходимой составляющей любого ПО, предназначенного для использования в интерактивном режиме, является пользовательский интерфейс — совокупность средств, методов и правил, регламентирующая и обеспечивающая взаимодействие человека с программным продуктом и компьютером.

Графический пользовательский интерфейс (ГПИ) (graphical user interface, GUI) — это разновидность пользовательского интерфейса, элементы которого (меню, кнопки, списки и т. п.) представлены на дисплее и выполнены в виде графических изображений. В настоящее время ГПИ широко распространены, а их элементы реализованы на основе различных метафор, что облегчает понимание и освоение функциональных возможностей программ пользователями.

Метафора "рабочий стол" используется в большинстве современных ГПИ, виртуальный рабочий стол основан на аналогии с офисным, поэтому пользователи легко понимают принципы работы с расположенными на нем папками и файлами [2]. Вместе с тем наблюдается тенденция к расширению функциональности виртуального рабочего стола и развитию соответствующей метафоры — специальные визуальные и анимационные эффекты применяются для симуляции трехмерного пространства, соответствующего привычной среде деятельности человека. Виртуальный рабочий стол является основной (первичной) областью работы пользователя с компьютером и совершенствование соответствующих принципов организации человеко-компьютерного взаимодействия оказывает положительное влияние на все аспекты эксплуатации ПК, ЭВМ и ПО.

Steven Sinofsky, руководитель подразделения Microsoft по разработке Windows и Windows Live, в августе 2011 г. опубликовал в своем блоге "Building Windows 8" статью "Designing for Metro style and the desktop" о новом ГПИ данной операционной системы. Приведем перевод цитат о рабочем столе: "В обозримом будущем рабочий стол будет по-прежнему играть ключевую роль в жизни многих пользователей. Поэтому мы собираемся его усовершенствовать. Мы готовы обсуждать наш дизайнерский выбор с пользователями, но в то же время хотим принимать решения в более широком контексте непревзойденной функциональности рабочего стола" [3]. Внимание к данному вопросу руководителей одной из ведущих фирм разработчиков ПО подтверждает актуальность поиска решений по развитию среды виртуального рабочего стола.

Ментальная модель, метафора и среда рабочего стола

Организация человеко-компьютерного взаимодействия основана на формировании у пользователей ментальной модели ПО — концептуального представления об особенностях и принципах его работы. В психологии ментальной моделью называют трудно формализуемую совокупность эмпирических знаний, которая формируется в сознании человека при взаимодействии с некоторым объектом.

Формирование у пользователя ментальной модели происходит на основе опыта эксплуатации компьютера и ПО, а также требует учета особенностей взаимодействия человека с объектами реального мира. Для переноса знаний пользователя о реальном окружающем мире в среду человеко-компьютерного взаимодействия применяется концепция метафор [2].

В общем случае метафора — это понятие, переносящее свойства или признаки одного объекта на другой для выявления их сходства или аналогии. Например, Baecker так характеризует метафоры: "Метафоры помогают проектировщикам, так как использование метафор позволяет им структурировать элементы интерфейса по аналогии с известной пользователям областью" [4].

В. Д. Магазанчик так говорит об использовании метафор в ГПИ [5]: "В большинстве случаев не следует заставлять пользователя создавать свою субъ-

ективную модель системы, а хорошо бы воспользоваться субъективными, уже готовыми моделями, которые были построены по другому поводу, но могут использоваться для работы с ГПИ. Таковым средством является метафора".

Метафора рабочего стола рассматривает дисплей как аналог столешницы, на которой могут находиться объекты (документы, папки) и дополнения, обеспечивающие работу пользователя, например, корзина и ярлыки. Документы и папки открываются в окнах, которые являются аналогами их бумажных копий.

Метафору рабочего стола разработал Alan Kay в Xerox PARC в 1970 г., ее первая реализация была выполнена в 1984 г. в компьютере Apple Macintosh. С расширением возможностей компьютерной техники и мультимедиа данная метафора потребовала пересмотра и развития.

В терминологии ГПИ рабочий стол (desktop — крышка стола, столешница) — это основное окно графической среды пользователя вместе с добавляемыми в него объектами и фоновым изображением. В некоторых средах (Windows, KDE, GNOME и т. п.) рабочему столу соответствует определенный каталог. Рабочий стол предоставляет пользователю первичную рабочую область — заполняет экран и формирует визуальный фон для всех выполняемых операций [6].

Среда рабочего стола (desktop environment) — это разновидность ГПИ, основанная на соответствующей метафоре и обеспечивающая пользователю область, называемую рабочим столом. В настоящее время разработано множество сред рабочего стола, различающихся трактовками его метафоры и используемыми визуальными и анимационными эффектами, их примерами являются BumpTop, GNOME, KDE, Xfce, LXDE, EDE, IRIX Interactive Desktop, OpenWindows, Ambient desktop, Mezzo, ROX Desktop, Unity и т. п.

Из-за существенной модификации исходной метафоры рабочего стола, трактовавшейся как плоскость размещения объектов, в настоящее время применяется термин виртуальный рабочий стол, характеризующий решения, при реализации которых среда рабочего стола расширяется вне физических пределов дисплея с помощью специальных функциональных возможностей ПО.

Актуальные решения в организации виртуального рабочего стола

Рассмотрим актуальные решения в организации виртуального рабочего стола, примеры реализующих их приложений и трактовок данной метафоры.

BumpTop — среда рабочего стола с широкими возможностями трехмерных анимационных эффектов, например, круговой обзор, изменение размера объектов, их размещение на "стенах" и т. д. [7] (рис. 1). Метафора рабочего стола может трактоваться как трехмерное пространство, ограниченное столешницей и офисными перегородками, аналогом которых является фоновое изображение.

Режим переключения задач в **Windows Vista** и **Windows 7** реализует отображение окон в трехмерном пространстве рабочего стола, что облегчает их обзор (рис. 2). В данном случае метафора рабочего стола может трактоваться как пространство размещения окон, ограниченное фоновым изображением.

Eaglemode — приложение, реализующее масштабируемый интерфейс (Zooming User Interface, Zoomable User Interface), в котором рабочая область представляет собой плоскость размещения объектов, свойства и содержание которых становятся доступны по мере их "приближения" при увеличении масштаба (рис. 3). В данном приложении метафора рабочего стола может трактоваться как масштабируемая плоскость отображения объектов.

Kool Desktop Environment (KDE), Cube Desktop Switcher — приложение, реализующее режим отображения рабочих столов на гранях вращаемого куба (рис. 4). В данном приложении метафора рабочего стола может трактоваться как ограниченное фоновым изображением пространство, содержащее инструмент взаимодействия с набором первичных рабочих областей.

Интерфейс **"Metro" Windows 8** ориентирован на сенсорные устройства (рис. 5). Вместо типового

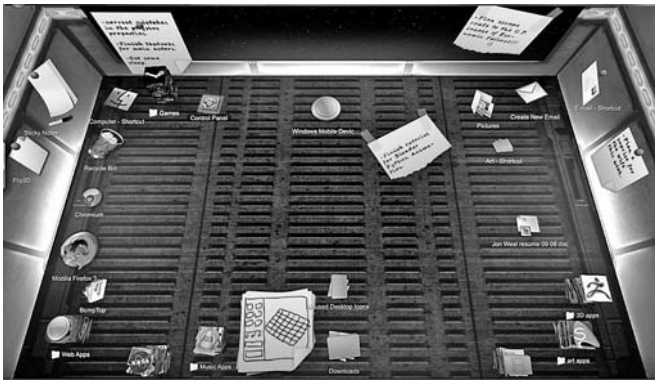
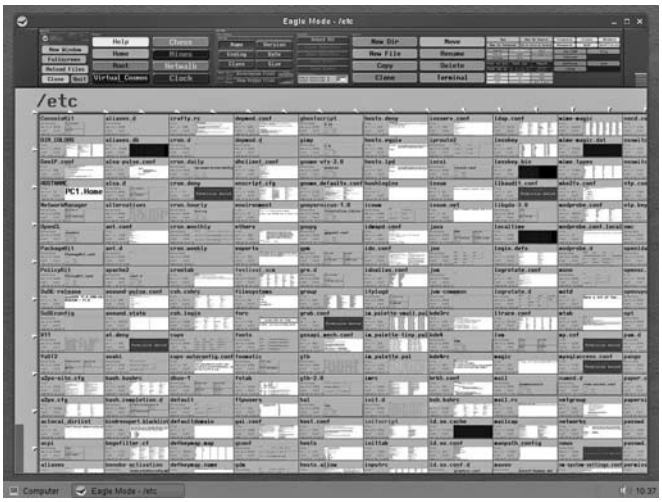


Рис. 1. Интерфейс среды рабочего стола BumpTop

Рис. 2. Режим переключения задач в Windows Vista и Windows 7



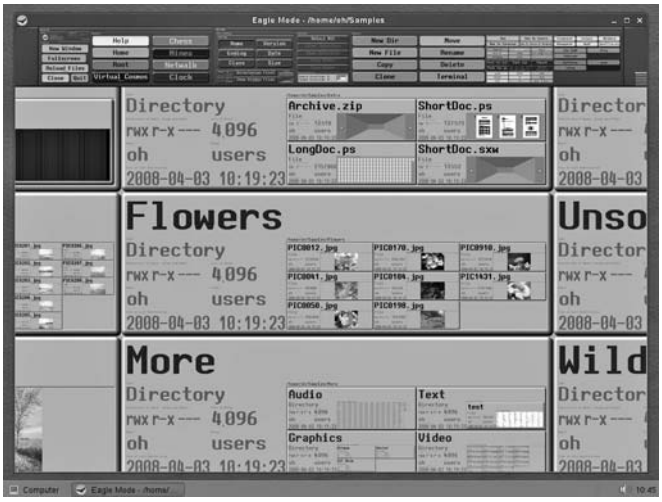


Рис. 3. Масштабируемый интерфейс приложения Eaglemode

Рис. 4. Интерфейс приложения KDE Cube Desktop Switcher

Рис. 5. Среда рабочего стола Windows 8 в стиле "Metro"

рабочего стола с кнопкой "Пуск" и панелью задач реализован стартовый экран с интерактивными панелями. Данные панели отображают названия и фрагменты приложений, предусмотрены опции их прокрутки, группировки, а также изменения размеров и расположения. Метафора рабочего стола может трактоваться как плоскость размещения панелей с заставками и данными доступных пользователю приложений.

Различные среды виртуального рабочего стола доступны для большинства операционных систем (ОС), они различаются трактовками соответствующей метафоры, функциональными возможностями и анимационными эффектами, но имеют общую особенность — фоновое изображение является нефункциональным элементом, визуальным фоном, ограничением информационного пространства.

Далее представлено описание разработанной автором программной модели, реализующей прототип оригинальной среды виртуального рабочего стола, исключающей определяемое его фоновым изображением ограничение на доступное пользователю информационное пространство.

Программная модель Interactive Desktop Environment (InDeviron)

Данная модель не является полнофункциональной средой рабочего стола, но позволяет оценить новшества и удобства работы пользователя, обеспеченные предложенными решениями, а также сделать вывод о целесообразности реализации ее функциональных возможностей в современных средах рабочего стола. Основными новшествами модели InDeviron являются следующие опции:

- создание дополнительных рабочих столов и переходов между ними (основным считается рабочий стол операционной системы);
- выделение на фоновом изображении рабочего стола областей произвольной формы и определение для них распознаваемых воздействий пользователя;
- определение для выделенных областей событий, наступающих при реализации пользователем соответствующих воздействий, — открытие папки, файла, Интернет-адреса или переход к другому рабочему столу;

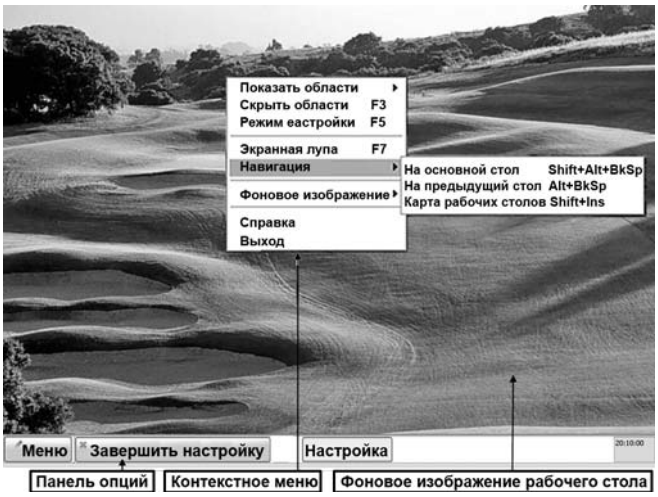


Рис. 6. Пример интерфейса модели InDeviron

- реализация событий и переходов между рабочими столами в результате выполнения определенных воздействий на области фонового изображения;
- визуальное отображение выделенных областей только в режиме настройки модели, что позволяет реализовать в режиме ее эксплуатации как обычную функциональность рабочего стола, так и интерактивное взаимодействие с его фоновым изображением.

Реализация перечисленных новшеств обеспечивает интерактивность фонового изображения рабочего стола и предоставляет пользователю возможности настройки его функциональности.

Интерфейс модели InDeviron с примером фонового изображения представлен на рис. 6. Ключевыми элементами, реализующими ее основные функциональные возможности, являются: рабочий стол (содержит фоновое изображение), контекстное меню (вызывается нажатием правой кнопки мыши) и панель опций (отображается вместо панели задач ОС в режиме настройки).

Контекстное меню модели содержит следующие пункты:

- показать области — реализует визуальное отображение выделенных пользователем областей фонового изображения (могут быть отображены все области либо отдельно соответствующие файлам, папкам, интернет-адресам и дополнительным рабочим столам);
- скрыть области — отменяет визуальное отображение выделенных областей;
- режим настройки — выполняет переход в режим настройки модели;
- экранная лупа — обеспечивает отображение в верхней части рабочего стола увеличенного изображения области текущего расположения курсора;
- навигация — отвечает за вызов карты переходов между рабочими столами, а также за переход к основному и предыдущему рабочим столам;
- фоновое изображение — содержит типовые опции позиционирования фонового изображения на рабочем столе;
- справка — открывает файл с текстом описания модели.

Режим настройки модели. Настройка модели осуществляется с помощью панели опций (рис. 7), содержащей следующие разделы:

- меню — вызывает меню опций, аналогичное контекстному меню;
- завершить настройку — выполняет выход из режима настройки;
- настройка — вызывает меню настроек модели.

На рис. 8 приведено окно "Сведения об областях", в столбцах таблицы представлены их параметры: номера, цвета выделения и соответствующие коды, указанные при создании краткие описания, ссылки на связанные объекты и их типы (файлы, папки, Интернет-адреса или рабочие столы), комбинации клавиш в соответствующих воздействиях. Отмеченные в первом столбце области можно отобразить и скрыть на фоновом изображении рабочего стола, редактировать или удалить с помощью соответствующих кнопок в нижней левой части окна.

На рис. 9 представлено окно "Создать область", обеспечивающее выбор и ввод следующих параметров:

- цвет области — вызывает стандартное окно выбора и настройки цветов (цветовое выделение области может быть определено автоматически);
- форма области — предусмотрено три варианта выделения областей на фоновом изображении: произвольная форма (определяется траекторией перемещения курсора мыши), прямоугольник и овал;
- описание области — ввод краткого текстового описания;

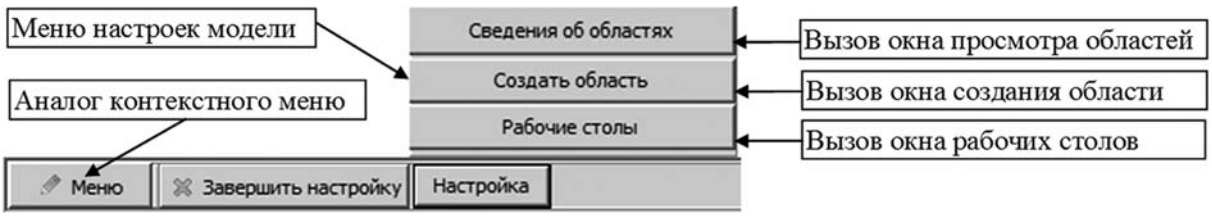


Рис. 7. Панель опций

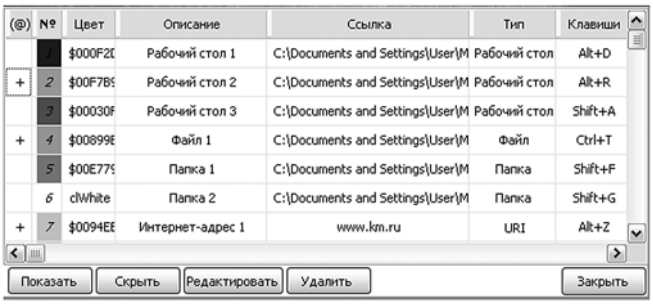


Рис. 8. Окно "Сведения об областях"

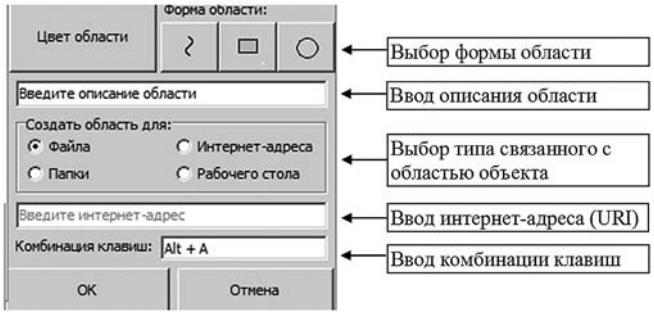


Рис. 9. Окно "Создать область"

- тип связанного объекта — для файла и папки предусмотрено диалоговое окно выбора объекта, для Интернет-адреса соответствующий URI требуется ввести в текстовое поле, для рабочего стола реализовано окно выбора дополнительных (созданных пользователем) рабочих столов;
- комбинация клавиш — после установки курсора в данное поле требуется нажать комбинацию клавиш, соответствующую распознаваемому воздействию, выполняемому совместно с нажатием кнопки мыши.

Для создания области требуется выполнить следующие действия:

1. Выбрать цвет, форму, тип связанного объекта и ввести текст описания.

2. Задать комбинацию клавиш в распознаваемом для области воздействии.

3. Перемещением курсора мыши выделить область (любого размера) на фоновом изображении рабочего стола.

4. В диалоговом окне выбрать связанный с областью объект или ввести URI (соответствующее событие будет определено автоматически).

На рис. 10 представлен пример визуального выделения областей фонового изображения в режиме настройки модели. В режиме эксплуатации модели области не выделяются, но при выполнении пользователем в их пределах соответствующих воздействий реализуются определенные для них события.

На рис. 11 представлено окно "Рабочие столы", содержащее таблицу и элементы управления, позволяющие создавать, редактировать и удалять дополнительные рабочие столы, а также переходить между ними. Для создания рабочего стола требуется указать путь к папке, содержащей размещаемые на нем объекты, путь к файлу с фоновым изображением и краткое описание.

Работа с моделью. Для реализации события, соответствующего конкретной области фонового изображения, пользователь должен нажать заданную при ее создании комбинацию клавиш и кликнуть кнопкой мыши в ее пределах.

Если области соответствуют файл, папка или Интернет-адрес, то файл будет открыт в соответствующем его формату приложении, папка — в окне операционной системы, а Интернет-адрес — в выбранном по умолчанию браузере. Если области соответствует рабочий стол, то будет реализован режим перехода между рабочими столами (рис. 12) — в центре дисплея отобразится уменьшенное фоновое изображение соответствующего рабочего стола, а текущее фоновое изображение будет затемнено. Воздействие мышью на область в центре дисплея реализует переход на соответствующий рабочий стол, а воздействие на затемненную область реализует выход из режима перехода между столами.

Для упрощения навигации между рабочими столами в контекстном меню реализовано подме-

Рис. 10. Пример визуального выделения областей в режиме настройки

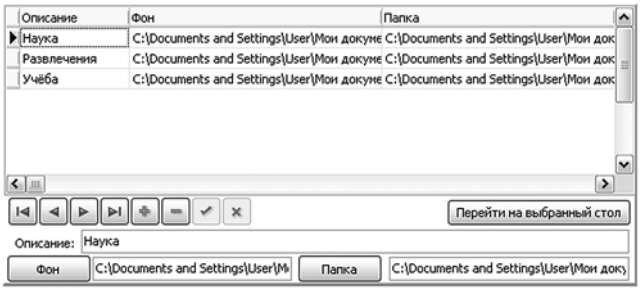


Рис. 11. Окно "Рабочие столы"

Рис. 12. Пример режима перехода между рабочими столами

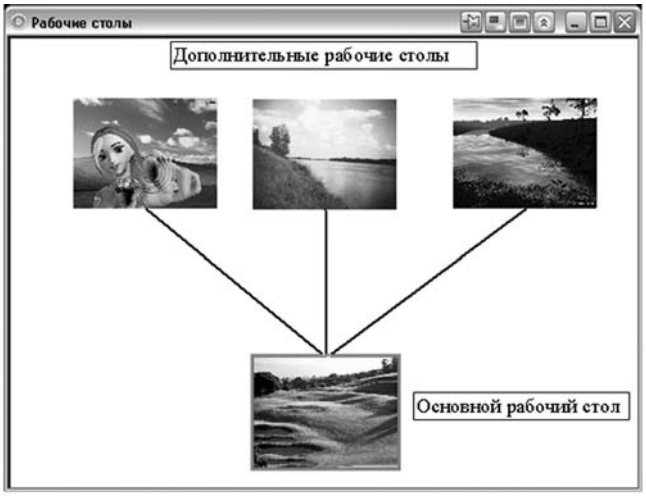


Рис. 13. Пример карты переходов между рабочими столами

ню "Навигация" (см. рис. 6), содержащее следующие пункты:

- на основной стол — служит для перехода к основному рабочему столу ОС;
- на предыдущий стол — выполняет переход к предыдущему рабочему столу;
- карта рабочих столов — отображает автоматически формируемую карту переходов между рабочими столами (рис. 13).

