ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

7(179) 2011

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 612.017, 004.89

Ю. И. Еременко, д-р техн. наук, проф.,
А. И. Глущенко, канд. техн. наук, ассистент,
Старооскольский технологический институт
(филиал) ФГОУ ВПО "Национальный
исследовательский технологический
университет "МИСиС",
e-mail: strondutt@mail.ru

О решении неформализуемых и плохоформализуемых задач методами иммунных алгоритмов

Рассматривается аппарат иммунных сетей, основные подходы к его формализации и реализации, а также возможности его применения для решения слабоформализуемых и неформализуемых задач.

Ключевые слова: *иммунная сеть, сингулярное разложение, отрицательный отбор, клональная селекция, dendritic cell algorithm*

Введение

В настоящее время для решения слабоформализуемых и неформализуемых задач, довольно часто возникающих как в задачах управления производством, так и в других сферах человеческой деятельности, применяют как классические методы, так и интеллектуальные. Анализ тематики публикаций и защищенных диссертаций позволяет сделать вывод о том, что в последние годы интеллектуальные методы (нейронные сети, нечеткая логика и т. д.) в решении этого класса задач приобретают все большую популярность.

Однако и в рамках искусственного интеллекта, помимо ставших уже классическими направлений, возникают и активно развиваются новые. Одним из таких подходов являются иммунные сети.

Иммунная система человека достаточно давно привлекает внимание исследователей наличием ряда важных особенностей: распределенностью, отсутствием центрального узла, наличием памяти, способностью к обучению, порогового механизма, распознавания "свой — чужой" [1].

Основной принцип работы иммунной системы заключается в том, что при попадании в систему антигена (чужеродной клетки) осуществляется его распознавание в целях определения вредоносности и класса (если антиген такого типа раньше встречался). После этого проводится построение иммунного ответа путем создания Т- и В-лимфоцитов (антител). Они взаимодействуют с антигеном, уничтожая его. Если антиген ранее не встречался, то система обучается, подбирая подходящую структуру антитела, а затем сохраняет результат в памяти.

Такая уникальность иммунитета обратила на себя внимание не только иммунологов, но и ученых ряда других областей. В результате возник ряд предположений о возможности использования механизмов, заложенных в иммунной системе человека, для решения задач в области робототехники, защиты информации, распознавания образов и пр. [1]. Для проверки такого рода предположений необходимо было математически описать процессы, происходящие в иммунной системе.

В настоящее время выделяются три основных подхода к формализации аппарата искусственных иммунных сетей:

1. Искусственные иммунные сети (*Artificial immune networks*) — направление, основанное на работах Д. Дасгупты, Л. Н. де Кастро, Фон Зюбена. Первые публикации относятся к 1999 г.

2. Иммунокомпьютинг — направление, основанное на работах А. О. Тараканова. Первые публикации относятся к 1999 г.

3. Теория опасности (*Danger theory*) — направление, основанное на работах У. Айкелина. Первые публикации относятся к 2002 г.

Рассмотрим эти направления подробнее.

Искусственные иммунные сети

Рассмотрим общий ***алгоритм*** построения и работы иммунной сети.

Определение значения понятий "антитело" и "антиген" для рассматриваемой задачи. Например, при решении задачи управления рассогласование входного сигнала и сигнала обратной связи для системы управления будет антигеном, а управляющее воздействие на объект — антителом, при распознавании образов — входной образ в виде последовательности генов — антигеном, а распознанный образ — антителом.

Формирование врожденного иммунитета. Если для задачи известен начальный набор антигенов, то необходимо сформировать антитела, покрывающие его. Если же такой набор неизвестен (задача защиты информации), то необходимо сформировать набор антигенов, не вступающих во взаимодействие с защищаемой информацией.

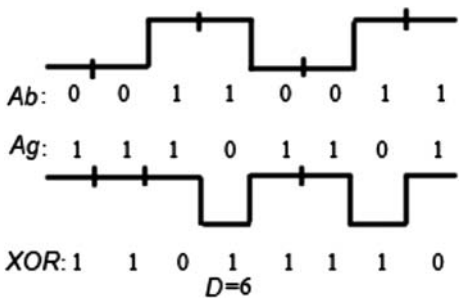

Рис. 1. Вычисление расстояния с помощью меры Хэмминга

Для этих целей служит алгоритм отрицательного отбора (*negative selection algorithm*) [2]. В рамках данного алгоритма проводятся генерация антител, а затем их проверка на совпадение с известными антигенами (или несовпадение с защищаемой информацией). Данный алгоритм может иметь несколько реализаций. Рассмотрим его подробнее.

Антитело и антиген в большинстве случаев представляются в виде последовательности генов (бинарной, небинарной). Для определения степени аффинности (связи) между ними вычисляется расстояние. В большинстве случаев для этого используются следующие метрики:

1) евклидово расстояние

$$D = \sqrt{\sum_{i=1}^{L} (ab_i - ag_i)^2}; \qquad (1)$$

2) манхэттенское расстояние

$$D = \sqrt{\sum_{i=1}^{L} |ab_i - ag_i|}; \qquad (2)$$

3) мера Хэмминга

$$D = \sum_{i=1}^{L} \delta, \text{ где } \delta = \begin{cases} 1, \text{ если } ab_i \neq ag_i; \\ 0 \text{ в противном случае.} \end{cases} \qquad (3)$$

Здесь D — расстояние; ab_i — значение i-го гена антитела; ag_i — значение i-го гена антигена; L — длина антитела и антигена.

На рис. 1 приведен пример вычисления значения расстояния с помощью меры Хэмминга.

Стоит заметить, что антитело не является копией антигена по значениям своих генов (операция *XOR*). Формула (3) применяется, если антиген и антитело представимы в виде бинарных последовательностей, в противном случае применяются формулы (1) и (2). Полученное значение расстояния подается на функцию связи антитела (в большинстве случаев используются пороговая и сигмоидальная функции (рис. 2)).

Считается, что антитело распознало антиген, если вычисленная аффинность оказалась выше заранее предустановленной величины. Таким образом реализуется пороговый механизм работы системы.

При генерации антител важным вопросом является определение их оптимального числа. Его увеличение замедляет работу системы, поскольку на каждом шаге проводится последовательная сверка новых данных со всеми существующими антителами. Повышение порога аффинности ведет к увеличению числа антигенов, распознаваемых одним антителом. Однако это ведет и к снижению точности распознавания.

Общее число уникальных антигенов и антител может быть представлено как k^L, где k — объем алфавита и L — длина антигена (антитела). Тогда число антигенов, распознаваемых одним антителом при заданном пороге аффинности ε, может быть вычислено как

$$C = \sum_{i=0}^{\varepsilon} \frac{L!}{i!(L-i)!}, \qquad (4)$$

где C — зона покрытия антитела.

Основываясь на формуле (4), при длине антигенов L и пороге аффинности ε минимальное число

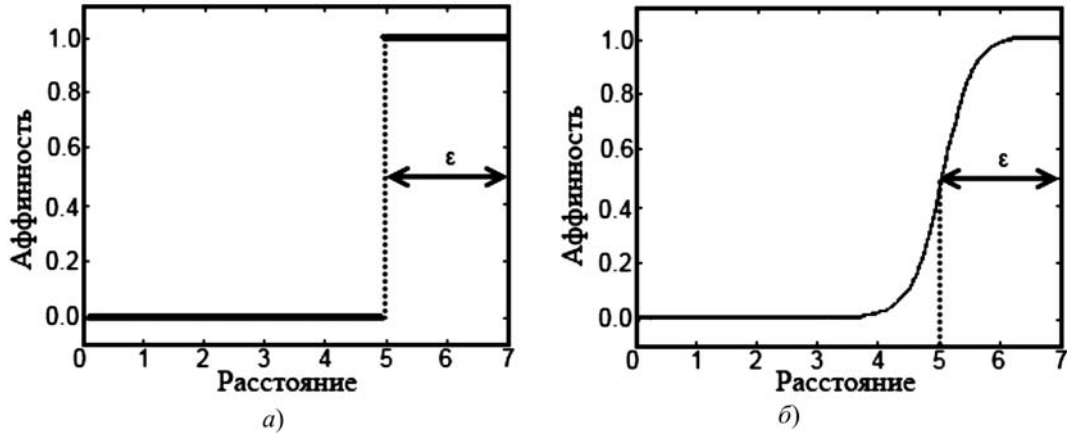

Рис. 2. Зависимость между аффинностью и расстоянием Хэмминга для антитела длины $L = 7$ с порогом аффинности ε = 2: *а* — пороговая функция, *б* — сигмоидальная функция

антител N, необходимое для распознавания всех известных антигенов, может быть вычислено как

$$N = \left\lceil \frac{k^L}{C} \right\rceil.$$

После определения значений указанных параметров осуществляется генерация антител с помощью алгоритма отрицательного отбора. Классический подход заключается в случайной генерации N антител с соблюдением следующего требования: каждый известный антиген распознается хотя бы одним антителом.

Построение иммунной сети (формирование приобретенного иммунитета, клональная селекция и ее разновидности). Полученный в результате выполнения предыдущего шага алгоритма набор антител позволит справляться с известными антигенами. Однако естественный иммунитет обладает способностью к обучению на новых антигенах и запоминанию вновь сгенерированных антител.

Для этих целей в рамках искусственных иммунных сетей применяется алгоритм клональной селекции (*clonal selection algorithm*).

В случае обнаружения нового антигена, если он не был распознан ни одним из существующих антител, осуществляется построение нового. Причем основой для него служит антитело с максимальным среди существующих уровнем аффинности к новому антигену.

Классический алгоритм клональной селекции включает в себя следующие шаги:

1. Для вновь обнаруженного антигена Ag_j ($Ag_j \in Ag$) вычисляется значение аффинности для всех существующих антител Ab.

2. В вектор $f = \{f_j\}$ длины N вносятся результаты вычислений п. 1.

3. Проводится выбор n максимальных значений аффинности, соответствующие им антитела из множества антител Ab формируют новое множество $Ab^j_{\{n\}}$.

4. n выбранных антител клонируются пропорционально значению их аффинности по отношению к антигену Ag_j (чем выше аффинность, тем больше число клонов). Клонированные антитела формируют множество C^j.

5. Все элементы множества C^j проходят процедуру мутации, причем число мутирующих генов в антителе обратно пропорционально его аффинности с антигеном Ag_j. Создается множество C^{j*} клонов антител, прошедших мутацию.

6. Вычисляется аффинность f_j^* прошедших мутацию клонов C^{j*} по отношению к антигену Ag_j.

7. Среди элементов множества C^{j*} выбирается антитело Ab_j^*, имеющее максимальную аффинность по отношению к антигену Ag_j. Если аффинность выбранного антитела по отношению к антигену Ag_j больше любого элемента вектора f, то антитело добавляется в иммунную память.

Существует несколько разновидностей приведенного выше алгоритма [2] (алгоритм адаптивной клональной селекции, оптимизационный иммунный алгоритм и др.). Подробное сравнение данных алгоритмов и описание их применения приведены в работе [3].

Данный подход применялся [2] при построении систем защиты информации и компьютерных сетей, мониторинга процессов ОС *UNIX*, для решения задач обнаружения неисправностей и медицинской диагностики, в робототехнике и при построении систем управления.

Иммунокомпьютинг

Иммунокомпьютинг представляет собой подход, базирующийся на принципах обработки информации молекулами белков и иммунными сетями [4]. Математической основой иммунокомпьютинга является идея формального белка (ФБ).

В соответствии с [4] пространственная структура скелета белка может быть геометрически представлена подобно изображенной на рис. 3, где k — номер повторяющегося фрагмента.

Формальный белок представляет собой упорядоченную пятерку:

$$P = \langle n,\ U,\ Q,\ V,\ v \rangle.$$

Здесь: $n > 0$ — количество связей; $U = \{\varphi_k, \psi_k\}$, $k = 1, \ldots, n$, где $-\pi \leqslant \varphi_k \leqslant \pi$, $-\pi \leqslant \psi_k \leqslant \pi$ — множество углов; $Q = \{Q_0, Q_k\}$ — множество единичных кватернионов, где $Q_0 = Q_1 Q_2 \ldots Q_n$ — результирующий кватернион формального белка; $V = \{v_{ij}\}$, $i = 1, 2, 3, 4$, $j \geqslant i$ — множество коэффициентов, где v — функция, определенная над элементами

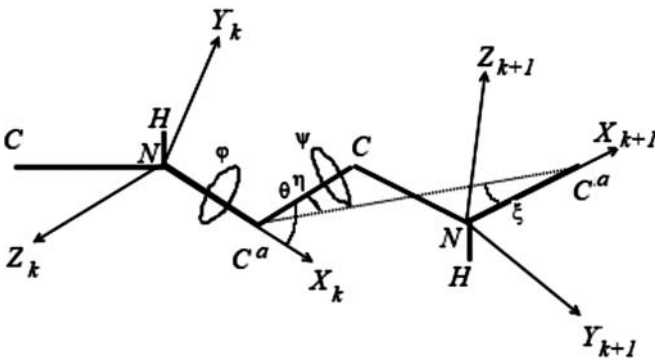

Рис. 3. Пространственная конфигурация скелета белка

результирующего кватерниона Q_0 посредством следующей квадратичной формы:

$$v = -\sum_{j>i} v_{ij} q_i q_j. \quad (5)$$

Функция (5), называемая свободной энергией ФБ, может быть представлена в векторно-матричной форме:

$$v = -[Q]^{\mathrm{T}} v [Q].$$

Главным условием функционирования белка является его связывание с другим белком либо другой молекулой. Основной биофизической характеристикой связи между белками является величина свободной энергии. Чем ниже свободная энергия, тем сильнее связь.

Соответственно, свободная энергия взаимодействия между ФБ называется *энергией связи* и определяется билинейной формой:

$$w(P, Q) = -[P]^{\mathrm{T}} W [Q],$$

где $[P]$, $[Q]$ — Q-векторы соответственно первого и второго белков; $W = \{w_{ij}\}$ — матрица связи, где w_{ij} — заданные коэффициенты, $i, j = 1, 2, 3, 4$.

Также известно, что экстремальные значения билинейной формы определяются так называемым сингулярным разложением матрицы. Таким образом, P- и Q-векторы определяются путем сингулярного разложения исходной матрицы, содержащей данные о конкретной задаче.

Решение задачи распознавания образов методом иммунокомпьютинга. Определим *образ* как n-мерный вектор-столбец $X = [x_1, ..., x_n]^{\mathrm{T}}$, где $x_1, ..., x_n$ — вещественные числа и т — транспонирование.

Определим *распознавание образов* как отображение $f(X) \to \{1, ..., c\}$ любого образа X в одно из целых чисел 1, ..., c (*классы*).

Задача распознавания образов может быть сформулирована следующим способом.

Дано: 1) число классов c; 2) набор из m обучающих образов: $X_1, ..., X_m$; 3) класс любого обучающего образа: $f(X_1) = c_1, ..., f(X_m) = c_m$; 4) произвольный n-мерный вектор Z. Найти: класс вектора Z: $f(z) = ?$

Алгоритм обучения

1. Сформировать обучающую матрицу $A = [X_1, ..., X_m]^{\mathrm{T}}$ $(m \times n)$.

2. Вычислить максимальное сингулярное число s, а также левый и правый сингулярные векторы L и R обучающей матрицы по следующей итеративной схеме:

$$L_{(0)} = [1...1]^{\mathrm{T}};$$

$$R^{\mathrm{T}} = L^{\mathrm{T}}_{(k-1)} A,\ R_{(k)} = R/|R|, \text{ где } |R| = \sqrt{r_1^2 + \ldots + r_n^2};$$

$$L = AR_{(k)},\ L_{(k)} = L/|L|, \text{ где } |L| = \sqrt{l_1^2 + \ldots + l_m^2};$$

$$s_{(k)} = L^{\mathrm{T}}_{(k)} AR_{(k)},\ k = 1, 2, ...,$$

до выполнения условия $|s_{(k)} - s_{(k-1)}| < \varepsilon$,

$$s = s_{(k)},\ L = L_{(k)},\ R = R_{(k)}.$$

3. Хранить сингулярное число s.

4. Хранить правый сингулярный вектор R (как *"антитело-пробу"*).

5. Для всякого $i = 1, ..., m$ хранить компоненту l_i левого сингулярного вектора L и класс c_i, соответствующий обучающему образу X_i.

Распознавание

1. Для всякого n-мерного образа Z вычислить его энергию связи с R:

$$w(z) = Z^{\mathrm{T}} R/s$$

(s — это хранимое сингулярное число, а R — это хранимый правый сингулярный вектор обучающей матрицы A).

2. Выбрать элемент l_i, который имеет минимальное расстояние d (соответственно, максимальное *сродство* $1/d$) с w:

$$d = \min_i |w - l_i|,\ i = 1, ..., m.$$

3. Считать класс c_i искомым классом образа Z.

Применение описанного подхода. В работе [5] показано, что в реальных задачах распознавания образов (на примерах экологического мониторинга и лазерной физики) иммунная сеть превосходит нейронные сети и генетические алгоритмы как минимум в 40 раз по быстродействию и в 2 раза по безошибочности распознавания. Данные результаты (табл. 1) получены на задачах сравнительно малой размерности ($17 \times 23 \times 6$ для экологического атласа и 19×5 для лазерного диода). На основе этих результатов было предположено, что еще большее преимущество будет достигнуто в задачах большой размерности (например, 51608×41 [4]), где при-

Таблица 1

Сравнение результатов решения задачи экологического мониторинга

Алгоритм	Иммунные сети	Нейронные сети
Объем обучающей выборки, примеры	11	11
Время обучения на ПК типа Pentium-4 1,8 GHz, с	<1	45
Максимальная ошибка на обучающей выборке	0	0
Объем текстовой выборки, примеры	391	391
Суммарная ошибка на тестовой выборке	137	187
Средняя ошибка на один пример	0,35	0,48

менение нейронных сетей становится достаточно сложным.

Кроме того, данный подход применялся для прогнозирования риска возникновения чумы и распознавания голосовых сигналов.

Однако следует отметить, что данная методика применима лишь в случае распознавания образов, когда число классов распознавания не меняется со временем. Компонентам левого сингулярного вектора ставятся в соответствие распознаваемые классы объектов. Это делает невозможным получение выхода, отличного от известных классов.

Теория опасности

В 1994 г. иммунолог П. Матзингер доказал, что иммунная система отвечает на присутствие молекул, известных как сигналы опасности, которые являются побочными продуктами незапланированной смерти клеток (некроза) [6]. Дендритные клетки чувствительны к усилению сигналов опасности, что приводит к их созреванию и иммунному ответу.

Дендритная клетка имеет три состояния: незрелое (поиск антигена и оценка его опасности), полузрелое (антиген найден и оценен как безопасный), зрелое (антиген найден и оценен как опасный).

Дендритные клетки воспринимают четыре типа сигналов [6]:

1) *PAMP* (*Pathogen-Associated Molecular Patterns*) — наличие чужеродных клеток;

2) сигналы опасности — возникают в результате внезапной смерти клеток;

3) безопасные сигналы — сигналы о нормальной ожидаемой смерти клеток;

4) сигнал о воспалении — не приводит к моментальному иммунному ответу. Но усиливает три остальных сигнала. Иммунный ответ возникает в результате сочетания сигналов (рис. 4, толщина линии пропорциональна величине весового коэффициента).

Выходными сигналами дендритной клетки являются три типа сигналов:

1) со-стимуляция (сигнал о необходимости передачи обнаруженного антигена для дальнейших действий);

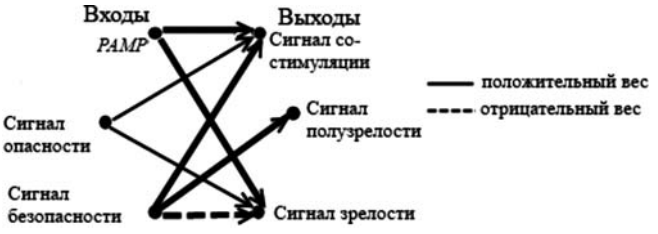

Рис. 4. Абстрактная модель обработки сигналов дендритной клеткой

2) сигнал о полузрелом состоянии дендрита (обнаружен безопасный антиген — запомнить и не реагировать);

3) сигнал о зрелом состоянии дендрита (обнаружен опасный антиген — запомнить и активировать лимфоциты).

Переход дендрита из незрелого состояния в полузрелое или зрелое определяется уровнем сигнала на его выходах и происходит при превышении некоторого установленного порога.

Изначально дендрит находится в незрелом состоянии.

Общая форма функции преобразования входных сигналов в выходные приведена ниже:

$$Output = (P_\omega \sum_i P_i + D_\omega \sum_i D_i + S_\omega \sum_i S_i)(1 + I), \quad (6)$$

где P_ω — это вес сигналов PAMP; D_ω — вес сигналов опасности; S_ω — вес сигналов безопасности; P_i, D_i, и S_i — входные сигналы вида *PAMP* (P), опасности (D), безопасности (S) соответственно; I — сигнал воспаления.

Каждый из выходных сигналов рассчитывается по формуле (6). Для этого используются различные весовые коэффициенты. Соотношения используемых при этом весовых коэффициентов получены эмпирически и приведены в табл. 2.

Гринсмит и др. [6] предложили алгоритм дендритных клеток (DCA), который включает в себя понятия сигналов опасности, безопасности и *PAMP*, влияющих на выходной сигнал дендритной клетки.

Общий вид алгоритма может быть представлен следующим образом.

Пусть входом алгоритма является множество S — это набор данных, для которых необходимо определить, опасны они или нет, а выходом — множество D — данные, помеченные как опасные и безопасные. Тогда алгоритм имеет следующий вид:

1) создать начальную популяцию дендритных клеток (D);
2) создать набор для хранения "мигрировавших" дендритов (перешедших из незрелого состояния в полузрелое или зрелое) (M);
3) для всех данных из набора S выполнить:
   3.1) создать набор дендритных клеток P, случайным образом выбранных из набора D;

Таблица 2

Соотношения между весовыми коэффициентами входных сигналов дендритной клетки. Вес сигнала PAMP используется для определения остальных весов

Сигнал	PAMP	Опасность	Безопасность
Со-стимуляция	W_1	$W_1/2$	$W_1 \cdot 1{,}5$
Сигнал полузрелости	0	0	1
Сигнал зрелости	W_2	$W_2/2$	$-W_2 \cdot 1{,}5$

3.2) для всех дендритных клеток из набора P выполнить:
   3.2.1) добавить текущий элемент данных s_i для анализа;
   3.2.2) определить уровни всех трех входных сигналов;
   3.2.3) на основе (6) вычислить выходные сигналы дендрита;
   3.2.4) переместить дендрит из множества D в множество M и добавить новый дендрит в D, если уровень сигнала на выходе со-стимуляции превысил установленный порог;
3.3) для всех дендритов в M выполнить:
   3.3.1) пометить дендрит как полузрелый, если уровень сигнала на выходе о полузрелом состоянии выше, чем на выходе о зрелом состоянии, иначе — пометить дендрит как зрелый;
3.4) подсчитать, сколько получено дендритов в полузрелом состоянии, а сколько — в зрелом; если первое число больше второго, то пометить s_i как безопасный, в противном случае — как опасный;
3.5) поместить текущий элемент данных во множество M;
4) закончить выполнение алгоритма.

Области применения дендритного алгоритма (*DCA*)

Данный алгоритм был создан для решения проблем обнаружения вторжений в вычислительные сети [6]. В ранних работах по данному алгоритму было высказано предположение о возможности его применения к решению задач построения самообучающихся систем. Однако была установлена его чувствительность к порядку предъявляемых примеров.

Кроме того, алгоритм был применен к решению задачи обнаружения сканирования портов и процессов. Оатс и др. [7] предложили использовать *DCA* для распознавания объектов мобильными роботами. *DCA* используется для классификации объектов на основе сигналов, получаемых с сенсоров робота в реальном времени.

Заключение

Таким образом, в исследованиях, посвященных искусственным иммунным сетям, можно выделить три основных направления. Все они основаны на различных идеях классической иммунологии.

В основе каждого из направлений лежит своя математическая модель, определяющая область применения разработанных алгоритмов.

Следует отметить, что до сих пор основной решаемой задачей остается обеспечение безопасности вычислительных сетей и компьютеров. При этом не все преимущества иммунной системы используются в полной мере (например, высокая скорость обучения, пороговый механизм).

Полученные к настоящему моменту результаты позволяют говорить о том, что по такому параметру, как скорость обучения, иммунные сети превосходят нейронные, что позволяет надеяться на решения проблемы длительности оперативного дообучения, свойственной нейронной сети, что в свою очередь позволит решить проблему управления технологическими объектами в реальном масштабе времени.

В целом, описанные возможности иммунных сетей позволяют сделать предположение о необходимости расширения областей применения приведенных выше алгоритмов.

Список литературы

1. **De Castro L. N., Von Zuben F. J.** Artificial Immune Systems: Part I — Basic Theory and Applications. Technical Report. 1999. 89 p.
2. **Dasgupta D., Nino L. F.** Immunological Computation: Theory and Applications. Boca Ration: CRC Press, 2008. 298 p.
3. **Cutello V., Narzisi G., Nicosia G., Pavone M.** Clonal Selection Algorithms: A Comparative Case Study using Effective Mutation Potentials, optIA versus CLONALG // Proceedings of ICARIS 2005. Berlin: Springer — Verlag. 2005. P. 31—48.
4. **Tarakanov A. O., Skormin V. A., Sokolova S. P.** Immunocomputing: Principles and Applications. New York.: Springer, 2003. 230 p.
5. **Tarakanov A. O., Tarakanov Y. A.** A comparison of immune and genetic algorithms for two real-life tasks of pattern recognition // Lecture Notes in Computer Science. Berlin: Springer—Verlag, 2004. Vol. 3239. P. 236—249.
6. **Greensmith J., Aickelin U., Tedesco G.** Information Fusion for Anomaly Detection with the DCA // Journal of Information Fusion, 2008. P. 138—152.
7. **Oates R., Greensmith J., Aickelin U.** The application of a dendritic cell algorithm to a robotic classifier // Proceedings of ICARIS'07. Berlin: Springer—Verlag. 2007. P. 204—215.

УДК 519: 816

В. В. Сафронов, д-р техн. наук, проф.,
ОАО "КБ Электроприбор", г. Саратов,
e-mail: svv@kbep.ru

Сравнительная оценка методов "жесткого" ранжирования и анализа иерархий в задаче гипервекторного ранжирования систем

Поставлена задача гипервекторного ранжирования систем. Показаны общие принципы ее решения, особенности применения метода "жесткого" ранжирования и метода анализа иерархий для различных правил свертки. Приведен численный пример.

Ключевые слова: *гипервекторное ранжирование, критерии, свертка критериев, метод анализа иерархий*

Введение. В ходе исследования систем (технических, технологических, информационных и т. п.) возникает необходимость использования векторных и многовекторных компонент [22—25]. Задачи принятия решений сводятся в этом случае к задачам многовекторного и гипервекторного ранжирования (ГВР). В работах [20, 21] осуществлены постановки задач гипервекторного ранжирования, рассмотрены характерные особенности такого класса задач, дан метод решения, основанный на методе "жесткого" ранжирования (МЖР).

Вместе с тем, отечественными и зарубежными учеными накоплен солидный опыт решения задач многокритериальной оптимизации и ранжирования. Разработаны методы, которые широко применяются в прикладных задачах: "жесткого" ранжирования [21]; анализа иерархий Т. Саати [18]; турнирной таблицы [26]; Борда [26]; равномерной оптимальности; справедливого компромисса; идеальной точки в пространстве критериев [6], минимаксный [3, 4] и многие другие [1—11, 13—16, 26]. Очевидна целесообразность применения известных методов многокритериального ранжирования в целях решения более сложной задачи гипервекторного ранжирования.

В настоящей статье рассмотрены постановка задачи гипервекторного ранжирования, общие принципы ее решения, особенности применения методов "жесткого" ранжирования и анализа иерархий и их сравнительная оценка.

Для однозначного понимания введем следующие определения:

Определение 1. Многокритериальными называют задачи, в которых векторный критерий представляет собой упорядоченное множество скалярных компонент.