С помощью карты пользователь может перемещаться между рабочими столами, для обозначения которых использованы соответствующие уменьшенные фоновые изображения, текущий рабочий стол выделен рамкой. Для перехода на следующий рабочий стол требуется:

1. Нажать комбинацию клавиш, заданную при создании соответствующей области фонового изображения текущего рабочего стола.

2. Дважды кликнуть кнопкой мыши по изображению требующегося стола.

Заключение

Модель InDeviron является актуальной и оригинальной разработкой, соответствующей передовому уровню организации интерактивного взаимодействия пользователя с компьютером. Реализованные в ней решения позволяют развить концепцию организации человеко-компьютерного взаимодействия, основанную на использовании метафоры и среды виртуального рабочего стола, исключают определяемое фоновым изображением ограничение на доступное информационное пространство, обеспечивают его интерактивность и позволяют создавать дополнительные первичные рабочие области. При этом метафора рабочего стола может трактоваться как интерактивная рабочая область размещения объектов, обладающая настраиваемой функциональностью.

Представленная модель является прототипом, позволяющим оценить предложенные новшества, а также обосновать целесообразность их реализации, в том числе в современных средах виртуального рабочего стола. При интегрировании функциональных возможностей модели в операционную систему или среду рабочего стола набор реализуемых событий (открытие файла, папки и Интернет-адреса) может быть расширен. Представленные в модели решения могут быть реализованы также в рассмотренных в статье приложениях для обеспечения интерактивности фонового изображения их рабочих столов.

Список литературы

1. **Гращенко Л. А., Фисун А. П.** Теоретические и практические основы человеко-компьютерного взаимодействия: базовые понятия человеко-компьютерных систем в информатике и информационной безопасности. Орел: ОГУ, 2004. 169 с.

2. **Мандел Т.** Дизайн интерфейсов. М.: ДМК Пресс, 2005. 416 с.

3. **Sinofsky S.** Designing for Metro style and the desktop. URL: http://blogs.msdn.eom/b/b8/archive/2011/08/31/designing-for-metro-style-and-the-desktop.aspx [дата обращения 12.01.12].

4. **Baecker R. M., Gridin J., Buxton W., Greenberg S.** Designing to fit Human capabilities // Readings in Human-Computer Interaction: Toward the year 2000. San Francisco, 1995.

5. **Магазанчик В. Д.** Человеко-компьютерное взаимодействие: учебн. пособие. М.: Университетская книга; Логос, 2007. 256 с.

6. **Гультяев А. К., Машин В. А.** Проектирование и дизайн пользовательского интерфейса. СПб.: КОРОНА принт, 2004. 352 с.

7. **Agarawala A. and Balakrishnan R.** Keepin' It Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen. University of Toronto. URL: www.dgp.toronto.edu.

УДК 004.327.12

С. С. Садыков, д-р техн. наук, проф.,
e-mail: sadykovss@yandex.ru,
С. В. Савичева, аспирант,
e-mail: savicheva-svetlana2010@yandex.ru
Муромский институт (филиал) ФГБОУ ВПО
"Владимирский государственный университет
имени А. Г. и Н. Г. Столетовых"

Распознавание плоских объектов при их наложении

Предложен алгоритм идентификации двух наложенных реальных плоских объектов на основе метода Байеса. В качестве основного признака использованы значения кривизны в точках контура. Дополнительными признаками являются длины выпуклых и вогнутых участков и коэффициенты выпуклости и вогнутости контура объекта. Работа алгоритма показана на примерах.

Ключевые слова: *идентификация наложенных реальных объектов, значение α-функции, метод Байеса, распознавание, признак, кластер*

Введение

В настоящее время автоматические системы распознавания (АСР) находят широкое применение в различных областях промышленности. При этом основной задачей АСР является узнавание объектов, расположенных в поле зрения видеодатчика. Эта задача включает в себя по степени сложности несколько подзадач:

- распознавание одиночного объекта;
- распознавание нескольких изолированных друг от друга объектов;
- распознавание двух и более соприкасающихся и наложенных друг на друга объектов.

На практике возможно возникновение любой из перечисленных ситуаций по отдельности и в совокупности.

На данный момент задача распознавания одиночных и изолированных объектов решена с высокой степенью достоверности путем использования несложных алгоритмов идентификации [1—3]. Как правило, основные трудности возникают при решении задачи распознавания соприкасающихся и наложенных объектов различных типов. Это связано с тем, что при наложении один объект может закрыть большую часть другого, поэтому часто закрытый объект идентифицировать не удается.

Ниже предлагается алгоритм идентификации двух плоских объектов при их наложении.

Алгоритм идентификации

Алгоритм идентификации состоит из двух этапов.
Этап 1. Обучение.
Этап 2. Распознавание.

Этап обучения

А. *Формирование эталонных α-функций исходных объектов.*

Пусть используется n исходных объектов. Для каждого исходного объекта вычисляются значения α-функции.

Под α-функцией понимается последовательность значений кривизны, вычисленной в каждой точке дискретного контура изображения объекта [1, 4—7].

Б. *Генерация эталонов наложенных объектов для каждого сочетания исходных объектов и формирование классов наложенности.* Размер поля для генерации был выбран равным 800 × 800 точек. При этом размер рабочей области (поля зрения видеодатчика) составляет 799 × 799 точек. Такое ограничение было введено для того, чтобы избежать ситуации, когда объект касается краев поля зрения, поскольку в таких случаях объект считается не полностью вошедшим в кадр, и в АСР он не анализируется.

Исходной информацией для генерации сочетаний плоских наложенных объектов являются координаты точек x и y исходных объектов и их угол наклона. Значения параметров задаются с помощью нескольких датчиков случайных чисел. При этом значения угла наклона изменяются в пределах от 0 до 360° (шаг равен 1°), а значения координат x и y для каждого объекта — в пределах от 0 до 799 (шаг равен 1) [8]. Для исследования выбираются случаи касания и наложения объектов.

В. *Нормализация изображений сгенерированных наложенных объектов.* Под нормализацией понимается приведение изображения, полученного с видеодатчика, к некоторому стандартному виду с использованием группы преобразований, связывающих эталонные и входные изображения.

В настоящее время в литературе описан целый ряд алгоритмов нормализации изображений в системах технического зрения (СТЗ). При этом к данным алгоритмам предъявляются два основных требования:

- операция перехода от входного изображения к нормализованному должна быть простой для

реализации на программном и аппаратном уровне;
- операция перехода должна быть помехоустойчивой.

Наибольшее распространение в СТЗ получила последовательная нормализация. Это связано с ограниченными возможностями СТЗ. Под последовательной нормализацией понимается многошаговая процедура, в которой на каждом шаге осуществляется нормализация только одной группы базовых преобразований (смещение, масштабирование, поворот).

В данной работе нормализация осуществляется с использованием алгоритма, основанного на методе главных компонент, описанного в работе [9].

Достоинством данного алгоритма является возможность снижения размерности, поскольку уже несколько первых главных компонент содержат большую часть информации.

Алгоритм состоит в следующем.

Пусть бинарное изображение объекта $B_0(x, y)$ содержит N точек. Эти точки будем рассматривать как множество реализаций X случайного вектора $x = [i, j]^{\mathrm{T}}$, где i — номер строки, соответствующей данной точке изображения; j — номер столбца.

Под преобразованием первых главных компонент (ПГК) понимают преобразование вида:

$$Y = V^{\mathrm{T}}(X - X_0), \qquad (1)$$

где $X_0 = E\{X\}$ — среднее значение результатов измерения всех параметров случайного вектора x; V — собственные векторы ковариационной матрицы

$$C = E\{(X - X_0)(X - X_0)^{\mathrm{T}}\}. \qquad (2)$$

В результате преобразования (1) получим изображение $B_1(x, y)$, на котором объект повернется так, что направления главных осей объекта (собственных векторов) будут совпадать с осями x и y соответственно.

Г. *Вычисление признаков объектов и формирование на их основе векторов-признаков эталонов наложенных объектов.* Основным признаком, используемым при идентификации, являются значения α-функции контуров изображений наложенных объектов.

На основе α-функции вычисляются дополнительные признаки: длины вогнутых и выпуклых участков контура объекта и коэффициенты вогнутости и выпуклости контура.

Под вогнутым участком контура объекта понимается участок кривой, на котором значение кривизны точек, составляющих этот участок, отрицательны.

Под выпуклым участком контура объекта понимается участок кривой, на котором значение кривизны точек, составляющих этот участок, положительны.

Длины выпуклых и вогнутых участков определяются по формуле [10]:

$$l_{\text{вогн, вып}} = 1k^4 + \sqrt{2}\,k^d,$$

где k^4 — число четырехсвязных точек; k^d — число d-связных точек.

Коэффициенты вогнутости и выпуклости контура вычисляются так [10]:

$$k_{\text{вогн, вып}} = \frac{L_{\text{вогн, вып}}}{l},$$

где $L_{\text{вогн, вып}} = \sum_{i=1}^{k_{l(l\text{вогн, вып})}} l_{i\text{вогн, вып}}$ — суммарная длина вогнутых (выпуклых) участков контура; $k_l(l_{\text{вогн, вып}})$ — число вогнутых (выпуклых) участков контура; l — общая длина контура.

Поскольку признаки объектов представлены в разных единицах измерения, то определить расстояние между объектами не удается. В таком случае согласно статистическому анализу, данные подвергаются нормировке, которая переводит их в безразмерные величины.

Нормировка значений признаков представляет собой переход к введению новой условной единицы измерения, допускающей формальное сопоставление объектов. Наиболее распространенным способом нормирования ввиду простоты и удобства использования является следующий:

$$z = \frac{x}{x_{\max}},$$

где $x_{\max}$ — наибольшее значение признака x.

Таким образом, на этапе обучения формируется K кластеров. В состав каждого кластера Q_r ($r = 1, \ldots, K$) входят значения M_r эталонных векторов-признаков для каждого сочетания n исходных объектов. Общее число эталонов составляет M.

Векторы-признаки каждого объекта состоят из двух частей. В общем виде они могут быть представлены следующим образом:

x_1	x_2	...	x_n
α_1	α_2	...	α_n

y_1	y_2	y_3	y_4
$l_{\text{вогн}}$	$l_{\text{вып}}$	$k_{\text{вогн}}$	$k_{\text{вып}}$

Этап распознавания

Пусть имеется обучающая выборка, состоящая из N изображений наложенных объектов для каждого из K кластеров.

Для каждого изображения в кластере сформирован n-компонентный вектор-признак $X^* = [x_1, x_2, \ldots, x_n]$.

Задача состоит в сравнении векторов-признаков экзаменационного (неизвестного) объекта со всеми векторами-признаками эталонов по кластерам наложенных объектов с использованием метода Байеса.

Решение о принадлежности принимается на основе следующего правила:

Объект с номером i и вектором-признаком X считается принадлежащим к кластеру Q в случае максимума апостериорной вероятности, т. е.*

$$X^* \in Q^*, \text{ если } P_i(Q^*|X^*) = \max P_i(Q_r|X^*),$$

где

$$P_i(Q_i|X^*) = P(Q_r)\frac{P(X^*|Q_r)}{P(X^*)}. \qquad (3)$$

Входящие в формулу (3) компоненты определяются следующим образом.

1. $P(Q_r) = \dfrac{M_r}{M}$ — априорная вероятность существования r-го класса, где M_r — число объектов r-го класса в обучающей выборке; M — общее число объектов в обучающей выборке.

2. $P(X^*|Q_r) = \sum\limits_{j=1}^{N} P(X_j^*|Q_r)$ — вероятность принятия j-м вектором-признаком значения X_j^*.

Значение $P(X_j^*|Q_r)$ определяется следующим образом:

$$P(X_j^*|Q_r) = \frac{M_{rj}}{M},$$

где M_{rj} — число объектов обучающей выборки, принадлежащих r-му классу, у которых j-й признак принимает значение X_j^*; M — общее число объектов в обучающей выборке.

3. $P(X^*) = \sum\limits_{j=1}^{K} P(Q_r)P(X^*|Q_r)$ — вероятность возникновения в обучающей выборке конкретного сочетания конкретных векторов-признаков.

Экспериментальные исследования

Для статистического исследования разработанного алгоритма идентификации наложенных двух реальных плоских объектов использовалась выборка, включающая по 2000 изображений для каждого сочетания исходных объектов. Изображения исходных 10 объектов и примеры наложения для некоторых сочетаний объектов приведены на рис. 1, 2.

Векторы-признаки некоторых наложенных объектов приведены в табл. 1.

В табл. 2 и на графике (рис. 3) приведена часть результатов идентификации наложенных объектов по α-функции.

Проведенные эксперименты показали:

1) данный алгоритм является в большинстве случаев нестабильным;

2) процент относительной ошибки распознавания составляет 15 %.

Для улучшения результатов необходимо ввести дополнительные признаки, которые бы позволили анализировать не весь контур объекта, а лишь отдельные его участки.

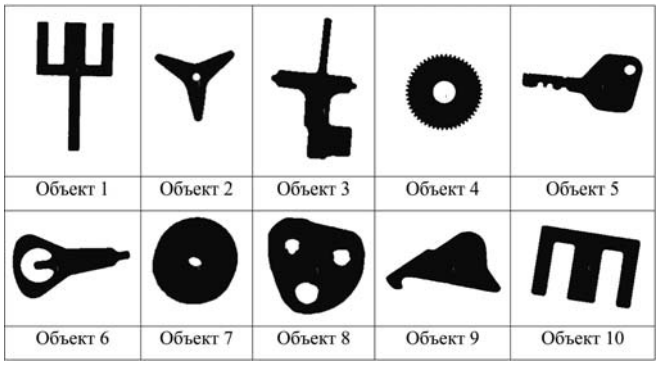


Рис. 1. Изображения исходных объектов

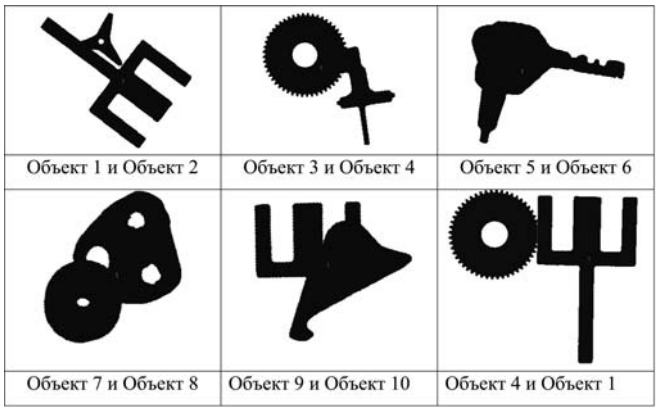


Рис. 2. Примеры изображений касающихся и наложенных объектов

Таблица 1

Значения векторов-признаков

Признак	Примеры векторов-признаков				
	1	2	3	...	*N*
α_1	−0,333	−0,667	−0,667	...	−0,333
α_2	0	0,667	0,333	...	0
α_3	−0,333	−0,667	0	...	0
α_4	0,333	−0,333	−0,667	...	0
...	...	...	...	...	...
α_{25}	−0,333	0	0,667	...	0
...	...	...	...	...	...

Таблица 2

Результаты идентификации наложенных объектов по α-функции

№ п.п.	Сочетания объектов	Число испытаний	Относительное значение ошибки распознавания
1	1 и 1	2000	0,9
2	1 и 2	2000	0,84
...	...	...	...
21	3 и 4	2000	0,83
22	3 и 5	2000	0,89
...	...	...	...
41	6 и 7	2000	0,83
42	6 и 8	2000	0,89
...	...	...	...
54	9 и 10	2000	0,83
55	10 и 10	2000	0,83

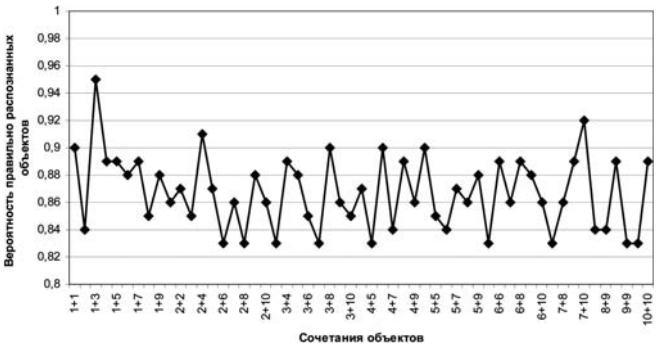


Рис. 3. График изменения вероятности правильно распознанных объектов в зависимости от сочетания объектов

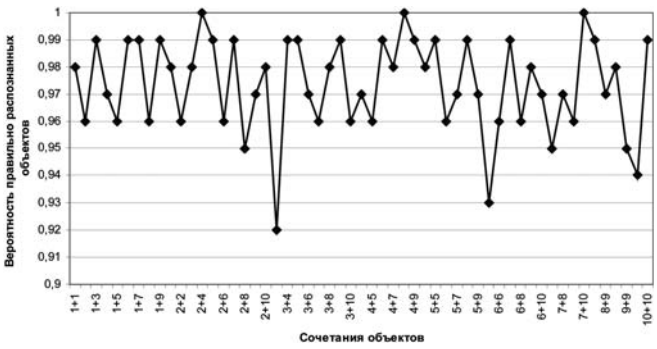


Рис. 4. График изменения вероятности правильно распознанных объектов в зависимости от сочетания объектов

Таблица 3

Результаты идентификации наложенных объектов и с использованием дополнительных признаков

№ п.п.	Сочетания объектов	Число испытаний	Относительное значение ошибки распознавания
1	1 и 1	2000	0,98
2	1 и 2	2000	0,96
...	...	...	...
21	3 и 4	2000	0,92
22	3 и 5	2000	0,99
...	...	...	...
41	6 и 7	2000	0,93
42	6 и 8	2000	0,96
...	...	...	...
54	9 и 10	2000	0,94
55	10 и 10	2000	0,99

Часть результатов идентификации наложенных объектов с использованием дополнительных признаков приведена в табл. 3 и на графике (рис. 4).

Заключение

На основе предложенного алгоритма было создано программное приложение, которое позволяет распознавать два наложенных как однотипных, так и разнотипных объекта на основе метода Байеса.

В качестве основного признака были использованы значения α-функции. Дополнительными признаками являются длины выпуклых и вогнутых участков, коэффициенты выпуклости и вогнутости контуров объектов.

Экспериментально были получены данные по распознаванию различных сочетаний исходных объектов. Исследования показали, что алгоритм распознавания наложенных объектов на основе значений α-функции является нестабильным. Максимальный процент ошибки составляет 17 % (см. табл. 2, рис. 3).

Введение дополнительных признаков позволило минимизировать число ошибок при идентификации. Как видно из табл. 3 максимальная относительная ошибка распознавания не превосходит 8 %. Исходя из анализа проведенных исследований можно сделать вывод, что алгоритм дает хорошие результаты и может быть использован для практического применения в промышленных СТЗ.

Основное число ошибок при распознавании наложенных объектов приходится на ситуации с большим процентом наложения (от 60 %). В результате получается, что большая часть одного объекта практически полностью закрыта другим объектом, и его идентифицировать не удается.

Алгоритм совершенствуется в целях его дальнейшего использования для идентификации трех реальных плоских объектов и минимизации временных затрат на его работу.

Список литературы

1. **Садыков С. С., Савичева С. В.** Идентификация реальных плоских объектов на основе единственного признака точек их внешних контуров // Информационные технологии. 2011. № 8. С. 13—16.

2. **Садыков С. С., Савичева С. В.** Идентификация реальных плоских объектов на основе их сигнатуры // Вестник компьютерных и информационных технологий. 2012. № 1. С. 17—20.

3. **Садыков С. С., Савичева С. В.** Алгоритм идентификации реальных плоских объектов с использованием значений их *r*-функций // Надежность и качество-2011: Труды Международного симпозиума / Под ред. Н. К. Юркова. Пенза: Изд-во Пенз. Гос. ун-та, 2011. С. 123—127.

4. **Садыков С. С., Савичева С. В.** Алгоритм идентификации плоских объектов с использованием минимального числа признаков // Автоматизация и современные технологии. 2011. № 7. С. 3—6.

5. **Савичева С. В.** Экспериментальное исследование алгоритма идентификации плоских объектов // Алгоритмы, методы и системы обработки данных: сборник научных трудов. Вып. 15. Муром: Изд.-полиграфический центр МИ ВлГУ, 2010. С. 153—160.

6. **Садыков С. С., Савичева С. В., Веденин А. С.** Экспериментальное исследование алгоритма идентификации наложенных объектов на основе алгоритмов трансформации контура и α-функции // Алгоритмы, методы и системы обработки данных: Электронный научный журнал. Вып. 1 (19). — Муром: Муромский институт (филиал) ВлГУ, 2012. С. 22.

7. **Садыков С. С., Савичева С. В., Комков В. А.** Сравнение алгоритмов распознавания наложенных объектов на основе α-функции и на основе особых участков // Алгоритмы, методы и системы обработки данных: Электронный научный журнал. Вып. 1 (19). — Муром: Муромский институт (филиал) ВлГУ, 2012. С. 23.

8. **Садыков С. С., Савичева С. В.** Исследование наложенности плоских объектов в поле зрения СТЗ // Приборостроение. 2012. № 2. С. 14—19.

9. **Бабаян П. В., Фельдман А. Б.** Распознавание объектов на изображениях при наблюдении из космоса // Вестник РГРТУ. 2008. № 4 (вып. 26). С. 20—28.

10. **Садыков С. С., Стулов Н. Н.** Методы и алгоритмы выделения признаков объектов в системах технического зрения. М.: Горячая линия — Телеком, 2005. 204 с.

УДК 519.688

Т. О. Перемитина, канд. техн. наук, науч. сотр.,
С. В. Лучкова, аспирант,
e-mail: sonetta27@gmail.com,
Институт химии нефти СО РАН, Томск

Применение программного комплекса "Нечеткая система на основе эволюционной стратегии" для задачи импутирования

Рассматривается модель восстановления данных, реализованная в программном комплексе "Нечеткая система на основе эволюционной стратегии" для задачи импутирования. Описывается нечеткая система, эволюционная стратегия и задача импутирования. Приведены результаты экспериментальных исследований.

Ключевые слова: *нечеткая система, эволюционная стратегия, задача импутирования*

Введение

Многие исследования связаны со сбором и обработкой данных, представленных в виде таблиц наблюдений. Данные из этих таблиц используют в различных задачах анализа и задачах построения моделей прогноза. Но часто, по самым различным причинам, эти таблицы содержат пропущенные значения.

Большинство алгоритмов не могут обрабатывать неполные данные, так как получаются неадекватные модели, либо модели вообще построить невозможно. Поэтому импутирование, или восстановление данных является очень важным моментом в обработке данных.

Анализ преимуществ и недостатков известных алгоритмов восстановления для решения задачи импутирования показал, что наиболее оптимальными методами восстановления пропущенных значений являются методы, основанные на нечетких моделях. Основное преимущество этих моделей — снятие требований нормальности распределения, однородности и полноты данных. Все перечисленные требования к исходным данным должны выполняться для применения статистических методов восстановления пропущенных значений, что усложняет процесс предварительной подготовки выборки данных, замедляет процесс анализа в целом и снижает точность.

Цель данной работы — описать программный комплекс "Нечеткая система на основе эволюционной стратегии" для задачи импутирования и проанализировать его работу.

Модель восстановления

За основу модели возьмем нечеткую систему типа синглтон [1], в которой i-е правило выглядит следующим образом:

$$\text{IF } x_1 = A_{1i} \text{ AND } \ldots \text{ AND } x_m = A_{mi} \text{ THEN } y = r_i,$$

где A_{ij} — лингвистический терм, которым оценивается переменная x_i; r_i — действительное число, которым оценивается выход y.

Нечеткая система осуществляет отображение $f:\Re^m \to \Re$, заменяя оператор нечеткой конъюнкции произведением, а оператор агрегации нечетких правил сложением. Получаем выходное значение $F(\mathbf{x})$:

$$F(\mathbf{x}) = \frac{\sum\limits_{j=1}^{m} r_j \cdot \prod\limits_{i=1}^{n} \mu_{A_{ij}}(x_i)}{\sum\limits_{j=1}^{m} \prod\limits_{i=1}^{n} \mu_{A_{ij}}(x_i)},$$

где $\mathbf{x} = [x_1, \ldots, x_n]^{\mathrm{T}} \in \Re^n$ — значение i-го входа, $\mu_{A_{ij}}(x_j)$ — функция принадлежности нечеткого терма A_{ij}; r_j — выходное значение в j-м правиле.

Функции принадлежности для нечетких систем представляют собой субъективное представление эксперта о предметной области. Часто такая субъективность помогает снизить степень неопределенности при решении слабо формализованных задач. Существует свыше десятка типовых форм кривых для задания функций принадлежности. Наибольшее распространение получили треугольная, трапецеидальная и гауссова функции принадлежности. В работе для идентификации воспользуемся треугольной функцией, которая определяется тройкой чисел (a, b, c), и ее значение в точке x вычисляется согласно выражению:

$$\mu_{A_{ij}}(x_j) = \begin{cases} 1 - \dfrac{b-x}{b-a}, & a \leqslant x \leqslant b, \\ 1 - \dfrac{x-b}{c-b}, & b \leqslant x \leqslant c, \\ 0, \text{ в остальных случаях.} \end{cases}$$

Параметры функции принадлежности термов входных переменных нечеткой системы можно представить в виде вектора [1]. Например, при n входных переменных, определенных на t термах с треугольными функциями принадлежности, описываемыми тройкой параметров, в модели типа синглтон вектор параметров будет выглядеть следующим образом:

$$\theta_n = [a_{11}b_{11}c_{11}\ldots a_{1t}b_{1t}c_{1t}a_{21}b_{21}c_{21}\ldots \\ \ldots a_{2t}b_{2t}c_{2t}\ldots a_{n1}b_{n1}c_{n1}\ldots a_{nt}b_{nt}c_{nt}],$$

где a_{ij}, c_{ij}, b_{ij} — параметры треугольной функции принадлежности формулы (1) i-й лингвистической переменной j-го терма.

Для настройки вектора параметров нечеткой системы воспользуемся методом эволюционной стратегии.

Эволюционная стратегия — эвристический метод оптимизации в разделе эволюционных алгоритмов, основанный на адаптации и эволюции. Стратегия основана на механизмах естественного отбора и наследования. В ней используется принцип "выживания" наиболее приспособленных особей. Преимущества алгоритма перед другими методами оптимизации заключаются в параллельной обработке множества альтернативных решений [2, 3].

Алгоритм работает с популяцией особей (хромосом), каждая из которых представляет собой упорядоченный набор параметров задачи, подлежащих оптимизации. Основной характеристикой каждой особи является ее мера приспособленности.

При поиске решения в эволюционной стратегии вначале происходит мутация и скрещивание особей для получения потомков, затем детерминированный отбор без повторений лучших особей из общего поколения родителей и потомков.

Работа алгоритма представляет собой итерационный процесс, который продолжается до выполнения условия останова — выполнение заданного числа поколений.

Постановка задачи восстановления пропусков в интерпретации нечеткой системы

Рассмотрим особенности структуры таблиц данных для использования предложенного метода. Пусть $\mathbf{X} = (X_1, X_2, ..., X_n)$ — вектор входных параметров; m — число записей в таблице; $\mathbf{A} = (a_{ij})_{i=1,j=1}^{m\ \ n}$ — матрица исходной информации. Она имеет пропуски, обозначенные звездочками (табл. 1). Допускается по одному пропуску на запись, так как пропущенное значение будет являться выходным значением для данной записи.

Таким образом, одно пропущенное значение в данных восстанавливается по значениям всех имеющихся записей.

Алгоритм восстановления [4] будет выглядеть так:

Вход: таблица наблюдений с пропусками в записях.

Шаг 1. Задаем параметры нечеткой системы.

Таблица 1

Структура входной информации

Число записей	X_1	X_2	X_3	...	X_{n-1}	X_n
1	a_{11}	a_{12}	a_{13}	.	*	a_{1n}
2	a_{21}	a_{22}	a_{23}	.	a_{2n-1}	a_{2n}
...	.	.	.	.	.	.
m	a_{m1}	a_{m2}	a_{m3}	.	a_{mn-1}	a_{mn}

Шаг 2. Загружаем входные данные (таблицу наблюдения).

Шаг 3. Выбираем параметры алгоритма эволюционной стратегии.

Шаг 4. Применяем эволюционную стратегию.

Шаг 5. Отбираем лучшую хромосому. Если достигнуто условие выхода — *Шаг 7*, иначе *Шаг 4*.

Шаг 6. Подставляем в базу правил записи с пропусками из таблицы наблюдения и восстанавливаем пропуск на основе сформированной базы правил и лучшей хромосомы.

Шаг 7. Выводим решения.

Выход: таблица наблюдений с восстановленными значениями.

Выбор средства реализации

В качестве средства реализации программного комплекса "Нечеткая система на основе эволюционной стратегии" для задачи импутирования был выбран язык объектно-ориентированного программирования *C#*, а средой разработки — *Microsoft Visual Studio 2010.*

C# — это объектно-ориентированный язык программирования общего назначения. *C#* дает разработчикам, занимающимся написанием кода, широкие возможности и языковую поддержку для создания сложных приложений [5]. К тому же *C#* — один из языков программирования, который может использоваться для создания приложений, выполняемых в среде .NET CLR. Этот язык — результат эволюции языков *C* и *C++*, созданный компанией *Microsoft* специально для использования на платформе .NET. Этот язык включает в себя самую полную поддержку структурного, компонентно-ориентированного и объектно-ориентированного программирования, которую только можно ожидать от современного языка [6].

Среда разработки *Microsoft Visual Studio 2010* — это набор инструментов и средств, предназначенных для помощи разработчикам программ любого уровня квалификации в решении сложных задач. *Visual Studio* улучшает процесс разработки и упрощает разработку высокоэффективных программ. Средства *Visual Studio* позволяют разработчикам работать с большей отдачей и затрачивать меньше усилий на повторяющиеся задачи. В версиях *Visual Studio* постоянно появляются новые средства, позволяющие разработчикам сосредоточиться на решении основных проблем, а не на рутинной работе. Например, дополнение *ReSharper*, созданное для повышения эффективности работы, проводит статистический анализ кода в масштабе всего решения, предусматривает дополнительные средства автозаполнения, навигации, поиска, подсветки синтаксиса, форматирования, оптимизации и генерации кода. Также *Visual Studio* разрабатывается таким образом, чтобы обеспечить высокую надежность и совместимость. *Visual Studio* обладает удачным сочетанием

безопасности, масштабируемости и взаимодействия. В *Visual Studio* всегда поддерживаются новейшие технологии, и по возможности, обеспечивается обратная совместимость.

Описание структуры программного комплекса

После изучения основных задач программы и особенностей нечетких систем было спроектировано и реализовано приложение для выполнения основных функций нечеткой системы: ее построения, загрузки, сохранения, параметрической настройки методом эволюционной стратегии, расчета ошибок вычисления. И для решения задачи импутирования были добавлены модуль для имитации восстановления пропусков с помощью метода скользящего экзамена [7] и модуль восстановления пропусков.

Требования к системе:

1) универсальность классов системы;

2) универсальность методов и функций;

3) оптимальность хранения информации и удобное обращение к ней;

4) расширяемость, возможность добавлять новые методы к базовой сборке.

Визуальная часть программного комплекса

В среде визуального программирования *Microsoft Visual Studio 2010* был реализован программный комплекс. Входными данными является таблица наблюдений или загруженная нечеткая модель из XML-файла, выходными — набор параметров нечеткой системы, обеспечивающий наиболее адекватное построение модели, таких как таблица наблюдений, база правил, параметры функции принадлежности, параметры методов оптимизации и значения основных ошибок, по которым оценивается адекватность модели.

Для настройки входных данных в программе реализованы два типа оператора скрещивания: многоточечный и унифицированный. Представлены также (μ, λ) и $(\mu + \lambda)$ — эволюционные стратегии [8, 9]. Селекция — случайный отбор, турнирный отбор, рулеточный отбор и стратегия элитаризма. И предоставлен многоточечный случайный метод мутации, который в частном случае также является и одноточечным.

Результаты также можно сохранить в XML-файле. Запись в файл происходит путем сериализации объекта данного класса (чтение путем десериализации). Такой способ работы с файлами обеспечивает надежность, расширяемость и простоту.

Визуально-описательно структура может быть представлена как дерево элементов. Элементы XML описываются тегами. Таким образом, описание XML-структуры представления нечеткой системы типа синглтон [10] представлено так:

<FuzzySystem/> — корневой тег, включает в себя три основных тега: **<Variables/>**, **<Rules/>**, **<Table/>**.

- **<Variables/>** — описание лингвистических переменных (ЛП). Является тегом контейнером для тегов **<Variable/>**. Атрибуты: *Count* — число ЛП.
  - **<Variable/>** — описание ЛП. Атрибуты: *Name* — имя ЛП; *Min* — минимальное значение области определения ЛП; *Max* — максимальное значение области определения ЛП.

  Подтеги: **<Terms/>**.
- **<Terms/>** — тег-контейнер для тегов **<Term/>**. Атрибуты: *Count* — число термов в ЛП.
- **<Term/>** — описание терма. Атрибуты: *Name* — имя терма; *TermType* — тип терма (возможные типы: TriangleTerm, GaussTerm, ParabolaTerm, TrapezeTerm). Подтеги: **<Params/>**.
  - **<Params/>** — тег-контейнер для термов, описываемых тегом **<Param/>**. Число параметров терма зависит от его типа. Так, для треугольного терма — три параметра, для параболического терма — два параметра.
- **<Param/>** — параметры терма. Атрибуты: *Number* — номер параметра; *Value* — значение параметра.
- **<Rules/>** — описание базы правил (БП) нечеткой системы. Атрибуты: *Count* — число правил в БП. Подтеги: **<Rule/>**.
  - **<Rule/>** — описывает правило в БП.

  Подтеги: **<Antecedent/>**, **<Consequent/>**.
- **<Antecedent/>** — тег-контейнер для тегов **<AntecedentPair/>**.
- **<AntecedentPair/>** — тег, описывающий антецедент правила. Атрибуты: *Variable* − имя ЛП; *Term* — имя терма.
- **<Table/>** — содержит описание таблицы наблюдений. Атрибуты: *Count* — число наблюдений в таблице.

  Подтеги: **<Row/>**.
  - **<Row/>** Содержит информацию об одной строке таблицы наблюдений. Подтеги: **<Cells/>**, **<Result/>**:
- **<Cells/>** Содержит информацию о значении входных ЛП. Подтеги **<Cell/>**:
- **<Cell/>** информация о значении входной переменной в данной строке таблицы наблюдений. Атрибуты: *VarName* — имя переменной для которой определено значение; *Value* — значение переменной
- **<Result/>** значение результирующий переменной. Атрибуты: *VarName* — имя результирующей переменной; *Value* — значение результирующей переменной.

Постановка эксперимента

Разработанный программный комплекс был применен на данных вязкопарафинистой нефти, содержащих 141 запись по восьми характеристикам. В полную таблицу были специально введены пропуски, такой подход позволяет рассчитывать точность восстановления, так как мы можем сравнить полученный результат с "пропущенными" данными.

Таблица 2

Оптимальный параметр нечеткой системы

Число термов на параметр	Ошибка вычисления		
	Средняя квадратичная (СКО)	Средняя абсолютная (САО)	Максимальная (МО)
3	0,06451	0,442859	2,29
4	0,04226	0,329707	2,09
5	**0,04195**	**0,334178**	**1,84**

На первом этапе был подобран оптимальный параметр нечеткой системы (НС) — число термов функции принадлежности на входной параметр. Как видно из табл. 2, оптимальное число термов будет 5.

Далее был запущен модуль восстановления пропусков с турнирным и элитарным алгоритмом селекции в методе эволюционной стратегии (ЭС). На этом этапе исследовалась как работа модели восстановления, так и параметры работы программного комплекса (затраченное вычислительное время и загрузка процессора). В программном комплексе предусмотрено разделение операций модели на потоки, для вычисления эффективности такого шага мы использовали ПК1: Intel Core 2 Duo, 2.0 ГГц, 2 Гбайт ОП, MS Win7 Premium и ПК2: Intel Core i7, 2,2 ГГц, 8 Гбайт ОП, MS Win7 Premium. Результаты представлены в табл. 3, где вторая графа отображает затраченное вычислительное время на 80 итераций, т. е. это время, которое было затрачено на улучшение решения, полученного нечеткой системой за счет настройки параметров системы методом эволюционной стратегии (т. е. значения граф "СКО НС" и "САО НС" в значения "СКО НС + ЭС" и "САО НС + ЭС" соответственно), а третья графа "ЦП, %" отображает загрузку процессора.

Как видно из табл. 3, многопоточное разделение операций модели приводит к тому, что вычислительное время снижается в 2,7 раз, а загрузка ЦП в 3,8 раз, что является хорошим ускорением вычислений, а ошибки вычислений в среднем уменьшаются на 7 %.

Таблица 3

Результаты исследования работы модели и программного комплекса

ПК	Время, ч	ЦП, %	Ошибка вычисления			
			СКО НС	СКО НС + ЭС	САО НС	САО НС + ЭС
Элитарный алгоритм селекции						
ПК1	50:17:59	50	1,786795	1,618248	4,536805	3,620211
ПК2	18:35:01	13	2,031218	1,219684	4,725026	3,425808
Турнирный алгоритм селекции						
ПК1	49:11:06	50	2,298067	0,883802	4,882955	2,634588
ПК2	18:25:57	13	2,362424	1,269812	5,194073	3,667147

Заключение

В данной работе была представлена модель восстановления данных на основе нечеткой системы, которая позволяет снять ограничения нормальности, полноты и однородности с входных данных, требуемых статистическими моделями. Были представлены нечеткая система, эволюционная стратегия и описана постановка задачи импутирования в интерпретации нечеткой системы. Также был описан программный комплекс, который позволяет решать задачу импутирования на основе данной модели.

Было проведено исследование работы модели и программного комплекса на данных о вязкопарафинистой нефти. И как было написано выше, многопоточное разделение операций нашей модели позволило снизить как вычислительное время, так и загрузку процессора ПК в 2,7 и 3,8 раз соответственно, что является важным критерием при обработке данных большого объема.

Список литературы

1. **Ходашинский И. А., Гнездилова В. Ю., Дудин П. А., Лавыгина А. В.** Основанные на производных и метаэвристические методы идентификации параметров нечетких моделей // Труды VIII международной конференции "Идентификация систем и задачи управления". SICPRO '08. Москва, 2009.
2. **Рутковская Д., Пилиньский М., Рутковский Л.** Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия — Телеком, 2006. 383 с.
3. **Hoche S., Wrobel S.** A Comparative Evaluation of Feature Set Evolution Strategies for Multinational Boosting // Proc. 13th Int. Conf. on ILP, 2003.
4. **Лучкова С. О.** Идентификация нечеткой системы методом эволюционной стратегии // Сб. трудов Всероссийского конкурса научно-исследовательских работ студентов и аспирантов в области информатики и информационных технологий в рамках Всероссийского фестиваля науки. — Белгород, 2011. С. 92—101.
5. **Шилдт Г.** C#: полное руководство.: пер. с англ. М.: Вильямс, 2011. 1056 с.
6. **Vision** Studio 2010 (EN): официальная страница URL: http:// www.microsoft.com/visualstudio/en-us/products/2010/default.mspx
7. **Загоруйко Н. Г.** Методы распознавания и их применение. М.: Сов. радио, 1972. 216 с.
8. **Schwefel H.-P.** Numerical Optimization of Computer Models. John Wiley & Sons, 1981.
9. **Schwefel H.-P.** Evolution and Optimum Seeking. New York: John Wiley & Sons, 1995.
10. **Дудин П. А., Горбунов И. В., Боровков А. В.** Унифицированное представление параметров нечеткой системы // Материалы докладов Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых "Научная сессия ТУСУР-2011". Томск: В-Спектр, 2010. Ч. 2. С. 168—170.

УДК 004.3

Н. И. Червяков[1], д-р техн. наук, проф., зав. каф.,
М. С. Афонин[2], аспирант,
М. Г. Бабенко[1], канд. физ.-мат. наук, мл. науч. сотр.,
П. А. Ляхов[1], аспирант
[1] ФГБОУ ВПО "Ставропольский государственный университет"
[2] ФГАОУ ВПО "СКФУ"

Аналитический обзор методов и алгоритмов распараллеливания арифметических операций с точками эллиптической кривой на основе нейросетевого подхода

Представлен подход к построению нейронных сетей для организации аппаратного или программного ядра криптографической системы на основе эллиптических кривых. Представлены результаты, доказывающие целесообразность распараллеливания базовых операций эллиптической криптографии посредством нейронных сетей высоких порядков.

Ключевые слова: *нейронная сеть, криптография, эллиптическая кривая*

Введение

Преимуществом вычислений над группой точек эллиптической кривой для криптографии является размер секретного ключа, который на порядок меньше ключа криптографических схем, основанных на вычислениях над конечными полями (RSA, схема Эль—Гамаля). Однако алгоритмы выполнения арифметических операций над группой точек эллиптической кривой требуют ускорения при программной или аппаратной реализации. Основная операция эллиптической криптографии (умножение точки эллиптической кривой на скаляр) может быть представлена в виде последовательности операций сложения и удвоения точек. Важной задачей является ускорение базовых данных примитивов эллиптической арифметики. В работе представлен подход, использующий как известный принцип поиска оптимальных для ускорения вычислений проективных координат, так и совершенно новый подход, связанный с использованием нейронных сетей высоких порядков. Показано, что нейронные сети высоких порядков подходят для распараллеливания операций эллиптической криптографии, причем выбор порядка не является тривиальной задачей. Ускорение операций за счет массового параллелизма нейронных сетей позволит использовать эллиптические кривые для защиты информации в реальном масштабе времени.

Эллиптическая криптография

Операции над точками эллиптической кривой определяются над полем Галуа $GF(p)$, где p — простое число ($p > 3$). Все арифметические операции выполняются по модулю p. Эллиптическая кривая E задается выражением $y^2 = x^3 + ax + b$, где $4a^2 + 27b^2 \neq 0$ и $x, y, a, b \in GF(p)$. Также существует один элемент, называемый бесконечной точкой и обозначаемый символом *"O"*, который выступает в роли аддитивной единицы:

$$\forall P(x, y) \in E: P + O = P.$$

Для сложения точек эллиптической кривой существуют следующие правила:

- $O = -O$,
- $P(x, y) + O = P(x, y)$,
- $P(x, y) + P(x, -y) = O$.