Определение 2. Многовекторными называют задачи, в которых векторный критерий представляет собой упорядоченное множество векторных компонент, а каждая векторная компонента — упорядоченное множество скалярных компонент.

Определение 3. Гипервекторными называют задачи, в которых векторный критерий представляет собой упорядоченное множество многовекторных компонент, каждая многовекторная компонента — упорядоченное множество векторных компонент, а каждая векторная компонента — упорядоченное множество скалярных компонент.

Особенностями многовекторных и гипервекторных задач являются:

- численные значения и векторных, и многовекторных компонент не известны — соотношения между векторными компонентами подсистем и многовекторными компонентами систем определяются в ходе решения задачи;
- коэффициенты важности назначаются отдельно для многовекторных компонент, векторных компонент и для каждого множества скалярных критериев каждой векторной компоненты.

Заметим, что для рассматриваемого класса задач проводить ранжирование вариантов только по скалярным критериям не вполне корректно в силу различного характера свойств системы (подсистемы), отражаемых векторными компонентами, в которые входят скалярные критерии.

Кроме того, при большом числе анализируемых скалярных критериев значения коэффициентов важности становятся малыми, как и их влияние на выбор эффективных систем.

1. Постановка задачи гипервекторного ранжирования

Введем необходимые в дальнейшем обозначения:

$S = \{S_\alpha, \alpha = \overline{1, n}\}$ — множество систем;

$S_D \subseteq S$ — множество допустимых систем, для которых, в зависимости от специфики системы, должны выполняться некоторые дисциплинирующие условия: неравенства, равенства, логические условия и т. п.;

$K_{\varepsilon ji}(S_\alpha)$ — i-й скалярный критерий j-й векторной компоненты, которая входит в многовекторную компоненту с номером ε ($\varepsilon = \overline{1, \mathrm{E}}$, $j = \overline{1, r_\varepsilon}$, $i = \overline{1, r_{\varepsilon j}}$). Здесь E — число многовекторных компонент; r_ε — число векторных компонент в много-

векторной компоненте с номером ε; $r_{\varepsilon j}$ — число скалярных критериев в j-й векторной компоненте, которая, в свою очередь, входит в многовекторную компоненту с номером ε.

$K_{\varepsilon j}(S_\alpha) = \{K_{\varepsilon ji}(S_\alpha),\ i = \overline{1, r_{\varepsilon j}}\}$; $K_\varepsilon(S_\alpha) = \{K_{\varepsilon j}(S_\alpha),\ j = \overline{1, r_\varepsilon}\}$, $K(S_\alpha) = \{K_\varepsilon(S_\alpha),\ \varepsilon = \overline{1, \mathrm{E}}\}$ — соответственно множество скалярных, векторных и многовекторных компонент, характеризующих систему $S_\alpha \in S_D$;

$A_{\varepsilon j} = \{a_{\varepsilon ji},\ i = \overline{1, r_{\varepsilon j}}\}$, $A_\varepsilon = \{a_{\varepsilon j},\ j = \overline{1, r_\varepsilon}\}$, $A = \{a_\varepsilon,\ \varepsilon = \overline{1, \mathrm{E}}\}$, — соответственно, множество коэффициентов важности скалярных, векторных и многовекторных компонент, причем $\sum_{\varepsilon = 1}^{\mathrm{E}} a_\varepsilon = 1$, $\sum_{j = 1}^{r_\varepsilon} a_{\varepsilon j} = 1$, $\sum_{i = 1}^{r_{\varepsilon j}} a_{\varepsilon ji} = 1$, $j = \overline{1, r_\varepsilon}$, $\varepsilon = \overline{1, \mathrm{E}}$;

$P = \{S^0_{k_1}, S^0_{k_2}, ..., S^0_{k_{n^\pi}}\}$ — упорядоченное множество эффективных систем (кортеж Парето), $P \subseteq S_D$; элементы кортежа ранжированы в соответствии с решающими правилами так, что выполняется условие $S^0_{k_1} \succ S^0_{k_2} \succ ... \succ S^0_{k_i} \succ ... \succ S^0_{k^\pi}$, где "$\succ$" — знак отношения доминирования, $k_i \in \{1, 2, ..., n\}$. Длина кортежа равна n^π.

Допустим, известны множества A, A_ε, $A_{\varepsilon j}$, S, $K_{\varepsilon j}(S_\alpha)$, $D(S_\alpha)$, ($\alpha = \overline{1, n}$; $\varepsilon = \overline{1, \mathrm{E}}$; $j = \overline{1, r_\varepsilon}$), решающие правила. Требуется найти кортеж Парето P, для элементов которого справедливо

$$K(S^0_{k_i}) = \min_{S_\alpha \in S_D} K(S_\alpha),\ S^0_{k_i} \in P. \qquad (1)$$

2. Принципы решения задачи гипервекторного ранжирования

Рассмотрим принципы решения задачи гипервекторного ранжирования, которые не зависят от принимаемого решающего правила.

1. *Представление критериев в виде иерархической структуры.*

Скалярные критерии располагаем на нижнем (третьем) уровне иерархии и объединяем в векторные компоненты (второй уровень иерархии). Векторные компоненты — в многовекторные (первый уровень иерархии), а многовекторные — в гипервекторную компоненту (корневая вершина).

2. *Решение задачи многокритериального ранжирования по скалярным критериям каждой векторной компоненты.*

В результате решения указанной задачи будут построены частные кортежи Парето, которые позволяют однозначно определить расположение вариантов сложных систем S_α относительно других вариантов по каждой векторной компоненте. Причем выявляются как доминирующие (доминируемые), так и эквивалентные варианты.

3. *Получение количественных оценок векторных компонент.*

При использовании различных методов получаем оценки векторных компонент, которые зависят от специфики методов. Назовем такие числа *псевдозначениями* (рангами) векторных компонент.

Во всех случаях неэффективные варианты не исключаем из рассмотрения.

4. *Ранжирование систем по совокупности многовекторных компонент.*

Введение рангов либо количественных оценок векторных компонент позволяет применить один из методов многокритериального ранжирования. Число обращений к методу будет равно числу многовекторных компонент. В результате решения задачи получаем расположение вариантов по совокупности многовекторных компонент, что позволяет построить соответствующие частные кортежи Парето.

5. *Построение кортежа Парето.*

Введение рангов либо количественных оценок многовекторных компонент позволяет применить один из методов многокритериального ранжирования. В итоге и будет построен искомый кортеж Парето.

3. Особенности применения некоторых методов для решения задачи гипервекторного ранжирования

3.1. Метод "жесткого" ранжирования

Без потери общности изложение будем проводить для систем S_α, $\alpha = \overline{1, n}$, свойства которых задают с помощью критериев $K_j(S_\alpha)$, $j = \overline{1, r}$.

В ходе решения задачи будем анализировать множество упорядоченных пар систем S_k, S_l ($k = \overline{1, n}$; $l = \overline{1, n}$; $k \neq l$), а результат анализа заносить в специальную оценочную матрицу $\|C_{kl}\|$. Сущность метода заключается в следующем.

1. На основе попарного сравнения систем S_k, S_l ($k = \overline{1, n}$; $l = \overline{1, n}$; $k \neq l$) определяем элементы C_{kl} оценочной матрицы $\|C_{kl}\|$. Значения элементов C_{kl} подбирают таким образом, чтобы отсечь неэффективные системы.

У эквивалентных систем S_k, S_l все соответствующие критерии равны. Полагаем $C_{kl} = 1$, $C_{lk} = 1$.

К числу неэффективных систем отнесем варианты, у которых:

а) все значения критериев k-й системы хуже, чем у l-й системы, тогда полагаем $C_{kl} = N_2 \gg 1$;

б) значения $m (m < r)$ критериев k-й системы хуже соответствующих значений критериев l-й системы при равных соответствующих значениях остальных критериев этих систем; тогда полагаем $C_{kl} = N_3$, $1 \ll N_3 < N_2$.

Если же для систем k, l имеем лучшие, худшие и, возможно, равные критерии, то значение C_{kl} определим по методу, изложенному в работах [15, 16].

Обозначим N_{kl}^{+}, N_{kl}^{-}, $N_{kl}^{=}$ соответственно подмножества номеров лучших, худших и равных критериев для каждой пары вариантов систем S_k, S_l ($k = \overline{1, n}$; $l = \overline{1, n}$, $k \neq l$). Будем осуществлять попарное сравнение систем S_k, S_l на основе анализа критериев $K_j(S_k)$, $K_j(S_l)$, $j = \overline{1, r}$. Для возможных значений подмножеств номеров N_{kl}^{+}, N_{kl}^{-}, $N_{kl}^{=}$ введем следующие значения элементов оценочной матрицы $\|C_{kl}\|$:

если $N_{kl}^{+} = \varnothing$, $N_{kl}^{-} = \varnothing$, $N_{kl}^{=} = \{\overline{1, r}\}$,
то $C_{kl} = 1$, $C_{lk} = 1$; (2)

если $N_{kl}^{+} = \{\overline{1, r}\}$, $N_{kl}^{-} = \varnothing$, $N_{kl}^{=} = \varnothing$,
то $C_{kl} = N_2$, $C_{lk} = 0$; (3)

если $N_{kl}^{+} = \varnothing$, $N_{kl}^{-} = \{\overline{1, r}\}$, $N_{kl}^{=} = \varnothing$,
то $C_{kl} = 0$, $C_{lk} = N_2$; (4)

если $N_{kl}^{+} \neq \varnothing$, $N_{kl}^{-} = \varnothing$, $N_{kl}^{=} \neq \varnothing$,
то $C_{kl} = N_3$, $C_{lk} = 0$; (5)

если $N_{kl}^{+} = \varnothing$, $N_{kl}^{-} \neq \varnothing$, $N_{kl}^{=} \neq \varnothing$,
то $C_{kl} = 0$, $C_{lk} = N_3$; (6)

если $N_{kl}^{+} \neq \varnothing$, $N_{kl}^{-} \neq \varnothing$, $|N_{kl}^{=}| \geqslant 0$, (7)

то определим C_{kl} в виде [15, 16]:

$$C_{kl} = \sum_{j \in N_{kl}^{+}} a_j \left(\sum_{j \in N_{kl}^{-}} a_j \right)^{-1}, \; C_{lk} = C_{kl}^{-1}. \qquad (8)$$

2. Для формулировки решающих правил введем характерные числа: H_l — число элементов в l-м столбце оценочной матрицы, значения которых больше единицы; M_l — число элементов в l-м столбце той же матрицы, значения которых меньше единицы; $C_{kl\max}$ — максимальное значение элемента в l-м столбце матрицы $\|C_{kl}\|$.

3. Для реализации "жесткого" ранжирования перейдем от одношагового процесса поиска приоритетного расположения систем к многошаговому процессу [2].

Решающие правила "жесткого" ранжирования

3.1. Ранжирование необходимо проводить среди эффективных систем по шагам. Число шагов $t \leqslant (n - 1)$.

3.2. На каждом шаге t ($t = 1, 2, ..., n - 1$) необходимо:

найти числа $H_l^{(t)}$, $M_l^{(t)}$, $C_{kl\max}^{(t)}$ и определить лучшую систему S_j с минимальным значением $H_j^{(t)}$;

номер j занести в множество P;

исключить из оценочной матрицы j-ю строку и j-й столбец.

Если системы с номерами $l_j \in L_{k(t)} = \{l_1, l_2, ..., l_j, ..., l_{k(t)}\}$ имеют одинаковые минимальные значения $H_{l_j}^{(t)}$, то лучшей является система S_{l_j} с максимальным значением $M_{l_j}^{(t)} = \max\limits_{l_j \in L_{k(t)}} M_{l_j}^{(t)}$.

3.3. Если системы с номерами $l_j \in L_{k(t)} = \{l_1, l_2, ..., l_j, ..., l_{k(t)}\}$ имеют соответственно одинаковые значения $H_{l_j}^{(t)}$, $M_{l_j}^{(t)}$, то лучшей является система S_{l_j} с минимальным значением $C_{l_j}^{(t)} = \min\limits_{l_j \in L_{k(t)}} C_{kl_j\max}^{(t)}$.

3.4. Если лучшие системы имеют соответственно равные значения $H_l^{(t)}$, $M_l^{(t)}$, $C_{kl\max}^{(t)}$, то такие системы считают эквивалентными.

Подробно метод изложен в работе [21].

Докажем две *теоремы*, имеющие важное прикладное значение.

Теорема 1. Если в l-м ($l \in \{\overline{1, n}\}$) столбце оценочной матрицы максимальный элемент равен значению N_3 или значению N_2, то l-й вариант системы не принадлежит множеству эффективных решений.

Доказательство. Из условия теоремы следует, что хотя бы для одного из вариантов k ($k \in \{\overline{1, n}\}$, $k \neq l$) выполняется одно из условий (3), (5). Таким образом, вариант l доминируется вариантом k. Значит, согласно определению множества Парето, l-й вариант не может принадлежать множеству эффективных решений. *Теорема доказана.*

Теорема 2. Множество неэффективных систем не зависит от значений коэффициентов важности критериев.

Доказательство. Из теоремы 1 следует, что если l-й вариант принадлежит множеству неэффективных решений, то $C_{kl\max} = N_3$ или $C_{kl\max} = N_2$.

В этом случае хотя бы один из элементов C_{kl} оценочной матрицы принимает одно из значений:

N_3, когда вариант системы k имеет по сравнению с вариантом системы l только лучшие и равные значения критериев (условие (5));

N_2, когда вариант системы k имеет по сравнению с вариантом системы l только лучшие значения критериев (условие (3)).

Значения N_3, N_2 введены автономно и не зависят от коэффициентов важности критериев. *Теорема доказана*.

3.2. Метод анализа иерархий

Американским математиком Т. Саати разработан метод решения многокритериальных задач, названный методом анализа иерархий (МАИ) [18]. Идеи метода были изложены еще в книге [17]. В оригинале, на английском языке, книга была опубликована в 1968 году. Метод нашел широкое применение для анализа сложных систем [18, 19, 25]. Свое дальнейшее развитие МАИ получил в очередной книге автора [19], изданной в нашей стране в 2008 году. При изложении метода Т. Саати делает предположение, что в общем случае матрицы парных сравнений не являются совместными (состоятельными), хотя и раскрывает свойства состоятельных матриц, показывает механизм их получения, но осознанно уходит от этих свойств [17].

Российский ученый В. Д. Ногин предложил модификацию МАИ, основанную на условии совместности матрицы парных сравнений [12]. В этом случае достаточно просто вычисляются собственный вектор и собственные значения матрицы, а, следовательно, и коэффициенты важности частных целей. Им же предложено использовать иную свертку критериев, а именно, максиминную свертку Ю. Гермейера [3, 12], которую автор называет нелинейной сверткой [12].

Поскольку МАИ широко освещается в книгах и журнальных статьях, ограничимся кратким изложением основных этапов метода и раскроем особенности применения метода для решения задачи ГВР. В основе МАИ лежат следующие построения [18]:

1. Структуризация задачи — определение целей, критериев, альтернатив.

2. Формирование матриц парных сравнений для критериев в целом (применительно к уровню 2) и по отдельным критериям (применительно к уровню 3). Здесь экспертами используется специальная шкала относительной важности (шкала сравнения), разработанная Т. Саати [17—19].

3. Вычисление коэффициентов важности (локальных приоритетов) на основе анализа каждой из матриц парных сравнений. С этой целью определяют собственные векторы матриц парных сравнений и осуществляют нормирование векторов. Проверяется согласованность суждений экспертов на основе вычисления индекса согласованности и отношения согласованности (ОС). ОС не должно превышать 20 %.

4. Определение глобальных приоритетов (количественного индикатора качества) каждой из систем (альтернатив), ранжирование систем и выбор наилучшей системы.

Рассмотрим теперь особенности применения МАИ для решения задачи гипервекторного ранжирования. Методика справедлива как для линейной свертки, так и для максиминной (нелинейной) свертки Ю. Гермейера.

Методика решения задачи гипервекторного ранжирования МАИ

1. Провести анализ исходной информации, формирование критериев оценок систем, построить матрицы парных сравнений, обладающие свойством совместности.

2. Ранжировать системы методом анализа иерархий по множеству скалярных критериев каждой векторной компоненты.

3. Вычислить значения глобальных приоритетов векторных компонент (построить частные кортежи Парето по векторным компонентам).

4. Ранжировать системы методом анализа иерархий по множеству векторных компонент.

5. Определить значения глобальных приоритетов многовекторных компонент (построить частные кортежи Парето по многовекторным компонентам).

6. Ранжировать системы методом анализа иерархий по множеству многовекторных компонент. Построить кортеж Парето.

7. Провести анализ результатов решения.

8. В случае необходимости уточнить исходные данные, изменить элементы матриц парных сравнений. Перейти к шагу 2. В противоположном случае перейти к шагу 9.

9. Конец решения.

К сожалению, применение МАИ с линейной и нелинейной свертками может привести к получению неэффективных решений. В соответствии с теоремой С. Карлина применение линейной свертки справедливо, когда множество векторных оценок строго выпукло, ограничено и замкнуто [6, 7, 8], т. е. для очень узкого класса задач. На этот факт еще раз обратил внимание исследователей, применяющих для решения многокритериальных задач метод анализа иерархий, В. Д. Ногин. Им предложено вместо линейной свертки применять нелинейную свертку [12]. Ю. Б. Гермейером доказана теорема о построении Парето-оптимальных решений для невыпуклых многокритериальных задач. Однако, как отмечено в работах [3, 7], если на частные критерии не накладывать никаких дополнительных ограничений, то решения, получаемые по Гермейеру, могут быть и не оптимальными по

Парето. В работе [10] показано, что в число возможных решений входит и неэффективное решение.

В целях устранения этих проблем предлагается применять специальные критерий и методику. Для их формулировки введем необходимые определения.

Определение 4. *Опорный кортеж Парето* P — это упорядоченное множество только эффективных вариантов, построенное в ходе решения задач многокритериального, многовекторного или гипервекторного ранжирования с использованием метода "жесткого" ранжирования.

Определение 5. *Псевдокортеж Парето* P_{nq} — это упорядоченное множество эффективных и неэффективных вариантов, построенное в ходе решения задач многокритериального, многовекторного или гипервекторного ранжирования с использованием метода, отличного от МЖР, $q = \overline{1, Q}$.

В частном случае в псевдокортеж Парето входят только эффективные варианты.

Определение 6. *Истинный кортеж Парето* P_{uq} — это упорядоченное множество эффективных вариантов, построенное на основе псевдокортежа Парето, у которого исключены неэффективные варианты, $q = \overline{1, Q}$.

Допустим, что с использованием МЖР, а также других, интересующих нас методов из заданного множества, построены, соответственно, опорный кортеж Парето P и q псевдокортежей P_{nq}, $q = \overline{1, Q}$. Справедлив следующий критерий построения истинных кортежей Парето P_{uq}, $q = \overline{1, Q}$.

Критерий. Для построения истинных кортежей Парето необходимо и достаточно из соответствующих псевдокортежей Парето выбрать, не нарушая порядок следования, лишь варианты, номера которых указаны в опорном кортеже Парето.

Иначе

$$P_{uq} = (P_{nq} \cap P,\ q = \overline{1, Q}\ ). \qquad (9)$$

Доказательство. *Необходимость.* В соответствии с теоремой 1 в опорный кортеж Парето входят только эффективные варианты. Следовательно, выбор указанного кортежа является оправданным и необходимым условием решения задачи.

Достаточность. После выполнения операции (9) в истинные кортежи Парето войдут лишь эффективные варианты, которые включены в опорный кортеж Парето, и никакие другие. Отличие в общем случае будет заключаться лишь в порядке следования эффективных вариантов, который зависит от конкретного решающего правила.

Для корректного решения задачи предлагается следующая *методика построения истинных кортежей Парето.*

1. Решить задачу ГВР с использованием МЖР и иных методов, в частности МАИ. В результате:

а) с помощью МЖР будет построен опорный кортеж Парето и определено подмножество неэффективных систем;

б) иными методами будут построены псевдокортежи Парето.

2. С учетом информации об эффективных системах, которые имеются в кортеже P, исключить из псевдокортежей Парето неэффективные системы. В итоге получим истинные кортежи, в которых расположены только эффективные системы, в порядке, определяемом конкретным методом многокритериального ранжирования.

4. Численный пример

Обратимся к примеру, приведенному в работе [24]. Допустим, необходимо построить упорядоченное множество эффективных моделей разработки программного обеспечения (ПО) (кортеж Парето) для проекта, предполагающего автоматизировать процесс некоторой гипотетической системы управления коммуникациями в организации. Значения критериев для десяти возможных моделей разработки ПО приведены в таблице.

Значения критериев, характерные для моделей разработки ПО

Критерии	Модели разработки ПО									
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Характеристики требований к проекту (K_1)										
K_{11}	1	1	2	2	2	1	2	1	2	2
K_{12}	1	1	2	1	3	1	2	1	1	2
K_{13}	1	1	1	1	2	1	1	1	1	2
K_{14}	1	1	2	1	2	2	1	1	1	2
Характеристики команды разработчиков (K_2)										
K_{21}	1	1	2	1	2	1	2	2	2	2
K_{22}	2	2	2	1	1	1	2	2	1	1
K_{23}	1	1	1	1	2	2	1	1	1	2
K_{24}	1	1	2	2	2	1	1	1	2	2
Характеристики пользователей и заказчика (K_3)										
K_{31}	1-2	1-2	1-2	2-3	3-4	3-4	2-3	1-2	2-3	2-4
K_{32}	1	1	1	1	2	2	2	1	1	3
Характеристики типов проектов и рисков (K_4)										
K_{41}	2	2	1	2	1	2	1	1	2	2
K_{42}	2	2	1	1	2	2	2	2	1	1
K_{43}	3	3	3	3	1	2	3	3	3	2
K_{44}	1-2	1-2	1-3	1-2	1-2	1-2	1-2	1-2	1-2	2-3
K_{45}	1-3	2-3	2-3	2-3	2-3	1-2	2-3	2-3	1-2	1-2
K_{46}	1-3	3-4	3-4	2-3	1-2	1-2	3-4	2-3	3-4	1-4
Характеристики процесса разработки (K_5)										
K_{51}	1	1	1	3	2	1	1	1	3	2
K_{52}	2	2	1	1	1	1	2	1	1	1
K_{53}	2	2	2	3	3	3	2	2	3	1
K_{54}	1	1	1	1	2	2	2	1	1	2

Задачу гипервекторного ранжирования будем решать с использованием трех методов: "жесткого" ранжирования; МАИ с линейной сверткой критериев; МАИ с нелинейной сверткой критериев.

В результате решения получено:

1. При использовании МЖР опорный кортеж Парето $P = \langle S_7, S_5, S_3, S_{10}, S_2\rangle$. Модели S_1, S_4, S_6, S_8, S_9 оказались неэффективными.

2. При использовании МАИ с линейной сверткой псевдокортеж Парето $P_{n1} = \langle S_7, S_2, S_1, S_9, S_5, S_4, S_{10}, S_3, S_6, S_8\rangle$.

3. При использовании МАИ с нелинейной сверткой получим псевдокортеж Парето $P_{n2} = \langle S_4 \sim S_7 \sim S_9, S_6 \sim S_{10}, S_5, S_1 \sim S_2 \sim S_3 \sim S_8\rangle$.

Нетрудно видеть, что если в качестве решающего правила применяется МАИ с различными свертками как линейной, так нелинейной, то в первую тройку лучших систем могут попасть и заведомо неэффективные системы. Более того, эффективные системы могут располагаться после неэффективных (например, неэффективная система S_1 на третьем месте в псевдокортеже P_{n1}, неэффективная система S_4 на первом месте в псевдокортеже P_{n2}). Применяя предлагаемые *критерий* и *методику*, получим истинные кортежи Парето: $P_{u1} = \langle S_7, S_2, S_5, S_{10}, S_3\rangle$, $P_{u2} = \langle S_7, S_{10}, S_5, S_2 \sim S_3\rangle$, в которые входят только эффективные системы.

Заключение

Рассмотрены общие принципы решения задач гипервекторного ранжирования. Раскрыты особенности решения задач при использовании:

метода "жесткого" ранжирования;

метода анализа иерархий при различных свертках критериев: линейной и нелинейной.

Решения, получаемые с использованием метода анализа иерархий, как и многих других методов, могут быть и не оптимальными по Парето. Численный пример еще раз подтвердил справедливость указанных выводов.

На наш взгляд, для класса задач, которые могут быть решены с помощью метода анализа иерархий, целесообразно применение метода "жесткого" ранжирования в целях сравнительной оценки, сопоставления результатов и отсеивания неэффективных решений.

Сформулированный критерий построения эффективных вариантов и соответствующая методика позволяют получать корректные решения задач многокритериального, многовекторного и гипервекторного ранжирования при использовании в качестве решающего правила метода анализа иерархий.

Список литературы

1. **Батищев Д. И.** Методы оптимального проектирования: учеб. пособие для вузов. М.: Радио и связь, 1984. 248 с.
2. **Белкин А. Р., Левин М. Ш.** Принятие решений: комбинаторные модели аппроксимации информации. М.: Наука, 1990. 160 с.
3. **Гермейер Ю. Б.** Введение в теорию исследования операций. М.: Наука, 1971. 383 с.
4. **Гуткин Л. С.** Оптимизация радиоэлектронных устройств. М.: Сов. радио, 1975. 368 с.
5. **Денисов А. А., Колесников Д. Н.** Теория больших систем управления: учеб. пособие для вузов. Л.: Энергоиздат, Ленингр. отд-ние, 1982. 288 с.
6. **Дубов Ю. А., Травкин С. И., Якимец В. Н.** Многокритериальные модели формирования и выбора вариантов систем. М.: Наука, 1986. 296 с.
7. **Захаров И. Г.** Обоснование выбора. Теория практики. СПб.: Судостроение, 2006. 528 с.
8. **Карлин С.** Математические методы в теории игр, программировании и экономике. М.: Сов. радио, 1964. 838 с.
9. **Ларичев О. И.** Наука и искусство принятия решений. М.: Наука, 1979. 200 с.
10. **Михалевич В. С., Волкович В. Л.** Вычислительные методы исследования и проектирования сложных систем. М.: Наука, 1982. 286 с.
11. **Моисеев Н. Н.** Математические задачи системного анализа. М.: Наука. Гл. ред. физ.-мат. лит., 1981. 488 с.
12. **Ногин В. Д.** Упрощенный вариант метода анализа иерархий на основе нелинейной свертки критериев // Журнал вычислительной математики и математической физики. 2004. Т. 44, № 7. С. 1259—1268.
13. **Перегудов Ф. И., Тарасенко Ф. П.** Введение в системный анализ: учеб. пособие для вузов. М.: Высш. школа, 1989. 367 с.
14. **Подиновский В. В., Гаврилов В. М.** Оптимизация по последовательно применяемым критериям. М.: Сов. радио, 1975. 192 с.
15. **Руа Б.** К вопросу принятия многокритериального решения / Перевод № А-10849. М.: Всесоюзный центр переводов научно-технической литературы и документации, 1977. 10 с.
16. **Руа Б.** Проблемы и методы решений в задачах с многими целевыми функциями // Вопросы анализа и процедуры принятия решений: сборник ст. М.: Мир, 1976. С. 20—58.
17. **Саати Т. Л.** Математические модели конфликтных ситуаций / Пер. с англ. под ред. И. А. Ушакова. М.: Сов. радио, 1977. 304 с.
18. **Саати Т. Л.** Принятие решений. Метод анализа иерархий / Пер. с англ. М.: Радио и связь, 1993. 320 с.
19. **Саати Т. Л.** Принятие решений при зависимостях и обратных связях: Аналитические сети. Пер. с англ. / Науч. ред. А. В. Андрейчиков, О. Н. Андрейчикова. М.: Издательство ЛКИ, 2008. 360 с.
20. **Сафронов В. В.** Гипервекторное ранжирование сложных систем // Информационные технологии. 2003. № 5. С. 23—26.
21. **Сафронов В. В.** Основы системного анализа: методы многовекторной оптимизации и многовекторного ранжирования. Саратов: Научная книга, 2009. 329 с.
22. **Сафронов В. В., Григорьев И. В., Ткачук А. В.** и др. Решение задач совершенствования системы образования с использованием методов ранжирования // Информационные технологии. 2008. № 11. С. 52—57.
23. **Сафронов В. В., Жебраков А. С.** Использование математического аппарата гипервекторного ранжирования для выбора энергосиловых установок летательных аппаратов // Вестник Самарского государственного аэрокосмического университета им. академика С. П. Королева. 2009. № 3 (19). Ч. 1. С. 74—82.
24. **Сафронов В. В., Федорец О. Н.** Метод построения эффективных моделей разработки программного обеспечения // Информационные технологии. 2010. № 1. С. 34—39.
25. **Семенов С. С., Харчев В. Н., Иоффин А. И.** Оценка технического уровня образцов вооружения и военной техники. М.: Радио и связь, 2004. 552 с.
26. **Трахтенгерц Э. А.** Компьютерная поддержка принятия согласованных решений // Приложение к журналу "Информационные технологии". 2002. № 3. 24 с.