Операция сложения двух произвольных точек эллиптической кривой выполняется следующим образом:

$$P(x_1, y_1) + P(x_2, y_2) = P(x_3, y_3),$$

где $x_3 = \lambda^2 - x_1 - x_2$; $y_3 = \lambda(x_1 - x_3) - y_1$; $\lambda = (y_2 - y_1)/(x_1 - x_2)$ — угловой коэффициент касательной к эллиптической кривой, используемой для определения результата сложения.

Операция удвоения точки:

$$P(x_1, y_1) + P(x_1, y_1) = P(x_3, y_3),$$

где $x_3 = \lambda^2 - 2x_1$; $y_3 = \lambda(x_1 - x_3) - y_1$; $\lambda = (3(x_1)^2 + a)/(2y_1)$.

Умножение точки $P(x, y) \in E$ на скаляр k над $GF(p)$ определяется серией сложений:

$$Q = [k]P = \underbrace{P + P + \ldots + P}_{k \text{ раз}}.$$

В табл. 1 представлено число модулярных операций в случае использования аффинных координат [6].

Основной процесс шифрования заключается в том, что открытое сообщение m преобразуется в

Таблица 1

Число арифметических операций, требуемых для сложения и удвоения точек эллиптической кривой в аффинных координатах

Операции по модулю p	Сложение точек	Удвоение точек
Сложение (С)	6	4
Умножение (У)	3	4
Мультипликативная инверсия числа (И)	1	1
Общее число	6С + 3У + 1И	4С + 4У + 1И

Таблица 2

Число арифметических операций, требуемых для сложения и удвоения точек эллиптической кривой в координатах $P(X, Y, Z)$, соответствующих $P(X/Z^2, Y/Z^3)$

Операции сложения точек	Сложность	Операции удвоения точки	Сложность
$\lambda_1 = x_1 z_2^2$, $\lambda_2 = x_2 z_1^2$	4У	$\lambda_1 = 3x_1^2 + a z_1^4$, $z_3 = 2y_1 z_1$	5У + 1С
$\lambda_3 = \lambda_1 - \lambda_2$, $\lambda_4 = y_1 z_2^3$	2У + 1С	$\lambda_2 = 4X_1 Y_1^2$, $x_3 = \lambda_1^2 - 2\lambda_2$	3У + 1С
$\lambda_5 = y_2 z_1^3$, $\lambda_6 = \lambda_4 - \lambda_5$	2У + 1С	$\lambda_3 = 8y_1^4$, $\lambda_4 = \lambda_2 - 2x_3$	1У + 1С
$\lambda_7 = \lambda_1 + \lambda_2$, $\lambda_8 = \lambda_4 + \lambda_5$	2С	$y_3 = \lambda_1\lambda_4 - \lambda_3$	1У + 1С
$z_3 = z_1 z_2 \lambda_3$, $x_3 = \lambda_6^2 - \lambda_7\lambda_3^2$	5У + 1С	Итого	10У + 4С
$\lambda_9 = \lambda_7\lambda_3^2 - 2x_3$, $y_3 = (\lambda_9\lambda_6 - \lambda_8\lambda_3^3)/2$	3У + 2С		
Итого	16У + 7С		

Таблица 3

Число арифметических операций, требуемых для сложения и удвоения точек эллиптической кривой в координатах $P(X, Y, Z)$, соответствующих $P(X/Z, Y/Z)$

Операции сложения точек	Сложность	Операции удвоения точки	Сложность
$\lambda_1 = x_1 z_2$, $\lambda_2 = x_2 z_1$	2У	$\lambda_1 = 3x_1^2 + a z_1^2$, $\lambda_2 = y_1 z_1$	3У + 1С
$\lambda_3 = \lambda_2 - \lambda_1$, $\lambda_4 = y_1 z_2$	1У + 1С	$\lambda_3 = x_1 y_1 \lambda_2$, $\lambda_4 = \lambda_1^2 - 8\lambda_3$	3У + 1С
$\lambda_5 = y_2 z_1$, $\lambda_6 = \lambda_5 - \lambda_4$	1У + 1С	$x_3 = 2\lambda_4\lambda_2$, $y_4 = \lambda_1(4\lambda_3 - \lambda_4) - 8(y_1\lambda_2)^2$	4У + 2С
$\lambda_7 = \lambda_1 + \lambda_2$, $\lambda_8 = \lambda_6^2 z_1 z_2 - \lambda_3^2\lambda_7$	5У + 2С	$z_3 = 8\lambda_2^3$	2У
$z_3 = z_1 z_2 \lambda_3^3$, $x_3 = \lambda_8\lambda_3$	3У	Итого	12У+4С
$\lambda_9 = \lambda_3^2 x_1 z_2 - \lambda_8$, $y_3 = \lambda_9\lambda_6 - \lambda_3^3 y_1 z_2$	3У + 2С		
Итого	17У + 6С		

точку P_m на эллиптической кривой. Соответствующий шифротекст включает пару точек:

$$C_m = \{kG,\ P_m + kP_B\},$$

где k — случайное положительное целое число, сгенерированное источником сообщений A; G — базовая точка (генератор); $P_B = n_B G$ — открытый ключ места назначения сообщений B и n_B — секретный ключ места назначения B.

Процесс дешифрования можно описать выражением

$$C_m\{P_m + kP_B\} - n_B(kG) = P_m + kn_B G - n_B kG = P_m.$$

В основном время на выполнение шифрования затрачивается на выполнение умножения kG и kP_B. Поэтому увеличение скорости шифрования/дешифрования связано с уменьшением времени на операции умножения точки на скаляр.

Нейросетевой метод ускорения эллиптической криптографии

Наиболее трудоемкой операцией из сигнатуры {С, У, И} над $GF(p)$ является инверсия по модулю, которую можно исключить, перейдя к другим координатам. В работе [1] представлен сравнительный анализ сложности операции умножения точки эллиптической кривой на скаляр, реализованной в различных координатах: проективных, Якобиана, модификациях координат Якобиана, Чудновского и др.

Рассмотрим две формы проективных координат:

- проективные координаты $P(X, Y, Z)$, эквивалентные аффинным координатам $P(X/Z^2, Y/Z^3)$;
- проективные координаты $P(X, Y, Z)$, эквивалентные аффинным координатам $P(X/Z, Y/Z)$.

В табл. 2 представлены характеристики операций над точками в проективных координатах первой формы, а в табл. 3 — второй формы. Операции в таблицах представлены так, чтобы было видно независимые ветви алгоритма.

Из табл. 2 и 3 видно, что наибольший вес в базовых операциях эллиптической криптографии имеет умножение по модулю p (характер поля Галуа). Кроме того, все операции соответствуют сигнатуре конечного кольца, что позволяет использовать уже известные способы распараллеливания алгоритмов, например, нейронные сети. Для демонстрации данной возможности в табл. 2 и 3 операции были расположены в порядке появления независимых ветвей алгоритма, начиная с верхнего столбца. Кроме того, содержимое табл. 2 и 3 позволяет определить число арифметических операций по модулю для последовательного исполнения базовых операций при умножении точки на скаляр $k = \sum_{j=0}^{l-1} k_j 2^j$, $k_j \in \{0, 1\}$. Поскольку трудоемкость операции умножения чисел большой разрядности по модулю значительно превышает трудоемкость операции сложения по моду-

лю, то сравнительную оценку будем проводить без учета аддитивной операции. Тогда для первой формы потребуется

$$S_{\text{послед}}^{P(X/Z^2,\ Y/Z^3)} = l(10\text{У} + 16\text{У}) \text{ операций},$$

для второй формы

$$S_{\text{послед}}^{P(X/(Z),\ Y/Z)} = l(12\text{У} + 15\text{У}) \text{ операций},$$

где l — разрядность скаляра k в двоичном представлении.

Известное m-арное ($m = 2^r$) последовательное представление скаляра [2] позволяет сократить число операций для первой формы в

$$\max\left(\frac{S_{\text{послед}}^{P(X/Z^2,\ Y/Z^3)}}{S_{\text{послед.}m\text{-арное}}^{P(X/Z^2,\ Y/Z^3)}}\right) = \max\left(\frac{26\text{У}}{r \times 10\text{У} + 16\text{У}}\right) \approx 2{,}6 \text{ раз},$$

для второй формы

$$\max\left(\frac{S_{\text{послед}}^{P(X/(Z),\ Y/Z)}}{S_{\text{послед.}m\text{-арное}}^{P(X/(Z),\ Y/Z)}}\right) = \max\left(\frac{27\text{У}}{r \times 12\text{У} + 15\text{У}}\right) \approx 2{,}25 \text{ раз}.$$

Получение более высоких показателей производительности возможно с помощью распараллеливания алгоритма. В данной задаче себя хорошо зарекомендовали формируемые нейронные сети, позволяющие использовать принципы массового параллелизма [3]. Для построения нейронных сетей данного типа требуется перевести задачу в нейросетевой логический базис, что является нетривиальной задачей.

Известно [4], что вычислительный процесс над $GF(p)$ переносится на нейроподобную вычислительную архитектуру, которая получила название "нейронные сети конечного кольца" (НСКК). Простое арифметическое расширение $GF(p^n)$ также реализуется на базе НСКК [5]. Однако вычисления над группой точек эллиптической кривой с использованием только базиса НСКК реализовать тяжело, поскольку базовые операции эллиптической арифметики включают умножение значения сигнала на значение сигнала, а не только на весовой коэффициент связи нейронов.

Формальные нейроны, лежащие в основе НСКК, относятся к нейронам первого порядка. На выходе такого нейрона результат нелинейного преобразования взвешенной суммы входных сигналов

$$y = f\left(\sum_{i=1}^{n} w_i x_i\right),$$

где y — сигнал на выходе формального нейрона; x_i — сигнал на входе нейрона; w_i — весовой коэффициент; f — функция активации нейрона.

Существуют нейроны высоких порядков, к которым относятся Sigma-Pi-сети, Pi-Sigma-сети, сети с функциональными связями, нейронные сети второго порядка, нейронные сети с нейронами-умножителями. Общим для сетей высоких порядков является наличие в формальной модели не только операции взвешенного сложения сигналов, но и операции взвешенного перемножения сигналов:

$$y_N = f\left(\sum_{i=0}^{n}\sum_{j=i}^{n}\sum_{l=j}^{n}\sum_{m=1}^{n} \ldots\ldots \sum_{y=x}^{n}\sum_{z=y}^{n} w_{ijlm\ldots z} x_i x_j x_l x_m x \ldots\ldots x_z\right).$$

Нейрон первого порядка

Нейрон второго порядка

Нейрон третьего порядка

Нейрон N-го порядка

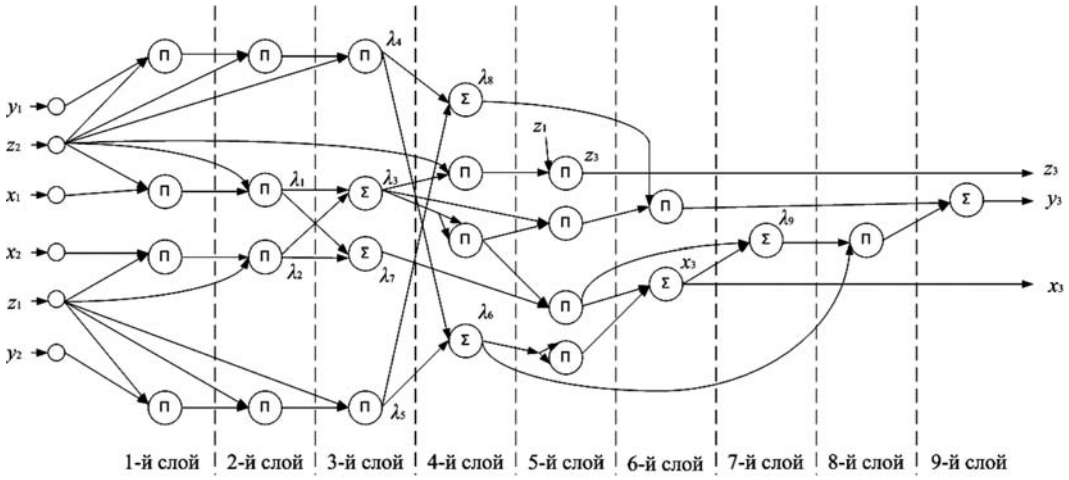


Рис. 1. Нейронная сеть второго порядка для сложения точек эллиптической кривой в проективной форме, соответствующей $P(X/Z^2, Y/Z^3)$ в аффинных координатах

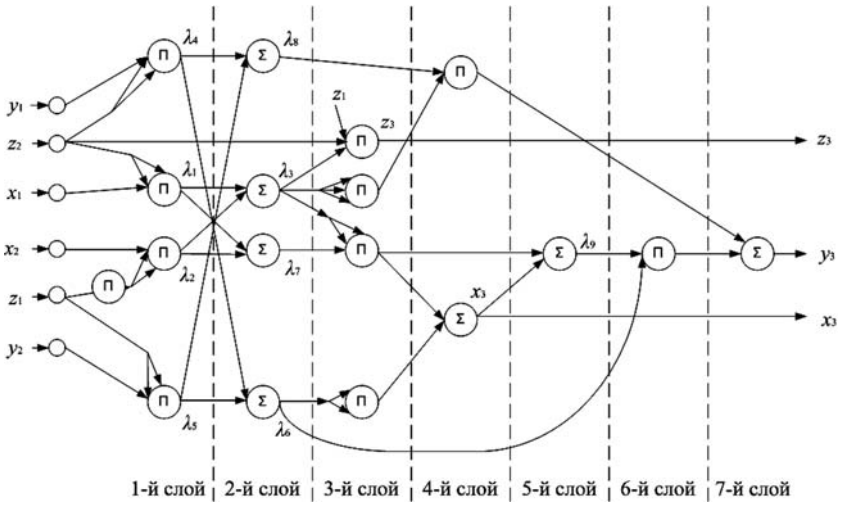


Рис. 2. Нейронная сеть третьего порядка для сложения точек эллиптической кривой в проективной форме, соответствующей $P(X/Z^2, Y/Z^3)$ в аффинных координатах

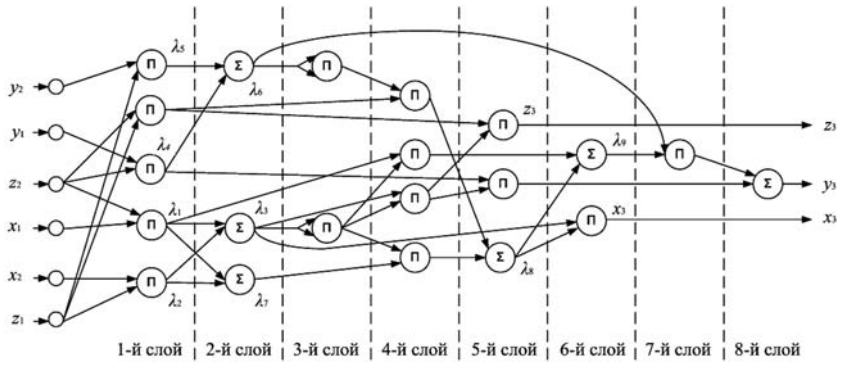


Рис. 3. Нейронная сеть второго порядка для сложения точек эллиптической кривой в проективной форме, соответствующей $P(X/Z, Y/Z)$ в аффинных координатах

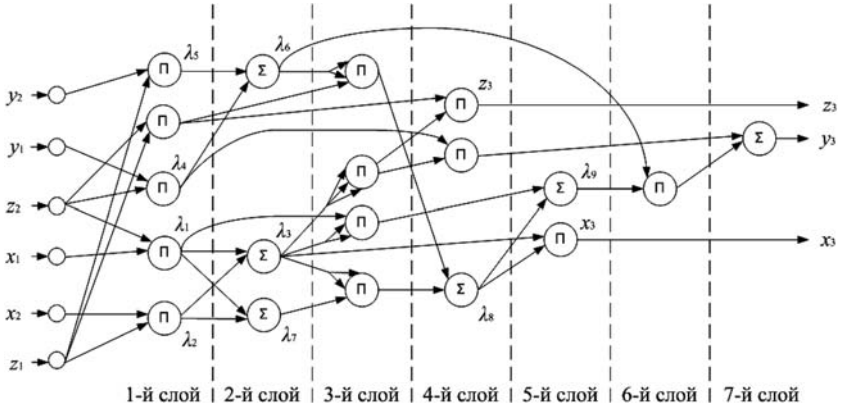


Рис. 4. Нейронная сеть третьего порядка для сложения точек эллиптической кривой в проективной форме, соответствующей $P(X/Z, Y/Z)$ в аффинных координатах

Таблица 4

Требуемые для операции сложения точек эллиптической кривой затраты на реализацию нейронных сетей высоких порядков

Тип формального нейрона	Число нейронов для первой формы		Число нейронов для второй формы	
	НС второго порядка	НС третьего порядка	НС второго порядка	НС третьего порядка
Нейроны-умножители Нейроны-сумматоры	18 7	10 7	15 6	13 6

Исходя из табл. 1 и 2 необходимо использовать нейронную сеть с порядком, равным или выше 2, поскольку максимальная степень, встречающаяся при вычислениях, равна 4 в случае первой проективной формы и равна 3 в случае второй проективной формы. При использовании нейронной сети второго порядка увеличивается число слоев, но уменьшается сложность реализации нейрона, в противоположность предельному случаю четвертого порядка.

На рис. 1 представлена сформированная в соответствии с табл. 1 нейронная сеть второго порядка для выполнения операции сложения точек. Нейронная сеть третьего порядка для первой формы (см. табл. 2) представлена на рис. 2. Кроме уменьшения числа слоев при увеличении порядка нейронов уменьшается число синаптических межслойных связей, что увеличивает надежность и упрощает аппаратную реализацию нейронной сети. Весовые коэффициенты нейронных сетей, изображенных на рис. 1 и 2, могут быть определены из табл. 2 посредством сопоставления строк данной таблицы со слоями нейронных сетей. На рис. 3 и 4 представлены нейронные сети второго и третьего порядков, соответственно, для второй формы проективных координат (табл. 3).

Нейроны с буквой "П" — нейроны взвешенного произведения, нейроны с буквой "Σ" — нейроны взвешенного сложения. Нейроны сети второго порядка имеют два входа, когда нейроны сети третьего порядка имеют три входа. Каждый нейрон выполняет операцию по модулю p.

Нейронная сеть второго и третьего порядков для второй формы (см. табл. 2) представлена на рис. 1 и 2 соответственно. Видно, что использование второй проективной формы для нейронной сети второго порядка позволяет сократить общее число нейронов, межнейронных связей и число слоев по сравнению с реализацией сети на базе первой проективной формы.

В табл. 4 представлены сравнительные затраты на реализацию нейронных сетей для сложения точек эллиптической кривой.

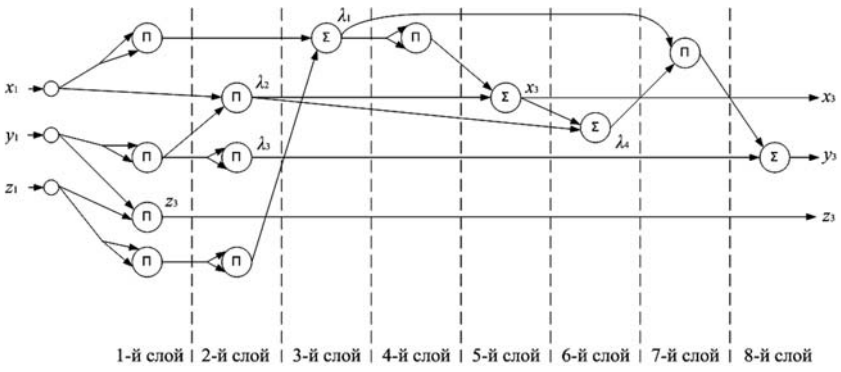


Рис. 5. Нейронная сеть второго порядка для удвоения точек эллиптической кривой в проективной форме, соответствующей $P(X/Z^2, Y/Z^3)$ в аффинных координатах

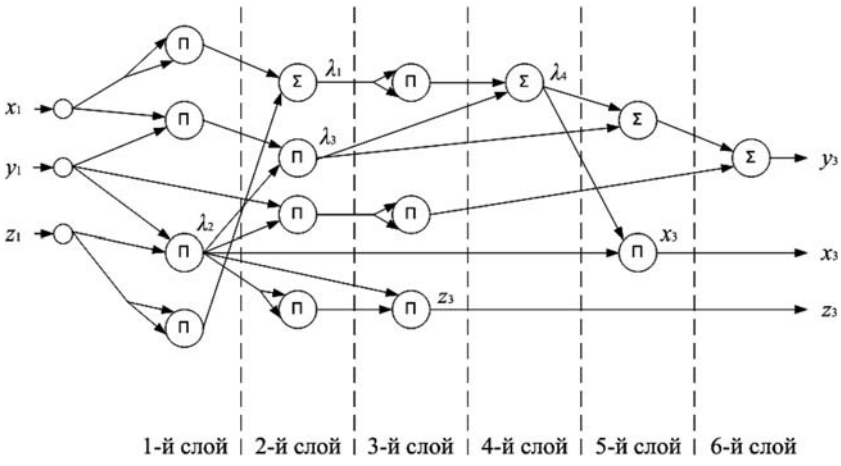


Рис. 6. Нейронная сеть второго порядка для удвоения точек эллиптической кривой в проективной форме, соответствующей $P(X/Z, Y/Z)$ в аффинных координатах

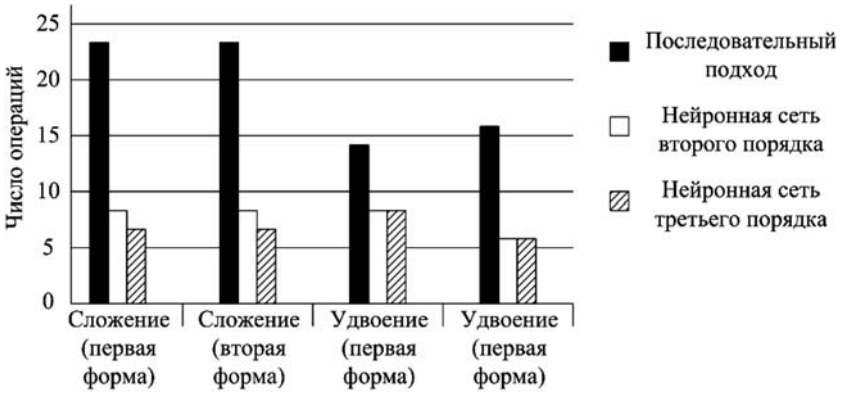


Рис. 7. Сравнение последовательного и параллельного подходов к реализации базовых операций эллиптической криптографии

Таблица 5

Требуемые для операции удвоения точек эллиптической кривой затраты на реализацию нейронных сетей второго порядка

Тип формального нейрона	Число нейронов для первой формы	Число нейронов для второй формы
Нейроны-умножители	9	11
Нейроны-сумматоры	4	4

Операция удвоения точки также можно реализовать в базисе нейронных сетей высоких порядков (рис. 5, 6). Однако использование нейронных сетей выше второго порядка не целесообразно, поскольку ведет к неэффективным аппаратурным затратам: увеличивается сложность нейронов без уменьшения числа слоев.

Требуемые для операции удвоения точек эллиптической кривой затраты на реализацию нейронных сетей второго порядка приведены в табл. 5.

Каждый слой представленных нейронных сетей соответствует одной операции по модулю p. На диаграмме рис. 7 представлены сравнение сложности последовательного и параллельного подходов.

Таким образом, полученные нейросетевые вычислительные структуры высоких порядков обеспечивают ускорение базовых операций эллиптической криптографии. Усложнение отдельного нейрона при переходе к сетям высокого порядка не является критерием отказа от ее использования. Если сравнивать со структурой нейронных сетей первого порядка, которые имеют десятки и сотни связей с предыдущими нейронами, то выбор в пользу нейронных сетей высокого порядка будет однозначен.

Список литературы

1. **Bernstein D. J., Lange T.** Performance evaluation of a new coordinate system for elliptic curves. URL: http://cr.yp.to/newelliptic/newelliptic-20070522.pdf. 2007. 17 p.

2. **Blake I., Seroussi G., Smart N.** Elliptic curves in cryptography. Cambridge University Press: New York, 1999.

3. **Галушкин А. И.** Нейронные сети. Основы теории. М.: Горячая Линия Телеком, 2010. 480 с.

4. **Червяков Н. И., Сахнюк П. Л., Шапошников Л. В., Макоха А. Н.** Нейрокомпьютеры в остаточных классах. Кн. 11: учеб. пособие для вузов. М.: Радиотехника, 2003. 272 с.

5. **Калмыков И. А.** Математические модели нейросетевых отказоустойчивых вычислительных средств, функционирующих в полиномиальной системе классов вычетов / Под ред. Н. И. Червякова. М.: ФИЗМАТЛИТ, 2005. 276 с.

6. **Ростовцев А. Г., Маховенко Е. Б.** Теоретическая криптография. СПб.: НПО "Профессионал", 2004. 480 с.

УДК 004.93

О. П. Архипов, канд. техн. наук, директор,
З. П. Зыкова, канд. физ.-мат. наук, зав. лаб.,
ОФ ИПИ РАН,
e-mail: arkhipov@yandex.ru

Коррекция детализации представлений *RGB*-изображений на периферийных устройствах ПЭВМ

Рассматривается задача коррекции детализации представлений RGB-изображений на мониторе и в отпечатках принтера на основе цифрового описания (RGB-характеризации) цветовосприятия пользователя ПЭВМ. Приводится алгоритм для решения этой задачи. Обсуждаются некоторые особенности реализации этого алгоритма и приложения полученных результатов, связанные с определением персонифицированных предварительных преобразований цветных изображений в целях улучшения детализации их представлений на периферийных устройствах ПЭВМ. Приведены иллюстрирующие примеры.

Ключевые слова: *Lab-контраст, градации, детализация, тоновоспроизведение*

Введение

В полиграфии для улучшения восприятия отпечатков изображений применяют коррекцию их детализации с помощью процедуры, параметры которой определяются на основе данных, полученных при определении градаций на тестовых тоновых шкалах и формировании соответствующих равноконтрастных зависимостей. Это уменьшает потери "в тенях" и "в светах", улучшает плавность и точность тоновоспроизведения. Коррекция выполняется в интересах массового потребителя, поэтому при расчетах используется модель "стандартного наблюдателя", с помощью которой учитываются особенности цветовосприятия большинства наблюдателей.

В рамках данной работы аналогичный подход применяется для решения задачи персонифицированной (в соответствии с цветовосприятием произвольного наблюдателя) коррекции детализации отображения цветных изображений — их визуального представления на периферийных устройствах ПЭВМ. Коррекция выполняется с помощью предварительных преобразований цветных изображений.

Параметры преобразований

Параметры преобразований определяются на основе данных, полученных при обработке результатов визуального или виртуального градационного тестирования цветовосприятия произвольным пользователем отображений пикселей тоновых шкал. При визуальном тестировании в качестве градаций выбирают пиксели, отображения которых имеют минимальные визуальные различия, а при виртуальном (*Lab*-тестировании) — пиксели, цифровое описание отображений которых удовлетворяет стандартному *Lab*-критерию цветоразличия. В любом случае последовательность градаций — это последовательность пикселей, в которой соседние компоненты имеют приблизительно одинаковый контраст.

При градационных преобразованиях изменяется плотность размещения градаций на различных фрагментах тоновых шкал. Это влечет аналогичное изменение детализации отображений тоновых шкал и соответствующих фрагментов общих изображений.

Например, при равноконтрастных градационных преобразованиях градации распределяются равномерно, что влечет приближенно равномерное соотношение контрастов отображений соседних пикселей преобразованных тоновых шкал [1].

Большей (меньшей) плотности градаций между двумя произвольными пикселями тоновой шкалы соответствует больший (меньший) контраст между их отображениями.

Различные способы определения градаций приводят к результатам, имеющим различную степень персонификации. Результаты, полученные на основе визуального тестирования, соответствуют цветовосприятию пользователя в большей степени, чем результаты, полученные при *Lab*-тестировании.

При визуальном тестировании зависимость контраста от плотности градаций более предсказуема, а функциональные зависимости предварительных градационных преобразований в большей степени переносятся на контрастные функциональные зависимости фрагментов отображений преобразованных цветных изображений.

Однако на практике большие временны́е, а в случае печатающих устройств, и другие ресурсные затраты на реализацию визуального тестирования могут оказаться неприемлемыми. В связи с этим в рамках данной работы рассматриваются предварительные преобразования изображений, функции

которых определяются по данным, полученным при *Lab*-градационном тестировании.

Предполагается, что компьютерная система пользователя (наблюдателя) функционирует при фиксированных условиях цветовоспроизведения на периферийном устройстве. Обозначим F функцию цветовосприятия наблюдателем отображения *RGB*-пикселей на каком-либо периферийном устройстве ПЭВМ. Аргументами этой функции являются пиксели из *RGB*-пространства, а значениями — пиксели из цветового пространства наблюдателя.

Обозначим Ψ функцию *RGB*-характеризации цветовосприятия [2—5]. Поскольку линии уровня функции Ψ приближенно совпадают с соответствующими зонами толерантности в цветовом пространстве наблюдателя (линиями уровня функции F), то, используя функцию Ψ, можно получить цифровое описание восприятия отображения исходного изображения $\Psi(Img)$.

Пусть последовательность пикселей $\{x_j\} \subset Img$ представляет некоторый фрагмент изображения, а $x_{j'}$ и $x_{j''}$ — произвольные пиксели этой последовательности. Если известны *Lab*-координаты пикселей $(L_{j'}, a_{j'}, b_{j'})$, $(L_{j''}, a_{j''}, b_{j''})$, то в качестве цифрового описания их контраста можно использовать величину $E(x_{j'}, x_{j''})$,

$$E(x_{j'}, x_{j''}) = \sqrt{(L_{j'} - L_{j''})^2 + (a_{j'} - a_{j''})^2 + (b_{j'} - b_{j''})^2}.$$

Детализацию отображения всего фрагмента можно описать с помощью двумерной последовательности контрастов $\{E(\Psi(x_{j'}), \Psi(x_{j''}))\}$.

Пусть M — множество пикселей *RGB*-куба, координаты которых кратны семнадцати, $\{T_i\}$ — тестовая совокупность ступенчатых тоновых шкал, проходящих через вершины *RGB*-куба, начиная с черного пикселя (0, 0, 0) и заканчивая белым пикселем (255, 255, 255) [1],

$$T_i = t_{i,j},\ t_{i,j} \in M,\ i = 0, 1, 2, ..., 6,\ j = 0, 1, ..., J_i^T,$$

где T_0 — Gray-шкала; T_1 — Red-шкала; T_2 — Yellow-шкала; T_3 — Green-шкала; T_4 — Cyan-шкала; T_5 — Blue-шкала; T_6 — Magenta-шкала.

Для каждой тоновой шкалы T_i определим ее носитель S_i — совокупность всех *RGB*-пикселей, последовательно расположенных от начала до конца тоновой шкалы на ломаной линии, соединяющей компоненты тоновой шкалы:

$$\{T_i\} \subset \{S_i\} = \{s_{i,j}\} \subset \bigcup_{m=0}^{J_i^T - 1} [t_{i,m}, t_{i,m+1}],\ j = 0, 1, ..., J_i^S.$$

Обозначим $e(x_{j'}, x_{j''})$ функцию описания восприятия контраста вида

$$e(x_{j'}, x_{j''}) = E(\Psi(x_{j'}), \Psi(x_{j''})).$$

Пусть f_l — функции преобразования носителей тоновых шкал, при которых первый и последний пиксели ступенчатых тоновых шкал и их носителей неподвижны,

$$t_{i,0} = s_{i,0} = f_l(s_{i,0}),\ t_{i,J_i^T} = s_{i,J_i^S} = f_l(s_{i,J_i^S}),$$

а распределение контрастов на ступенчатых тоновых шкалах удовлетворяет следующей системе уравнений:

$$\frac{e(f_l(t_{i,j}), f_l(t_{i,j+1}))}{\sum_{m=0}^{J_i^T - 1} e(f_l(t_{i,m}), f_l(t_{i,m+1}))} = \frac{\tau_{l,i}(t_{i,j}, t_{i,j+1})}{\sum_{m=0}^{J_i^T - 1} \tau_{l,i}(t_{i,m}, t_{i,m+1})}, \quad (1)$$

где $\tau_{l,i}$ — некоторые функции, значениями которых являются положительные числа.

Графики функций $\tau_{l,i}(t_{i,j}, t_{i,j+1})$ и $e(f_l(t_{i,j}), f_l(t_{i,j+1}))$ как функций от j, подобны, поскольку при любом j из (1) следует

$$\tau_{l,i}(t_{i,j}, t_{i,j+1}) = d_{l,i} \cdot e(f_l(t_{i,j}), f_l(t_{i,j+1})),$$

где

$$d_{l,i} = \frac{\sum_{m=0}^{J_i^T - 1} \tau_{l,i}(t_{i,m}, t_{i,m+1})}{\sum_{m=0}^{J_i^T - 1} e(f_l(t_{i,m}), f_l(t_{i,m+1}))}.$$

Следовательно, задавая вид графиков функций $\tau_{l,i}$, можно корректировать детализацию ступенчатых тоновых шкал, их носителей, а также соответствующих фрагментов общих цветных изображений.

Значения *RGB*-координат пикселей $t_{i,j}$, $f_l(t_{i,j})$ являются целыми числами. Уже по одной этой причине точное решение системы уравнений (1) невозможно. Для приближенного решения уравнений (1) могут быть предложены различные способы. В рамках данной работы предлагается алгоритм приближенного вычисления функций f_l с помощью градаций.

Вычисление градаций

Обозначим $\{G_i\}$ последовательности градаций, выбранных на носителях $\{S_i\}$ по стандартному критерию цветоразличия:

$$G_i = \{g_{i,j}\},\ j = 0, 1, ..., J_i^G,\ e(g_{i,j}, g_{i,j+1}) > E',$$
$$j = 0, 1, ..., J_i^G - 1,$$

где E' — некоторое положительное число, обычно равное пяти.

Значения градаций могут быть вычислены различными способами. Кратко опишем один из них. В качестве первой градации выберем первый компонент последовательности S_i:

$$g_{i,0} = s_{i,0}.$$

В качестве следующей градации следует выбрать первый (пусть $s_{i,j'}$) из тех компонентов $s_{i,j}$, $j \geqslant 1$, который удовлетворяет критерию цветоразличия:

$$g_{i,1} = s_{i,j'},\ e(g_{i,0},\ s_{i,j'}) > E'.$$

Далее из компонентов носителя $s_{i,j}$, $j > j'$, в качестве следующей градации также выбирается первый пиксель (пусть $s_{i,j''}$), удовлетворяющий критерию цветоразличия:

$$g_{i,2} = s_{i,j''},\ e(g_{i,1},\ s_{i,j''}) > E'.$$

Аналогичным образом можно получить все оставшиеся градации.

Если последняя выбранная таким образом градация совпадает с последним компонентом носителя s_{i,J_i^S}, то процесс определения градаций можно считать завершенным. Иначе, в зависимости от значения контраста следует либо увеличить число градаций, добавив к их совокупности s_{i,J_i^S}, либо оставить число градаций прежним, но заместить последнюю градацию пикселем s_{i,J_i^S}.

Вычисление функции преобразования с помощью градаций

Обозначим $\{s_{i,n(l,i,j)}\}$ последовательность пикселей, определяющих местоположение градаций $g_{i,j}$ на носителях $\{S_i\}$ после преобразования f_l. Потребуем, чтобы для $s_{i,n(l,i,j)}$ удовлетворялись соотношения, аналогичные соотношениям (1) для пикселей ступенчатых тоновых шкал:

$$\frac{e(g_{i,j},\ g_{i,j+1})}{\sum\limits_{m=0}^{J_i^G-1} e(g_{i,m},\ g_{i,m+1})} = \frac{\tau_{l,i}(s_{i,n(l,i,j)},\ s_{i,n(l,i,j+1)})}{\sum\limits_{m=0}^{J_i^G-1} \tau_{l,i}(s_{i,n(l,i,m)},\ s_{i,n(l,i,m+1)})}, \quad (2)$$

где

$$g_{i,j} = f_l(s_{i,n(l,i,j)}),\ j = 0,\ 1,\ ...,\ J_i^G - 1,$$
$$g_{i,0} = s_{i,0} = f_l(s_{i,0}),\ g_{i,J_i^G} = s_{i,J_i^S} = f_l(s_{i,J_i^S}).$$

В качестве функций $\tau_{0,i}$ выберем функцию определения расстояний в *RGB*-пространстве:

$$\tau_{0,i}(s_{i,j'},\ s_{i,j''}) = \rho(s_{i,j'},\ s_{i,j''}) = \sum_{m=j'}^{j''-1} \rho(s_{i,m},\ s_{i,m+1}).$$

Из соотношений (2) имеем

$$\frac{e(g_{i,j},\ g_{i,j+1})}{\sum\limits_{m=0}^{J_i^G-1} e(g_{i,m},\ g_{i,m+1})} = \frac{\rho(s_{i,n(0,i,j)},\ s_{i,n(0,i,j+1)})}{\sum\limits_{m=0}^{J_i^G-1} \rho(s_{i,n(0,i,m)},\ s_{i,n(0,i,m+1)})}. \quad (3)$$

Поскольку

$$\sum_{m=0}^{J_i^G-1} \rho(s_{i,n(0,i,m)},\ s_{i,n(0,i,m+1)}) = \rho(s_{i,0},\ s_{i,J_i^S}),$$

то

$$\rho(s_{i,n(0,i,j)},\ s_{i,n(0,i,j+1)}) = d'_{0,i} \cdot e(g_{i,j},\ g_{i,j+1}), \quad (4)$$

где

$$d'_{0,i} = \frac{\rho(s_{i,0},\ s_{i,J_i^S})}{\sum\limits_{m=0}^{J_i^G-1} e(g_{i,m},\ g_{i,m+1})}.$$

Пусть градации вычислены, тогда могут быть вычислены и значения $d'_{0,i}$. Это позволяет последовательно определить новое местоположение градаций. Так, при $j = 0$ из выражения (4) имеем уравнение с известной правой частью

$$\rho(s_{i,0},\ s_{i,n(0,i,1)}) = d'_{0,i} \cdot e(g_{i,0},\ g_{i,1}).$$

Последовательно при $j' = \{1,\ 2,\ ...\}$ вычисляя суммы

$$\rho(s_{i,0},\ s_{i,j'}) = \sum_{m=0}^{j'-1} \rho(s_{i,m},\ s_{i,m+1}),$$

можно найти такое значение j', при котором будет выполнено неравенство

$$\rho(s_{i,0},\ s_{i,j'}) \geqslant d'_{0,i} \cdot e(g_{i,0},\ g_{i,1}).$$

Полагая $n(0,\ i,\ 1) = j'$, из соотношения (2) уже при $j = 1$ имеем уравнение с известной правой частью:

$$\rho(s_{i,n(0,i,1)},\ s_{i,n(0,i,2)}) = d'_{0,i} \cdot e(g_{i,1},\ g_{i,2}).$$

Отсюда определяем $n(0,\ i,\ 2)$, а затем и все оставшиеся $n(0,\ i,\ j)$. Таким образом определяются пиксели $\{s_{i,n(0,i,j)}\}$ — последовательность аргументов, для которых определены значения f_0.

Значения функции преобразования f_0 от компонентов носителя S_i, расположенных между пикселями $s_{i,n(0,i,j)}$, могут быть вычислены обычным путем с помощью интерполяции. Вследствие этого для соседних пикселей из $\{s_{i,k(j)}\}$, $j = 0,\ 1,\ ...,\ J_i^T$ — не-

которых подпоследовательностей носителей S_i, приближенно выполняются соотношения, аналогичные соотношениям (3) для пикселей, определяющих местоположение градаций после преобразования, а именно

$$\frac{e(f_0(s_{i,\,k(j)}), f_0(s_{i,\,k(j+1)}))}{\sum\limits_{m=0}^{J_i^T-1} e(f_0(s_{i,\,k(m)}), f_0(s_{i,\,k(m+1)}))} \approx \frac{\rho(s_{i,\,k(j)}, s_{i,\,k(j+1)})}{\sum\limits_{m=0}^{J_i^T-1} \rho(s_{i,\,k(m)}, s_{i,\,k(m+1)})}. \quad (5)$$

Выполнение условий (5) для равномерных последовательностей означает, что с помощью функции f_0 выполняются *Lab*-равноконтрастные градационные преобразования. Вследствие неизбежных погрешностей точная равноконтрастность не достижима, но, как показывает практика, при рассмотренном преобразовании диапазон изменения *Lab*-контрастов на ступенчатых тоновых шкалах существенно (в 2—4 раза) сужается по сравнению с исходным диапазоном. Это влечет улучшение детализации и тоновоспроизведения отображений, тоновых шкал и соответствующих фрагментов общих цветных изображений.

Далее рассмотрим вспомогательную систему уравнений вида

$$\frac{\tau_{l,\,i}(g'_{l,\,i,\,j}, g'_{l,\,i,\,j+1})}{\sum\limits_{m=0}^{J_{l,\,i}^{G'}-1} \tau_{l,\,i}(g'_{l,\,i,\,m}, g'_{l,\,i,\,m+1})} = \frac{\rho(s'_{i,\,n(l,\,i,\,j)}, s'_{i,\,n(l,\,i,\,j+1)})}{\sum\limits_{m=0}^{J_{l,\,i}^{G'}-1} \rho(s'_{i,\,n(l,\,i,\,m)}, s'_{i,\,n(l,\,i,\,m+1)})}, \quad (6)$$

где $\{g'_{l,\,i,\,j}\}, j = 0, 1, ..., J_{l,\,i}^{G'}$ — последовательности градаций, выбранных на носителях $\{S_i\}$ по критерию

$$\tau_{l,\,i}(g'_{l,\,i,\,j},\ g'_{l,\,i,\,j+1}) > E'', j = 0,\ 1,\ ...,\ J_{l,\,i}^{G'} - 1,$$

$$g'_{l,\,i,\,j} = \varphi_l(s'_{i,\,n(l,\,i,\,j)}), j = 0,\ 1,\ ...,\ J_{l,\,i}^{G'} - 1,$$

$$g'_{l,\,i,\,0} = \varphi_l(s_{i,\,0}),\ g'_{i,\,J_{l,\,i}^{G'}} = \varphi_l(s_{i,\,J_i^S}),$$

$$E'' = E' \cdot \frac{\sum\limits_{j=0}^{J_i^S-1} \tau_{l,\,i}(s_{i,\,j}, s_{i,\,j+1})}{\sum\limits_{j=0}^{J_i^S-1} e(s_{i,\,j}, s_{i,\,j+1})}.$$

После определения функции φ_l из (6) способом, аналогичным способу определения f_0 из выражения (3), можем утверждать справедливость соотношений, аналогичных соотношениям (5):

$$\frac{\tau_{l,\,i}(\varphi_l(s_{i,\,k(j)}), \varphi_l(s_{i,\,k(j+1)}))}{\sum\limits_{m=0}^{J_i^T-1} \tau_{l,\,i}(\varphi_l(s_{i,\,k(m)}), \varphi_l(s_{i,\,k(m+1)}))} \approx \frac{\rho(s_{i,\,k(j)}, s_{i,\,k(j+1)})}{\sum\limits_{m=0}^{J_i^T-1} \rho(s_{i,\,k(m)}, s_{i,\,k(m+1)})}. \quad (7)$$

Из соотношений (5) и (7) имеем

$$\frac{e(f_0(s_{i,\,k(j)}), f_0(s_{i,\,k(j+1)}))}{\sum\limits_{m=0}^{J_i^T-1} e(f_0(s_{i,\,k(m)}), f_0(s_{i,\,k(m+1)}))} \approx \frac{\tau_{l,\,i}(\varphi_l(s_{i,\,k(j)}), \varphi_l(s_{i,\,k(j+1)}))}{\sum\limits_{m=0}^{J_i^T-1} \tau_{l,\,i}(\varphi_l(s_{i,\,k(m)}), \varphi_l(s_{i,\,k(m+1)}))}. \quad (8)$$

Обозначим f_l функцию, удовлетворяющую условиям вида

$$f_0(s_{i,\,j}) = f_l(\varphi_l(s_{i,\,j})),$$

и пусть последовательность $\{s_{i,\,k(j)}\}$ такова, что

$$t_{i,\,j} = \varphi_l(s_{i,\,k(j)}),\ t_{i,\,j+1} = \varphi_l(s_{i,\,k(j+1)}),$$

тогда из (8) получаем

$$\frac{e(f_l(t_{i,\,j}), f_l(t_{i,\,j+1}))}{\sum\limits_{m=0}^{J_i^T} e(f_l(t_{i,\,m}), f_l(t_{i,\,m+1}))} \approx \frac{\tau_{l,\,i}(t_{i,\,j}, t_{i,\,j+1})}{\sum\limits_{m=0}^{J_i^T} \tau_{l,\,i}(t_{i,\,m}, t_{i,\,m+1})}.$$

Таким образом, построена искомая функция f_l, определенная на всей совокупности ступенчатых тоновых шкал, с помощью которой распределение контрастов на тоновых шкалах является приближенно подобным распределению значений функции $\tau_{l,\,i}$.