УДК 621.385.6.01

Е. Ю. Апарина, адъюнкт,
Военная академия связи им. С. М. Буденного,
А. Н. Бегаев, канд. техн. наук,
руководитель службы проектов,
В. Н. Куделя, д-р техн. наук, гл. специалист,
ОАО "Институт сетевых технологий",
e-mail: kvn@int.spb.ru

Проблемы и решения по доставке информации приложений реального времени в IP-сетях

Выполнен анализ проблем и существующих решений по доставке информации приложений реального времени в IP-сетях. Представлен общий механизм гарантированной доставки информации, позволяющий обеспечить требуемую вероятность доставки IP-пакетов с информацией приложений реального времени без существенного усовершенствования сетевого оборудования

Ключевые слова: *передача аудио-, видеосигнала и данных, IP-сеть, широковещательное телевидение, потери пакетов*

Введение

Операторы телекоммуникационных услуг стараются предоставлять разнообразные услуги с тем, чтобы максимально увеличить доход, получаемый с каждого пользователя, и сохранить конкурентоспособность. Интересы операторов в большей степени смещаются в сторону предложения взаимосвязанных услуг *triple-play* (передачи аудио, видеосигнала и данных) через IP-сети, при этом приложения реального времени представляют собой ключевой компонент для роста их бизнеса. В этой связи делаются значительные инвестиции в развитие IP-сетей для обеспечения широковещательной трансляции программ телевидения (IPTV), видео по запросу (VoD), видеосвязи высокой четкости (HD Video), голосовой связи высокой четкости (HD Audio) и т. д., чтобы дополнить свои уже существующие сервисы, предоставляя новые услуги приложений реального времени. Однако имеется ряд противоречий между предлагаемым в настоящее время качеством доставки информации приложений реального времени в IP-сетях и требованиями развлекательной индустрии (*entertainment-grade*) к качеству изображения и звука.

Способность операторов предложить доступные и качественные телекоммуникационные услуги для провайдеров аудио- и видеоконтента (контент-провайдеров) становится жизненной необходимостью, залогом сохранения конкурентоспособности в отрасли и мобилизует разработчиков на поиски решений по гарантированной доставке потоков информации в реальном времени по IP-сети одновременно миллионам потребителей.

1. Проблемы доставки аудио- и видеопотоков по IP-сетям

Основная проблема для операторов телекоммуникационных услуг заключается в том, что аудио- и видеопотоки создают совсем другой профиль нагрузки, нежели традиционные приложения, для которых IP-сети изначально проектировались. Для обычных приложений типичны кратковременные всплески активности с очень большими объемами передаваемой информации, в то время как приложения для коммуникации в реальном времени загружают сеть непрерывным и относительно равномерным потоком данных. Другая характерная особенность потоков в реальном времени — их чувствительность к задержкам: если пакет с данными задержится или потеряется, то сторона получателя сразу отреагирует ухудшением звука или изображения. Даже небольшие сбои при передаче пакетов могут привести к значительному снижению качества.

Согласно отраслевым нормам, качество восприятия видеоизображения считается приемлемым, если в течение двух часов передачи случается не более одного видимого ухудшения изображения, поэтому для контент-провайдеров, предоставляющих, например, услуги телевидения высокой четкости (*High Definition Television*, HDTV) максимальный уровень потерь пакетов 10^{-6} является базисным требованием на рынке телекоммуникационных услуг. Для стандартного телевидения (*Standard Definition TV*, SDTV) необходимый уровень потерь пакетов составляет не более 10^{-5}.

Для иллюстрации сути проблемы пакетных потерь рассмотрим эту проблему на примере доставки видеосигналов широковещательного телевидения. Видеосигнал передается по IP-сети в сильно сжатом виде. Именно по этой причине потеря даже единственного IP-пакета с видеоконтентом может привести к заметному ухудшению качества видеоизображения.

Видеокодеки обычно получают поток в стандарте *Moving Pictures Expert Group-2* (MPEG-2) или MPEG-4, который состоит из трех типов кадров: **I**, **P** и **B**.

Кадры типа **I** являются внутренними кадрами (*intra frames*) и содержат информацию для описания целого кадра в пределах видеопотока. Это дубликаты отдельных кадров, которые могут использоваться для воссоздания всей информации об изображении в пределах потока.

Тип **P** — это кадры предсказания (*predictive frames*); они используют информацию предыдущих кадров типа **I** или **P** для воссоздания самих себя в качестве полного изображения.

Кадры типа **B** являются кадрами двунаправленного предсказания (*bidirectional*), для полной прорисовки они нуждаются в информации как из предыдущих, так и из последующих кадров **I** и (или) **P** последовательности.

Для создания группы изображений кодер или декодер стандарта MPEG использует в потоке последовательность кадров типа **I** и следующих за ними кадров типа **P** или **B**. Группа изображений — это некоторое число кадров в промежутке между следующими друг за другом кадрами типа **I**.

Кодирование последовательности группы изображений в потоке MPEG позволяет значительно снизить требования к пропускной способности, достигая более управляемых диапазонов скорости передачи, характерных для сегодняшних сетей. Обратной стороной снижения нагрузки на сеть является зависимость качества изображения от надежности доставки пакетов по сети. Утеря кадра типа **I** может привести к внедрению в передачу явно видимых ложных изображений (называемых артефактами), таких как пикселизация, укрупнение составных частей видеоизображения, что серьезно ухудшает впечатление потребителя от видеопросмотра.

Таким образом, одной из основных проблем при использовании контент-провайдерами телекоммуникационных услуг является проблема наличия пакетных потерь IP-сети, которая обычно связывается с принципиальной "ненадежностью" IP-протокола, а ее решение — с наличием высокоэффективных механизмов защиты от потерь.

2. Сети нового поколения. Решения по надежной доставке потоков информации реального времени

Архитектура IP-сети нового поколения (IP-NGN) Cisco для широковещательного телевидения поддерживает разнесение путей для доставки двух потоков "живой" видеопрограммы как от одиночного, так и от двойного источника для обеспечения максимальной надежности [1]. Получение видеоизображения от двух источников (*dual-live*) подразумевает возможность для клиента получать идентичный видеоконтент с одного или двух источников по двум раздельным путям (рис. 1, см. вторую сторону обложки). При возникновении ошибки в видеопотоке получатель может переключиться с одного потока или источника на другой. Схема *dual-live*, использующая разнесение путей, позволяет избежать перерывов в передаче, которые могут возникнуть вследствие затрат времени на пересчет маршрута и в результате потерь кадров типа **I** в потоке широковещательной передачи видеоизображений формата MPEG по сети.

Для того чтобы архитектура *dual-live* была эффективной, предлагается применять географическое разнесение первичного и вторичного видеоисточников и использовать механизм типа *anycast* на стороне получателя.

Очевидно, что это решение позволит оператору поддерживать непрерывность широковещательной видеотрансляции и оставить максимально благоприятное впечатление у зрителя от просмотра популярных видеопрограмм. Однако это решение не является каким-либо общим инструментом надежной доставки информации приложений реального времени по IP-сети. Кроме того, предлагаемое решение не позволяет предоставлять услуги широковещательной трансляции IPTV по IP-сети с уровнем потерь пакетов выше 10^{-5}. Этот недостаток исключает возможность использования IP-сетей обычной архитектуры, а также беспроводных сетей[1], для трансляции видеопрограмм.

Указанные недостатки являются одним из основных аргументов в пользу необходимости поиска новых решений по гарантированной доставке пакетов информации в реальном времени по IP-сети как перспективного механизма дальнейшего наращивания объема телекоммуникационных услуг на рынке сервисов приложений реального времени.

3. Решение по гарантированной доставке пакетов в IP-сети

Для информации приложений реального времени механизмы протокола контроля передачи (*Transmission Control Protocol*, *TCP*) оказываются непригодны для улучшения качества отправления, так как обеспечивается доставка данных, но не гарантируется время их доставки. Протокол передачи дейтаграмм пользователя (*User Datagram Protocol*, *UDP*), используемый приложениями реального времени, обеспечивает быстроту, но не гарантирует доставку данных.

Возникает закономерный вопрос: есть ли другие механизмы, обеспечивающие как быстроту, так и

[1] В беспроводных сетях потери пакетов случаются не только вследствие перегрузок, но и вследствие замирания, т. е. уровень потерь значительно выше, чем в проводных сетях. Кроме того, потери носят произвольный характер.

гарантированность доставки потоков данных реального времени по IP-сети?

В настоящее время применяются два основных подхода. Первый их них состоит в обеспечении доставки с требуемым качеством за счет поддержки всех необходимых параметров сети при глобальном усовершенствовании сетевого оборудования. Именно на этом подходе, в основном, и базируется решение по надежной доставке потоков широковещательной видеотрансляции в IP-NGN Cisco (рис. 1, см. вторую сторону обложки).

Второй подход основан на использовании таких механизмов, которые обеспечивали бы приемлемое качество при малой зависимости от характеристик сети. Из механизмов второго подхода наиболее распространенным и поддерживаемым во множестве сетей является приоритетное обслуживание. Это целый набор функций, посредством которых обеспечивается приоритет определенных типов данных в сети, таких как аудио или видео, по сравнению с менее чувствительными к времени передачи элементами, что позволяет влиять на уровень потерь пакетов приложений реального времени. Однако эксперты предполагают, что в ближайшие два-три года мировой ежемесячный IP-трафик вырастет до 11 эксабайт при среднегодовом темпе роста в сложных процентах (CAGR) более 56 %, и лидирующее положение в этом трафике будет занимать информация видеоприложений [1]. То есть для приложений реального времени обеспечить требуемый уровень потерь будет все труднее, а использование механизма приоритетного обслуживания уже и сейчас не дает возможности потребителям, получить качество видеотрансляций (*Quality of Experience*, *QoE*) хотя бы на уровне качества современных кабельных или спутниковых телевизионных сервисов.

Так как имеется множество приложений реального времени, необходим универсальный инструмент, который обеспечивал бы как время, так и вероятность доставки информации по IP-сети. Для достижения требуемого уровня потерь пакетов в IP-сети предлагается способ [2], представленный на рис. 2 (см. вторую сторону обложки).

Суть механизма гарантированной доставки пакетов с информацией реального времени по IP-сети заключается в том, что для повышения вероятности доставки пакетов до потребителя осуществляется одновременная отправка сервером гарантированной доставки как пакета, так и его копий по независимым кратчайшим маршрутам. Независимые маршруты — это маршруты, не имеющие общих элементов.

Сервер гарантированной доставки на основании информации о состоянии сети формирует несколько независимых маршрутов до каждого получателя аудио-, видеоконтента. Данный механизм может использоваться совместно с процедурой маршрутизации от источника (*Source Routing*). В соответствии с этой процедурой задается в отправляемом в сеть пакете полный маршрут его следования через все промежуточные маршрутизаторы.

Получение услуги реального времени в этом случае подразумевает возможность для клиента получать аудио- и видеоконтент, восстановленный из нескольких потоков. При этом потоки не обязательно должны быть идентичными. Например, вариации в совокупности потоков пакетов с кадрами **I**, **P** и **B** обеспечат возможность полного восстановления видеоконтента.

Предложенный механизм гарантирует как время, так и вероятность доставки пакета, не предъявляя дополнительных требований к вероятностно-временным характеристикам сети.

4. Обоснование механизма приращения вероятности доставки пакетов в IP-сети

Предложенный механизм гарантированной доставки пакетов IP-сети предполагает передачу одних и тех же пакетов по нескольким сетевым маршрутам. Сетевой маршрут включает в себя все транзитные узлы и ребра сети между узлом-отправителем и узлом-получателем.

Тогда критерием доставки пакета является наличие хотя бы одного из нескольких маршрутов доставки как самого пакета, так и его копий между рассматриваемыми узлами (связность маршрутами узла-отправителя и узла-получателя). Предположим, что имеется перечень возможных маршрутов доставки для пакета и его копий между узлом-отправителем s и узлом-получателем t в виде списка элементов, входящих в каждый маршрут. Вероятность доставки пакета P_m любым маршрутом μ_m, где $m = \|M\|$ — мощность множества маршрутов между узлом-отправителем и узлом-получателем, можно вычислить по формуле последовательного соединения $P_m = r_s \cdot p_{s2} \cdot r_2 \cdot p_{23} \cdot \ldots \cdot r_i \cdot p_{ij} \cdot \ldots \cdot p_{kt} \cdot r_t$, где r_i — надежностные характеристики узла, например коэффициент оперативной готовности; $p_{ij} = 1 - q_{ij}$, а q_{ij} — вероятность потери пакета k-м ребром маршрута, при этом q_{ij} учитывает по формуле последовательного соединения как вероятность потери пакета, обусловленную надежностными характеристиками ребра, так и вероятность потери пакета вследствие искажений и перегрузок сети.

В этом случае искомая вероятность доставки пакета P_{st} зависит от вероятности доставки пакета и его копий по каждому маршруту. Таким образом, вероятности доставки пакета ставится в соответствие вполне определенная вероятность связности маршрутами узла-отправителя и узла-получателя.

В общем случае маршруты будут зависимы, поскольку любой элемент сети может входить в разные маршруты. То есть P_{st} зависит от вероятности доставки пакета по каждому отдельному маршруту

и от вариантов пересечений этих маршрутов по общим элементам. Обозначим вероятность доставки пакета, которая обеспечивается первыми маршрутами i из m, через P_i, $i = 1, ..., m$. Добавление очередного $(i + 1)$-го маршрута с вероятностью доставки копии пакета P_{i+1} приведет к увеличению вероятности доставки пакета, которая будет определяться объединением двух событий: имеется хотя бы один маршрут (либо из i первых, либо $(i + 1)$-й). Вероятность наступления этого объединенного события с учетом возможной зависимости наличия $(i + 1)$-го маршрута и первых i

$$P_{i+1} = P_i + P_{i+1} - P_{i+1}P_{i/(i+1)}, \quad (1)$$

где $P_{i/(i+1)}$ — вероятность наличия хотя бы одного из первых i маршрутов при условии, что имеется $(i + 1)$-й маршрут.

Из определения условной вероятности [3] $P_{i/(i+1)}$ следует, что при ее расчете совместную вероятность исправной работы всех элементов за время прохождения пакета и возможность потери пакета вследствие искажений в этих элементах, входящих в $(i + 1)$-й маршрут, необходимо положить равной единице. Для удобства представим последний член выражения (1) в следующем виде

$$P_{i+1} * P_i, \quad (2)$$

где $(*)$ — операция символьного умножения, которая означает, что при перемножении вероятности доставки пакета каждым элементом, входящим в первые i маршруты, и общих с $(i + 1)$-м маршрутом, заменяются единицей. С учетом (2) можно переписать выражение (1):

$$\Delta P_{i+1} = P_{i+1} * Q_i, \quad (3)$$

где $\Delta P_{i+1} = P_{i+1} - P_i$ — приращение вероятности доставки пакета при введении $(i + 1)$-го маршрута доставки копии пакета; $Q_i = 1 - P_i$ — вероятность того, что произойдет потеря пакета на первых i маршрутах.

Учитывая, что приращение вероятности доставки ΔP_{i+1} численно равно уменьшению вероятности потери ΔQ_{i+1}, получим следующее уравнение в конечных разностях:

$$\Delta Q_{i+1} = P_{i+1} * Q_i. \quad (4)$$

Решением уравнения (4) является функция

$$Q_i = (1 - P_1) * (1 - P_2) * ... * (1 - P_i). \quad (5)$$

Из соотношений (1)—(5) очевидно, что приращение вероятности доставки пакета при введении $(i + 1)$-го маршрута доставки копии пакета ΔP_{i+1} тем больше, чем меньше общих элементов имеют первые i маршруты с $(i + 1)$-м. Из этого следует, что наибольший эффект от соотношения приращения вероятности доставки к числу маршрутов (копии пакетов) $\frac{\Delta P_{i+1}}{m}$ проявляется при доставке копий пакета по независимым маршрутам.

В случае независимых маршрутов операция символьного умножения совпадает с обычным умножением и выражение (5) дает вероятность потери пакета между узлом-отправителем и узлом-получателем, так как передача пакета и его копий в сети осуществляется по маршрутам, не имеющим общих элементов:

$$Q_{st} = (1 - P_1) \times (1 - P_2) \times ... \times (1 - P_m). \quad (6)$$

Таким образом, из приведенных выше математических соотношений (1)—(6) следует, что предложенный механизм гарантированной доставки пакетов в IP-сети позволяет, с введением очередного маршрута для доставки копий пакетов, увеличить вероятность доставки пакета P_{st} без принятия специальных мер по снижению (корректировке) потерь по каждому маршруту.

Так, для обеспечения двумя независимыми маршрутами приемлемого качества восприятия видеоизображения HDTV максимально гарантируемый уровень потерь пакетов 10^{-3} в IP-сети будет достаточным.

5. Математическая постановка задачи гарантированной доставки пакетов в IP-сети

Механизм гарантированной доставки пакетов базируется на варианте управления потоками, отличном от существующего в IP-сетях. Он предусматривает отправку источником s пакетов информации в адрес получателя t по "совокупности независимых маршрутов". Необходимость предоставления услуг реального времени миллионам потребителей подразумевает наличие динамических процедур выработки таких маршрутов до каждого получателя. В математическом плане постановка задачи выработки нескольких независимых маршрутов может быть поставлена следующим образом.

Заданы: граф $G[N, A]$ с выделенными вершинами s и t, а также положительные целые числа J и K (J, $K \leqslant |N|$).

Требуется: найти в графе $G[N, A]$ не менее J маршрутов из s в t, попарно не имеющих общих узлов и включающих не более K дуг.

Задача в данной постановке известна [4] под названием "максимальное количество ограниченных непересекающихся путей" и входит в список NP-полных задач. Однако авторы считают, что это не является обстоятельством "непреодолимой силы" [5], а разработка универсального инструмента гарантированной доставки информации приложений реального времени придаст совершенно новые свойства IP-сетям даже с обычной архитектурой.

Заключение

Предлагаемый механизм гарантированной доставки пакетов по IP-сети можно применять для приложений реального времени, к которым относятся системы управления технологическими процессами, управление промышленным оборудованием, распределенное интерактивное моделирование, аудио- и видеоконференции, передача видео для немедленного воспроизведения, удаленная медицинская диагностика, телефония, некоторые игры и т. д. Также этот механизм может быть рекомендован к использованию в существующих и создаваемых сетях, обеспечивающих функционирование командных и контрольных центров, систем и центров управления, систем электронного документооборота и т. д., которые предъявляют повышенные требования к вероятностно-временным характеристикам доставки команд, сигналов управления, информации состояния, технологических сигналов и т. д.

Список литературы

1. **Захаров М.** Требования к IP-сетям нового поколения для организации масштабируемого и надежного сервиса широковещательной трансляции IPTV. — Сайт "Cisco на русском или просто о сложном": Брошюры, 2007.
2. **Куделя В. Н.** Способ гарантированной доставки блоков данных в коммутируемой сети с потерями // Заявка на изобретение № 2010117467. — М: Роспатент, 2010.
3. **Мизин И. А., Богатырев В. А., Кулешов А. П.** Сети коммутации пакетов / Под ред. В. С. Семенихина. М.: Радио и связь, 1986. 408 с.
4. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
5. **Куделя В. Н.** Методы перебора на графах и задачи управления телекоммуникационной сетью // Информационные технологии. 2008. № 4. С. 2—7.

УДК 004.738.5

Р. Т. Касумова, ст. науч. сотр.,
Институт информационных технологий
Национальной академии наук Азербайджана, Баку,
departl@iit.ab.az, kasumova-rena@rambler.ru

Сравнительный анализ географических доменов верхнего уровня сети Интернет

Представлен анализ географических доменов верхнего уровня. Проведен мониторинг правил регистрации географических доменов около 250 стран различного уровня развития. Определено число людей на один домен в 25 странах, представляющих наибольший интерес и значимость, указаны особенности географических доменов верхнего уровня и рассмотрены принципы разрешения споров, связанных с доменными именами.

Ключевые слова: *домен, Интернет-протокол, система доменных имен, администратор, регистратор, регистрант*

Введение

Одним из самых интенсивно развивающихся направлений в области информационных технологий (ИТ) является Интернет. Сегодня около 1 800 млн человек являются пользователями сети Интернет (www.internetworldstats.com). Каждый компьютер, подключаемый к сети Интернет, идентифицируется уникальным кодом (адресом), представляющим собой определенный набор цифр. По своему техническому значению такой код организует IP-адрес (Internet Protocol) данного компьютера, по которому происходят поиск и взаимодействие компьютеров в сети. В силу возникающих сложностей для пользователей сети в проведении операций созданная система доменных имен (Domain Name Sistem, DNS) представляет собой наиболее важный компонент Интернет-структуры, выражая адреса компьютеров в Интернете в виде IP-адресов. Домен (с английского *domain*, произошел от латинского слова *dominium*, обозначающий владение, территория) является логическим уровнем Интернета. Доменные имена состоят из областей символов, разделенных точками [1, 2].

На сегодняшний день Интернет-корпорация по присвоенным именам и номерам (The Internet Corporation for Assigned Names and Numbers, ICANN, www.icann.org) несет ответственность за управление DNS и распределение IP-адресов. Первоначально организацией ICANN было создано 250 доменов верхнего уровня: национальные (географические) домены верхнего уровня, принадлежащие 243 странам (country code Top Level Domain, ccTLD): .az, .ru, .us и др., и семь доменов общего пользования (generic Top Level Domain, gTLD): .com, .org, .net, .edu, .int, .gov, .mil. С 2005 г. специальные спонсируемые домены ограниченного пользования (Sponsored Top-Level Domains, sTLDs) получили одобрение ICANN: .travel, .jobs, .cat, .asia, .mobi, .tel, .museum, .aero, .coop. Национальные двухбуквенные домены (ccTLD) представляют собой двухбуквенные доменные имена верхнего (первого) уровня. Присваиваются (делегируются) национальные домены по кодам стран в соответствии со стандартом ISO 3166-1 [3].

1. Правила регистрации доменных имен и их сравнительный анализ

Для ознакомления с веб-ресурсом и выхода в сеть пользователи, т. е. физическое или юридическое лицо, государственные, местные органы самоуправления, компании и частные предприниматели, регистрируют доменные имена. В отношении регистрируемого имени предусмотрены электронные шаблоны и программы (Whois) для точной проверки полученного имени. Whois-сервис содержит открытую и закрытую информацию о доменах и их владельцах.

В процессе регистрации доменных имен задействованы следующие участники (за исключением доменов GOV, MIL, EDU и INT, к регистрации которых предъявляются особые требования) [4, 5]:

- некоммерческая организация ICANN;
- регистратура домена верхнего уровня;
- регистраторы, перечисленные на странице www.icann.org/registrars;
- организации, уполномоченные регистратором на регистрацию доменов в данной зоне;
- регистранты.

Существуют определенные правила регистрации доменных имен в доменах общего пользования, а также в географических доменах верхнего уровня, принадлежащих странам. Следует отметить, что при регистрации доменного имени мы сталкиваемся с административными и рядом технических ограничений. Правила во всех национальных доменах практически одинаковые. Так, в доменах верхнего уровня (GEOGRAPHIC, GENERIC) общие правила регистрации домена следующие:

- доменное имя должно начинаться и заканчиваться буквами латинского алфавита и цифрами;
- в доменном имени в качестве промежуточных символов могут быть буквы латинского алфавита, цифры или дефис;
- дефис не может использоваться в начале и конце имени, минимальная длина доменного имени в зависимости от зоны должна составлять 2—3 символа, а максимальная длина — 63 (в некоторых странах до 127);
- если какое-либо доменное имя, например "nauka", регистрируется в зоне AZ, то имя веб-сервера будет показано как www.nauka.az, а почтовые адреса будут заканчиваться на *@nauka.az (в зоне .com соответственно как www.nauka.com);
- доменные имена в зонах верхнего уровня, принадлежащих странам (например, AZ, TR, RU и т. д.), регистрируются как минимум на один год. Срок разовой регистрации доменного имени максимум составляет 10 лет;
- домены в зонах COM, NET, ORG, İNFO, BİZ, CC, TV, NAME можно регистрировать на срок от одного года до 10 лет;
- права по использованию доменного имени предоставляются только зарегистрированному лицу и т. п.

При разработке правил регистрации доменных имен большинство регистратур консультируются с местными Интернет-сообществами. Как правило, функционируют наблюдательные и консультационные комитеты, занимающиеся подобного рода вопросами. В эти комитеты входят представители правительств, патентных ведомств, операторов, бизнес-сообществ, а также независимые юристы [6, 7].

В 2005 г. Советом европейских национальных регистратур доменов верхнего уровня (Council of European National Top Level Domain Registries, CENTR, www.centr.org) было проведено исследование регистратур (администраторов национальных доменов), входящих в CENTR. По своему юридическому статусу все крупные регистратуры и большинство средних регистратур являются частными компаниями (18 из 27), среди них девять являются фондами, три — ассоциациями, один — кооперативом. Из девяти оставшихся (средних) регистратур семь являются подразделениями научных институтов и университетов, занимающихся вопросами развития научных сетей (к ним относятся регистратуры Хорватии, Латвии, Литвы, Польши, Ирана, Румынии, Словении).

Испанские (ES) и финские (Fİ) регистратуры контролируются государственными органами. Двое из восьми маленьких регистратур относятся к государственным структурам: регистратуры Ватикана (VA) и Афганистана (AF). Остальные являются либо подразделениями академических сетей, либо частными компаниями.

В большинстве стран (например, в Российской Федерации) регистрацией доменных имен занимаются специальные регистрационные компании. А администраторы доменов разрабатывают правила регистрации, обеспечивают деятельность регистраторов и технического центра, обслуживающего реестр домена. Регистрацией доменов в Беларуси (BY) занимается Государственный центр безопасности информации при президенте Республики Беларусь. Регистратор Франции (FR) AFNIC (Association Frangaise pour le Nommage Internet en Cooperation) учрежден при участии правительства Франции, за представителями которого закреплена часть мест в правлении. Администратором национального домена Бельгии (BE) является специализированная организация DNS Belgium VZM, учрежденная Федерацией промышленности Бельгии, ассоциацией операторов и общественным объединением пользователей. В состав правления в качестве наблюдателя входят также представители Министерства связи и Министерства экономического развития Бельгии.

В ряде стран применяют *определенные ограничения при регистрации доменных имен*. Данные ограничения отражаются в правилах регистрации этих стран. Анализы показывают, что требования по регистрации доменных имен могут варьироваться в зависимости от политики, осуществляемой государством. Так, в Канаде (CA), Венгрии (HU), Словении (SI), Финляндии (FI), Голландии (NL), Ватикане (VA), Исландии (IS), Норвегии (NO) и Кипре (CY) домен не может быть открыт нерезидентом. Кроме того, в Словении и Венгрии не допускается регистрация доменов со стороны частных лиц. В Израиле (IL), Норвегии (NO), Словении (SI), Венгрии (HU) и Кипре (CY) существуют ограничения по числу доменов, регистрируемых на одно лицо. В независимом государстве Эритрея (ER), расположенном в Африке, регистрация доменов второго уровня находится под строгим контролем правительства. Домены, прошедшие регистрацию, используются только для отправления и принятия по электронной почте.

Длина доменов третьего уровня в Австрии (AT) может составлять как минимум один символ (например, t3.co.at или s.or.at), а длина доменов второго уровня должна быть не менее трех символов и не превышать лимит 63 символов. В Нидерландах (NL), Норвегии (NO), Японии (JP) возможна регистрация доменов, состоящих только из цифр, например www.1234.nl, или цифр, разделенных дефисом, например www.12-34.nl. В Нидерландах, начиная с февраля 2008 г., запущена открытая регистрация цифровых имен. В зоне NL пользователь, регистрирующий домен, может находиться только на территории государства, а физическое лицо может зарегистрировать лишь одно доменное имя. Регистрация доменного имени возможна только через участников фонда, войти в который могут любые организации, зарегистрированные и находящиеся на территории Европейского союза. В случае если это правило нарушается в какой-либо форме, то администратор зоны немедленно аннулирует всю регистрацию незаконных дополнительных доменных имен. Стоимость регистрации зависит от каждого конкретного регистратора (www.domain-registry.nl).

В Великобритании (UK) с 1996 г. домены регистрирует некоммерческая организация Nominet UK, имеющая более 2000 участников. Все решения принимает управляющий совет компании. Согласно регистрационным правилам Великобритании регистрация проводится только в доменах второго уровня общего пользования: CO.UK, ORG.UK, NET.UK, LTD.UK и PLC.UK, SCH.UK.

В доменной зоне Антарктиды (AQ) регистрация доменных имен началась 26 февраля 1992 г. В зоне AQ можно зарегистрировать доменное имя на двухлетний срок, а затем регистрацию можно продлить еще на два года. Каждый пользователь может зарегистрировать только одно доменное имя. Передача домена невозможна. В домене Антарктиды владельцем доменного имени могут быть государственная организация страны-участницы Антарктического соглашения и любое заинтересованное лицо. При этом ему необходимо самому находиться в Антарктиде или иметь документ от руководителя какой-либо из действующих антарктических станций с подтверждением того, что данный "желающий" был в Антарктике или собирается ее посетить. Для желающего открытие и продление доменного имени проводится бесплатно.

Необходимо отметить, что *для регистрации доменов необходимо заплатить определенную сумму*. Для каждой зоны эта сумма различна и может меняться. Как правило, это связано с формой собственности регистрирующего органа регистрации, числом регистрируемых доменов, общей маркетинговой политикой, в частности, с участием в регистрационном процессе провайдеров услуг Интернет (Internet service provider, ISP) или регистраторов.

В Канаде (CA) можно регистрировать доменные имена с минимальной длиной в два знака, максимальной длиной в 50 знаков, а стоимость составляет 8,5 долл. В этой зоне срок регистрации доменов охватывает от одного года до 10 лет. В доменной зоне остров Воссоединения (RE) регистрационная цена зависит от регистратора доменного имени, являющегося членом AFNIC (Association Française pour le Nommage Internet en Coopération). Доменное имя не должно нарушать права человека, право на интеллектуальную собственность, а также правила добросовестной конкуренции.

В Швейцарии (CH) и Германии (DE) доменные имена могут состоять из нестандартных букв (à, ã, ä, ð, å, æ, ø и т. п.) в виде умлаутов, ударных и других диакритических знаков. В зонах DE и CH регистрантом может стать как физическое, так и юридическое лицо, вне зависимости от того, резидент он данной страны или нет. Для зоны DE цена регистрации составляет 58 евро, а для зоны CH — 24 долл. В доменной зоне Лихтенштейна (LI) также допускается использование букв с диакритическими знаками. Регистрантом может стать любой человек или организация из любой страны. Стоимость регистрации составляет 27 CNF (включая НДС для жителей Швейцарии и без НДС для остальных заказчиков).

Администратор доменной зоны Китая CNNIC (China Internet Network Information Center, www.cnnic.net.cn) поддерживает регистрацию мультиязычных доменных имен, в данном случае состоящих из китайских иероглифов. Минимальная длина такого доменного имени — один знак, максимальная — 20 символов. Стоимость регистрации определяется аккредитивными регистраторами независимо.

Необходимо отметить, что с конца 2008 г. ICANN, хоть и частично, но решает проблемы многоязыковых доменов. На 36-й конференции ICANN, проведенной в столице Южной Кореи — Сеуле, было принято решение о регистрации доменных имен на русском, китайском, арабском, индийском, корейском, иврите и прочих алфавитах, не основанных на латинской графике (30.10.2009). К сожалению, не предусматривается рассмотрение вопросов регистрации доменных имен на языках тюркоязычных народов.

Необходимо отметить, что на XI съезде дружбы, братства и сотрудничества тюркских государств и обществ, проведенном в Баку 17—19 ноября 2007 г., Институт информационных технологий Национальной академии наук Азербайджана выдвинул предложение о включении в статус sTLD таких доменных имен, как .turan, .turk, что было принято положительно и включено в рекомендации съезда.

В Турции Интернет-именами занимается коллегиальная организация в Ближневосточном техническом университете (Middle East Technical University, Department of Computer Engineering). Стоимость регистрации и перерегистрации доменного имени варьируется в пределах 5—25 турецких лир в год (www.metu.edu.tr).

Восемь регистраторов регистрируют домены в независимом порядке: Кипр, Хорватия, Израиль, Исландия, Латвия, Люксембург, Мальта и Ватикан. В этих странах регистрация осуществляется только администраторами национальных доменов. Стоимость регистрации в зоне Кипра (CY) составляет 25 евро, (за 2 года 45 евро), в Хорватии (HR) — 25 долл., в Израиле (İL) за 2 года — 60 долл. (оплата ведется кредитной карточкой), в Исландии (IS) — 35 евро, в Латвии (LV) — 14 долл., в Люксембурге (LU) — 40 евро, в Мальте (MT) — 11,65 евро. В Ватикане регистрация бесплатна и осуществляется напрямую через администратора зоны. Домен был создан в 1995 г. и до настоящего времени зарезервирован для официальных сайтов Ватикана.

Сравнительный анализ географических доменов верхнего уровня показывает, что пути решения *споров, возникших в связи с регистрацией и управлением доменных имен*, различны. Рассмотрение споров, возникших в связи с регистрацией доменных имен и их управлением, находит свое отражение в правилах регистрации стран. Администратор/Регистратор зоны несет ответственность за выполнение этих правил. Рассмотрение споров в связи с доменными именами может быть решено посредством нескольких методов [8, 9, 10]:

- в рамках Единой политики разрешения споров по доменным именам, принятой ICANN (Uniform Domain Name Dispute Resolution Policy, UDRP);
- обращением в международные арбитражные центры;
- обращением в суды согласно соответствующему законодательству страны;
- путем переговоров с использованием конфликтующими сторонами законных методов.

Надо отметить, что администраторы зон некоторых стран не несут ответственности за использование доменов другими лицами. Все споры решаются сторонами самостоятельно, без вмешательства администратора зоны. В число таких стран входят Албания (AL), Босния и Герцеговина (BA), Бангладеш (BD), Бельгия (BE), Бенин (BJ), Белиз (BZ), Швейцария (CH), Кипр (CY), Германия (DE), Франция (FR), Люксембург (LU), Ливия (LY), Мавритания (MR), Мексика (MX), Монголия (MN), Монако (MC), Оман (OM) и т. д.

2. Распределение доменов относительно численности населения в ccTLD и gTLD

Согласно статистическому отчету, проведенному компанией VeriSign, на конец первого квартала 2010 г. в Интернете насчитывалось более 193 млн доменных имен, что примерно на 11 млн доменов больше, чем за аналогичный период 2009 г. Из них 76,3 млн относятся к национальным, а около 114 млн доменов — к доменам общего пользования. В первую десятку национальных доменов с наибольшим числом регистраций вошли: Германия (DE), Китай (CN), Великобритания (UK), Нидерланды (NL), Евросоюз (EU), Российская Федерация (RU), Аргентина (AR), Бразилия (BR), Токелау

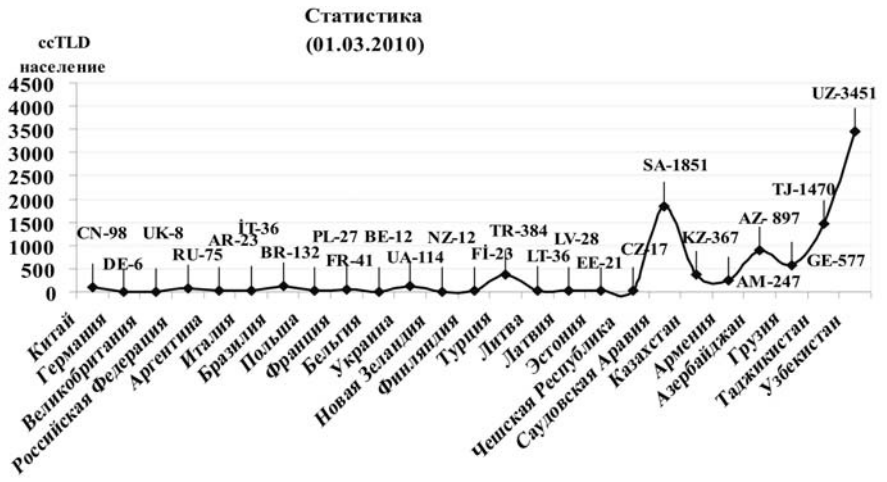

Рис. 1. Распределение доменов (число людей на один домен) в 25 странах ccTLD

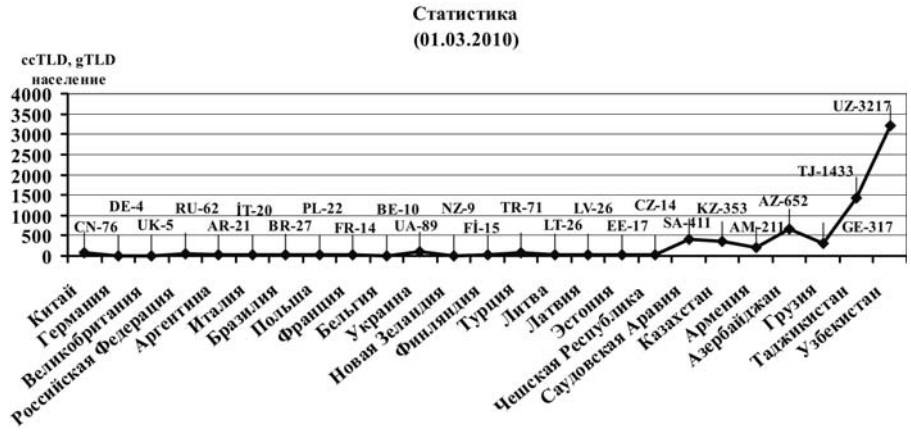

Рис. 2. Распределение доменов (число людей на один домен) верхнего уровня по 25 странам

(TK) и Италия (IT). На эти домены приходится 62 % доменных имен, зарегистрированных во всех 240 ccTLD.

В отчете компании VeriSign за 01.06.2010 г. представлено, что в зоне CN число доменов, резко понизившись, достигло 8,8 млн. Столь низкие темпы роста специалисты VeriSign прежде всего связывают с уменьшением числа регистраций и продлений в домене Китая (CN) (http://www.verisign.com/domain-name-report-junel0.pdf). На рис. 1 показано число людей на один домен по ccTLD в 25 странах за 01.03.2010 г.

В целях уточнения числа людей на один домен в каждой стране проведен мониторинг за первый квартал 2010 г. В таблице представлены результаты статистического анализа, проведенного по некоторым параметрам доменов в 25 странах с различными уровнями развития.

На рис. 2 продемонстрированы результаты мониторинга доменов верхнего уровня в 25 странах за 01.03.2010 г.

3. Особенности национального домена

Национальные домены в зависимости от регистрационного ограничения делятся на *открытые*, *условно открытые* и *закрытые* домены. В этом делении находят свое отражение политика и интересы каждой страны. Открытые домены развиваются наиболее активно. Это связано с их доступностью: в открытых доменах владельцами доменных имен могут стать как резиденты страны, так и нерезиденты, а также физические и юридические лица.

В ряд государств открытых доменов входят Германия (DE), Россия (RU), Конго (CG), Камерун (CM), Кокосовые острова (CC), Микронезия (FM), Гренада (GD), Гвиана (GY), Индия (IN), Кыргызстан (KG), Коморские острова (KM), Лихтенштейн (LI), Литва (LT), Латвия (LV), Монголия (MN), Новая Зеландия (NZ), Голландия (Nl), Панама (PA), Филиппины (PH), Польша (PL), Пуэрто-Рико (PR), Руанда (RW), Сербия (RS),

Результаты статистического анализа, проведенного по некоторым параметрам доменов в 25 странах с различными уровнями развития

Страны	ccTLD	Население	Число доменов ccTLD	Число доменов gTLD	Общее число доменов	Число людей на один домен ccTLD	Число людей на один домен gTLD	Число людей на один домен ccTLD + gTLD
Китай	.cn	1 338 612 968	13 700 000	3 904 217	17 604 217	98	343	76
Германия	.de	82 329 758	13 500 000	6 172 635	19 672 635	6	13	4
Великобритания	.uk	61 113 205	8 000 000	4 156 756	12 156 756	8	15	5
Российская Федерация	.ru	140 041 247	1 870 000	387 7500	2 257 750	75	361	62
Аргентина	.ar	40 913 584	1 800 000	154 798	1 954 798	23	264	21
Италия	.it	58 112 621	1 600 000	1 286 087	2 886 087	36	45	20
Бразилия	.br	198 739 269	1 500 000	5 769 187	7 269 187	132	34	27
Польша	.pl	34 882 919	1 300 000	292 184	1 592 184	27	119	22
Франция	.fr	62 150 775	1 500 000	2 881 944	4 381 944	41	22	14
Бельгия	.be	10 414 336	900 000	180 220	1 080 220	12	58	10
Украина	.ua	45 400 395	400 000	107 848	507 848	114	421	89
Новая Зеландия	.nz	4 213 418	350 000	116 848	466 848	12	36	9
Финляндия	.fi	5 250 275	230 000	125 798	355 798	23	42	15
Турция	.tr	76 805 524	200 000	887 801	1 087 801	384	87	71
Литва	.lt	3 555 179	100 000	35 106	135 106	36	101	26
Латвия	.lv	2 231 503	80 000	6683	86 683	28	334	26
Эстония	.ee	1 299 371	63 100	13 843	76 943	21	94	17
Чешская Республика	.cz	10 211 904	600 000	147 771	747 771	17	69	14
Саудовская Аравия	.sa	28 686 633	15 500	54 234	69 734	1851	529	411
Казахстан	.kz	15 399 437	42 000	1655	43 655	367	9305	353
Армения	.am	2 967 004	12 000	2094	14 094	247	1417	211
Азербайджан	.az	9 000 000	10 034	3767	13 801	897	2389	652
Грузия	.ge	4 615 807	8000	6552	14 552	577	704	317
Таджикистан	.tj	7 349 145	5000	130	5130	1470	56 532	1433
Узбекистан	.uz	27 606 007	8000	582	8582	3451	47 433	3217

Судан (SD), Токелау (TK), Тувалу (TV), Туркменистан (TM), Узбекистан (UZ), Западная Самоа (WS), Ангилья (AI) и др.

В некоторых странах в с связи наличием определенных ограничений, накладываемых на регистрантов, национальные домены относятся к категории условно открытых доменов. В таких доменных зонах при регистрации доменов второго уровня бывают ограничения. Например, во Франции (FR) владельцем доменного имени может быть только резидент страны, а в государстве Бенин (BJ) регистрировать домен разрешается только юридическим лицам. Домен Эстонии (EE) до 5.07.2010 г. входил в список условно открытых доменов. С 05.07.2010 г. в связи с проведением реформ в зоне национального домена Эстонии большинство ограничений на регистрацию было отменено. В результате либерализации регистрационных правил в зоне EE частные лица, юридические лица и нерезиденты могут регистрировать любое число доменов. Стоимость регистрации составляет приблизительно 18 евро. К условно открытым доменам относятся Албания (AL), Бангладеш (BD), Венгрия (HU), Италия (İT) и другие страны.

Закрытые национальные домены обычно развиваются медленно. В доменных зонах Ирландии (IE), Швеции (SE) регистрантами могут быть только резиденты или организации, прошедшие регистрацию на территории данного государства. KY домены Каймановых островов относятся к закрытым доменам. Только резидент Каймановых островов может регистрировать доменные имена. К другим закрытым доменам относятся домены Sent Pyer и Makvelonun PM. В доменной зоне SI Словении регистраторами могут быть только юридические лица, находящиеся на территории Словении. Одна организация не может регистрировать более 20 доменов.

Имеются некоторые национальные домены, которые позиционируются на рынке не как национальные, а как ассоциативные домены. Двухбуквенный домен зачастую похож на знакомые всем аббревиатуры. Например, домен FM (Федеративный штат Микронезия) может подойти пользователям, которые принадлежат к радиосреде или радиобизнесу. Домен CD (Демократическая Республика Конго) позиционируется как домен для проектов, связанных с музыкальной индустрией. Домен DJ (Джибути) необходим ди-джеям, TV (Тувалу) — телевизионным компаниям, TM (Туркменистан) — владельцам товарных знаков, MD (Молдова) — медицинским учреждениям, Филиппины активно продвигают свой домен PH как телефонный справочник.

Список литературы

1. **Венедрюхин А. А.** Доменные войны. СПб.: Питер, 2009. 224 с.
2. **Храмцов П. Б.** Лабиринт Internet. М.: Электронинформ, 1996.
3. **Davis G.** The New Thing, Uniform Domain-Name Dispute-Resolution Policy of the Internet Corporation for Assigned Names and Numbers // The Journal of World Intellectual Property. 2000. Vol. 3. N 4. P. 525—554.
4. **Серго А. Г.** Доменные имена. М.: Бестселлер, 2006. 368 с.
5. **Федоренко Н. В., Пархоменко П. Н., Пархоменко Н. Г.** Особенности адресной организации сети Интернет и арбитражные споры вокруг доменных имен // Защита информации. Конфидент. 2004. № 4. С. 12—14.
6. **Кравченко В. Ф.** Домен — не адрес, а средство индивидуализации // Интеллектуальная собственность. 2001. № 2. С. 66—70.
7. **Наумов В. Б.** Споры, связанные с нарушением прав на объекты интеллектуальной собственности в сети Интернет // Арбитражные споры. 2001. № 1 (13). С. 81—88.
8. **Вацковский Ю. Ф.** Доменные споры. Защита товарных знаков и фирменных наименований. М.: Статут, 2009. 192 с.
9. **Милютин З. Ю.** "Киберсквотинг" как злоупотребление правом // Хозяйство и право. 2005. № 12. С. 60—62.
10. **Кирий Л. Л.** Охрана интеллектуальной собственности в национальных доменах Интернета // Патенты и лицензии. 2001. № 7. С. 33—40.

УДК 532.5 + 536.24

К. В. Максименко-Шейко[1],
канд. физ.-мат. наук, ст. науч. сотр.,
А. В. Толок[2],
д-р техн. наук, проф., зав. кафедрой,
e-mail: atol@norbert.ru,
Т. И. Шейко[1],
д-р техн. наук, проф., зав. отделом
[1] ИПМаш им. А. Н. Подгорного НАН Украины;
ХНУ им. В. Н. Каразина
[2] МГТУ "Станкин";
ИПУ им. В. А. Трапезникова РАН

R-функции в фрактальной геометрии

На основе конструктивных средств теории R-функций, суперпозиции функций, рекурсивных процедур и свойстве подобия фигур разработана методика и построены уравнения ряда объектов фрактальной геометрии: салфетка и ковер Серпинского, губка Менгера, кривая Коха, снежинка и крест Коха и др.

Ключевые слова: *фрактальная геометрия, подобие, R-функции, суперпозиции функций*

Введение

В настоящее время фракталы широко применяются в радиотехнике при проектировании антенных устройств (кривая Коха и ковер Серпинского) и волноводов (снежинка Коха), в компьютерной графике и при сжатии изображений. В физике фракталы естественным образом возникают при моделировании нелинейных процессов, таких как турбулентное течение жидкости, сложные процессы диффузии-адсорбции и т. п. Фракталы используются при моделировании пористых материалов, например в нефтехимии. В биологии они применяются для моделирования популяций и для описания систем внутренних органов (например, системы кровеносных сосудов). Большой интерес вызывают задачи математического моделирования физико-механических полей в областях фрактальной природы.

Однако В. Л. Рвачев в своих работах (например, [4]), описывая типы геометрических объектов, исключил из рассмотрения такие "геометрические монстры, как Канторово множество, ковер Серпинского" и другие объекты фрактальной геометрии. В работе [5] впервые были построены уравнения границ некоторых объектов фрактальной геометрии. В данной работе на основе конструктивных средств теории R-функций [6], суперпозиций и рекурсивных процедур разработана методика и построены новые уравнения ряда объектов фрактальной геометрии.

Основная часть

Рассмотрим самый простой детерминированный фрактал, который образуется при прибавлении квадратов к вершинам других квадратов и называется *"коробка"*, где и инициатор, и генератор — квадраты. Его фрактальная размерность $\frac{\ln 8}{\ln 3} = 1{,}8927$. Построим уравнение основного рисунка для квадрата со стороной $2a$:

$$\omega_0(x, y) = \frac{a^2 - x^2}{2a} \wedge_0 \frac{a^2 - y^2}{2a} \geqslant 0;$$

$$\omega_{01}(x, y) = \omega_0(x - 2a, y - 2a) \geqslant 0;$$

$$\omega_{02}(x, y) = \omega_0(x + 2a, y - 2a) \geqslant 0;$$

$$\omega_{03}(x, y) = \omega_0(x + 2a, y - 2a) \geqslant 0;$$

$$\omega_{04}(x, y) = \omega_0(x + 2a, y + 2a) \geqslant 0;$$

$$\omega_1(x, y) = \omega_0 \vee_0 \omega_{01} \vee_0 \omega_{02} \vee_0 \omega_{03} \vee_0 \omega_{04} \geqslant 0.$$

Теперь построим итерационный процесс, в результате которого получим:

$$\omega_{k-1,1}(x, y) = \omega_{k-1}(x - 6a, y - 6a) \geqslant 0;$$

$$\omega_{k-1,2}(x, y) = \omega_{k-1}(x + 6a, y - 6a) \geqslant 0;$$

$$\omega_{k-1,3}(x, y) = \omega_{k-1}(x + 6a, y - 6a) \geqslant 0;$$

$$\omega_{k-1,4}(x, y) = \omega_{k-1}(x + 6a, y + 6a) \geqslant 0;$$

$$\omega_k(x, y) = \omega_{k-1} \vee_0 \omega_{k-1,1} \vee_0 \vee_0 \omega_{k-1,2} \vee_0 \omega_{k-1,3} \vee_0 \omega_{k-1,4} \geqslant 0.$$

На рис. 1 (см. третью сторону обложки) построены картины линий уровня функции $\omega_k(x, y) \geqslant 0$, задающей фрактал "коробка" для различных значений k.

Одним из свойств фракталов является самоподобие. Возьмем, например, *треугольник (или салфетку) Серпинского.* Для его построения из центра равностороннего треугольника "вырезают" треугольник. Повторяют эту же процедуру для трех образовавшихся треугольников (за исключением центрального), и так до бесконечности. Если теперь взять любой из образовавшихся треугольни-

ков и увеличить его, то получим точную копию целого. В данном случае имеет место полное самоподобие. В этом фрактале инициатор и генератор, как и в предыдущем случае, одинаковы. При каждой итерации добавляется уменьшенная копия инициатора к каждому углу генератора и т. д. Если при создании этого фрактала провести бесконечное число итераций, он бы занял всю плоскость. Поэтому его фрактальная размерность $\frac{\ln 9}{\ln 3} = 2$. Запишем уравнение правильного треугольника в виде

$$\omega_0(x, y) = -\sqrt{x^2 + y^2}\cos\left(\frac{2}{3}\arcsin\left(\sin\frac{3\theta}{2}\right)\right) + R \geqslant 0,$$

$$\text{или } \omega_0(x, y) = -x1 + R \geqslant 0,$$

где $x1 = r\cos\mu$; $y1 = r\sin\mu$; $\mu(\theta) = \frac{2}{3}\arcsin\left(\sin\frac{3\theta}{2}\right)$;

$r = \sqrt{x^2 + y^2}$; $\theta = \operatorname{arctg}\frac{y}{x}$; R — радиус вписанной окружности. Тогда

$$\omega_1(x, y) = \omega_0(-2(x1 - R), 2y1)/2 \geqslant 0$$

и, соответственно,

$$\omega_k(x, y) = \omega_{k-1}(2(x1 - R), 2y1)/2 \geqslant 0, (k = 2, 3, ...).$$

Заметим, что в данном случае R-функции не используются. На рис. 2 (см. третью сторону обложки) построены картины линий уровня функции $\omega_k(x, y) \geqslant 0$, задающей салфетку Серпинского для различных значений k.

Построим уравнение фрактальной области *ковер Серпинского*. Для этого исходный прямоугольник со сторонами $2a \times 2b$ разбивается на девять равновеликих прямоугольников, из которых исключается центральный. Оставшиеся прямоугольники подвергаются той же процедуре и т. д. Фрактальная размерность построенной области $\frac{\ln 8}{\ln 3} = 1{,}89$. Если

$$f_1 = \frac{a^2 - x^2}{2a} \geqslant 0; f_2 = \frac{b^2 - y^2}{2b} \geqslant 0,$$

то $\omega_0 = f_1 \wedge_0 f_2 \geqslant 0$ — предфрактал нулевого уровня. Построим вспомогательные функции, пользуясь свойством самоподобия:

$$\omega_1(x, y) = \frac{\overline{\omega_0(3x, 3y)}}{3} \geqslant 0;$$

$$\omega_k(x, y) = \frac{\omega_{k-1}(3\mu_{hx}, 3\mu_{hy})}{3} \geqslant 0..., (k = 2, 3, ...),$$

где $\mu_{hx} = \frac{h_x}{\pi}\arcsin\left(\sin\frac{\pi x}{h_x}\right)$, $\mu_{hy} = \frac{h_y}{\pi}\arcsin\left(\sin\frac{\pi y}{h_y}\right)$,

$h_x = \frac{2a}{3}$, $h_y = \frac{2b}{3}$.

Тогда $K_\omega(x, y) = \omega_0(x, y) \wedge_0 \omega_1(x, y) \wedge_0 \omega_2(x, y) \wedge_0 \wedge_0 ... \wedge_0 \omega_k(x, y) \geqslant 0$. На рис. 3 (см. третью сторону обложки) построены картины линий уровня функции $K_{\omega_k}(x, y) \geqslant 0$, задающей ковер Серпинского для различных значений k.