Согласованное градационное преобразование *RGB*-куба

Пусть известна f_l — функция градационного преобразования тоновых шкал T_i и их носителей S_i. Для определения значения этой функции от произвольного пикселя *RGB*-куба $P_0 = (r_0, g_0, b_0)$, не принадлежащего множеству шкал $\{S_i\}$, через точку P_0 проведем плоскость перпендикулярно серой шкале (Gray-шкале). Координаты точки $P_1 = \{r_1, g_1, b_1)$

пересечения этой плоскости и соответствующей диагонали *RGB*-куба равны друг другу, причем

$$r_1 = g_1 = b_1 = \frac{r_0 + g_0 + b_0}{3}.$$

Обозначим через P_2 и P_3 такие точки пересечения проведенной плоскости и ломаных линий тоновых шкал, которые и являются ближайшими к P_2 и составляют вместе с точкой P_1 невырожденный треугольник $\Delta P_1P_2P_3$, содержащий P_0.

Для каждой из точек P_j, $j \in \{1, 2, 3\}$, существует такая пара точек P'_j и P''_j на соответствующем носителе $S_{i(j)}$, что $P_j \in [P'_j, P''_j]$. Поскольку значения $f_l(P'_j)$ и $f_l(P''_j)$ известны, то с помощью интерполяции могут быть вычислены и значения $f_l(P_j)$.

Обозначим через A_l матрицу размером $[3 \times 3]$ линейного преобразования треугольника $\Delta P_1P_2P_3$ в треугольник $\Delta f_l(P_1)f_l(P_2)f_l(P_3)$. Заметим, что в силу невырожденности треугольника $\Delta P_1P_2P_3$ матрица A_l существует, единственна и может быть вычислена по известным алгоритмам из системы девяти уравнений с девятью неизвестными (компонентами матрицы A_l):

$$A_lP_j = f_l(P_j),\ j \in \{1, 2, 3\}.$$

После вычисления матрицы A_l полагаем

$$f_l(P_0) = A_lP_0,$$

что в силу произвольности пикселя P_0 означает полное определение функции градационного преобразования пикселей *RGB*-куба, согласованного с градационным преобразованием ступенчатых тоновых шкал $\{T_i\}$.

Обозначим через *Img* произвольное *RGB*-изображение. Влияние предварительного преобразования f_l на детализацию отображения *Img* можно оценить при визуальном сравнении отображений изображений *Img* и $f_l(Img)$, а также при вычислении *Lab*-контраста пикселей изображений $\Psi(Img)$ и $\Psi_l(f_l(Img))$.

Примеры коррекции детализации

Для изготовления иллюстраций было создано специальное программное обеспечение *T*_12, функционирующее на базе одного компьютера типа *PC IBM* с оболочкой *Windows XP* с цветным лазерным принтером *HP Color LaserJet* 4700*n* и цветным монитором *FLATRON L*1950*SQ*.

С помощью *T*_12 была вычислена функция Ψ — функция *RGB*-характеризации цветовосприятия отображений цветных пикселей на мониторе пользователя с такой аномалией цветового зрения как протанопия. Характеристики цветовосприятия протанопа получены из выходных данных программы ColorOracle [6].

В качестве тестового изображения *Img* использовалось изображение из ступенчатых тоновых шкал $Img = \{T_i\}$ (рис. 1, см. вторую сторону обложки). Цифровое описание восприятия данного изображения имеет вид в соответствии с рис. 2 (см. вторую сторону обложки).

Предварительные преобразования изображения выполнялись с помощью функций f_0 и f_1. В качестве образцов для преобразования контрастов применялись функции

$$\tau_{0,l}(s_{i,j'}, s_{i,j''}) = \rho(s_{i,j'}, s_{i,j''}),\ \tau_{1,l}(s_{i,j'}, s_{i,j''}) = E(s_{i,j'}, s_{i,j''}).$$

После равноконтрастного градационного преобразования f_0 восприятие изображения соответствует рис. 3, после f_1 — рис. 4 (см. вторую сторону обложки).

Как показывает практика, соотношение значений *Lab*-контрастов соответствует визуальному восприятию. Например, для части *Magenta*-шкалы (T_6) вычисленные последовательности контрастов имеют вид в соответствии с рис. 5 (см. вторую сторону обложки).

Заключение

Рассмотрена задача коррекции детализации отображений *RGB*-изображений на периферийных устройствах ПЭВМ. Приведен алгоритм для решения этой задачи, который может быть реализован с помощью специального программного обеспечения в автоматическом, и при необходимости, в ручном режиме под управлением пользователя.

Результаты работы имеют важное практическое значение, поскольку позволяют создать программный инструмент, дающий возможность каждому пользователю корректировать детализацию отображений цветных изображений в соответствии с собственными предпочтениями.

Список литературы

1. **Архипов О. П., Зыкова З. П.** Равноконтрастные градационные преобразования ступенчатых тоновых шкал // Информационные системы и технологии. 2011. № 4. С. 39—46.

2. **Архипов О. П., Зыкова З. П.** Интеграция гетерогенной информации о цветных пикселях и их цветовосприятии // Информатика и ее применения. 2010. Т. 4. Вып. 4. С. 14—25.

3. **Архипов О. П., Зыкова З. П.** Функциональное описание индивидуального цветовосприятия // Информационные системы и технологии. 2010. № 5. С. 5—12.

4. **Архипов О. П., Зыкова З. П.** RGB-характеризация пространства цветовосприятия // Системы и средства информатики. Вып. 20. — М.: ИПИ РАН, 2010. № 1. С. 73—90.

5. **Архипов О. П., Зыкова З. П.** Многокритериальный выбор тестового множества при исследовании цветовосприятия // Информационные технологии. 2011. № 2. С. 67—73.

6. **Color** Oracle. Institute of Cartography, ETH Zurich, 2008. URL: http://colororacle.cartography.ch.

УДК 004.021

С. И. Протасов, аспирант,
e-mail: stanislav.protasov@gmail.com,
А. А. Крыловецкий, канд. физ.-мат. наук, доц.,
С. Д. Кургалин, д-р физ.-мат. наук, доц.,
Воронежский государственный университет

Подход к решению задачи ректификации стереоизображений по сцене без калибровки камер

Рассматривается метод предобработки изображений в системах стереозрения, основанный на модификации алгоритма стабилизации видео [1]. Метод описывает ректификацию изображений как набор последовательных преобразований, каждое из которых находится как решение оптимизационной задачи. Описываются математическая модель, соответствующая основным положениям подхода к стабилизации видео, оптимизационные методы нахождения параметров преобразований и статистические алгоритмы уточнения решения. Предлагаемый подход не требует предварительной калибровки камер и использует только изображения сцены. Практической областью применения данного решения являются персональные системы стереозрения.

Ключевые слова: *стереозрение, машинное зрение, стереоизображения, калибровка, камеры, стабилизация видео*

Введение

Ректификация изображений (*image rectification*) является одной из важных задач при создании системы стереозрения. Под ректификацией обычно понимают приведение набора изображений к общей координатной системе так, чтобы горизонтальные линии на изображениях соответствовали одной плоскости. Как правило, данной задаче уделяется мало внимания вследствие того, что она решается однократно для конкретной системы камер. Тем не менее, системы стереокамер становятся вся более востребованными в различных областях [2]. В персональных системах стереозрения источником изображений могут быть камеры, установленные в мобильных телефонах, ноутбуках и др. В данной статье мы предлагаем методику ректификации изображений без необходимости калибровки и взаимной калибровки камер. Предлагаемый подход позволит конечному пользователю получить ректифицированный стереопоток, например, подключив дополнительную веб-камеру к ноутбуку.

Постановка задачи

Предлагаемая в статье модель решения является адаптацией задачи стабилизации видеопотока к ректификации изображений. Основой задачи стабилизации видеопотока является определение параметров перемещения камеры между последовательными кадрами или эквивалентная ей задача определения перемещения изображения. Под перемещением изображения понимается набор необходимых трансформаций для превращения кадра в последующий. Как правило, рассматривается проекционная модель камеры или, в случае пренебрежения дисторсиями, вносимыми оптикой, модель камеры-обскуры. В общей постановке задачи учитываются все шесть степеней свободы камеры: перемещения вдоль трех осей декартовых координат, а также повороты вокруг этих осей [3]. Нахождение всех шести параметров перемещения является нетривиальной задачей, поэтому в модели применяются упрощения. Проводится ограничение вариаций углов до малых значений (это позволяет применять следствия первого замечательного предела и приводить задачу к линейной), а также исключаются некоторые степени свободы. В данной статье рассматривается решение аналогичной задачи с учетом специфики стереозрения. В предлагаемой нами модели плоскости изображения принимаются параллельными, т. е. оптические оси камер считаются коллинеарными. Такое упрощение правомерно в рамках практических применений, например, закрепление дополнительной камеры на крышке ноутбука. Следствием этого упрощения является исключение из рассмотрения двух поворотных степеней свободы. Еще одно допущение делается в отношении угла поворота камеры вокруг оптической оси: будем считать его малым.

Декомпозиция и анализ задачи

Рассмотрим математическое описание процесса трансформации изображения одной камеры в изображение другой с учетом принятых ограничений. В рамках модели четыре степени свободы можно трактовать как три преобразования вида $\mathbb{R}^2 \Rightarrow \mathbb{R}^2$ над точкой изображения $p \in \mathbb{R}^2$. Перемещение $T(p)$ вдоль осей, лежащих в плоскости изображения, можно описать единым вектором перемещения $v_0 = (d_x, d_y)^{\mathrm{T}}$:

$$T(p) = p + (d_x, d_y)^{\mathrm{T}}.$$

Вращение $R(p)$ камеры вокруг оптической оси эквивалентно повороту изображения на определенный угол Θ вокруг точки p_c:

$$R(p) = p_c + \begin{pmatrix} \cos\Theta & \sin\Theta \\ -\sin\Theta & \cos\Theta \end{pmatrix} (p - p_c).$$

Движение вдоль оптической оси $Z(p)$ можно описать через деформацию растяжения относительно точки p_c, учитывая при этом возможное изменение соотношения сторон:

$$Z(p) = p_c + \begin{pmatrix} kx & 0 \\ 0 & ky \end{pmatrix} (p - p_c),$$

где kx, ky — коэффициенты растяжения изображения вдоль осей абсцисс и ординат.

Результатом решения задачи ректификации является набор величин p_c, v_0, Θ, kx, ky. Все три преобразования являются линейными по отношению к точке, поэтому с точностью до коэффициентов они могут проводиться в любом порядке. В предлагаемом нами решении поиск параметров осуществляется в последовательности $T \to R \to Z$.

Решение задачи

Решение задачи состоит из четырех основных этапов, базируется на операциях с векторным полем перемещения точек и является задачей оптимизации. Линейность каждого из преобразований позволила нам разделить решение задачи на независимые этапы, каждый из которых также является задачей оптимизации. На первом этапе выполняются поиск и фильтрация точечных особенностей на изображении первой камеры. На втором этапе строится векторное поле перемещения точек и находятся параметры перемещения $T(p)$. На третьем этапе вычисляются параметры трансформации $R(p)$ с учетом предварительной оценки координат точки p_c. На последнем шаге определяются параметры преобразования $Z(p)$ и находятся поправки для координат точки p_c.

Поиск точечных особенностей

В задаче стабилизации видеопотока применяется расчет векторного поля перемещения точек изображения. Опорные точки поля находятся в узлах регулярной сетки. Для уменьшения вероятности некорректного определения перемещений мы предлагаем использовать в качестве опорных точки с максимальной дисперсией окрестности. Под дисперсией окрестности понимается дисперсия яркости пикселей в малой квадратной симметричной окрестности рассматриваемой точки. Для того чтобы опорные точки были распределены относительно равномерно, применяем фильтрацию через регулярную сетку. Таким образом, на каждую ячейку регулярной сетки будет приходиться не более одной точечной особенности, как и в оригинальном методе, но каждая из них будет обладать локально максимальной дисперсией. Ячейки, внутри которых дисперсия окрестности точек не превышает заданный порог, будут "пустыми". Кроме того, для снижения вероятности ложных срабатываний поиск точечных особенностей проводится с заданным отступом от края изображения, так как края изображения с первой камеры с высокой вероятностью будут отсутствовать на изображении со второй камеры, и наоборот.

Нахождение векторного поля перемещения

Для каждой точки на изображении с первой камеры, отобранной на предыдущем этапе, найдем соответствие на изображении со второй камеры. В общем случае камеры не являются идентичными, имеют различные оптику и чувствительность матрицы, а значит, изображения будут иметь различную яркость и цветовой баланс. Поэтому до поиска соответствий проводим цветовую калибровку изображений [4]. Поскольку в нашей модели заложена возможность поворота изображений на малый угол, то нам необходимо учитывать влияние поворота на функцию соответствия. Поэтому в качестве критерия подобия мы используем взвешенное среднеквадратичное отклонение (weighted MSE) [1]. Весовой коэффициент мы положили равным $(1 + \sqrt{|\Delta x| + |\Delta y|})^{-1}$. В результате для каждой изначально отобранной точки получили индивидуальный вектор ее перемещения T_p (рис. 1) и значение минимума wMSE, являющееся критерием достоверности результата. Для ускорения работы алгоритма мы применили одну из модификаций алгоритма поиска соответствий с адаптивным окном [5].

Полученное поле по-прежнему содержит в себе векторы, соответствующие ложным срабатываниям. Корректное определение векторного поля перемещений точек изображения является существенным для точности работы всех дальнейших этапов. Именно поэтому на данном этапе необходимо обеспечить минимизацию числа ошибочно найденных векторов. Нами предложен способ фильтрации векторов: изображение разбивается на достаточно крупные блоки, в рамках которых, тем не менее, угол отклонения каждого вектора перемещения от среднего вектора по блоку невелик. Задав порог максимального отклонения от среднего, мы можем избавиться от заведомо некорректных векторов. В нашем эксперименте мы использовали шесть блоков (3×2) и допуск угла отклонения $\pm\pi/6$. Величина угла отклонения получена в рамках проведенных численных экспериментов. На рис. 1 можно видеть векторы, которые были отсеяны в результате фильтрации (они изображены тонкими линиями).

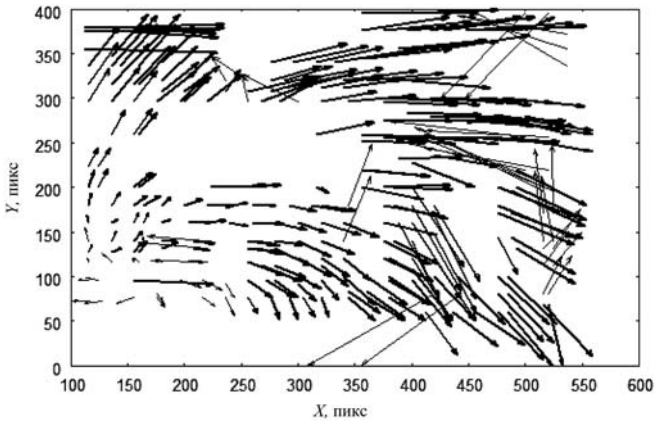


Рис. 1. Векторное поле перемещения точек с указанием отфильтрованных векторов

Для сцен, все точки которых являются равноудаленными от камер, средний вектор перемещения T_{avg} позволяет провести предварительную оценку точки p'_c перемещением центра изображения $p_{im} = \left(\frac{width}{2}, \frac{height}{2}\right)^{\mathrm{T}}$ с первой камеры:

$$p'_c = p_{im} + T_{avg}.$$

Нахождение угла поворота

Найденная оценка p'_c позволяет гарантировать, что после прибавления ко всем векторам v_p вектора $-T_{avg}$ искомая точка p_c будет находиться в пределах видимого изображения (рис. 2).

Лемма. Исходная область векторного поля перемещений точек, смещенная на вектор $-T_{avg}$, будет содержать в себе точку, относительно которой осуществляется операция деформации растяжения.

Доказательство (от противного).

Предположим, что точка p_c находится за пределами смещенной области по некоторой координате Q. Тогда средний вектор перемещения:

1) устремлен к точке p_c вдоль оси Q в случае, если коэффициент деформации по координате Q меньше 1;

2) устремлен от точки p_c вдоль оси Q в случае, если коэффициент деформации по координате Q больше 1;

3) в проекции на ось Q будет равен 0, если коэффициент деформации по координате Q равен 1.

В первом и втором случаях мы получаем противоречие, так как новый средний вектор перемещения $\frac{1}{n}\sum_n (v_p - T_{avg}) = \frac{1}{n}\sum_n v_p - T_{avg} = T_{avg} - T_{avg} = 0$. В третьем случае рассматривается тривиальная операция деформации, которая может осуществляться относительно любой точки, что также противоречит изначальному утверждению. *Лемма доказана.*

Найдем очередное приближение точки p''_c. Воспользуемся гипотезой о малости угла поворота. В данном случае закономерно предположение, что две прямые, параллельные осям координат и проходящие через искомую точку, разбивают плоскость изображения на четыре квадранта знакопостоянства проекций векторов перемещения на оси координат. Воспользуемся численным интегрированием с различным шагом вдоль осей для нахождения точки p''_c. Глобальные максимумы на графике (рис. 3) соответствуют координатам искомого приближения.

Поиск угла поворота мы предлагаем осуществлять по принципу максимизации суммы модулей скалярных произведений единичных векторов перемещения

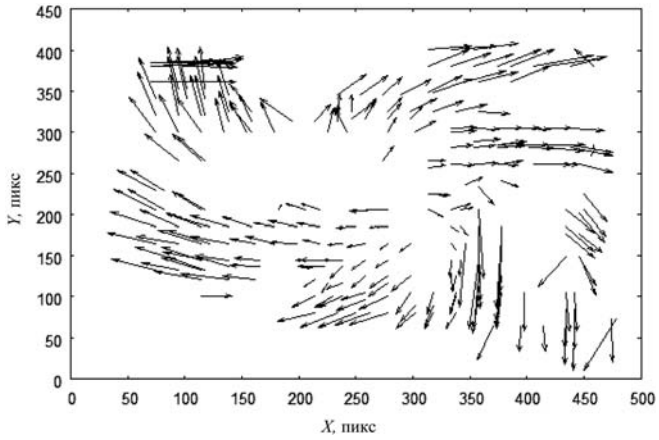


Рис. 2. Векторное поле перемещения точек после смещения

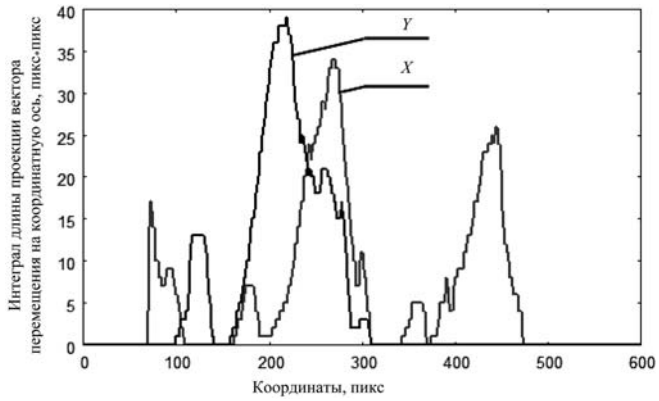


Рис. 3. Поиск координат центральной точки для операции поворота

$nv_p = \frac{v_p}{|v_p|}$ на единичные векторы "направления на точку" $t_p = \frac{p - p_c}{|p - p_c|}$. Максимизация суммы модулей скалярных произведений нормированных векторов гарантирует минимизацию среднего угла отклонения для малых значений угла отклонения вне зависимости от размеров изображений:

$$\forall\left(\Theta{:}|\Theta| < \frac{\pi}{2}\right) : \sum_n |(nv_p \cdot t_p)| = \sum_n |\cos\Theta_n|$$

при $\sum_n |\cos\Theta_n| \to n$, $\sum_n \Theta_n \to 0$.