Трехмерным аналогом ковра Серпинского является *губка Менгера*. Губка Менгера имеет фрактальную размерность $\frac{\ln 20}{\ln 3} = 2{,}73$, поскольку состоит из 20 равных частей, каждая из которых подобна всей губке с коэффициентом подобия 1/3. Уравнение куба со стороной $2a$ имеет вид

$$\omega_0(x, y, z) = \frac{a^2 - x^2}{2a} \wedge_0 \frac{a^2 - y^2}{2a} \wedge_0 \frac{a^2 - z^2}{2a} \geqslant 0.$$

Уравнение трех центральных сквозных отверстий строится следующим образом:

$$\omega b_1 = \frac{1}{3}\overline{\left(\frac{a^2 - 9x^2}{2a} \wedge_0 \frac{a^2 - 9y^2}{2a}\right)\vee_0}$$

$$\overline{\vee_0\left(\frac{a^2 - 9z^2}{2a} \wedge_0 \frac{a^2 - 9x^2}{2a}\right)\vee_0\left(\frac{a^2 - 9z^2}{2a} \wedge_0 \frac{a^2 - 9y^2}{2a}\right)} \geqslant 0,$$

а уравнения транслированных самоподобных отверстий —

$$\omega b_k = \frac{1}{3^k}\overline{\left(\frac{a^2 - 3^{2k}\mu_{xk}^2}{2a} \wedge_0 \frac{a^2 - 3^{2k}\mu_{yk}^2}{2a}\right)\vee_0}$$

$$\overline{\vee_0\left(\frac{a^2 - 3^{2k}\mu_{zk}^2}{2a} \wedge_0 \frac{a^2 - 3^{2k}\mu_{xk}^2}{2a}\right)\vee_0}$$

$$\overline{\vee_0\left(\frac{a^2 - 3^{2k}\mu_{zk}^2}{2a} \wedge_0 \frac{a^2 - 3^{2k}\mu_{yk}^2}{2a}\right)} \geqslant 0, (k = 2, 3, ...),$$

где $\mu_{xk} = \frac{h_k}{\pi}\arcsin\sin\frac{\pi x}{h_k}$; $\mu_{yk} = \frac{h_k}{\pi}\arcsin\sin\frac{\pi y}{h_k}$;

$\mu_{zk} = \frac{h_k}{\pi}\arcsin\sin\frac{\pi z}{h_k}$; $h_k = \frac{2a}{3^{k-1}}$; $\omega_k = \omega_0 \wedge_0 \omega b_1 \wedge_0 \wedge_0 \omega b_2 \wedge_0 ... \wedge_0 \omega b_k \geqslant 0$.

На рис. 4 приведена визуализация уравнений поверхностей губки Менгера для различных значений k, выполненная в условиях эксплуатации системы РАНОК [7].

Типичным детерминированным фракталом является *кривая Коха*. Процесс ее построения выглядит следующим образом: берем единичный отрезок, разделяем на три равные части и заменяем средний интервал равносторонним треугольником без этого сегмента. В результате образуется ломаная, состоящая из четырех звеньев длины 1/3. На

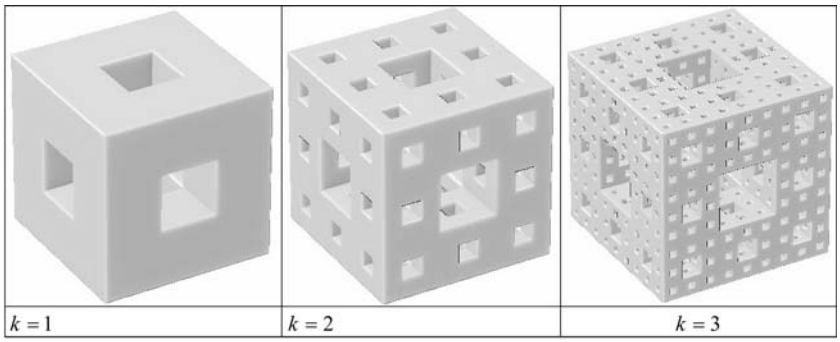

Рис. 4. Визуализация уравнений поверхности губки Менгера для различных значений *k*

следующем шаге повторяем операцию для каждого из четырех получившихся звеньев. Предельная кривая является кривой Коха. Эта кривая была описана в 1904 г. шведским математиком Хельге фон Кохом, который, изучая работы Карла Вейерштрасса и Георга Кантора, натолкнулся на описание некоторых странных кривых с необычным поведением. Кривая Коха примечательна тем, что нигде не имеет касательной, т. е. нигде не дифференцируема, хотя всюду непрерывна. Такие "ущербные" функции были построены Вейерштрассом лишь для того, чтобы показать своим скептически настроенным коллегам (в том числе ужаснувшемуся Эрмиту), что такие функции (непрерывные и недифференцируемые) действительно существуют. Однако другие математики увидели в них новый свет. Например, Больцман в 1898 г. писал, что недифференцируемые функции могли быть изобретены физиками, так как в статистической механике имеются проблемы, для решения которых "недифференцируемые функции абсолютно необходимы". Жан Перрен пошел еще дальше: в 1906 г. он, предвосхищая отношение к такого рода математическим монстрам, заявил, что "кривые, не имеющие касательных, являются общим правилом, а гладкие кривые — интересным, но весьма частным случаем".

Кривая Коха не спрямляема, не имеет самопересечений. Она имеет фрактальную размерность, которая равна $\frac{\ln 4}{\ln 3} \approx 1{,}26$, поскольку она состоит из четырех равных частей, каждая из которых подобна всей кривой с коэффициентом подобия 1/3. Выполним построение на интервале $-3a \leqslant x \leqslant 3a$. Тогда

$$\omega_0 = -y \geqslant 0;\ \omega_1 = \omega_0 \vee_0 (f_1 \wedge_0 f_2) \geqslant 0;$$

$$f_1 = \frac{1}{2}(x\sqrt{3} - y + a\sqrt{3}) \geqslant 0;$$

$$f_2 = \frac{1}{2}(-x\sqrt{3} - y + a\sqrt{3}) \geqslant 0;$$

$$\omega_{21} = \omega_1(3(x + 2a),\ 3y) \geqslant 0;$$

$$\omega_{22} = \omega_1\left(3\left(\frac{x + a/2}{2} + \left(y - \frac{a\sqrt{3}}{2}\right)\frac{\sqrt{3}}{2}\right),\right.$$

$$\left.3\left(-(x + a/2)\frac{\sqrt{3}}{2} + \left(y - \frac{a\sqrt{3}}{2}\right)\frac{1}{2}\right)\right) \geqslant 0;$$

$$\omega_2 = (\omega_{21}(x, y) \vee_0 \omega_{22}(x, y)) \wedge_0$$
$$\wedge_0 (\omega_{21}(-x, y) \vee_0 \omega_{22}(-x, y)) \geqslant 0;$$

$$\omega_{k1} = \omega_{k-1}(3(x + 2a),\ 3y) \geqslant 0;$$

$$\omega_{k2} = \omega_{k-1}\left(3\left(\frac{x + a/2}{2} + \left(y - \frac{a\sqrt{3}}{2}\right)\frac{\sqrt{3}}{2}\right),\right.$$

$$\left.3\left(-(x + a/2)\frac{\sqrt{3}}{2} + \left(y - \frac{a\sqrt{3}}{2}\right)\frac{1}{2}\right)\right) \geqslant 0;$$

$$\omega_k = (\omega_{k1}(x, y) \vee_0 \omega_{k2}(x, y)) \wedge_0$$
$$\wedge_0 (\omega_{k1}(-x, y) \vee_0 \omega_{k2}(-x, y)) \geqslant 0,\ (k = 3, 4, \ldots).$$

На рис. 5 (см. третью сторону обложки) приведены картины линий уровня функции $\omega_k(x, y) \geqslant 0$, задающей кривую Коха для различных значений k.

Зная уравнение кривой Коха, можно построить, например, тела вращения (рис. 6).

Три копии кривой Коха, построенные (остриями наружу) на сторонах правильного треугольника, образуют замкнутую кривую, называемую *снежинкой Коха*. Снежинка Коха, или триада Коха, является математической моделью кривой побережья, с которой работал Ричардсон. Размерность Хаусдорфа—Безиковича снежинки Коха составляет $\frac{\ln 4}{\ln 3} \approx 1{,}26$, т. е. она больше топологической размерности линии (равной единице), но меньше евклидовой размерности плоскости, на которой она расположена. Отсюда следует, что снежинка Коха представляет собой линию

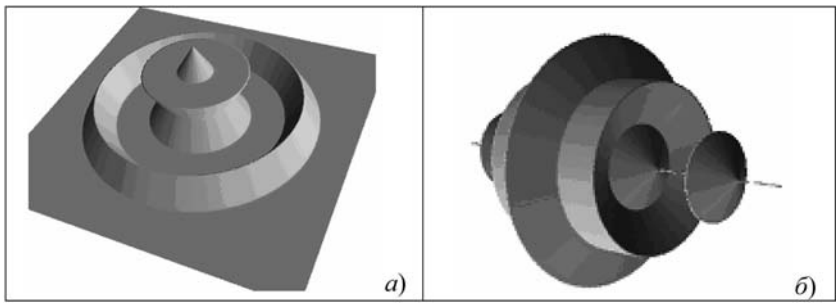

Рис. 6. Тела вращения, построенные с помощью кривой Коха:
а — ось вращения — *Oy* при $k = 2$; *б* — ось вращения — *Ox* при $k = 2$

бесконечной длины, ограничивающую конечную площадь. Итальянский математик Э. Чезаро, удивленный внутренней бесконечностью и самоподобием снежинки Коха, писал в 1905 г.: "Если бы она была одарена жизнью, то можно было бы лишить ее жизни, только уничтожив кривую в целом. В противном случае она бы возрождалась снова и снова из глубины своих треугольников, как это делает жизнь во Вселенной". Зная уравнение кривой Коха $\omega_k(x, y) \geqslant 0$, можно построить уравнение снежинки Коха (рис. 7, см. четвертую сторону обложки), выполнив следующие преобразования:

$$\omega S_k = \omega_k(r\sin\mu,\ r\cos\mu - R) \geqslant 0, \qquad (1)$$

где $\mu(n\theta) = \frac{2}{n}\arcsin\left(\sin\frac{n\theta}{2}\right)$, $r = \sqrt{x^2 + y^2}$, $\theta = \operatorname{arctg}\frac{y}{x}$,

R — радиус окружности, вписанной в правильный n-угольник со стороной, равной отрезку, на котором строится кривая Коха.

Пользуясь данной методикой, можно строить фрактальные снежинки на сторонах различных правильных многоугольников, например, на сторонах квадрата (рис. 8, см. четвертую сторону обложки).

Можно построить *крест Коха* на сторонах квадрата, при этом проводя построение внутрь квадрата. Для этого выполним построение кривой Коха, заменяя средний интервал равносторонним треугольником, ориентированным вниз, без этого сегмента. Повторяем операцию для каждого из четырех получившихся звеньев и т. д. Метод R-функций позволяет легко получить такую кривую, взяв отрицание функции, построенной ранее (рис. 9, см. четвертую сторону обложки).

Зная уравнение переориентированной кривой Коха, можно построить уравнение креста Коха (рис. 10, см. четвертую сторону обложки), выполнив преобразования, аналогичные (1).

Рассмотрим *фрактал Леви*, предложенный французским математиком П. Леви. Для построения берется равнобедренный прямоугольный треугольник (рис. 11, см. четвертую сторону обложки при $k = 1$), а затем каждый катет заменяется подобным треугольником (рис. 11, см. четвертую сторону обложки при $k = 2$). Повторяя эту операцию, в пределе получим фрактал Леви. Выполним построение на интервале $-3a \leqslant x \leqslant 3a$:

$$\omega_1(x, y) = y \wedge_0 ((x - y + 3a) \wedge_0 (-x - y + 3a)) \geqslant 0;$$

$$\omega_{k1}(x, y) = \omega_{k-1}(x + y + 1{,}5a,\ -x + y - 1{,}5a) \geqslant 0;$$

$$\omega_{k2}(x, y) = \omega_{k1}(-x, y) \geqslant 0;$$

$$\omega_k(x, y) = \omega_{k1}(x, y) \vee_0 \omega_{k2}(x, y) \geqslant 0,$$
$$(k = 2, 3, 4, \ldots).$$

Выводы

Математический аппарат теории R-функций оказался весьма удобным для описания объектов фрактальной геометрии функциями $\omega(x) = 0$, $x \in E^n$, E^n — евклидово пространство, (или неравенствами $\omega(x) \geqslant 0$), где $\omega(x)$ имеет вид единого аналитического выражения. При этом были использованы следующие конструктивные средства:

- R-операции системы
$$\{R_0\} = \begin{cases} x \wedge_0 y = x + y - \sqrt{x^2 + y^2}; \\ x \vee_0 y = x + y + \sqrt{x^2 + y^2}; \\ \overline{x} = -x; \end{cases}$$
- суперпозиции функции $\omega(\mu_{hx}, \mu_{hy})$ с периодическими функциями $\mu_{hx} = \frac{h_x}{\pi}\arcsin\left(\sin\frac{\pi x}{h_x}\right)$, $\mu_{hy} = \frac{h_y}{\pi}\arcsin\left(\sin\frac{\pi y}{h_y}\right)$, позволяющие транслировать заданную функцию $\omega(x, y)$ вдоль осей с шагом h_x и h_y;
- суперпозиции функции $\omega(x1 - R, y1)$, (где $x1 = r\cos\mu$, $y1 = r\sin\mu$, $\mu(\theta) = \frac{2}{n}\arcsin\left(\sin\frac{n\theta}{2}\right)$, $r = \sqrt{x^2 + y^2}$, $\theta = \operatorname{arctg}\frac{y}{x}$), позволяющие транслировать заданную функцию $\omega(x, y)$ вдоль окружности радиуса R n раз;
- свойство подобия фигур, описанных уравнениями $\omega(x, y) = 0$ и $\frac{1}{K}\omega(Kx, Ky) = 0$, где K — коэффициент подобия;
- рекурсивные процедуры.

В этой статье построены лишь некоторые, наиболее известные объекты фрактальной геометрии. Разработанные методы позволили также построить дерево Пифагора и др.

Список литературы

1. **Мандельброт Б.** Фрактальная геометрия природы. М.: Институт компьютерных исследований, 2002.
2. **Пайтген Х.-О., Рихтер П. Х.** Красота фракталов. М.: Мир, 1993.
3. **Федер Е.** Фракталы. М.: Мир, 1991.
4. **Рвачев В. Л.** Теория R-функций и некоторые ее приложения. Киев: Наукова думка, 1982.
5. **Кравченко В. Ф., Басараб М. А.** Булева алгебра и методы аппроксимации в краевых задачах электродинамики. М.: Физматлит, 2004.
6. **Максименко-Шейко К. В.** R-функции в математическом моделировании геометрических объектов и физических полей. Харьков: ИПМаш НАН Украины, 2009.
7. **Толок А. В.** Графические образы-модели в информационных технологиях // Прикладная информатика. 2009. № 4 (22). С. 31—40.

УДК 519.2

Э. Ю. Орехов, канд. физ.-мат. наук, доц.,
Ю. В. Орехов, канд. техн. наук, доц.,
Уфимский государственный авиационный технический университет,
e-mail: emil.orekhov@bk.ru

Об оценке качества эвристического алгоритма на конечной массовой задаче

Предложена характеристика качества работы эвристического алгоритма на конечной массовой задаче. Обсуждаются способы и приводятся примеры получения и оценивания введенной характеристики качества в зависимости от имеющейся информации о критерии качества данного алгоритма на индивидуальных задачах данной массовой задачи.

Показано, что в типичной ситуации возможно получение лишь статистической оценки характеристики качества, и обоснован способ ее статистического оценивания, базирующийся на равновероятной генерации индивидуальных задач данной массовой задачи. Приведен пример конкретного равновероятного генератора индивидуальных задач конечной массовой задачи целочисленного раскроя-упаковки.

Ключевые слова: *эвристический алгоритм, конечная массовая задача, характеристика качества эвристического алгоритма, статистическая оценка характеристики качества, равновероятная генерация*

Введение

Трудности, связанные с получением решения задач переборного характера и обусловленные *NP*-трудностью и большой размерностью этих задач, привели к практике построения и широкого использования эвристических алгоритмов, базирующихся как на эвристиках общего характера, так и на специфических приемах, учитывающих особенности конкретных задач, а также использующих эффекты взаимодействия эвристик различных уровней и типов [1].

Однако эвристические алгоритмы, преодолевая в той или иной степени "проклятие размерности", не гарантируют, вообще говоря, не только отыскания достаточно хорошего с точки зрения пользователя решения задачи, но и получения ее допустимого решения. Поэтому определение эффективности эвристического алгоритма, предназначенного для решения определенного класса задач, актуально как с точки зрения пользователя, получающего возможность оценить затраты на получение необходимого решения, так и с точки зрения разработчика, получающего возможность целенаправленного конструирования эвристического алгоритма на основе оценивания эффективности различных эвристик и их комбинаций.

Сложившаяся к настоящему времени практика оценки качества эвристических алгоритмов основана на тестировании интересующего алгоритма на некоторой конечной выборке задач данного класса с последующей, как правило, качественной интерпретацией полученных результатов. При этом тестовая выборка обычно формируется одним из следующих способов.

1. Элементы тестовой выборки представляют собой задачи, возникающие в ходе протекания определенного реального (например технологического) процесса. Позитивными свойствами такой выборки являются ее объективность (репрезентативность относительно множества возникающих задач), а также практически отсутствующие затраты на ее получение. Негативными свойствами являются неопределенность в описании класса решаемых задач, а также зачастую недостаточный объем имеющейся выборки.

2. Элементы тестовой выборки представляют собой задачи, специально отобранные экспертами. Как правило, это такие задачи из рассматриваемого класса, которые наиболее "труднорешаемы" для уже опробованных алгоритмов. Примером является библиотека задач прямоугольного раскроя-упаковки [2]. Позитивными свойствами такой выборки являются малые затраты на ее получение, а также возможность ее использования для совершенствования конструируемого алгоритма в сторону повышения его эффективности именно на выделенных "труднорешаемых" задачах. Основными негативными свойствами являются необъективность (нерепрезентативность) выборки по отношению к исследуемому классу задач и неопределенность именно того класса задач, по отношению к которому данная выборка репрезентативна.

3. Элементы тестовой выборки получены путем так или иначе организованной случайной генерации задач интересующего класса; пример такого случайного генератора описан в работе [3]. Основным позитивным свойством этого способа получения выборки является ненулевая вероятность попадания в нее любой задачи класса, однако вопрос о репрезентативности такой выборки по-прежнему остается открытым.

Таким образом, понятия класса решаемых задач и репрезентативности выборки во всех случаях оказываются не связанными друг с другом.

Другая трудность, возникающая при оценке качества эвристических алгоритмов, обусловлена отсутствием единого, общепринятого подхода к их тестированию, когда каждый исследователь исполь-

зует собственные показатели качества алгоритма и процедуры их оценки; это приводит к проблемам, связанным с сопоставимостью результатов, полученных разными авторами.

Цель статьи — формирование подхода к определению качества эвристического алгоритма на основе уточнения понятий класса задач, конкретной задачи данного класса, введения понятий критерия эффективности эвристического алгоритма на конкретной задаче класса, характеристики качества эвристического алгоритма на данном классе задач, а также построение процедур вычисления этой характеристики и обоснование способа ее статистического оценивания, базирующегося на равновероятной генерации задач из данного класса.

В дальнейшем изложении понятие класса задач отождествимо с понятием массовой задачи, а понятие конкретной задачи данного класса — с понятием индивидуальной задачи [4]. Далее предполагаем также, что понятие индивидуальной задачи, а, следовательно, и множество индивидуальных задач данной массовой задачи однозначно определены.

1. Определение характеристики качества эвристического алгоритма на конечной массовой задаче

Пусть П — массовая задача, а D_{Π} — множество всех индивидуальных задач, получающихся из данной массовой задачи [4]. Будем говорить, что данная массовая задача П параметризована, если выполнены следующие условия.

1. Определен конечный набор действительных переменных (параметров) $x = (x_1, ..., x_m)$, в параметрическом пространстве R^m которых:

- некоторые точки $x \in R^m$ определяют индивидуальные задачи $I_{\Pi} \in D_{\Pi}$;
- любой индивидуальной задаче $I_{\Pi} \in D_{\Pi}$ соответствует по крайней мере одна точка $x \in R^m$.

2. В параметрическом пространстве R^m определена ограниченная область $D \subset R^m$ такая, что между точками $x \in D$ и индивидуальными задачами $I_{\Pi} \in D_{\Pi}$ имеет место взаимно однозначное соответствие.

В качестве примера параметризации рассмотрим следующую массовую задачу.

Имеется последовательность, состоящая из N различных натуральных чисел от 1 до N, и задано натуральное число n, $1 \leqslant n \leqslant N$. Нужно определить номер данного числа n в имеющейся последовательности.

При заданных N, n индивидуальная задача определяется заданием любой последовательности — перестановки чисел {1, ..., N}, в которой число n находится на определенном месте последовательности. Таким образом, данная массовая задача содержит N различных индивидуальных задач.

Пусть $x = (x_1, ..., x_N)$ — набор параметров, принимающих каждый натуральные значения; x_i — число, находящееся на i-м месте рассматриваемой последовательности, $i = 1, ..., N$; R^N — соответствующее параметрическое пространство точек x. Очевидно, что точки $x \in R^N$, имеющие различные значения координат $1 \leqslant x_i \leqslant N$, определяют индивидуальные задачи данной массовой задачи, а любой индивидуальной задаче с $x_i = n$ соответствуют $(N - 1)!$ этих точек.

Область $D \subset R^N$, каждая точка которой взаимно однозначно соответствует индивидуальной задаче данной массовой задачи, может быть определена как

$$D = \left\{ \begin{array}{l} x^1 = (n, 1, 2, ..., n-1, n+1, ..., N-1, N), \\ x^2 = (1, n, 2, ..., n-1, n+1, ..., N-1, N), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x^n = (1, 2, ..., n-1, n, n+1, ..., N-1, N), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x^{N-1} = (1, 2, ..., n-1, n+1, ..., N-1, n, N), \\ x^N = (1, 2, ..., n-1, n+1, ..., N-1, N, n). \end{array} \right\}.$$

В качестве другого примера параметризации рассмотрим массовую целочисленную задачу одномерного раскроя в следующей постановке [5].

Имеются одинаковые отрезки длины L (L — натуральное число), в которые нужно упаковать без перекрытия набор отрезков целочисленной длины $x_1, ..., x_m$; $1 \leqslant x_i \leqslant L$, $i = 1, ..., m$, так, чтобы число использованных отрезков длины L оказалось наименьшим.

При заданных L, m индивидуальная задача определяется заданием длин $x_1, ..., x_m$, $i = 1, ..., m$, упаковываемых отрезков.

Пусть $x = (x_1, ..., x_m)$ — набор параметров, принимающих каждый натуральные значения; x_i — длина i-го отрезка, $i = 1, ..., m$; R^m — соответствующее параметрическое пространство точек x. Очевидно, что точки $x \in R^m$, имеющие (не обязательно различные) значения координат $1 \leqslant x_i \leqslant L$, определяют индивидуальные задачи данной массовой задачи, а любой индивидуальной задаче, определяемой набором длин упаковываемых отрезков, соответствует некоторое число этих точек, лежащее в пределах от 1 до $m!$.

Так как любая перестановка чисел $x_1, ..., x_m$ соответствует одной и той же индивидуальной задаче, то множество всех индивидуальных задач рассматриваемой массовой задачи ставится во взаимно

однозначное соответствие точкам области D, определяемой следующей системой ограничений:

$$D = \{x = (x_1, ..., x_m): x_i = 1, 2, ...; i = 1, ..., m; \\ 1 \leqslant x_1 \leqslant ... \leqslant x_m \leqslant L\}.$$

Пусть H — эвристический алгоритм, предназначенный для решения данной массовой задачи П. Предполагая массовую задачу П параметризованной, введем в рассмотрение определенную в области $D \subset R^m$ действительную функцию $f(x)$ — критерий качества алгоритма H на индивидуальной задаче $I_\Pi \in D_\Pi$, образом которой в области $D \subset R^m$ является точка $x \in D$. Учитывая взаимно однозначное соответствие между индивидуальными задачами $I_\Pi \in D_\Pi$ и точками $x \in D$, будем в дальнейшем говорить об индивидуальных задачах $x \in D$.

Массовую задачу П будем называть конечной, если D_Π — конечное множество; в этом случае область D содержит конечное число точек, которое обозначим через V. Примерами конечных массовых задач являются сформулированные выше массовая задача определения номера заданного элемента в последовательности и массовая целочисленная задача одномерного раскроя при заданных L, m.

В дальнейшем изложении рассматриваются только конечные массовые задачи, что обусловлено следующими причинами.

1. Достаточно большая часть реальных проблем либо представляют собой конечные массовые задачи, либо могут быть вполне адекватно представлены массовой задачей этого типа.

2. В случае конечной массовой задачи все вводимые понятия наглядны, а используемый математический аппарат наиболее прост.

3. Обобщение полученных результатов на случай, когда все или часть параметров могут меняться непрерывно, не представляет принципиальных трудностей, хотя при рассмотрении массовых задач более общего вида возможно возникновение трудностей технического характера.

Введем в рассмотрение подмножество D_y множества точек $x \in D$, определенное как $D_y = \{x : x \in D, f(x) < y\}$, y — действительное число, и пусть V_y — число точек в D_y.

Качество работы алгоритма H на конечной массовой задаче П будем характеризовать всюду определенной действительной функцией

$$F(y) = V_y/V. \qquad (1)$$

Величина $F(y)$ есть доля индивидуальных задач $x \in D$, имеющих значение критерия качества f меньше заданного числа y.

Приведем примеры определения критерия качества $f(x)$ алгоритма H на индивидуальной задаче $x \in D$ и соответствующей характеристики качества $F(y)$ алгоритма H на массовой задаче П.

Пример 1. Пусть $f(x) = \begin{cases} 1, \textit{ если } H \textit{ решает} \\ \textit{задачу } x; \\ 0, \textit{ в противном случае.} \end{cases}$

Тогда

$$F(y) = \begin{cases} 0, \ y \leqslant 0; \\ a, \ 0 < y \leqslant 1, \ 0 \leqslant a \leqslant 1; \\ 1, \ y > 1, \end{cases}$$

где a — доля задач из D, не решаемых алгоритмом H; соответственно, $1 - a$ — доля задач из D, решаемых алгоритмом H. В этом случае из двух алгоритмов H_1, H_2 лучшим на D (т. е. на соответствующей D массовой задаче П) будет тот, у которого значение величины a меньше, т. е. H_1 лучше H_2, если $F_{H_1}(y) < F_{H_2}(y)$, $y \in (0, 1]$.

Пример 2. Пусть массовая задача П — это задача минимизации заданного критерия c на некотором заданном множестве, и пусть $\underline{c}(x) > 0$ — оценка снизу значения величины c на индивидуальной задаче $x \in D$, а $c(x)$ — наименьшее значение величины c на индивидуальной задаче $x \in D$, вычисляемое алгоритмом H. Положим

$$f(x) = \frac{c(x) - \underline{c}(x)}{\underline{c}(x)}.$$

Тогда

$$F(y) = \begin{cases} 0, \ y \leqslant 0; \\ a_1, \ 0 < y \leqslant b_1; \\ a_2, \ b_1 < y \leqslant b_2; \\ ..... \\ a_k, \ b_{k-1} < y \leqslant b_k; \\ 1, \ y > b_k, \end{cases}$$

где $0 < a_1 < ... < a_k < 1$, $0 < b_1 < ... < b_k$. В этом случае из двух алгоритмов H_1, H_2 лучшим на D (т. е. на соответствующей D массовой задаче П) будет H_1, если $F_{H_1}(y) \geqslant F_{H_2}(y)$, и хотя бы для одной точки $y_0 > 0$ будет $F_{H_1}(y_0) > F_{H_2}(y_0)$.

Таким образом, задача определения качества работы эвристического алгоритма H на массовой задаче П при заданном критерии f качества работы этого алгоритма на индивидуальной задаче сводится к отысканию функции F.

2. Решение задачи отыскания функции F

2.1. Непосредственное построение функции F

В качестве примера построения функции F рассмотрим описанную выше конечную массовую задачу определения номера заданного элемента в последовательности, для решения которой ис-

пользуется простой переборный алгоритм, сравнивающий заданное число n поочередно с элементами последовательности, начиная с первого, и останавливающийся на элементе, совпадающем с n. Пусть критерием качества f такого алгоритма на индивидуальной задаче данной конечной массовой задачи является число просмотренных алгоритмом элементов последовательности. Так как данная массовая задача содержит N индивидуальных задач и значение критерия f на индивидуальной задаче $x^i \in D$ есть $f(x^i) = i$, то получаем

$$F(y) = \begin{cases} 0, \ y \leqslant 1; \\ \frac{1}{N}, \ 1 < y \leqslant 2; \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \frac{i}{N}, \ i < y \leqslant i + 1; \\ \ldots\ldots\ldots\ldots \\ \frac{N-1}{N}, \ N - 1 < y \leqslant N; \\ 1, \ y > N. \end{cases}$$

Отметим, что найти $F(y)$ в данном случае оказалось легко благодаря очень простой структуре как рассматриваемой массовой задачи, так и используемого алгоритма и критерия качества f этого алгоритма на индивидуальной задаче, значение которого на каждой индивидуальной задаче априори известно.