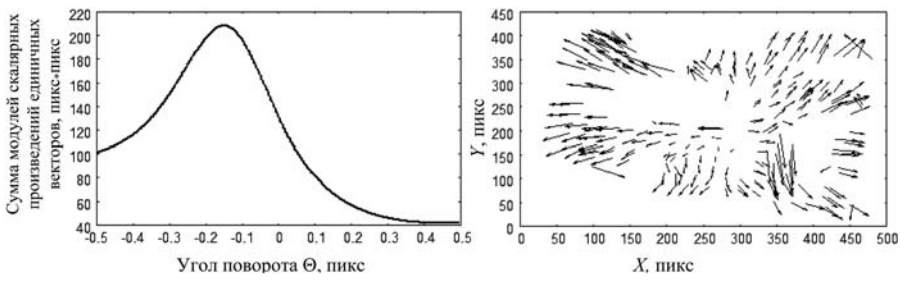


Рис. 4. Зависимость суммы модулей скалярных произведений от угла поворота и результирующее поле

Практически нам удалось показать, что данный метод является эффективным для определения угла поворота вокруг оптической оси (рис. 4) в случае, если оценка p_c'' является достаточно точной.

Предположение о малости угла Θ позволяет избежать ситуации, когда поворот изображения на угол π будет также максимизировать сумму. Найденный угол Θ вместе с оценкой p_c'' определяет преобразование поворота $R(p)$, одновременно уточняя преобразование перемещения $T(p)$.

Изменение размера изображения

В общем случае используемые камеры являются неидентичными, имеют различный угол обзора, а их плоскости изображений являются параллельными, но не совпадают. Преобразование $Z(p)$ учитывает все эти факторы и допускает изменение отношения сторон изображения. Мы предлагаем для нахождения коэффициентов kx и ky минимизировать сумму модулей векторов перемещения, получаемых после трансформации изображения:

$$(kx,\ ky) = \arg\min_{(kx,\ ky)} \sum_n |Z(R(T(p_n))) - p_n^*|, \quad (1)$$

где p_n^* — координаты соответствующей точки изображения со второй камеры. На рис. 5 приведен

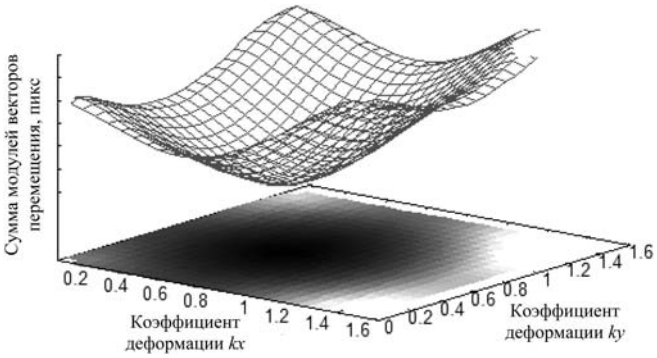


Рис. 5. Минимизация значения суммы модулей векторов в координатах (*kx*, *ky*)

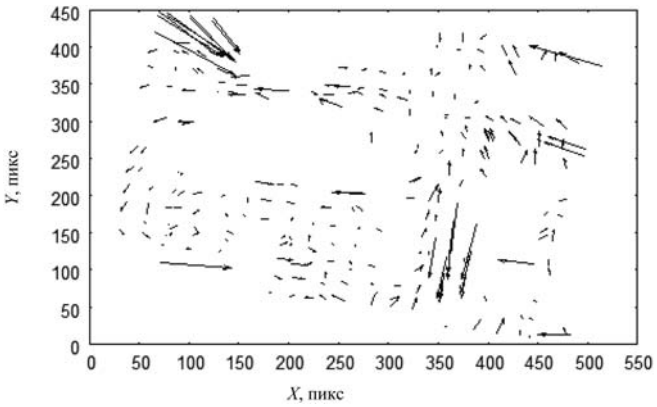


Рис. 6. Векторное поле перемещений точек после преобразования $Z(p)$

график функции (1), на котором хорошо заметен глобальный минимум, соответствующий оптимальному преобразованию $Z(p)$.

На рис. 6 изображена векторная карта перемещений, полученная после всех преобразований.

Трансформация изображения и уточнение перемещения

Нами были найдены параметры всех преобразований $T(p)$, $R(p)$, $Z(p)$. На данном этапе изображение с первой камеры может быть преобразовано в координаты второй камеры последовательным применением преобразований. Однако следует отметить, что найденное значение p_c'' является не окончательным решением, а всего лишь оценкой. Для ректификации изображений наиболее важным является размещение эпиполярных линий вдоль соответствующих строчек изображений. Ошибка в коэффициенте перемещения d_x не влияет на расположение строк изображений. Нам необходимо найти корректировку Δd_y для смещения d_y. Для этого мы предлагаем использовать дискретную функцию суммы яркостей вдоль строчек изображений $I_s(y) = \sum_n I(x, y)$, а затем найти смещение, минимизирующее сумму среднеквадратичного отклонения функции

$$\partial y = \arg\min_{dy} \sum_y [I_s^*(y) - I_s(y + \partial y)]^2, \quad (2)$$

где $I_s(y)$ и $I_s^*(y)$ — дискретные функции суммы яркостей изображений первой и второй камер соответственно.

В качестве альтернативного подхода мы предлагаем максимизировать взаимную корреляцию.

На рис. 7 можно увидеть "профили" функций $I_s(y)$ и $I_s^*(y)$. Подбор корректировки Δd_y позволяет

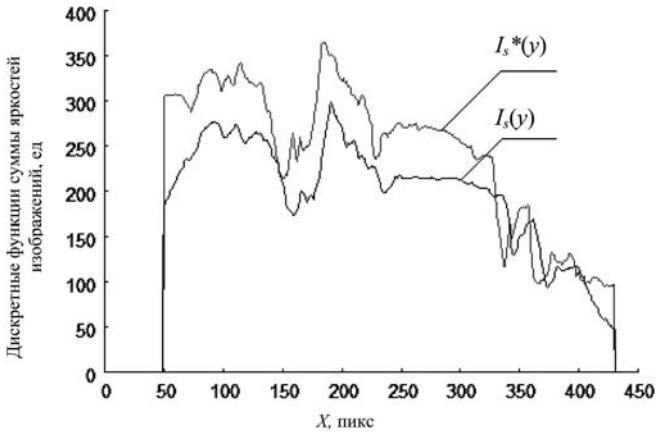


Рис. 7. Дискретные функции суммарной яркости $I_s(y)$ и $I_s^*(y)$

Рис. 8. Входная и результирующая стереопары

вычислить окончательные координаты $p_c = p''_c + (0, \Delta d_y)^{\mathrm{T}}$. На рис. 8 представлены результаты трансформации изображения.

Статистическая обработка

После нахождения преобразований T, R и Z в первом приближении мы можем увеличить точность коэффициентов этих преобразований. Так, на рис. 6 можно заметить векторы, существенно выделяющиеся длиной. Данные векторы являются следствием не исключенных ошибок алгоритма поиска точечных соответствий. Поскольку данные векторы легко отделить от прочих, исключим их из выборки и проведем поиск преобразований заново. Эксперимент показал, что в подавляющем большинстве случаев для фильтрации достаточно одной итерации: при повторной фильтрации по длине отбрасывается менее 5 % векторов.

Еще одним способом увеличения точности решения является поиск коэффициентов для нескольких независимых сцен. На рис. 9 в качестве примера показано значение угла поворота Θ камеры в радианах, найденное для различных кадров, при одинаковом позиционировании камер. После исключения выбросов можно воспользоваться средним значением коэффициента как окончательным приближением.

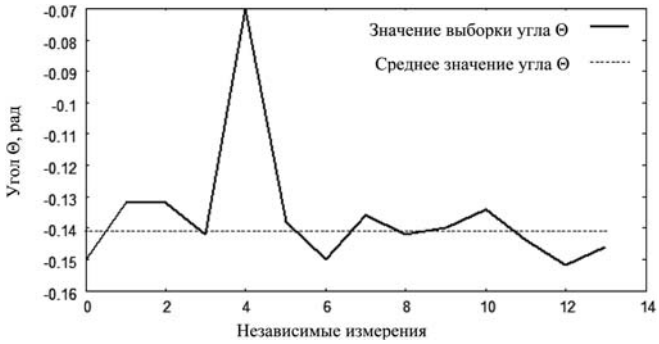


Рис. 9. Значения угла поворота для разных кадров

Заключение

Нами был разработан метод и алгоритм ректификации стереоизображений по сцене, основанный на модификации подхода к стабилизации камеры. Получаемые изображения могут применяться для создания стереовидеопотоков.

Предложенный метод содержит несколько ограничений. Так, на уровне модели исключены две степени свободы, а поворот изображения вокруг оптической оси камеры рассматривался в диапазоне $\pm \frac{\pi}{6}$. Поскольку в модель заложена возможность поворота камеры, мы использовали для сопоставления критерий взвешенного среднеквадратичного отклонения (weighted MSE) для увеличения достоверности сопоставления. Данный критерий, применяемый локально, может давать некорректные результаты на регулярных паттернах (одинаковые полосы, решетки, обои и т. п.). В качестве альтернативы можно рассматривать применение специальных изображений-паттернов с высокой цветовой дисперсией для нахождения параметров преобразований.

Поскольку найденные преобразования, полученные за один проход по единственной стереопаре, могут содержать неточные значения коэффициентов, мы предложили два статистических улучшения, которые повышают точность найденного решения. Во-первых, на рис. 6 видно, что часть векторов с относительно большими значениями нормы с высокой вероятностью являются результатом ошибочного срабатывания алгоритма wMSE; если отфильтровать такие векторы и повторить решение, то можно повысить точность найденных параметров преобразования. Во-вторых, выборка из нескольких решений, выполненных при одинаковой конфигурации камер, позволяет использовать математическое ожидание коэффициента в качестве окончательного решения.

Список литературы

1. **Chen Ting.** Video Stabilization Algorithm Using a Block-Based Parametric Motion Model. Stanford University, 2000. P. 3—4.

2. **Бондаренко С., Бондаренко М.** Создание 3D-изображений: теория и практика. URL: http://3domen.com/index.php?newsid=5794 (дата обращения 26.07.2012)

3. **Jin Jesse S., Zhu Zhigang, Xu Guangyou.** Digital Video Sequence Stabilization Based on 2.5D Motion Estimation and Inertial Motion Filtering // IEEE International Conference on Intelligent Vehicles. 2001. Vol. 7, Is. 4. P. 357—365.

4. **Протасов С. И., Крыловецкий А. А.** Использование web-камер как источника стереоизображений в реальном времени // Информатика: проблемы, методология и технологии: материалы XI международной научно-метод. конф., Воронеж, 10—11 февраля 2011 г. В 3-х т. Воронеж: ВГУ, 2011. Т. 2. С. 229—232.

5. **Kanade Takeo, Okutomi Masatoshi.** A stereo matching algorithm with an adaptive window: theory and experiment // IEEE Transactions on Pattern Analysis and Machine Intelligence. Sep. 1994. P. 920—932.

6. **Vermeulen Eddy.** Real-time Video Stabilization For Moving Platforms // 21st Bristol UAV Systems Conference — April. 2007.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ФИНАНСОВО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ

УДК 519.7

К. В. Павлов, студент,
e-mail: kirill.pavlov@phystech.edu,
Московский физико-технический институт

Алгоритм выбора многоуровневых моделей в задаче банковского кредитного скоринга

Решается задача классификации с использованием логистической регрессии. Предлагается новый подход, заключающийся в совместной кластеризации объектов и выборе признаков моделей. Результатом подхода является многоуровневая модель — набор моделей оптимальной сложности и разбиение объектов на группы, причем для объекта из определенной группы используется соответствующая этой группе модель. Для построения моделей использован EM-алгоритм. Предлагаемый алгоритм тестировался на данных по кредитным займам наличными.

Ключевые слова: *логистическая регрессия, многоуровневые модели, выбор моделей, EM-алгоритм*

Введение

Рассматривается задача классификации с двумя классами. Вводится предположение о виде условного распределения зависимой переменной. Это предположение называется гипотезой порождения данных. В соответствии с этой гипотезой строится регрессионная модель. Если гипотеза порождения не соответствует выборке, по которой строится функция регрессии, то качество классификации может быть низким. Один из способов повысить качество классификации заключается в использовании многоуровневых моделей.

Задача классификации решается с помощью моделей логистической регрессии. Эти модели являются частным случаем обобщенных линейных моделей [1], также рассматриваемых в работе. Впервые они были введены Джоном Нельдером и Робертом Веддербурном в 1972 г. [2]. В обобщенных линейных моделях предполагается, что объекты порождаются из экспонентного семейства распределений [3], к которому относится биномиальное распределение, соответствующее логистической регрессии.

При выборе модели логистической регрессии решается задача выбора признаков и оценки их правдоподобия. Известны следующие подходы к выбору признаков:

- шаговая регрессия [4, 5];
- метод итеративных перевзвешанных наименьших квадратов [6];
- метод порождения и выбора нелинейных регрессионных моделей [7, 8].

В случае, когда выборка описывается более чем одной моделью, предлагается использовать многоуровневый подход, согласно которому объекты разбиваются на несколько подмножеств и каждому подмножеству соответствует одна регрессионная модель.

В данной работе предложен метод выбора многоуровневой модели, использующий EM-алгоритм [6, 9, 10]. На E-шаге происходит отнесение объектов к моделям на основе оценки правдоподобия многоуровневой модели [11] в соответствии с некоторым критерием разбиения. На M-шаге происходит оценка наиболее вероятных параметров модели по объектам, которые к ней отнесли.

Преимуществом данного подхода является его способность описывать те выборки, которые затруднительно описывать одной моделью, и разбивать выборку в соответствии с выбранными моделями. Алгоритм тестировался на модельных и реальных данных. Реальные данные представляли собой истории кредитных займов наличными. Эксперименты показали преимущество использования многоуровневой модели по сравнению с использованием одной модели.

Постановка задачи

В задачах прогнозирования предполагается, что зависимая переменная y является случайной величиной, значение которой зависит от набора переменных $\mathbf{x} = (x_1, ..., x_n)$. Предполагается, что условное математическое ожидание $E(y|\mathbf{x})$ целевой переменной можно представить в следующем виде:

$$E(y|\mathbf{x}) = f(\mathbf{x}, \mathbf{w}).$$

Зависимость f является функцией регрессии от независимой переменной $\mathbf{x}$, называемой регрессором, и вектора параметров $\mathbf{w}$. Регрессионная выборка $D = \{(\mathbf{x}^i, y^i)\}_{i=1}^m$ — множество m пар, состоящих из вектора $\mathbf{x}^i = \{x_1^i, ..., x_n^i\}^{\mathrm{T}}$ и соответствующего этому вектору значения y^i.

Далее предполагается, что переменные определены на подмножестве действительных чисел: $\mathbf{x} \in \mathbf{X} \subseteq R^n$, $y \in \mathbf{Y} \subseteq R$. Индексы элементов i и компонент вектора независимой переменной j являются элементами конечных множеств $i \in I = \{1, ..., m\}$, $j \in J = \{1, ..., n\}$. Матрица плана $\mathbf{X}$ — матрица, строки которой есть компоненты независимой переменной $\mathbf{x}$, $\mathbf{X} = (\mathbf{x}^1, ..., \mathbf{x}^m)^{\mathrm{T}}$. Регрессионную выборку будем обозначать $D = (\mathbf{X}, \mathbf{y})$. Для нахождения функции регрессии используется понятие регрессионной модели.

Регрессионная модель — параметрическое семейство функций, отображающих декартово произведение областей определения объектов $\mathbf{X}$ и параметров $\mathbf{W}$ в область значений $\mathbf{Y}$ целевой переменной

$$f: \mathbf{X} \times \mathbf{W} \to \mathbf{Y}.$$

Если модель включает в себя несколько регрессионных моделей, то она называется многоуровневой. Многоуровневая регрессионная модель — набор регрессионных моделей f_k, $k = 1, ..., l$, такой что при разбиении множества индексов объектов $I \cup I_k$ для всех объектов с индексами из I_k используется модель f_k. Индексы моделей k являются элементами множества $K = \{1, ..., l\}$.

Обобщенные линейные модели

В основе обобщенных линейных моделей лежат следующие предположения. Во-первых, считается, что зависимая переменная $\mathbf{y}$ имеет экспонентную плотность распределения с вектором параметров $\boldsymbol{\theta}$,

$$p(\mathbf{y}|\boldsymbol{\theta}) = \exp(\mathbf{T}(\mathbf{y})^{\mathrm{T}}\boldsymbol{\eta}(\boldsymbol{\theta}) - b(\boldsymbol{\theta}) + c(\mathbf{y})),$$

где $\mathbf{T}$, $\boldsymbol{\eta}$, b и c — заданные функции. Оказывается, что в случае экспонентного распределения, и только в нем, $\mathbf{T}(\mathbf{y})$ является достаточной статистикой. Второе предположение заключается в том, что предиктор $\boldsymbol{\eta}$ линеен по координатам независимой переменной $\mathbf{x}$:

$$\boldsymbol{\eta} = \boldsymbol{\eta}(\boldsymbol{\theta}) = \mathbf{X}\mathbf{w}.$$

Предполагается также, что математическое ожидание $\boldsymbol{\mu} = (\mu_1, ..., \mu_m)$ зависимой переменной y есть монотонная функция вектора $\boldsymbol{\eta}$. При этом регрессионная модель имеет вид

$$E(\mathbf{y}|\boldsymbol{\theta}) = \boldsymbol{\mu} = f(\boldsymbol{\eta}) = f(\mathbf{X}\mathbf{w}).$$

Функция f называется функцией активации. В силу ее монотонности существует обратная функция f^{-1}, которая называется функцией связи.

В частном случае экспонентного распределения $\boldsymbol{\eta}(\boldsymbol{\theta}) = \boldsymbol{\theta}$, т. е. распределение имеет каноническую форму. Функция плотности при этом

$$p(\mathbf{y}|\boldsymbol{\theta}) = \exp(\mathbf{T}(\mathbf{y})^{\mathrm{T}}\boldsymbol{\theta} - b(\boldsymbol{\theta}) + c(\mathbf{y})). \quad (1)$$

Для случая канонической формы можно выписать выражения математического ожидания и дисперсии достаточной статистики зависимой величины

$$E(\mathbf{T}(\mathbf{y})) = \boldsymbol{\mu} = \nabla b(\boldsymbol{\theta});\ D(\mathbf{T}(\mathbf{y})) = \nabla\nabla^{\mathrm{T}} b(\boldsymbol{\theta}). \quad (2)$$

Логистическая регрессия

Решается задача классификации с двумя классами. Каждому объекту $\mathbf{x}$ необходимо приписать класс: ноль или единица. Задачу можно решить с помощью логистической регрессии, являющейся частным случаем обобщенных линейных моделей. Предполагается, что класс объекта y является случайной величиной из распределения Бернулли с параметром p, y: $B(p)$

$$y = \begin{cases} 1, & p; \\ 0, & 1 - p. \end{cases}$$

Покажем, что распределение Бернулли есть частный случай экспонентного распределения (1). Функция плотности $p(y|p)$ имеет вид

$$p(y|p) = p^y(1 - p)^{1 - y}.$$

Логарифмируя плотность $p(y|p)$ получим функцию правдоподобия

$$l(y|p) = y\log p + (1 - y)\log(1 - p).$$

Сгруппируем члены

$$l(y|p) = y\log\frac{p}{1 - p} + \log(1 - p).$$

Полученное выражение имеет форму экспонентного семейства (1) для случая $\mathbf{T}(\mathbf{y}) = y$:

$$\log p(y|p) = y\theta - b(\theta) + c(y)$$

со следующим соответствием: из вида первого слагаемого получим, что канонический параметр θ соответствует логистической функции параметра p:

$$\theta = \log\frac{p}{1 - p}.$$

Из полученного равенства можно найти обратную зависимость $p(\theta)$:

$$p = \frac{\mathbf{e}^{\theta}}{1 + \mathbf{e}^{\theta}} = \sigma(\theta);\ 1 - p = \frac{1}{1 + \mathbf{e}^{\theta}} = \sigma(-\theta). \quad (3)$$

Преобразуем второе слагаемое:

$$\log(1 - p) = \log\left(\frac{1}{1 + \mathbf{e}^{\theta}}\right) = -\log(1 + \mathbf{e}^{\theta}),$$

откуда определяется функция $b(\theta)$

$$b(\theta) = \log(1 + \mathbf{e}^{\theta}).$$

В случае распределения Бернулли $c(y) = 0$. Тем самым показана принадлежность логистических моделей обобщенным линейным.