В реальных, достаточно сложных ситуациях найти характеристику качества $F(y)$ точно не удается, но оказывается возможным получить приемлемую оценку этой величины.

2.2. Статистическое оценивание функции F

Заметим, что введенная функция F обладает всеми характеристическими свойствами функции распределения, и поэтому является функцией распределения некоторой дискретной случайной величины Y с конечным множеством возможных значений.

Каждому параметру x_i поставим в соответствие дискретную случайную величину X_i, $i - 1, \ldots, n$, и рассмотрим систему дискретных случайных величин $X = (X_1, \ldots, X_m)$, областью возможных значений которой является область D. Положим $Y = f(X) = f(X_1, \ldots, X_m)$ и поставим следующий вопрос: какому закону распределения должна подчиняться система случайных величин $X = (X_1, \ldots, X_m)$, чтобы случайная величина Y имела функцию распределения, определенную соотношением (1)?

Утверждение. Пусть система случайных величин равновероятно распределена в D, т. е. $P(X = x) = P(X_1 = x_1, \ldots, X_m = x_m) = \frac{1}{V}$ для любой точки $x \in D$. Тогда функция распределения $\hat{F}(y)$ случайной величины $Y = f(X)$ есть $F(y)$, определенная соотношением (1), для любой функции f.

Доказательство. По определению функции распределения случайной величины имеем

$$\hat{F}(y) = P(Y < y) = P(f(X_1, \ldots, X_m) < y) =$$

$$= \sum_{\substack{f(x_1, \ldots, x_m) < y \\ (x_1, \ldots, x_m) \in D}} P(X_1 = x_1, \ldots, X_m = x_m) =$$

$$= \sum_{\substack{f(x_1, \ldots, x_m) < y \\ (x_1, \ldots, x_m) \in D}} \frac{1}{V} = \frac{1}{V} \sum_{\substack{f(x_1, \ldots, x_m) < y \\ (x_1, \ldots, x_m) \in D}} 1 = \frac{V_y}{V} = F(y).$$

Замечание 1. Найденное решение поставленной задачи не является, вообще говоря, единственным. Пусть, например, $m = 1$, т. е. $X = X_1$, $D = \{1, 2, 3\}$, критерий f определен как $f(1) = f(2) = 0$, $f(3) = 1$, тогда

$$F(y) = \begin{cases} 0, \ y \leqslant 0; \\ \frac{2}{3}, \ 0 < y \leqslant 1; \\ 1, \ y > 1, \end{cases}$$

т. е. закон распределения случайной величины Y можно задать в виде $P(Y = 0) = \frac{2}{3}$, $P(Y = 1) = \frac{1}{3}$. Задав закон распределения X в виде $P(X = 1) = p_1$, $P(X = 2) = p_2$, $P(X = 3) = p_3$, с учетом определения $Y = f(X)$ получим

$$\begin{cases} p_1 + p_2 = \frac{2}{3} \\ p_3 = \frac{1}{3} \\ p_i \geqslant 0, \ i = 1, 2, 3, \end{cases}$$

т. е. решением задачи является любой из законов распределения случайной величины X вида $P(X = 1) = p_1$, $P(X = 2) = \frac{2}{3} - p_1$, $P(X = 3) = \frac{1}{3}$, где $0 \leqslant p_1 \leqslant \frac{2}{3}$. Конечно, равновероятный закон распределения X в D, определяемый значениями $p_1 = p_2 = p_3 = \frac{1}{3}$, тоже входит в бесконечное множество решений данной задачи. Отметим также, что все множество решений данной задачи удалось найти благодаря наличию полной информации о значениях функции f в каждой точке области D.

Замечание 2. Равновероятный закон распределения X в D будет, очевидно, единственным решением сформулированной задачи, если между элементами множеств D и D_f, где D_f — область значений f при заданной области определения D этой функции, имеет место взаимно однозначное соответствие.

Замечание 3. Из доказанного утверждения и сделанных замечаний следует, что равновероятный закон распределения X в D является единственным законом, который всегда входит во множество решений сформулированной задачи, в том числе и в условиях, когда функция f априори неизвестна (полностью или частично).

Учитывая изложенное выше, задача оценки качества эвристического алгоритма H на массовой задаче П, характеризуемого функцией $F(y)$, сводится к задаче оценки функции распределения $F(y)$ введенной случайной величины Y, являющейся некоторой функцией f системы дискретных случайных величин $X = (X_1, ..., X_m)$, равновероятно распределенной в D.

Функция f, как правило, априори неизвестна, но ее значение может быть вычислено для любого $x \in D$ путем прогонки алгоритма H на соответствующей x индивидуальной задаче $I_{\Pi} \in D_{\Pi}$, поэтому статистическая оценка $F(y)$ может быть получена следующим образом [6].

Пусть $y_1, ..., y_n$ — результаты n независимых реализаций случайной величины Y. Расположим их в порядке возрастания, обозначив через y_i^* i-е реализованное значение Y, тогда последовательность $y_1, ..., y_n$ можно записать в виде вариационного ряда $y_1^* \leqslant ... \leqslant y_n^*$. Определим эмпирическую функцию распределения

$$F_n(y) = \begin{cases} 0, & y \leqslant y_1^*; \\ \dfrac{i}{n}, & y_i^* < y \leqslant y_{i+1}^*, \ i = 1, ..., n-1; \\ 1, & y > y_n^*. \end{cases}$$

Отметим, что в ситуации до получения реализаций случайной величины Y эмпирическая функция распределения $F_n(y)$ для каждого значения y является случайной величиной.

По теореме Гливенко [6]

$$P\{\sup_{-\infty < y < \infty} |F_n(y) - F(y)| \xrightarrow{n \to \infty} 0\} = 1,$$

поэтому $F_n(y)$ при достаточно большом n дает хорошее представление об интересующей величине $F(y)$ и может рассматриваться в качестве статистической оценки $F(y)$.

Для получения независимых реализаций $y_1, ..., y_n$ случайной величины Y необходимо:

- получить n независимых реализаций $x^1, ..., x^n$ системы случайных величин $X = (X_1, ..., X_m)$, *равновероятно* распределенных в D;
- прогонкой алгоритма H на каждой из n индивидуальных задач, определяемых наборами значений параметров $x^1, ..., x^n$, найти

$$y_1 = f(x^1),$$
$$.....$$
$$y_n = f(x^n).$$

Отметим, что наряду с "функциональной" характеристикой качества $F(y)$ эвристического алгоритма H на массовой задаче П можно ввести в рассмотрение и более простые числовые характеристики качества, например, математическое ожидание $M[Y]$ и дисперсию $D[Y]$ случайной величины Y, статистические оценки которых

$$\tilde{M}\{Y\} = \frac{1}{n} \sum_{i=1}^{n} y_i,$$

$$\tilde{D}[Y] = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \tilde{M}[Y])^2$$

также могут быть вычислены по n независимым реализациям случайной величины Y.

Таким образом, необходимым этапом статистического оценивания характеристики качества $F(y)$ алгоритма H на массовой задаче П является решение задачи равновероятной генерации точек $x \in D$, соответствующих индивидуальным задачам $I_{\Pi} \in D_{\Pi}$.

2.3. Пример равновероятной генерации точек области D

Рассмотрим решение задачи равновероятной генерации точек $x \in D$, где $D = \{x = (x_1, ..., x_m)$: $x_i = 1, 2, ...$; $i = 1, ..., m$; $1 \leqslant x_1 \leqslant ... \leqslant x_m \leqslant L\}$ — область в параметрическом пространстве, соответствующая сформулированной выше массовой целочисленной задаче одномерного раскроя при заданных L, m [5]. В этом случае имеем [5]:

$$V = \frac{L(L+1)...(L+m-1)}{m!},$$

поэтому равновероятный закон распределения системы случайных величин $X = (X_1, ..., X_m)$ в D имеет вид

$$P(X_1 = x_1, ..., X_m = x_m) =$$
$$= \begin{cases} \dfrac{m!}{L(L+1)...(L+m-1)}, & (x_1, ..., x_m) \in D; \\ 0, & (x_1, ..., x_m) \notin D, \end{cases}$$

а условные законы распределения входящих в систему случайных величин таковы:

$$P(X_1 = x_1/X_2 = x_2, \ldots, X_m = x_m) = \begin{cases} \frac{1}{x_2}, x_1 = 1, \ldots, x_2, \\ 0, \text{ в противном случае;} \end{cases}$$

$$P(X_2 = x_2/X_3 = x_3, \ldots, X_m = x_m) = \begin{cases} \frac{2}{x_3(x_3+1)} x_2, x_2 = 1, \ldots, x_3, \\ 0, \text{ в противном случае;} \end{cases}$$

.....

$$P(X_i = x_i/X_{i+1} = x_{i+1}, \ldots, X_m = x_m) = \begin{cases} \frac{i}{x_{i+1}(x_{i+1}+1)\ldots(x_{i+1}+i-1)} x_i(x_i+1) \ldots \\ \ldots (x_i + i - 2), x_i = 1, \ldots, x_{i+1}, \\ 0, \text{ в противном случае;} \end{cases}$$

.....

$$P(X_{m-1} = x_{m-1}/X_m = x_m) = \begin{cases} \frac{m-1}{x_m(x_m+1)\ldots(x_m+m-2)} x_{m-1}(x_{m-1}+1) \ldots \\ \ldots (x_{m-1} + m - 3), x_{m-1} = 1, \ldots, x_m, \\ 0, \text{ в противном случае;} \end{cases}$$

$$P(X_m = x_m) = \begin{cases} \frac{m}{L(L+1)\ldots(L+m-1)} x_m(x_m+1) \ldots \\ \ldots (x_m + m - 2), x_m = 1, \ldots, L, \\ 0, \text{ в противном случае.} \end{cases}$$

Генерация реализаций случайных величин X_m, X_{m-1}, ..., X_1 осуществляется в данной последовательности в соответствии с приведенными безусловным для X_m и условными для остальных случайных величин системы X законами распределения с помощью стандартного алгоритма [7].

Заключение

Предложенный в работе подход основан на понятиях конечной массовой задачи, индивидуальной задачи, параметризации массовой задачи, критерия качества эвристического алгоритма на индивидуальной задаче и, наконец, характеристики качества данного эвристического алгоритма на данной конечной массовой задаче.

Интерпретация этой характеристики как функции распределения некоторой случайной величины позволяет сформулировать задачу ее статистического оценивания и решить эту задачу на основе равновероятной генерации точек области параметрического пространства данной массовой задачи, взаимно однозначно соответствующих индивидуальным задачам этой массовой задачи.

Приведенный пример равновероятного генератора для массовой целочисленной задачи линейного раскроя-упаковки показывает, что построение такого генератора является, вообще говоря, технически непростой задачей.

Отметим также, что предложенный подход без каких-либо изменений применим для оценки качества не только эвристических, но и любых других алгоритмов (точных и приближенных, детерминистских и стохастических) на конечных массовых задачах, а распространение данного подхода на массовые задачи других типов не наталкивается на трудности принципиального характера.

Список литературы

1. **Норенков И. П.** Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. 1999. № 1. С. 2—7.

2. **Bortfeld A.** A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. // Eur. J. Oper. Res. 2006. V. 172 (3). P. 814—837.

3. **Schwerin P., Waecher G.** The bin-packing problem: A problem generator and some numerical experiments with FFD packing and MTP // International transactions in operational research. 1997. V. 4, № 5/6. P. 337—389.

4. **Гэри М., Джонсон Д.** Вычислительные машины и трудноразрешаемые задачи. М.: Мир, 1982. — 416 с.

5. **Orekhov E. Yu, Orekhov Yu. V.** Equiprobable Generation of the Integer One-Dimensional Cutting-Packing Problem // Proc. of the 5th International Workshop on Computer Science and Information Technologies. Ufa, Russia. September 16—18, 2003. V. 2. P. 41—42.

6. **Гнеденко Б. В.** Курс теории вероятностей. 3-е изд. М.: Физматлит, 1961. 406 с.

7. **Бусленко Н. П., Голенко Д. И., Соболь И. М., Срагович В. Г., Шрейдер Ю. А.** Метод статистических испытаний (метод Монте-Карло) // Под ред. Ю. А. Шрейдера. — М.: Государственное изд-во физико-математической литературы, 1962. 332 с.

УДК 004.3.06:62-507

А. А. Ахи, студент,
А. С. Станкевич, доц.,
А. А. Шалыто, д-р техн. наук, проф., зав. каф.,
Санкт-Петербургский государственный
университет информационных технологий,
механики и оптики
e-mail: akhi@rain.ifmo.ru

Алгоритм построения флибов со 100 %-ной точностью предсказания

Предложен алгоритм построения флиба с минимальным числом состояний, который имеет 100 %-ную точность предсказания очередного значения окружающей среды.

Ключевые слова: *конечный автомат, автомат Мили, флиб*

Введение

В работах [1—3] рассмотрена задача о флибах, которая состоит в том, чтобы построить флиб — детерминированный конечный автомат Мили, предсказывающий значение некоторого битового параметра окружающей среды на основе ранее полученных данных.

В указанных работах рассматривалась автоматическая генерация автоматов с помощью генетических алгоритмов. Вопрос о необходимом и достаточном числе состояний автомата для 100 %-ной точности предсказания не рассматривался, хотя в работе [3] благодаря удачному выбору фитнесс-функции было значительно уменьшено их число по сравнению с другими работами.

В настоящей статье предложен алгоритм построения флиба с минимальным числом состояний, имеющего 100 %-ную точность предсказания очередного значения окружающей среды.

Постановка задачи

Задача состоит в моделировании простейшего существа, способного предсказывать изменения параметра среды, обладающего периодичностью. В качестве простейшей модели такого существа можно использовать конечный автомат Мили. В работе [1] такие конечно-автоматные модели были названы флибами (сокращение от *finite living blobs* — конечные живые капельки). На вход флиба подается переменная, которая принимает одно из двух значений — ноль или единица. Эта переменная соответствует состоянию окружающей среды в текущий момент времени. Рассматривается параметр среды, имеющий лишь два возможных значения. Флиб изменяет свое состояние и формирует значение выходной переменной, принимающей одно из двух указанных значений. Это значение соответствует возможному состоянию среды в следующий момент времени. Задача флиба — предсказать, какое на самом деле состояние окружающей среды наступит в следующий момент времени. Это можно выполнить благодаря периодичности изменений состояний среды.

Размер идеального флиба

Будем считать, что битовая маска s, задающая изменение окружающей среды, не является периодичной: $s \neq s_0^n$ ни для какого $n \geqslant 2$. В этом случае обозначим *zeros* число нулей в маске, а *ones* — число единиц.

Утверждение. Флиб, предсказывающий поведение среды с точностью 100 %, имеет не менее max(*zeros*, *ones*) состояний.

Доказательство. Докажем данное утверждение от обратного. Не умаляя общности, пусть *ones* $\geqslant$ *zeros* и пусть также существует автомат из менее чем *ones* состояний, который имеет точность 100 %. Тогда существуют две единицы, угадав которые, автомат приходит в одно и то же состояние. Пусть $|s| = n$, и эти две единицы оказались на i-й и j-й позициях в маске. Так как автомат оказался в одном и том же состоянии, угадав при этом один и тот же символ, то далее автомат будет себя вести одинаково в обоих случаях, так как флиб все время верно угадывает изменения параметра.

Однако в этом случае получаем, что $s[i + k] = s[j + k]$ при $\forall k \in N$. Следовательно, рассмотренная маска s является периодичной.

Таким образом, получаем противоречие. ◀

Теперь известно, что для 100 %-ной точности требуется хотя бы max(*zeros*, *ones*) состояний. Далее будет показано, как построить автомат, на котором достигается эта оценка.

Построение идеального флиба

Рассмотрим построение автомата на примере строки 1111010010. Здесь *zeros* = 4, *ones* = 6. Поэтому на основе приведенного выше утверждения флиб должен иметь не менее шести состояний. Сначала построим тривиальный автомат из $|s|$ состояний (рис. 1).

В верхнем слое изображены состояния, из которых существует переход только по единице, а в нижнем — только по нулю. Совместим состояния нижнего слоя с состояниями верхнего слоя так,

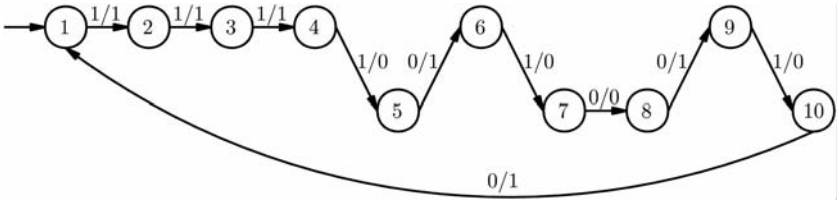

Рис. 1. Тривиальный автомат с $|s|$ состояниями

чтобы каждому состоянию из верхнего состояния соответствовало не более одного состояния из нижнего слоя. Ребра из нового состояния будут вести в совмещенные аналоги прежних состояний (рис. 2). Совмещение можно проводить по-разному. На этом рисунке приведены два примера таких совмещений.

Получившиеся автоматы имеют *ones* состояний. С помощью совмещения состояний можно построить автомат для любой маски.

Опишем ***алгоритм*** построения более формально. Пусть $n = \max(zeros, ones)$ — число состояний в автомате. Пронумеруем отдельно нули и единицы в строке. Будем считать, что строка зациклена. Построим автомат таким образом, чтобы i-е состояние отвечало за действия после i-го нуля и/или единицы. Ребра будут строиться по следующим правилам:

1. Если после i-го нуля в строке идет ноль, то проведем из i-го состояния в состояние с номером $i + 1$ ребро с пометкой 0/0.

2. Если после i-го нуля в строке идет j-я единица, то проведем из i-го состояния в состояние с номером j ребро с пометкой 0/1.

3. Если после i-й единицы в строке идет j-й нуль, то проведем из i-го состояния в состояние с номером j ребро с пометкой 1/0.

4. Если после i-й единицы в строке идет единица, то проведем из i-го состояния в состояние с номером $i + 1$ ребро с пометкой 1/1.

Полученный таким образом автомат имеет точность предсказания 100 %.

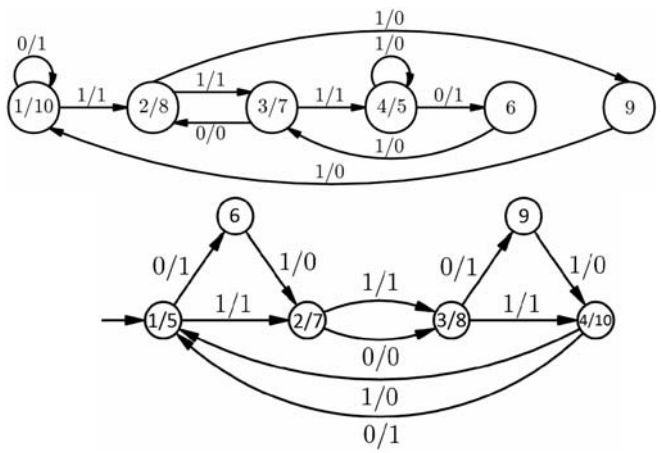

Рис. 2. Два автомата с *ones* состояниями

На рис. 3—7 приведен процесс построения автомата на основе предложенного алгоритма для строки 1111010010.

На рис. 3 показан первый шаг алгоритма — так как после первой единицы в строке стоит вторая единица, то по правилу 4 проводим ребро из вершины 1 в вершину 2 с пометкой 1/1.

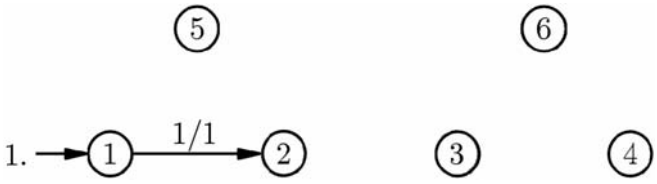

Рис. 3. Первый шаг алгоритма

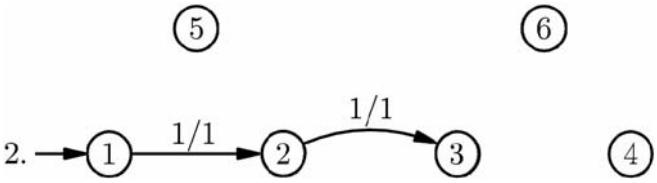

Рис. 4. Второй шаг

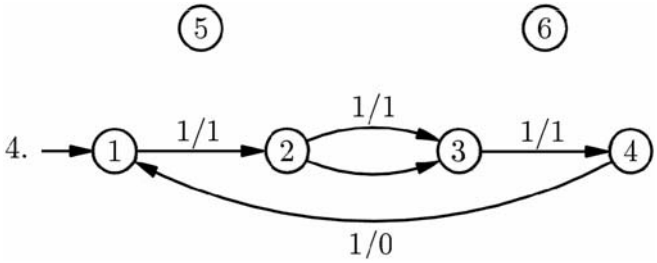

Рис. 5. Четвертый шаг

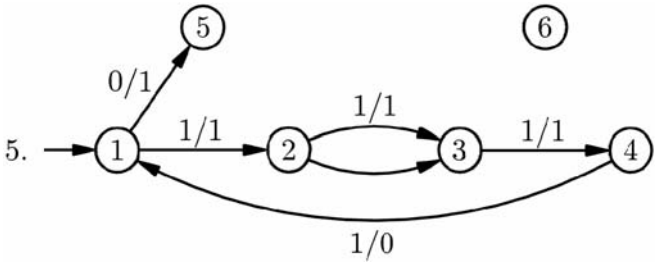

Рис. 6. Пятый шаг

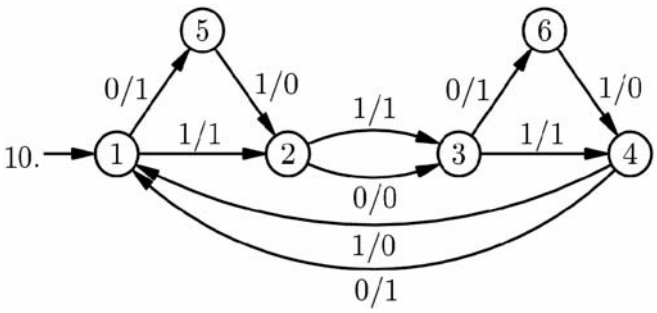

Рис. 7. Десятый шаг

На рис. 4 показан второй шаг алгоритма. При этом, так как после второй единицы в строке стоит третья единица, то по правилу 4 проводим ребро из вершины 2 в вершину 3 с пометкой 1/1.

Третий шаг не приводится, так как он схож с первыми двумя. На рис. 5 показан четвертый шаг алгоритма. При этом, так как после четвертой единицы в строке стоит первый нуль, то по правилу 3 проводим ребро из вершины 4 в вершину 1 с пометкой 1/0.

На рис. 6 показан пятый шаг алгоритма. При этом, так как после первого нуля в строке стоит пятая единица, то по правилу 2 проводим ребро из вершины 1 в вершину 5 с пометкой 0/1.

Шаги с шестого по девятый выполняются аналогичным образом. На рис. 7 показан последний, десятый шаг алгоритма. При этом, так как после четвертого нуля в строке стоит первая единица, то по правилу 2 проводим ребро из вершины 4 в вершину 1 с пометкой 0/1. Искомый автомат построен.

Таблица 1

Алгоритм	Число состояний	Точность предсказания, %
1	20	88
2	20	100
3	12	100
Предложенный	12	100

Таблица 2

Алгоритм	Число состояний	Точность предсказания, %
1	30	87
2	30	97
3	20	100
Предложенный	20	100

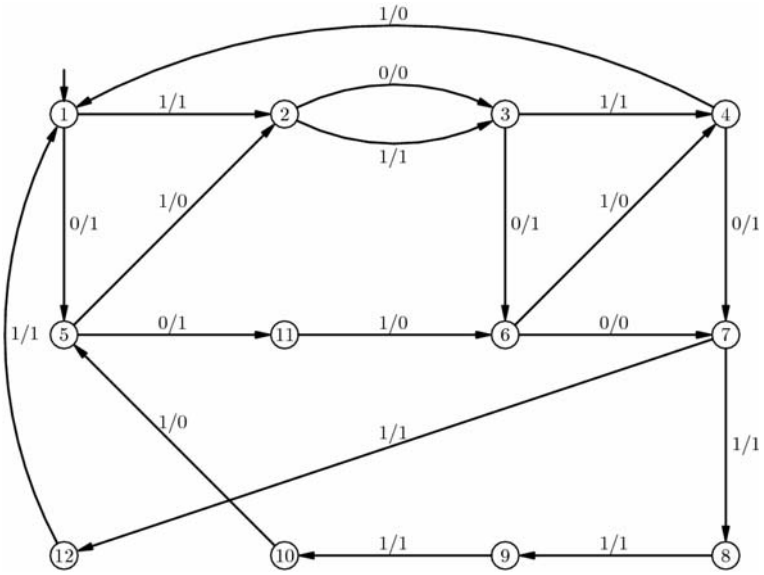

Рис. 8. Автомат с 12 состояниями для строки из 19 символов. Неиспользуемые переходы не изображены

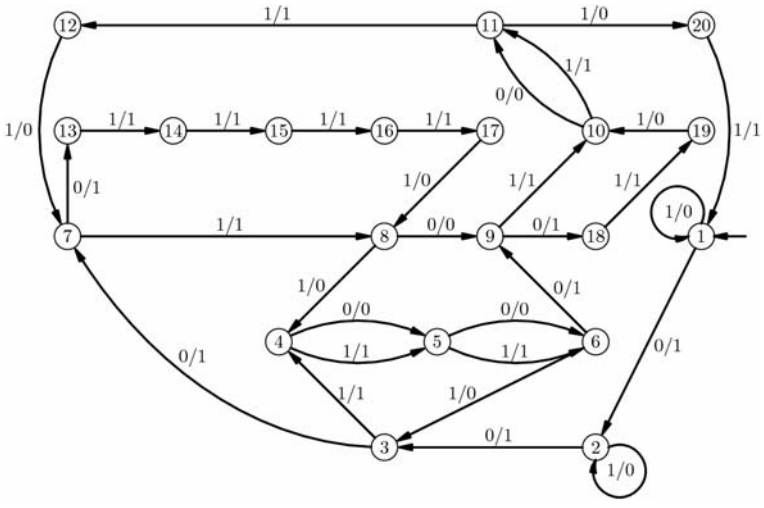

Рис. 9. Автомат с 20 состояниями для строки из 31. Неиспользуемые переходы не изображены

Эксперименты

Ниже приводятся результаты экспериментов, основанных на применении трех алгоритмов (алгоритмы 1—3), изложенных в работах [1—3], и предлагаемого алгоритма.

Эксперимент 1. Битовая маска из 19 символов: 1111010010111101001. Результаты эксперимента приведены в табл. 1.

Автомат, построенный предложенным алгоритмом, приведен на рис. 8.

Эксперимент 2. Битовая маска из 31 символа: 10101111011000111101111100110 01.

Результаты эксперимента приведены в табл. 2.

Автомат, построенный предложенным алгоритмом, приведен на рис. 9.

"Слепые" флибы

Описанный выше алгоритм позволяет построить автомат с минимальным числом состояний и точностью предсказания 100 %. Однако остается открытым вопрос о необходимом числе состояний автомата для получения любой другой фиксированной точности. Однако такую задачу можно решить для "слепых" флибов, которая и рассматривается ниже.

Определение и постановка задачи. "Слепой" флиб — автомат Мили, который пытается угадать поведение окружающей среды, однако на вход он получает не текущее состояние окружающей среды, а то значение, которое он выдал на предыдущем шаге.

Получается, что "слепой" флиб всегда считает, что он верно угадал значение битовой переменной и не "смотрит" на окружающую среду. Поведение такого флиба детерминировано. При этом сначала, возможно, флиб выдаст какую-то строку p_0, а затем будет периодически выдавать строку p. Будем называть p — периодом, а p_0 — предпериодом. Для простоты будем считать, что "сле-

пой" флиб не имеет предпериода: $p_0 = \varepsilon$. Задача состоит в том, чтобы для фиксированного числа состояний n и заданной строки s, описывающей поведение окружающей среды, построить "слепой" флиб, который имеет наибольшую точность предсказания.

Решение новой задачи. Из описанного выше следует, что для $n \geqslant \max(zeros, ones)$ максимальная точность составляет 100 % и автомат строится описанным выше алгоритмом (флиб, имеющий 100 %-ную точность ведет себя как "слепой"). Рассмотрим строку p. Заметим, что "слепой" флиб ведет себя в точности как "зрячий" флиб, который смотрит на окружающую среду p и имеет точность 100 %. Из этого следует, что $n \geqslant \max(z_p, o_p)$, где z_p и o_p — число нулей и единиц в p соответственно. Таким образом, получаем, что "слепой" флиб может выдавать любую строку p, содержащую не более n нулей и не более n единиц.

Задача свелась к задаче со строками: построить такую строку p, состоящую из не более n нулей и не более n единиц для того, чтобы величина $\dfrac{same(s^{|p|}, p^{|s|})}{|s||p|}$ была максимальна. Здесь $same(s^{|p|}, p^{|s|})$ — число совпадающих символов в строках $s^{|p|}$ и $p^{|s|}$.

Решение задачи со строками. Эта задача решается с помощью динамического программирования [4]. Сначала решим эту задачу для фиксированной длины p.

Решение при фиксированной длине. Положим $m = |p|$. Будем вычислять функцию a_{ij} — максимальное число совпадений, которое можно получить, зафиксировав первые $i + j$ символов p, среди которых i нулей и j единиц. При подсчете числа совпадений остальные символы p не рассматриваются.

Для вычисления значений этой функции понадобятся некоторые вспомогательные величины:

- b_i — число нулей, которые встречаются в s^m на позициях с номерами $mk + i$, где $0 \leqslant k < |s|$ — целое число. При этом $b_i = \sum\limits_{k=0}^{|s|} [s^m[mk + i] = 0]$;
- $c_i = |s| - b_i$ — число единиц, которые встречаются в s^m на позициях с номерами $mk + i$.

База динамики — $a_{00} = 0$, $a_{ij} = +\infty \ \forall i, j: i \neq 0 \vee j \neq 0$.

Переход осуществляется следующим образом: пусть известно значение a_{ij}, тогда:

- если $i < n$, то $a_{i+1,j} = \max(a_{i+1,j}, a_{ij} + b_{i+j+1})$. Этот переход соответствует попытке приписать к уже имеющейся части p нуль в конец;
- если $j < n$, то $a_{i,j+1} = \max(a_{i,j+1}, a_{ij} + c_{i+j+1})$. Этот переход соответствует попытке приписать к уже имеющейся части p единицу в конец.

В результате $x = \max\limits_{|p| - n \leqslant i \leqslant n} a_{i,\,|p| - i}$ — максимальное число совпадающих символов, которое можно получить при фиксированной длине p. По результатам вычислений можно получить саму строку p, на которой достигается такой результат.

Решение задачи в общем случае. В общем случае необходимо перебрать все возможные длины p от 1 до $2n$. Среди всех полученных решений для разных длин следует выбрать то, для которого величина $\dfrac{x(p)}{|p|}$ максимальна.

Воспользовавшись алгоритмом построения идеального флиба для строки p, получим желаемый оптимальный "слепой" флиб. Указанным образом для любого n можно узнать, с какой наибольшей точностью "слепой" флиб с не более чем n состояниями может угадывать поведение окружающей среды, задаваемой строкой s.

Заключение

Для некоторых строк автомат с минимальным числом состояний и 100 %-ной точностью предсказания был построен в работе [3] с помощью генетических алгоритмов. При этом на вычисления уходило много времени и не было гарантии достижения оптимального результата для произвольных строк. Алгоритм, предложенный в этой статье, строит автомат с минимальным числом состояний и 100 %-ной точностью предсказания с существенно меньшими временными затратами. Остается открытым вопрос о необходимом числе состояний автомата для получения любой другой фиксированной точности. Однако для "слепых" флибов эта задача в настоящей работе решена.

Список литературы

1. **Воронин О., Дьюдни А.** Дарвинизм в программировании // Мой компьютер. 2004. № 35. URL: http://www.mycomp.kiev.ua/text/7458

2. **Лобанов П. Г., Шалыто А. А.** Использование генетических алгоритмов для автоматического построения конечных автоматов в задаче о "Флибах" // Матер. 1-й Российской мультиконференции по проблемам управления. Сб. докл. 4-й Всероссийской научной конференции "Управление и информационные технологии" (УИТ-2006). СПб.: Изд. СПбГЭТУ "ЛЭТИ". 2006. URL: http://is.ifmo.ru/works/flib/

3. **Мандриков Е. А., Кулев В. А., Шалыто А. А.** Применение генетических алгоритмов для создания управляющих автоматов в задаче о "Флибах" // Информационные технологии. 2008. № 1. URL: http://is.ifmo.ru/download/2008-02-23_flibs.pdf

4. **Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.** Алгоритмы: построение и анализ. Гл. 15. Динамическое программирование. М.: Вильямс, 2005.

УДК 004.9; 528.8

А. В. Замятин, канд. техн. наук, доц.,
Национальный исследовательский
Томский политехнический университет,
e-mail: zamyatin@tpu.ru

Концепция региональной информационной системы аэрокосмического мониторинга с интеллектуальной распределенно-параллельной обработкой данных

Предложена концепция построения региональной информационной системы, предназначенной для комплексного решения современных задач аэрокосмического мониторинга. Отличительной особенностью системы является возможность использования данных дистанционного зондирования Земли с различными характеристиками, позволяющая с высокой степенью точности и оперативности выполнять сложную интеллектуальную обработку данных и создавать широкий спектр оригинальных информационных продуктов, направленных на решение задач аэрокосмического мониторинга. Значительное увеличение производительности обработки данных достигается применением методов и алгоритмов, адаптированных для распределенно-параллельных вычислений, применимых как на дорогостоящих суперкомпьютерах, так и на кластерах из недорогих ПЭВМ.

Ключевые слова: *аэрокосмический мониторинг, ландшафтный покров, высокопроизводительная система, распределенно-параллельные вычисления, интеллектуальная обработка данных, автоматизированная интерпретация, моделирование, сжатие аэрокосмических изображений*

Введение

С развитием космического и наземного сегментов систем дистанционного зондирования Земли (ДЗЗ), а также соответствующих информационных технологий передачи, сбора и обработки данных ДЗЗ, одним из основных методов экологического мониторинга становится *аэрокосмический мониторинг ландшафтного покрова* [1]. Традиционно дистанционный аэрокосмический мониторинг направлен на решение целого комплекса задач исследования состояния и динамики экосистем, прогноза протекающих в них деструктивных процессов и характеризуется оперативностью и точностью получаемых данных.

Решению задач в глобальном масштабе мониторинга в последние годы уделяется особое внимание и создан целый ряд успешно действующих систем и программ, обеспечивающих масштабную поддержку решения задач аэрокосмического мониторинга (EOSDIS в США, INFEO в Европе, "Природа" в России и др.).

С ростом доступности данных ДЗЗ различного спектрального и пространственного разрешения и с распространением коммерческого программного обеспечения обработки и интерпретации аэрокосмической информации, позволяющих проводить широкому кругу пользователей автоматизированный анализ явлений и процессов ландшафтного покрова в региональном масштабе, особую значимость в мире получили региональные системы и центры аэрокосмического мониторинга (далее — системы мониторинга), обеспечивающие прием, предварительную обработку, архивирование, каталогизацию аэрокосмических данных и их распространение конечным потребителям [1, 2]. Учитывая обширные территории Российской Федерации, в которой отдельные субъекты по площади порой соответствуют нескольким странам мира, необходимость в развитых региональных и территориальных центрах аэрокосмического мониторинга, ориентированных на региональные научные изыскания с учетом наземных наблюдений и другой доступной местной информации, повышающей адекватность проводимых исследований, очевидна. Однако традиционно вопросам создания наземных сегментов (в отличие от орбитальных) региональных и территориальных систем аэрокосмического мониторинга с центрами приема, обработки и распространения достаточного внимания не уделялось [2].

Типовая региональная система мониторинга

Существующие отдельные примеры построения региональных систем аэрокосмического мониторинга систем в России основаны в различной степени на современных информационных технологиях, типовом и оригинальном математическом и программном обеспечении и характеризуются высокой научно-технической сложностью [2, 3]. Наземный сегмент типовой системы аэрокосми-

ческого мониторинга с помощью соответствующих подсистем, как правило, обеспечивает [2]:

- *сбор заявок и распространение данных ДЗЗ* потребителям;
- *подготовку данных для управления съемкой* с летательного аппарата на основе соответствующих целеуказаний;
- *прием и регистрацию данных ДЗЗ* с заданной территории за сеанс связи;
- *предварительную обработку* с заданным уровнем;
- *архивацию и каталогизацию* принятых и обработанных данных ДЗЗ.

Следует отметить, что функции содержательной обработки данных для решения прикладных задач выполняются до сих пор, главным образом, рядовыми потребителями (после получения необходимых данных ДЗЗ по каналам связи) на собственных ПЭВМ со стандартными вычислительными возможностями.

В условиях значительных объемов данных ДЗЗ с широким спектром характеристик растут требования потребителей к точности, оперативности и глубине содержательной обработки данных, связанной с решением задач, характерных исключительно для интеллектуальных экспертных систем — интерпретации как формирования высокоуровневых выводов из набора данных, прогнозирования как проектирования возможных последствий ситуации, мониторинга как сравнения наблюдаемого и фактического поведения среды и др. [4]. При этом стандартные вычислительные средства пользователя для такой сложной и затратной обработки в значительной степени не пригодны. Хотя региональным системам мониторинга за счет концентрации ресурсов доступны значительные вычислительные мощности, типовых функциональных возможностей систем мониторинга, а также попыток совершенствования этих систем с наделением функциями тематической обработки данных на основе стандартного или оригинального математического и программного обеспечения явно недостаточно для решения современных задач аэрокосмического мониторинга и соответствия современным требованиям потребителей [2, 3].

В связи с этим совершенствование систем мониторинга следует проводить с учетом последних результатов исследований в области методов обработки данных ДЗЗ, модифицируя подсистемы предварительной и основной обработки данных и позволяя эффективно использовать доступные значительные вычислительные мощности. Системное решение этих проблем требует разработки концепции создания систем аэрокосмического мониторинга, учитывающей комплекс современных требований к таким системам [2].

Возможности совершенствования системы мониторинга

Современное состояние исследований в области информационных технологий и разработки математического и программного обеспечения сложной обработки данных ДЗЗ позволяет создавать системы мониторинга нового поколения. Эти системы следует наделить не существовавшими ранее возможностями создания оригинальных информационных продуктов, необходимых для решения современных задач аэрокосмического мониторинга, с учетом доступных вычислительных мощностей, а также обеспечения более развитого доступа к ним потребителей.

Так, сегодня существуют новые возможности сложной *интеллектуальной обработки данных*: автоматизированное дешифрирование мульти- и гиперспектральных аэрокосмических изображений (АИ) с различным пространственным и спектральным разрешением; построение моделей анализа ландшафтного покрова по разновременным данным и их использование при построении прогнозных ландшафтных карт; развитый пространственный анализ, выполняемый с использованием оригинальных и стандартных систем обработки и интерпретации данных ДЗЗ, современных геоинформационных систем (ГИС), реализованных в единой информационной среде.

Кроме сложной содержательной обработки данных по созданию различных информационных продуктов для конечных потребителей, функцией системы мониторинга, также непосредственно ориентированной на пользователя, является *архивирование и каталогизация* данных ДЗЗ и результатов их обработки. Поэтому совершенствование системы мониторинга именно в этой части представляется особенно значимым.

Архивирование, обеспечивающее надежное хранение огромных массивов данных ДЗЗ, было и во многом остается в техническом отношении сложнейшей задачей [2, 3]. В последнее время активно проводятся исследования в области сжатия данных ДЗЗ без потерь и с потерями, существенно повышающие эффективность сжатия, но не получившие пока практического применения в подсистемах архивирования данных ДЗЗ систем мониторинга [5]. Поэтому совершенствование систем мониторинга, основанное на современных подходах к сжатию и направленных на повышение эффективности не только архивного хранения, но передачи данных ДЗЗ по каналам связи, представляется крайне актуальным.

Каталогизация обеспечивает оперативный и дружественный доступ потребителей к архивам данных ДЗЗ и результатам их обработки. В связи с экстремально большим объемом данных для хра-

нения и поиска особое развитие функции каталогизации получили в глобальных системах сбора, архивирования и распространения данных ДЗЗ [2, 3]. Региональные информационные системы оперируют меньшими объемами данных и при этом обеспечивают пользователя функциями более глубокой предварительной оценки практической значимости АИ, реализуемой использованием изображений для предварительного просмотра (англ. *Preview Image*, *Quick Look*). Такие изображения, полученные с помощью методов сжатия с потерями, позволяют пользователю визуально оценить потребность в передаче по каналам связи конкретных оригинальных АИ.

Результаты исследований автора показывают возможность усовершенствования каталогизации за счет дифференцированного метода сжатия АИ с потерями, который позволяет проводить не только визуальную оценку, но и автоматизированную обработку восстановленных АИ с приемлемым уровнем точности [6]. Это позволит пользователю более адекватно оценить необходимость в конкретных АИ и возможности их автоматизированной обработки, избегая передачи по каналам связи громоздких оригинальных АИ, практическая значимость которых невысока.

Системы мониторинга характеризуются наличием регулярно пополняемых богатейших ресурсов данных ДЗЗ и результатов их обработки. Распространение современной дорогостоящей суперкомпьютерной техники, которой сегодня оснащают и центры приема, обработки и распространения данных ДЗЗ, а также наличие большого объема доступных вычислительных ресурсов в виде недорогих ПЭВМ, объединенных в локальных вычислительных сетях, делает возможным построение систем аэрокосмического мониторинга с использованием *высокопроизводительной распределенно-параллельной обработки* данных ДЗЗ. Значительное увеличение производительности обработки больших массивов данных ДЗЗ достигается не только за счет непосредственного использования вычислительно мощной программно-аппаратной среды кластера, но и путем адаптации применяемых методов и алгоритмов под распределенно-параллельные вычисления [5, 7, 9].

Современные возможности удаленного доступа на основе Интернет-технологии позволяют с учетом

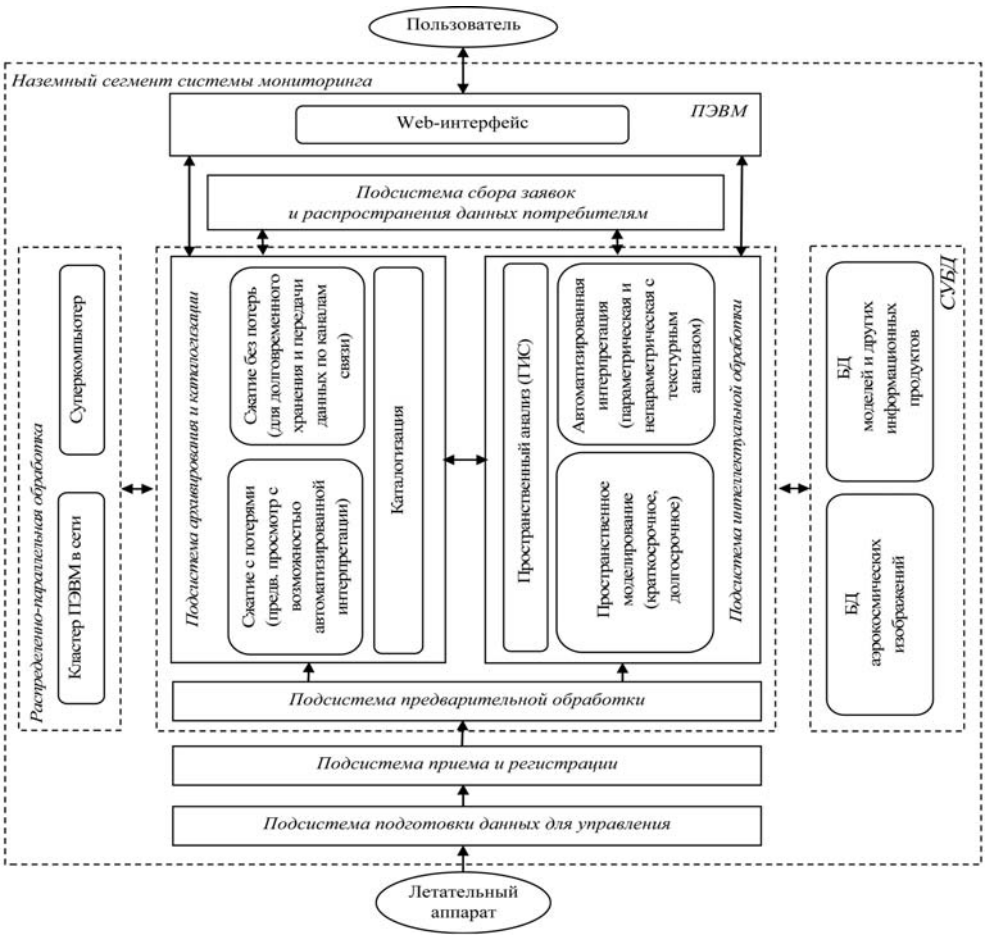

Рис. 1. Обобщенная структура региональной системы аэрокосмического мониторинга с интеллектуальной распределенно-параллельной обработкой данных

изложенных выше усовершенствований организовывать в системе мониторинга *удаленную интерактивную содержательную обработку данных ДЗЗ* и создать на мощных вычислительных ресурсах системы различные информационные продукты, необходимые для решения задач аэрокосмического мониторинга.

Требования к современной системе мониторинга

С учетом отмеченных выше недостатков сформулируем *основные требования* к современной системе аэрокосмического мониторинга:

- интеллектуальная обработка данных, обеспечивающая автоматизированную интерпретацию мульти- и гиперспектральных АИ с различным пространственным и спектральным разрешением, реализующая по данным разновременной съемки возможности краткосрочного и долгосрочного пространственного моделирования и прогнозирования динамики ландшафтного покрова, а также позволяющая выполнять развитый пространственный анализ исследуемой территории; экспертная поддержка принятия решений, позволяющая решать основные задачи аэрокосмического мониторинга специалистам с разным опытом и квалификацией;
- высокопроизводительная обработка данных ДЗЗ, реализуемая как на дорогостоящих суперкомпьютерах, так и на кластерах из недорогих ПЭВМ в локальной сети, с использованием методов и алгоритмов интеллектуальной обработки данных, адаптированных к распределенно-параллельным вычислениям;
- сжатие без потерь в целях повышения эффективности долговременного хранения и передачи данных по каналам связи, учитывающее специфику многоканальных данных ДЗЗ и обеспечивающее более высокие показатели степени сжатия, чем распространенные универсальные алгоритмы сжатия без потерь; сжатие данных ДЗЗ с потерями, позволяющее обеспечить не только высокие коэффициенты сжатия и возможности визуального оценивания восстановленных АИ, но и их автоматизированную интерпретацию с приемлемым уровнем точности, направленные на совершенствование архивации, каталогизации и распространения данных ДЗЗ потребителям;
- единая интегрированная информационная среда средств обработки данных с возможностями удаленного доступа и масштабируемости.

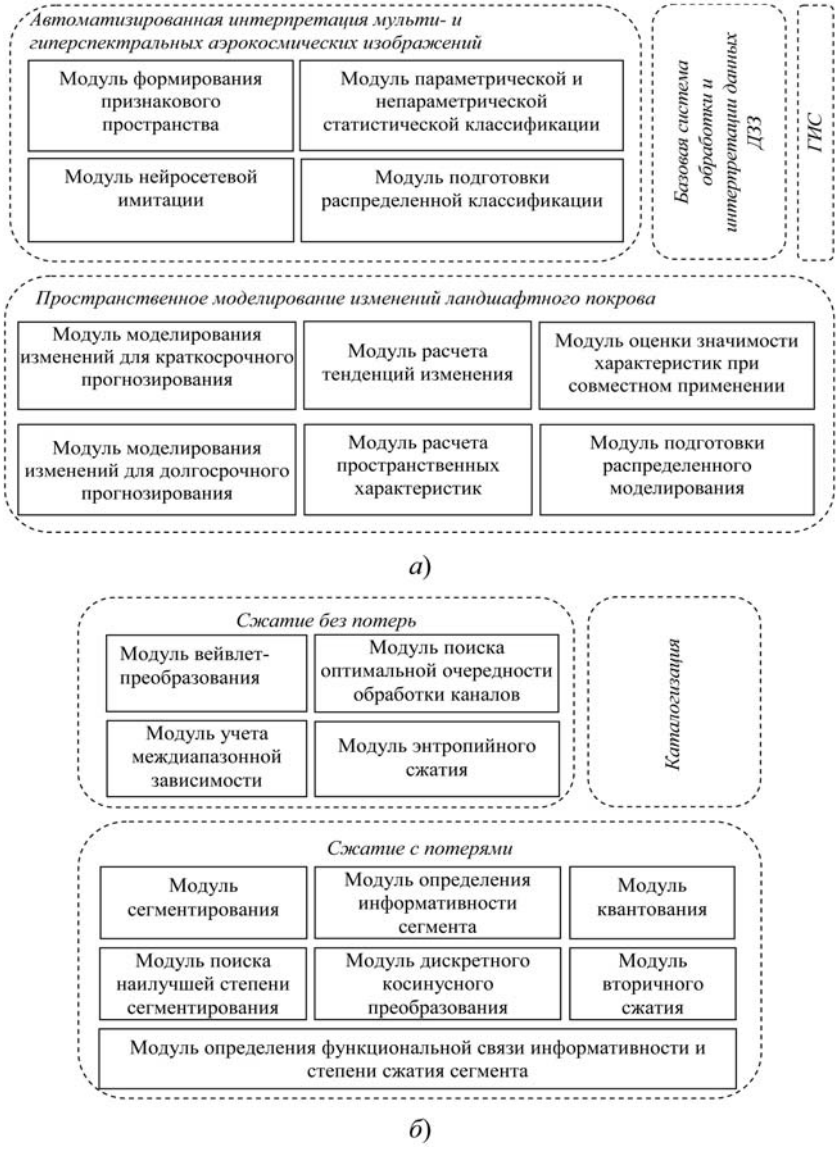

Рис. 2. Модульная структура подсистем основной обработки данных:
а — подсистема интеллектуальной обработки данных; *б* — подсистема архивирования и каталогизации

С учетом отмеченных выше недостатков типовых систем мониторинга и изложенных требований к современным системам предлагается структура региональной информационной системы аэрокосмического мониторинга с интеллектуальной распределенно-параллельной обработкой данных (рис. 1). Детализированная модульная структура подсистем основной обработки данных, модифицированных с учетом оригинальных результатов исследований, выполненных при непосредственном участии автора, представлена на рис. 2.

Технология практического применения системы

Система мониторинга, структура которой приведена на рис. 1 и 2, предполагает следующую обобщенную технологию применения. Летательный аэрокосмический аппарат, управление которым осуществляет *Подсистема подготовки данных для управления*, через *Подсистему приема и регистрации* передает за сеанс связи данные ДЗЗ, которые подвергаются обработке заданного уровня *Подсистемой предварительной обработки*.

Затем полученные данные передаются в *Подсистему архивирования и каталогизации*, которая реализует хранение данных в соответствующих базах данных под контролем системы управления базами данных (СУБД), а также обеспечивает структурированный доступ к данным ДЗЗ и продуктам их обработки. В этой подсистеме процедура архивирования предполагает режимы сжатия данных ДЗЗ без потерь и с потерями.

Сжатие без потерь реализуется на основе оригинального многоэтапного подхода, предложенного автором, увеличивающего коэффициент сжатия данных за счет учета междиапазонной зависимости и поиска очередности обработки каналов АИ [5]. Сжатие с потерями реализует предложенный автором подход к дифференцированному сжатию сегментированных АИ с оценкой информативности сегментов, обеспечивающий высокие коэффициенты сжатия и возможность автоматизированной интерпретации восстановленных АИ с приемлемым уровнем точности [6].

Такой подход позволяет не только существенно повысить эффективность использования доступных каналов связи при передаче данных ДЗЗ за счет более высоких коэффициентов сжатия, но и применить восстановленные после сжатия АИ не только в качестве изображений для предварительного просмотра, но и для автоматизированной интерпретации и построения предварительных тематических карт. Такие карты позволяют пользователям оценить необходимость и достаточность имеющихся в архиве данных ДЗЗ не только по восстановленным АИ, но и по результатам их автоматизированной обработки.

Из *Подсистемы архивирования и каталогизации* данные могут быть запрошены пользователем через *Подсистему сбора заявок и распространения данных потребителям* и получены для самостоятельной обработки. Также возможен вариант, при котором данные из *Подсистемы архивирования и каталогизации* поступают в *Подсистему интеллектуальной обработки данных*. Эта подсистема предназначена для получения тематических и прогнозных карт при решении задач автоматизированной интерпретации, краткосрочного и долгосрочного прогнозирования, проведения сопутствующего пространственного анализа, создания моделей специфических процессов или явлений, а также построения иных информационных продуктов, требуемых при решении прикладных задач аэрокосмического мониторинга.

Построение тематических карт по данным мульти- и гиперспектральной съемки возможно как базовыми средствами обработки и интерпретации данных ДЗЗ, так и на основе оригинального подхода с использованием текстурного анализа, параметрических и непараметрических (в том числе нейросетевых) алгоритмов классификации [5]. Получение ландшафтных прогнозных карт возможно с учетом особенностей краткосрочного и долгосрочного прогнозирования, с использованием различных пространственных характеристик и анализом их значимости при совместном применении, направленных на повышение адекватности моделирования в условиях сложной ландшафтно-классовой структуры [5, 8].

Возможность размещения в базе данных моделей и прочих информационных продуктов, направленных на решение задач аэрокосмического мониторинга, позволяет избежать их повторного создания. Совместно с изложенными выше автоматизированными процедурами обработки данных это способствует использованию накопленного ранее при построении моделей экспертного опыта и упрощенному практическому применению моделей специалистами с небольшой квалификацией, что дополняет возможности интеллектуальной обработки данных в системе.

В условиях существенных объемов данных ДЗЗ и их постоянного роста предварительная (подсистема предварительной обработки) и основная (подсистема интеллектуальной обработки данных и подсистема архивирования и каталогизации) обработка данных ДЗЗ требуют значительного повышения производительности. Оно реализуется с использованием дорогостоящей суперкомпьютерной техники, однако, как правило, без учета особенностей распределенно-параллельных вычислений и соответствующей адаптации алгоритмов обработки данных [2]. Это существенно снижает потенциальные возможности суперкомпьютерной техники и алгоритмов обработки данных, поэтому излагаемая технология предполагает использование таких алгоритмов, которые учитывают особенности распределенно-параллельных вычислений.

В части предварительной обработки данных различными научно-исследовательскими коллективами получен ряд значимых результатов исследований. Наиболее интересные результаты, которые с успехом могут быть практически применены в соответствующей подсистеме предварительной обработки, изложены в работе [7].

Распределенно-параллельный вариант автоматизированной интерпретации реализуется с помощью оригинальной технологии распределенной классификации многоканальных данных ДЗЗ, учитывающей особенности классификаторов с линейным разделением и оценкой условной плотности распределения и применимой для мульти- и гиперспектральных АИ [5].

Моделирование изменений ландшафтного покрова и построение прогнозных тематических карт реализуется на основе оригинального стохастического подхода, основанного на аппарате марковских цепей, клеточных автоматах с вероятностным определением правил функционирования, учитывающего особенности краткосрочного и долгосрочного прогнозирования [8]. Повышение адекватности и точности моделирования изменений ландшафтного покрова достигается применением набора пространственных характеристик с оценкой значимости их сочетаний при совместном применении [5]. Для значительного снижения вычислительных затрат алгоритма моделирования изменений ландшафтного покрова предложен оригинальный подход к распределенно-параллельной обработке, предварительные результаты исследований которого приведены в работе [9]. Особенностью подхода является изменение логики последовательного исполнения алгоритма моделирования, ведущее к некоторому отклонению в точности построения прогнозных карт.