Проверим значения математического ожидания и дисперсии. Для этого подставим в (2) полученные значения $b(\theta)$:

$$E(y) = b'(\theta) = \frac{\mathbf{e}^{\theta}}{1 + \mathbf{e}^{\theta}} = p.$$

Вторая производная даст дисперсию

$$D(y) = b''(\theta) = \frac{\mathbf{e}^{\theta}}{(1 + \mathbf{e}^{\theta})^2} = p(1 - p). \quad (4)$$

Настройка параметров модели

Рассмотрим распределение бернуллиевского случайного вектора $\mathbf{y}$ с независимыми компонентами $y_i : B(p_i)$. В рамках обобщенных линейных моделей натуральный параметр θ представляется как

$$\theta = \sum_{j=1}^{n} x_j w_j = \mathbf{x}^{\mathrm{T}}\mathbf{w}.$$

Функция плотности вектора $\mathbf{y}$ имеет вид

$$p(\mathbf{y}|\mathbf{w}) = \prod_{i=1}^{m} p_i^{y_i}(1 - p_i)^{1 - y_i}.$$

Определим функцию штрафа как минус логарифм правдоподобия:

$$E(\mathbf{w}) = -\ln p(\mathbf{y}|\mathbf{w}) = -\sum_{i=1}^{m} y_i \ln p_i + (1 - y_i)\ln(1 - p_i).$$

Будем минимизировать функцию по параметрам модели $\mathbf{w}$, оптимальное значение параметров $\mathbf{w}$ доставляет минимум $E(\mathbf{w})$, т. е.

$$w^* = \arg\min_{w \in R} E(\mathbf{w}).$$

Для подбора параметров $\mathbf{w}$ модели воспользуемся методом Ньютона—Рафсона, который на каждой итерации вычисляет квадратичную аппроксимацию функции, используя ее градиент и гессиан. Формула обновления весов

$$\mathbf{w}^{new} = \mathbf{w}^{old} - H^{-1}(\mathbf{w}^{old})\nabla E(\mathbf{w}^{old}).$$

Используя тождества

$$\frac{d\sigma(\theta)}{d\theta} = \sigma(1 - \sigma),\ p = \sigma(\mathbf{x}^{\mathrm{T}}\mathbf{w}),$$

вычислим градиент функции:

$$\nabla E(\mathbf{w}) = -\sum_{i=1}^{m} (y_i(1 - \sigma_i) - (1 - y_i)\sigma_i)\mathbf{x}_i =$$

$$= \sum_{i=1}^{m} (\sigma_i - y_i)\mathbf{x}_i = \mathbf{X}^{\mathrm{T}}(\boldsymbol{\sigma} - \mathbf{y}),$$

где $\sigma_i = \sigma(\mathbf{x}_i^{\mathrm{T}}\mathbf{w})$, $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_m)^{\mathrm{T}}$. Производная градиента даст гессиан функции штрафа:

$$H(\mathbf{w}) = \nabla\nabla^{\mathrm{T}}E(\mathbf{w}) = \sum_{i=1}^{m} \sigma_i(1 - \sigma_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} = \mathbf{X}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{X},$$

где введено обозначение $\boldsymbol{\Sigma}$ — диагональная матрица, $\Sigma_{ii} = \sigma_i(1 - \sigma_i)$. Используя формулу (4), заметим, что $\Sigma_{ii} = D(y_i)$, а так как компоненты вектора $\mathbf{y}$ по предположению независимы, то $\boldsymbol{\Sigma}$ является корреляционной матрицей.

Ковариационная матрица $\boldsymbol{\Sigma}$ положительно определена, а значит, и гессиан положительно определен (он является матрицей Грама в пространстве весов) из чего следует, что функция $E(\mathbf{w})$ выпукла и имеет единственный минимум.

Формула Ньютона—Рафсона для обновления весов для модели логистической регрессии

$$\mathbf{w}^{new} = \mathbf{w}^{old} - (\mathbf{X}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}(\boldsymbol{\sigma} - \mathbf{y}) =$$
$$= (\mathbf{X}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Sigma}(\mathbf{X}\mathbf{w}^{old} - \boldsymbol{\Sigma}^{-1}(\boldsymbol{\sigma} - \mathbf{y})) = (\mathbf{X}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{z};$$
$$\mathbf{z} = \mathbf{X}\mathbf{w}^{old} - \boldsymbol{\Sigma}^{-1}(\boldsymbol{\sigma} - \mathbf{y}). \qquad (5)$$

Формула (5) с точностью до вектора $\mathbf{z}$ совпадает с формулой нахождения параметров для случая взвешенной логистической регрессии, при этом объекты $\mathbf{x}$ взвешиваются своими дисперсиями $\boldsymbol{\Sigma}_{ii}$.

Процедура выбора модели

Выше был рассмотрен способ нахождения параметров одной модели. Пусть теперь имеется набор моделей. Определим объекты, по которым настраивать каждую модель.

Для настройки многоуровневых моделей при решении задачи классификации для объекта нужно выбрать соответствующую ему модель.

Это можно сделать на основе ее правдоподобия. Вероятность того, что объект $(\mathbf{x}^i, y^i)$ был порожден моделью f_k,

$$p(f_k|\mathbf{x}^i, y^i) = \frac{p(f_k, \mathbf{x}^i, y^i)}{p(\mathbf{x}^i, y^i)} = \frac{p(y^i|f_k, \mathbf{x}^i)p(f_k, \mathbf{x}^i)}{p(\mathbf{x}^i, y^i)}.$$

Априорная вероятность объекта $p(\mathbf{x}^i, y^i)$ одинакова для всех моделей. Величина $p(f_k, \mathbf{x}^i)$ называется априорной вероятностью модели. Предположим, что заранее нет никаких предпочтений в выборе моделей и априорные вероятности их равны. Пусть для объекта $(\mathbf{x}^i, y^i)$ имеются две модели-кандидата f_1 и f_2. Отношение их правдоподобий будет

$$\frac{p(f_1|\mathbf{x}^i, y^i)}{p(f_2|\mathbf{x}^i, y^i)} = \frac{p(f_1, \mathbf{x}^i, y^i)}{p(f_2, \mathbf{x}^i, y^i)} = \frac{p(y^i|f_1, \mathbf{x}^i)}{p(y^i|f_2, \mathbf{x}^i)} \frac{p(f_2, \mathbf{x}^i)}{p(f_1, \mathbf{x}^i)}.$$

Дробь, являющаяся вторым сомножителем, называется априорным отношением вероятностей моделей. Ввиду предполагаемого равенства априорных вероятностей дробь принимает значение единица. Принцип максимума правдоподобия модели связан с принципом максимума правдоподобия данных следующим образом:

$$k^* = \arg\max_{k \in \{1,...,l\}} p(y^i|f_k, \mathbf{x}^i).$$

Класс y объекта неизвестен, но от него зависит значение выражения $p(y^i|f_k, \mathbf{x}^i)$. В зависимости от различных значений класса y объекта $\mathbf{x}$ будут выбираться разные модели. Предположим, что модель зафиксирована и рассмотрим наихудший вариант, когда объект имеет класс, доставляющий минимум $p(y^i|f_k, \mathbf{x}^i)$ для данной модели. Минимумы правдоподобия данных можно найти для всех моделей и составить из них вектор $(p_{\min}(y^i|f_1, \mathbf{x}^i), ..., p_{\min}(y^i|f_l, \mathbf{x}^i))$. Выберем ту модель, для которой соответствующий элемент вектора минимальных вероятностей максимален:

$$k^* = \arg\max_{k \in \{1,...,l\}} \min_u p(u|f_k, \mathbf{x}^i).$$

В случае задачи логистической регрессии переменная класса u принимает всего два значения $u \in \{0, 1\}$. Вероятности принадлежности объектов к каждому классу выражаются через логистическую функцию (3). Перепишем решающее правило для выбора модели:

$$k^* = \arg\max_k \min\{\sigma(\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k), \sigma(-\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k)\}.$$

Минимум двухэлементного множества можно найти явно. Воспользуемся монотонностью сигмоидной функции. Минимальное значение будет то, у которого аргумент минимален. В данной ситуации аргументы равны по модулю и противоположны по значению. Перепишем решающее правило с использованием модуля:

$$k^* = \arg\max_k \min\{\sigma(\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k), \sigma(-\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k)\} =$$
$$= \arg\max_k \sigma(-|\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k|).$$

Воспользуемся еще раз монотонностью логистической функции:

$$k^* = \arg\max_k \sigma(-|\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k|) =$$
$$= \arg\max_k (-|\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k|) = \arg\min_k |\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k|. \qquad (6)$$

Заметим, что $\dfrac{|\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k|}{|\mathbf{w}_k|}$ есть расстояние от $\mathbf{x}^i$ до гиперплоскости с нормальным вектором $\mathbf{w}_k$ (рис. 1). С точностью до нормы вектора $\mathbf{w}_k$ объекты относятся к той модели, расстояние до разделяющей гиперплоскости которой минимально.

Алгоритм 1 EM-алгоритм для l моделей

Вход: $\mathbf{X} = \{\boldsymbol{x}_i^{\mathrm{T}}\}_{i=1}^m$ — матрица плана;
$\boldsymbol{y} = \{y_i\}_{i=1}^m$ — метки классов;
l — число моделей;
Выход: Набор моделей $(\mathrm{model}_k)_{k=1}^l$;
1: EmIrls($\mathbf{X}$, $\boldsymbol{y}$, l):
2: Инициализировать модели объектов случайно
3: **повторять**
4: M-шаг:
5: **для** $k = 1$ **до** l
6: Оценить параметры k-й модели;
$\mathbf{X}^k$ — объекты, отнесенные к k-й модели;
$\boldsymbol{y}^k$ — классы объектов k-й модели;
$\boldsymbol{w}_k = IRLS(\mathbf{X}^k, \boldsymbol{y}^k)$
7: E-шаг:
8: **для всех** $i = 1$ **до** m
9: $\mathrm{model}(\boldsymbol{x}_i) = \arg\min_k |\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_k|$
10: **пока** модели не стабилизируются

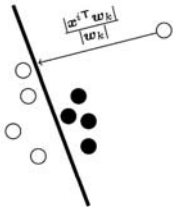


Рис. 1. Иллюстрация решающего правила отнесения объектов к моделям

Докажем, что минимизация $|\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k|$ эквивалентна максимизации $D(y^i|\mathbf{w}_k)$. Дисперсию объекта можно представить в виде

$$D(y^i|\mathbf{w}_k) = \sigma(\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k)(1 - \sigma(\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k)).$$

Это квадратичная функция по $\sigma(\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k)$. Введем обозначение: $z = (\mathbf{x}^{i\mathrm{T}}\mathbf{w}_k)$, тогда

$$\arg\min_z \sigma(z)(1 - \sigma(z)) = \arg\min_z \left|\frac{1}{2} - |z|\right| = \arg\min_z |z|,$$

что доказывает утверждение. Решающее правило (6) можно интерпретировать так: объекты относятся к модели, дисперсия относительно которой максимальна.

Построение многоуровневой модели

Для построения многоуровневой модели будем использовать алгоритм 1 со следующими шагами:

M-step

На *M*-шаге настраиваются параметры моделей с помощью логистической регрессии и метода Ньютона—Рафсона.

E-step

На *E*-шаге происходит отнесение объекта к моделям на основании их правдоподобия. Решающее правило имеет вид

$$k^* = \arg\min_k |\mathbf{x}_i^{\mathrm{T}}\mathbf{w}_k|.$$

Шаги повторяются итеративно до тех пор, пока модели не стабилизируются.

Численный эксперимент

Алгоритм тестировался на модельных и реальных данных. В обоих случаях исходные данные делились на обучающую и контрольную выборки в отношении 0,7:0,3 соответственно. На обучающей выборке настраивались параметры алгоритма, на контрольной измерялись критерии качества.

Модельные данные представляли собой два кластера на плоскости, в каждом из которых объекты разных классов распределены нормально и линейно разделимы, однако сама выборка линейно разделимой не является. Алгоритм выявил наличие двух моделей и безошибочно классифицировал объекты гиперплоскостями. На рис. 2 изображены объекты. Крестики и нолики обозначают классы, прямые являются разделяющими гиперплоскостями, построенными алгоритмом.

Реальные данные представляли собой кредитные истории займа наличными. Выборка содержала данные о 6000 клиентах, каждый из которых описывался 50 признаками, такими как возраст заемщика, его заработная

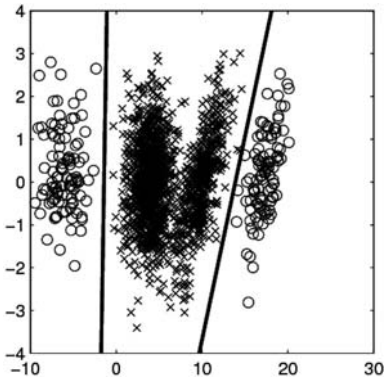


Рис. 2. Классификация модельной выборки

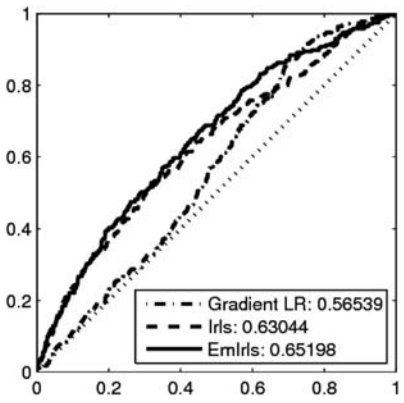


Рис. 3. ROC-кривые и значения площади под кривой для различных моделей

плата, наличие машины и т. д. В качестве функции качества использовалась площадь под ROC-кривой, построенной по контрольной выборке. Метод сравнивался с логистической регрессией, настроенной градиентным спуском и итеративным перевзвешивающим методом наименьших квадратов. На рис. 3 изображены ROC-кривые, построенные по результатам работы каждого алгоритма. По оси абсцисс откладывается доля ошибочных положительных классификаций (false positive rate), по оси ординат — доля правильных положительных классификаций (true positive rate). ROC-кривые для предложенного алгоритма, метода перевзвешенных наименьших квадратов и логистической регрессии изображены, соответственно, сплошной, штриховой и штрих-пунктирной линиями. Предложенный алгоритм показал лучший результат.

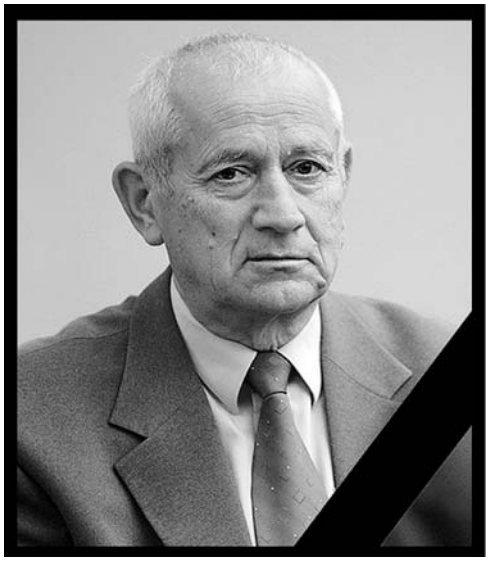
Заключение

Предложен алгоритм выбора многоуровневых моделей. Его работа проиллюстрирована на реальных и синтетических данных. В работе показано преимущество использования многоуровневой модели по сравнению с использованием одной модели на примере классификации линейно неразделимой выборки и реальных данных.

Список литературы

1. **Lee Y., Nelder J., Pawitan Y.** Generalized Linear Models with Random Effects. Boca Raton. Taylor and Francis Group. 2006.
2. **Nelder J., Wedderburn R.** Generalized Linear Models // Journal of the Royal Statistical Society. 1972. P. 370—384.
3. **Ивченко Г. И., Медведев Ю. И.** Введение в математическую статистику. М.: ЛКИ. 2010.
4. **Tibshirani R. J.** Regression shrinkage and selection via the lasso // Journal of the Royal Statistical Society. 1996. Vol. 32, № 1. P. 267—288.
5. **Стрижов В. В., Крымова Е. А.** Выбор моделей в линейном регрессионном анализе // Информационные технологии. 2011. № 10. С. 21—26.
6. **Bishop C.** Pattern Recognition and Machine Learning. Springer. Information Science and Statistics. 2006.
7. **Strijov V., Weber G. W.** Nonlinear regression model generation using hyperparameter optimization // Computers and Mathematics with Applications. 2010. № 60 (4). P. 981—988.
8. **Стрижов В. В.** Методы индуктивного порождения регрессионных моделей. М.: ВЦ РАН. 2008.
9. **Павлов К. В., Стрижов В. В.** Выбор многоуровневых моделей в задачах банковского кредитного скоринга. Математические методы распознавания образов. 2011. С. 58—161.
10. **Павлов К. В.** Алгоритм выбора многоуровневых моделей в задаче восстановления логистической регрессии // Труды 54-й научной конференции МФТИ. 2011. С. 104—106.
11. **Strijov V., Krymova E., Weber G. W.** Evidence optimization for consequently generated models // Mathematical and Computer Modelling. 2011. № 2 (17).

ПАМЯТИ КОЛЛЕГИ И ТОВАРИЩА



НОРЕНКОВ
Игорь Петрович

(19.08.1933—31.12.2012)

Редколлегия журнала "Информационные технологии" понесла тяжелую утрату... На 80-м году ушел из жизни известный ученый, блестящий преподаватель, талантливый организатор и замечательный человек, главный редактор журнала "Информационные технологии" — **Игорь Петрович Норенков**.

Основатель и до последнего времени руководитель кафедры систем автоматизированного проектирования МГТУ им. Н. Э. Баумана, Заслуженный деятель науки и техники РФ, лауреат Государственной премии СССР, академик РАЕН, доктор технических наук, один из ведущих специалистов в области современных компьютерных и информационных технологий, Игорь Петрович внес неоценимый вклад в развитие российской науки и образование.

И. П. Норенков родился 19 августа 1933 г., в 1960 г. закончил Приборостроительный факультет МВТУ им. Н. Э. Баумана (ныне МГТУ им. Н. Э. Баумана) по специальности "Математические машины". В 1964 г. группой сотрудников кафедры "Электронные вычислительные машины и системы" под руководством И. П. Норенкова была создана первая отечественная программа анализа электронных схем (ПАЭС), которая дала жизнь целому семейству соответствующих программных средств для предприятий радиоэлектронной промышленности. В 1975 г. за большой вклад в теорию и практику методов автоматизированного проектирования электронных схем Игорь Петрович был удостоен Государственной премии СССР.

Игорь Петрович являлся одним из пионеров компьютеризации проектно-конструкторской деятельности, руководителем одной из ведущих отечественных школ в области моделирования и оптимизации. В 1980 г. И. П. Норенковым была сформулирована концепция многоаспектного моделирования, которая в дальнейшем послужила основой для создания не имеющего в то время аналогов в мире универсального программного обеспечения для моделирования и анализа динамических систем с физическими разнородными элементами. Эти результаты позволили разработать единую методологию преподавания студентам различных специальностей основ автоматизированного проектирования. В последние десятилетия в работах И. П. Норенкова все большее внимание уделялось проблемам автоматизации трудно формализуемых проектных процедур структурного синтеза на основе использования современных информационных технологий.

И. П. Норенков в 1982 г. основал в МВТУ им. Н. Э. Баумана одну из первых отечественных кафедр в области систем автоматизированного проектирования, которая занимает ведущее место в системе подготовки высококвалифицированных специалистов в области автоматизации проектирования. В 1995 г. он принял участие в организации и возглавил редакцию журнала "Информационные технологии", который за эти годы стал всероссийской трибуной для специалистов в области современных информационных технологий.

Все знали И. П. Норенкова как высокоинтеллегентного ученого, которого отличали широкий научный кругозор, постоянное стремление двигаться вперед, доброжелательность, отзывчивость. В памяти друзей и коллег он останется очень светлым человеком.

Редколлегия и редакция журнала "Информационные технологии"
выражают глубокое соболезнование родным и близким покойного.
Вечная память об Игоре Петровиче Норенкове навсегда сохранится в наших сердцах.

CONTENTS

An algorithm for the identification of two superimposed flat real objects based on the Bayesian method. The main criterion used in the value of the curvature of the contour points. Additional features are the length of the convex and concave, and the coefficients of convexity and concavity of the contour of the object. The algorithm is illustrated by examples.

Keywords: identification of real objects superimposed, the value of α-function, the Bayesian method, detection, indication, a cluster

In this paper we consider model for data recovery, implemented in software "Fuzzy system based on evolutionary strategy" for the data recovery problem. A fuzzy system, evolutionary strategies, data recovery problem are described.

Experimental researches results are presented.

Keywords: fuzzy system, evolutionary strategies, data recovery problem

The paper presents an approach to the construction of neural networks for the organization of hardware or software kernel cryptographic system based on elliptic curves. The results prove the feasibility of parallelization of the basic operations of the elliptic cryptography by means of neural networks of higher order.

Keywords: neural network, cryptography, elliptic curve

The correcting problem of detail presentations of RGB-images on monitor and on printer on the basis of digital description (RGB-characterizations) the color perception of PC user is considered. The algorithm for solving is given. Some complicity characteristics of this algorithm and applications of these results are discussed. This related to definitions of personified preliminary modifications RGB-images for the purpose of improving detail their representations on peripherals PC. Examples to illustrate are given.

Keywords: LAB-contrast, gradation, reproducing of details, reproducing of hues

This article is devoted to the method of initial image processing to use in stereovision systems. It is based on a modification of video stabilization approach. The method considers image rectification process as a sequence of transformations. Each transformation is found as a solution of optimization problem. The article describes mathematical model that fits main principles of video stabilization method, provides optimization techniques to find transformation parameters and describes statistical approach to solution refinement. The method provided does not require camera calibration and uses only scene images. This approach can be applied to create individual stereovision systems.

Keywords: stereo-vision, computer vision, stereoimages, camera calibration, video stabilization

Consider classification problem using logistic regression. The author proposes a new approach, which selects models and clusters the objects. The result of the approach is a multilevel model — set of logistic regression models of optimal complexity and partition of objects into clusters. A model is used for objects from the same cluster. The proposed model selection approach involves the Expectation-Maximization algorithm. The proposed approach is illustrated by a cash loan data set.

Keywords: logistic regression, multilevel models, model selection, EM-algorithm

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.

Корректор *Е.В. Комиссарова*.

Сдано в набор 05.12.2012. Подписано в печать 21.01.2013. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT213. 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.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".

105120, г. Москва, ул. Нижняя Сыромятническая, д. 5/7, стр. 2, офис 2.