Заключение

С учетом развития современных систем и программ глобального аэрокосмического мониторинга показано, что решение современных задач аэрокосмического мониторинга требует создания соответствующих региональных систем и центров, вопросам построения которых достаточного внимания не уделялось.

Приведены базовые функции наземного сегмента типовой региональной системы аэрокосмического мониторинга. Отмечено, что в соответствии с современными характеристиками данных дистанционного зондирования Земли и требованиями потребителей, типовых функциональных возможностей систем недостаточно и требуется их совершенствование. При этом отмечается, что пока нет концепции создания систем аэрокосмического мониторинга, которая учитывала бы комплекс современных требований к таким системам, наличие высокопроизводительной компьютерной техники, а также актуальное состояние исследований в области обработки данных для задач аэрокосмического мониторинга.

Показано, что основным потенциалом совершенствования систем аэрокосмического мониторинга являются применение более сложной интеллектуальной обработки данных при решении основных задач аэрокосмического мониторинга, использование сжатия аэрокосмических изображений с потерями и без потерь в процедурах архивирования, каталогизации и передачи данных по каналам связи, допускающего автоматизированную обработку с приемлемой потерей точности. Кроме того, для увеличения производительности обработки значительных объемов аэрокосмических данных существует необходимость не только в непосредственном использовании высокопроизводительной программно-аппаратной среды, но и в адаптации разработанных методов и алгоритмов обработки к распределенно-параллельным вычислениям.

С учетом изложенных возможностей совершенствования сформулированы основные требования к современной региональной системе аэрокосмического мониторинга с интеллектуальной распределенно-параллельной обработкой данных, предложена ее обобщенная структура, модульная структура подсистем основной обработки данных, а также технология практического использования системы.

Список литературы

1. **Книжников Ю. Ф., Кравцова В. И., Тутубалина О. В.** Аэрокосмические методы географических исследований. М.: Academia, 2004. 332 с.

2. **Копылов В. Н.** Основы создания центра космического мониторинга окружающей среды. Екатеринбург: Контур, 2006. 144 с.

3. **Лупян Е. А., Мазуров А. А., Назиров Р. Р.** и др. Универсальная технология построения систем хранения спутниковых данных. М.: Препринт ИКИ РАН, Пр. 2024, 2000. 22 с.

4. **Люггер Д. Ф.** Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е изд.: пер. с англ. М.: Вильямс, 2003. 864 с.

5. **Zamyatin A.** Advanced Processing of Remote Sensing Data for Land Use and Land Cover. Saarbrücken: LAP Lambert Academic Publishing AG & Co. KG, 2010. 232 p.

6. **Замятин А. В.** Дифференцированное сжатие аэрокосмических изображений с потерями // Информационные технологии. 2011. № 6. С. 60—65.

7. **Бучнев А. А., Пяткин В. П., Русин Е. В.** Распределенная высокопроизводительная обработка данных дистанционного зондирования Земли // Исследование Земли из космоса. 2007. № 4. С. 34—38.

8. **Замятин А. В.** Стохастический алгоритм моделирования для задач долгосрочного прогнозирования изменений ландшафтного покрова // Математическое моделирование. 2010. Т. 22, № 11.

9. **Афанасьев А. А., Замятин А.** В. Распределенные вычисления в задаче моделирования изменений ландшафтного покрова // Сб. трудов VIII Всероссийской научно-практической конференции "Молодежь и современные информационные технологии", Томск. Ч. 1. 2010. С. 77—78.

УДК 519.857

В. И. Струченков, д-р техн. наук, проф.,
e-mail: str1942@mail.ru,
А. Н. Козлов, аспирант,
А. С. Егунов, аспирант,
Московский государственный институт радиотехники, электроники и автоматики (технический университет)

Кусочно-параболическая аппроксимация плоских кривых при наличии ограничений специального вида[1]

При автоматизированном проектировании трасс линейных сооружений возникают задачи аппроксимации плоских кривых, заданных дискретно, последовательностью элементов определенного вида (отрезки прямых, дуги окружностей или парабол второй степени, а также клотоид) при наличии ограничений на параметры элементов. Число элементов аппроксимирующей кривой неизвестно. Рассматривается задача поэлементной аппроксимации, в которой элементами являются отрезки парабол второй степени (как частный случай — отрезки прямых), при наличии ряда ограничений. Задача решается с помощью динамического программирования.

Ключевые слова: *аппроксимация, ограничения, динамическое программирование*

Постановка задачи

Исходная плоская кривая задана последовательностью точек (не обязательно с равным шагом). Требуется аппроксимировать ее гладкой кривой, состоящей из прямолинейных и параболических (второй степени) элементов, на которые наложены линейные ограничения специального вида. Число элементов искомой кривой неизвестно и должно быть установлено в процессе решения задачи при выполнении всех ограничений и минимальном отклонении от исходной кривой. В качестве показателя качества аппроксимации может рассматриваться максимальное по абсолютной величине отклонение от исходной кривой или интеграл от квадрата разности исходной и аппроксимирующей функций.

Должны быть выполнены следующие ограничения:

- на отклонения искомой кривой от исходной в заданных точках;
- на первую производную искомой кривой во всех точках;
- на кривизну искомой кривой во всех точках;
- длины элементов не должны быть меньше заданных величин. В практических задачах это ограничение будет рассматриваться как ограничение на разность абсцисс конца и начала элемента;
- считаются заданным начальная и конечная точка искомой кривой, а также начальное и конечное направления;
- исходная и аппроксимирующая кривая являются графиками однозначных функций.

На границах элементы имеют общую касательную.

Отдельно рассматривается вариант, при котором угол между касательными на границах элементов не превосходит заданной величины. В этом случае допускается разрыв первой производной аппроксимирующей функции на стыках элементов, но ограничение второго типа сохраняется в том смысле, что ограничены значения производной слева и справа в точке ее разрыва.

Практический смысл задачи. Задачи поэлементной аппроксимации с ограничениями возникают при автоматизации проектирования трасс линейных сооружений (железные и автомобильные дороги, трубопроводы различного назначения, каналы и др.). Поскольку *число элементов искомой кривой неизвестно*, приходится решать проектную задачу в три этапа.

1. Ограничение на минимальную длину элемента игнорируется, проектная линия рассматривается в виде ломаной (как и исходная кривая с *заданным числом элементов*), все остальные ограничения заменяются их дискретными аналогами. Решается задача оптимизации по принятому критерию (например, минимум затрат на строительство сооружения).

2. Результат первого этапа (ломаная линия) преобразуется в последовательность элементов заданного вида с соблюдением всех ограничений, включая ограничение на минимальную длину элемента. Именно этот этап является предметом данной статьи. На этом этапе отклонения из-за аппроксимации невелики, примерно на порядок меньше, чем на первом этапе, так что вместо исходного критерия оптимизации рассматривается квадратичный критерий близости кривых (или минимум максимального отклонения). Результат аппроксимации является начальным приближением для следующего этапа. Для различных проектных задач элементами искомой линии могут

[1] Данная статья является продолжением статьи "Кусочно-линейная аппроксимация плоских кривых при наличии ограничений". См. журнал "Информационные технологии", № 12, 2010. С. 32—34.

быть отрезки прямых, парабол второй степени, окружностей и клотоид. [4]

3. *При известном числе элементов* и полученном начальном приближении решается задача оптимизации по исходному критерию.

На первом и третьем этапе в действующих САПР используются алгоритмы нелинейного программирования [1, 4], для которых число переменных (размерность задачи) должно быть задано.

Для задач проектирования трасс линейных сооружений не требуется высокая точность аппроксимации, так как нужно установить число элементов и их примерное положение, чтобы на третьем этапе получить окончательное решение.

Применение динамического программирования

Сначала рассмотрим простой случай, когда известно и *число элементов, и абсциссы их концов* и требуется найти параметры элементов. В области поиска разобьем сетку варьирования. Для этого на границах элементов относительно исходной ломаной вверх и вниз отложим заданное число дискретов (рис. 1). Величина дискрета задается исходя из требуемой точности решения задачи и вычислительных возможностей.

На первом шаге (рис. 2) в каждую точку вертикали 1 приходит только одна парабола, так как начальная точка A и начальное направление заданы. Уравнение параболы

$$y = ax^2 + bx + c.$$

Если задать ординату конца параболы y_C, то для определения параметров a, b и c имеем систему трех уравнений:

$$ax_A^2 + bx_A + c = y_A;$$

$$ax_C^2 + bx_C + c = y_C;$$

$$2ax_A + b = i_{\text{нач}}.$$

Здесь начальный уклон $i_{\text{нач}} = \text{tg}\alpha$, где α — угол заданного начального направления с осью X.

Находим последовательно

$$a = \frac{y_C - y_A - i_{\text{нач}}(x_C - x_A)}{(x_C - x_A)^2}; \qquad (1)$$

$$b = i_{\text{нач}} - 2ax_A \text{ и } c = y_A - ax_A^2 - bx_A.$$

Уклон (производная) в конце элемента

$$i_{\text{кон}} = 2(y_C - y_A)/(x_C - x_A) - i_{\text{нач}}. \qquad (2)$$

Если $(y_C - y_A)/(x_C - x_A) = i_{\text{нач}}$, то вместо параболы имеем прямую ($i_{\text{кон}} = i_{\text{нач}}$) как частный случай. При различных значениях y_C получаем различные очертания элемента (выпуклое, вогнутое, с вершиной внутри или вне элемента). Отметим, что при изменении y_C на Δ (шаг сетки) конечный уклон меняется на $2\Delta/(x_C - x_A)$. Естественно, что из всех парабол остаются только те, параметры которых удовлетворяют всем ограничениям, и для каждой допустимой параболы вычисляется оценка критерия (показателя качества аппроксимации). Ограничения на кривизну фактически сводятся к двусторонним ограничениям на параметр a, если уклоны много меньше единицы, что имеет место в задачах проектирования продольного профиля трасс линейных сооружений. В любом случае для каждого построенного варианта параболы легко проверить выполнение всех заданных ограничений.

Можно было бы не перебирать варианты y_C, а вычислить параметры a, b, c из условия минимума интеграла от квадрата разности исходной и аппроксимирующей кривой на рассматриваемом интервале. Если при этом нарушаются ограничения, то следует скорректировать полученное значение и вычислить соответствующее ему y_C. Однако все равно придется разбивать сетку относительно полученной точки, которая близка к точке на исходной кривой с той же абсциссой, поэтому в использовании наилучшего (*локально!*) приближения особого смысла нет.

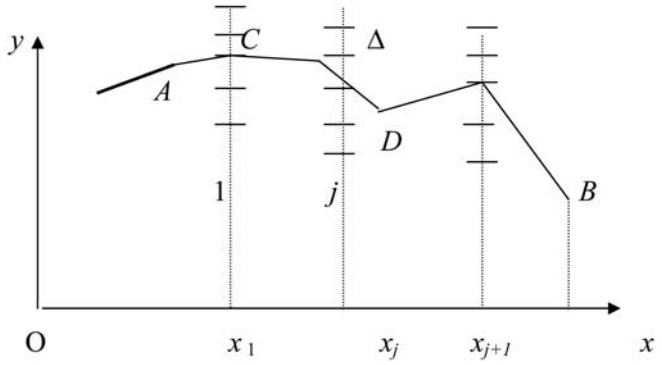

Рис. 1. Сетка варьирования. *ACDB*-исходная линия

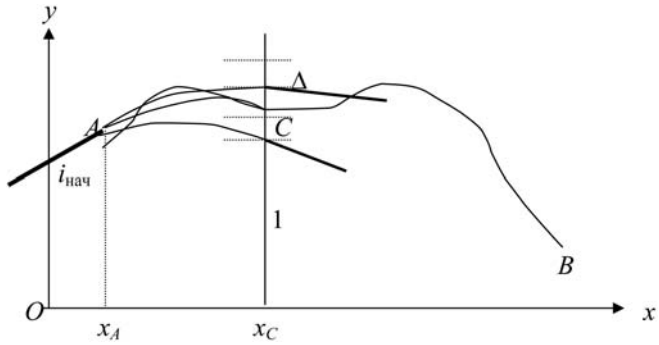

Рис. 2. Варианты первой параболы

При построении второго и всех последующих элементов каждая точка на вертикали в конце последнего из построенных элементов должна рассматриваться как начальная. На первом элементе это была точка A с одним возможным значением начального уклона, на вертикали 1 таких точек несколько, но в каждой из них есть только одно значение начального уклона; на последующих вертикалях в каждой из точек есть несколько значений начального уклона. Если число точек на каждой вертикали равно m, то без учета ограничений, на второй вертикали в *каждой точке* оказываются m сходящихся элементов (каждый со своим конечным уклоном), на третьей соответственно m^2, потом m^3 и т. д.

В качестве "состояния системы" нельзя принять отдельную точку на вертикали, нужно еще принять и значение уклона (производной) в ней. Сравнивать можно только варианты, сходящиеся в точке с одним и тем же уклоном, и из всех таких вариантов оставлять для дальнейшего рассмотрения только один. При заданной длине, точнее разности абсцисс концов элементов L, такие варианты, как следует из формулы для конечного уклона (2), будут иметь место, если

$$2(y_{\text{н}}^2 - y_{\text{н}}^1)/L = i_{\text{н}}^1 - i_{\text{н}}^2.$$

Здесь $y_{\text{н}}^1$, $y_{\text{н}}^2$, $i_{\text{н}}^1$, $i_{\text{н}}^2$ — соответственно ордината и начальный уклон для двух вариантов элемента длиной L, сходящихся в одной точке. При таком правиле отбраковки резкое возрастание числа вариантов приводит к непреодолимым вычислительным сложностям даже при заданных длинах элементов.

Вместо этого будем считать сравнимыми сходящиеся в одной точке варианты, если их конечные уклоны близки, т. е. введем дискретность по уклонам. Это означает, что задача становится двухпараметрической, но по второму параметру (уклону) сетка варьирования строится в процессе счета.

На втором и всех последующих шагах поступим следующим образом (рис. 3).

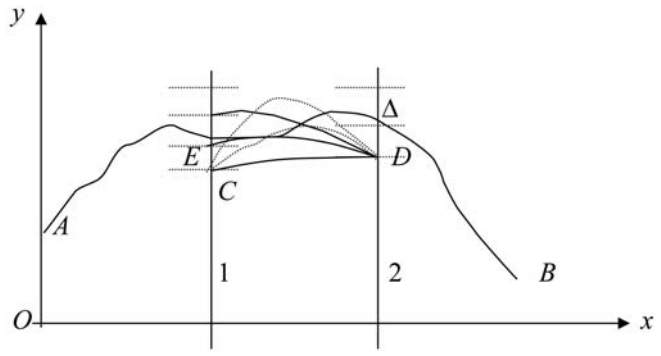

Рис. 3. Варианты очередной параболы

1. Берем точку уже рассмотренной вертикали (на втором шаге это вертикаль 1) с наименьшей ординатой (точка C) и строим допустимые параболы в точку с наименьшей ординатой следующей вертикали (точка D). Все *допустимые* по ограничениям варианты оставляем для дальнейшего анализа и для каждого из них вычисляем величину критерия *суммарно* от начала кривой (точка A) до точки D.

2. Берем следующую точку на левой вертикали (точку E) и проводим из этой точки параболу в точку D с использованием одного из имеющихся начальных уклонов. Но при этом используем следующее правило отбраковки вариантов: если конечные уклоны парабол, сходящихся в одной точке (D), по абсолютной величине различаются незначительно ($|i_{\text{кон}}^2 - i_{\text{кон}}^1| < \varepsilon$), то остается вариант с меньшей величиной критерия.

Величина ε играет роль дискрета по уклонам и должна выбираться так, чтобы вызванное этой дискретностью *максимальное отклонение на следующем элементе* не превышало шага сетки Δ. Речь идет о том, что при подборе очередной параболы мы хотим знать, к чему приведет ошибка в задании начального уклона $i_{\text{нач}}$, который используется для расчета параметров a, b и c. Изменение $i_{\text{нач}}$ вызывает изменение этих параметров и, как следствие, получим другую параболу, но с теми же начальной и конечной точками. Разность ординат двух таких парабол достигает экстремального значения в средней точке.

Найдем максимальное отклонение в пределах элемента между двумя параболами, начальные уклоны которых отличаются на ε. Очевидно, это отклонение не изменится, если начало координат поместить в начальную точку элемента. При этом существенно упрощается расчет параметров. Итак, $x_A = 0$; $y_A = 0$; $b = i_{\text{нач}}$; $c = 0$; $x_C = L$; y_C задано. Для неизвестного параметра a имеем $y_C = aL^2 + i_{\text{нач}}L$. Отсюда следует, что $\delta aL = -\delta i_{\text{нач}}$, так как y_C не изменяется при изменении $i_{\text{нач}}$. Здесь $\delta i_{\text{нач}}$ — изменение начального уклона, а δa — вызванное им изменение параметра a. В средней точке элемента разность ординат двух парабол определяется выражением

$$\delta aL^2/4 + \delta i_{\text{нач}}L/2 = \delta i_{\text{нач}}L/4.$$

Это отклонение не должно превышать шага сетки Δ, поэтому $\delta i_{\text{нач}} < 4\Delta/L$. Дискрет по уклонам ε возьмем вдвое меньше $\varepsilon = 2\Delta/L_{\min}$, где $L_{\min}$ — минимальная длина элемента, так как в дальнейшем будут рассматриваться элементы различной длины, но рассмотрение элементов, длина которых больше $2L_{\min}$ особого смысла не имеет, по-

скольку каждый такой элемент может быть построен как совокупность двух элементов.

3. Поочередно рассматриваем все соединения точек левой вертикали с точкой *D* правой вертикали, из допустимых оставляем параболы по сформулированному правилу отбраковки. Запоминаем в точке *D* конечные уклоны, для каждого из них суммарную величину критерия и соответствующую точку левой вертикали.

4. Переходим к следующей точке правой вертикали и осуществляем те же действия.

В итоге на правой вертикали в каждой точке останется "веер" уклонов, которые на следующем шаге рассматриваются как начальные. Естественно, что на последнем шаге есть только одна точка на правой вертикали, это конечная точка *B*. Для нее определяется точка на предыдущей вертикали. А для этой точки запомнили уклон и точку на предыдущей вертикали, что и дает возможность восстановить всю линию обратным разворотом от точки *B*.

Если в точке *B* задано конечное направление, то это приходится учитывать при расчете возможных вариантов последней параболы.

Отметим, что в отличие от кусочно-линейной аппроксимации, кроме дискретности по ординатам, для кусочно-параболической аппроксимации нам пришлось вводить допуск (фактически дискрет) по уклонам.

Рассмотрим теперь поиск при неизвестном числе элементов. Кроме дискретности по вертикали Δ и по уклонам ε введем дискретность по длине элемента λ. Ее наличие характерно для проектных задач и осложняет использование методов нелинейного программирования. В данном случае, наоборот, эта дискретность упрощает задачу.

Возможная длина первого элемента принимает значения в интервале от $L_{\min}$ до $2L_{\min}$ с шагом λ. Это означает, что начальная точка *A* (см. рис. 2) может быть соединена с точками не на одной, а на нескольких вертикалях (рис. 4).

Каждая из возможных точек на этих вертикалях рассматривается как начало второго элемента. "Состояние системы" — по-прежнему точка и уклон, но число вертикалей ("шагов процесса") много больше, чем при известных длинах элементов. Более того, для запоминания связи с предыдущим состоянием надо запоминать и точку начала элемента, и номер вертикали, которой она принадлежит. Шаг поиска — это не очередной элемент, а очередная вертикаль, так как одна и та же точка на вертикали может являться концом траектории, состоящей из разного числа элементов. По-прежнему в каждом состоянии запоминается величина критерия. Сравнимыми считаются допустимые варианты, имеющие общую конечную точку и близкие значения уклонов в ней. При сравнении худший вариант отбраковывается, и все его продолжения не рассматриваются. Задача остается двухпараметрической, но число вариантов существенно больше, чем при заданных длинах элементов.

Последняя вертикаль отстоит от конечной точки *B* аппроксимируемой кривой на $L_{\min}$. После того как эта вертикаль достигнута, рассматриваются все допустимые соединения точки *B* с точками вертикалей, отстоящих от точки *B* на расстоянии не более чем на $2L_{\min}$, т. е. решается задача построения последнего элемента (рис. 5).

Последний элемент выбирается среди всех допустимых соединений с точкой *B*, так чтобы была минимальной величина критерия суммарно на всей траектории *AB*. Для этого на каждом допустимом варианте последнего элемента, включая элементы, имеющие общее начало, но разный уклон, вычисляется величина критерия, суммируется с записанной величиной критерия для начала элемента и путем сравнения сумм определяется наилучший вариант последнего элемента. Для этого элемента известна начальная точка (ордината), номер вертикали (абсцисса) и конечный уклон. По этим данным вычисляется начальный уклон (формула (2) применяется в "обратном направлении"), он же конечный уклон предпоследнего элемента и т. д. Отметим, что в итоге будет определено и число элементов, и все их параметры.

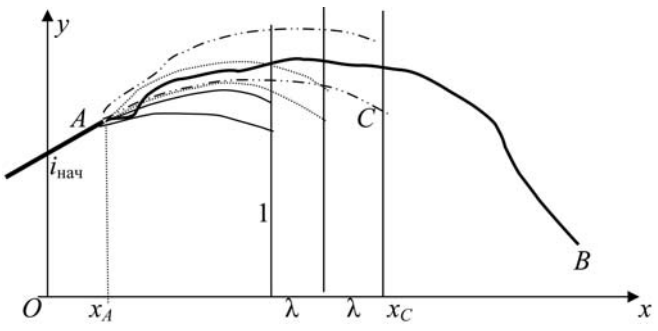

Рис. 4. Варианты первого элемента неизвестной длины

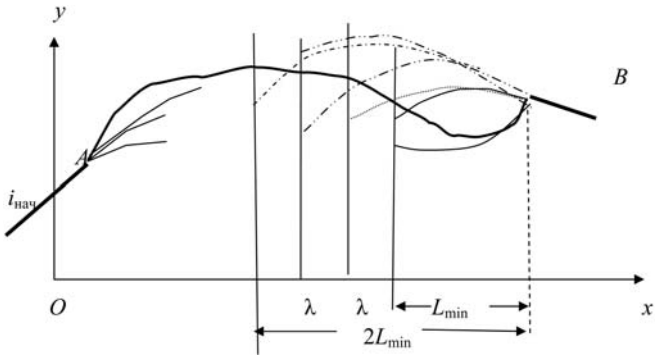

Рис. 5. Варианты последнего элемента неизвестной длины

Задача кусочно-параболической аппроксимации может ставиться и при дополнительном условии: при сопряжении элементов допускается отсутствие общей касательной, но образующаяся разность уклонов на стыках элементов по абсолютной величине не должна превышать заданной величины ε_1. Такая задача возникает, например, при проектировании продольного профиля автомобильных дорог низких категорий.

Рассмотренная выше задача кусочно-линейной аппроксимации является частным случаем при $\varepsilon_1 = 0$.

Оказывается, что это дополнительное условие не влечет существенных усложнений алгоритмов динамического программирования как при известном числе элементов аппроксимирующей кривой, так и в общем случае, когда неизвестны не только абсциссы границ элементов, но и их число.

Действительно, при построении элементов, начиная с первого, приходится рассматривать большее число вариантов. Так, при $\varepsilon_1 = 0$ из начальной точки A в каждую точку вертикали 1 (см. рис. 2) можно было построить только одну параболу при заданном начальном уклоне i_A. Теперь приходится рассматривать несколько значений начального уклона $i_{\text{нач}}$ в интервале $i_A - \varepsilon_1 \leqslant i_{\text{нач}} \leqslant i_A + \varepsilon_1$ с заданным шагом ε ($\varepsilon < \varepsilon_1$ используется и для отбраковки вариантов, сходящихся в одной точке). Число вариантов конечного уклона в каждой точке возрастает, но оно и ранее уже за несколько шагов достигало своего максимума, определяемого как $(i_{\max} - i_{\min})/\varepsilon$. Фактически это число в несколько раз меньше, так как в реальных задачах при большом различии уклонов соответствующие варианты просто не вписываются в полосу допустимых отклонений от исходной ломаной.

Состояние системы, как и при $\varepsilon_1 = 0$, определяется точкой на вертикали и конечным уклоном. Отбраковка вариантов проводится по тому же правилу с использованием ε. Запоминаются для каждого состояния те же данные: параметры (точка и уклон), величина критерия, номер точки на левой вертикали и номер вертикали начала элемента, если границы элементов неизвестны. Отличия возникают при восстановлении линии при обратном развороте, так как теперь конечный уклон элемента не обязательно совпадает с начальным уклоном следующего элемента. Придя в конечную точку B и выбрав наилучший вариант, вычисляем начальный уклон последнего элемента i_n, используя координаты точки B, уклон в точке B и координаты начала последнего элемента. Но в конце предпоследнего элемента могло и не быть такого уклона i_n среди всех, которые запомнили, так как при построении вариантов последнего элемента допускались и другие значения начальных уклонов в пределах допустимого перелома касательной ε_1. Но для каждого состояния хранится оценка критерия по наилучшему пути, что позволяет в допустимых пределах $i_n - \varepsilon_1 \leqslant i \leqslant i_n + \varepsilon_1$ выбрать наилучший конечный уклон предпоследнего элемента. Как и в точке B, получаем для новой точки конечный уклон, что и позволяет двигаться дальше к началу линии вплоть до точки A, где выбора уже нет.

Программная реализация

Описанный алгоритм проверялся на задаче проектирования продольного профиля автомобильных дорог различных категорий. Установлено, что из-за наличия ограничений число уклонов в "веере" растет не очень резко, так что решение задачи вполне реально даже при $\varepsilon = 0{,}0001(i_{\max} - i_{\min})$ и числе точек на вертикали 50 и более. Здесь $i_{\max}$, $i_{\min}$ — соответственно максимальный и минимальный уклоны, определяемые ограничениями на первую производную.

Поскольку целью аппроксимации является определение числа элементов и приближенных значений их параметров, т. е. получение начального приближения для программы оптимизации по методу нелинейного программирования [4], нет смысла в использовании малых величин дискретов. Поэтому программа позволяет решать реальные проектные задачи при шаге поиска по вертикали 0,02 м в области отклонений 0,5 м в обе стороны, шаге поиска по длине 10 м и длине участка 10 км.

Это позволило использовать в САПР новую программу вместо программы расчета элементов по эвристическому алгоритму последовательной локальной аппроксимации с возвратами.

Описанный алгоритм может быть обобщен на случай поиска аппроксимирующей кривой, состоящей из других элементов, например, отрезков прямых и окружностей для проектирования плана и продольного профиля дорог, трубопроводов и других сооружений. Соответствующие программы находятся в стадии разработки. Кроме того, алгоритм может иметь и другие приложения, далекие от проектирования трасс линейных сооружений.

Список литературы

1. **Аоки М.** Введение в методы оптимизации: Пер. с англ. М.: Наука, 1977.

2. **Беллман Р.** Динамическое программирование. М.: ИЛ, 1960.

3. **Беллман Р., Дрейфус С.** Прикладные задачи динамического программирования. М.: Наука, 1964.

4. **Струченков В. И.** Методы оптимизации в прикладных задачах. М.: Солон-Пресс, 2009.

УДК 621.382.26

И. Ю. Жуков, д-р техн. наук, доц.,
первый зам. директора — генеральный конструктор,
ОАО "Всероссийский научно-исследовательский
институт автоматизации управления
в непромышленной сфере им. В. В. Соломатина",
Д. М. Михайлов, аспирант, ассистент,
e-mail: mdmitry@bk.ru,
А. В. Стариковский, аспирант, ассистент,
Национальный исследовательский
ядерный университет "МИФИ"

Усовершенствованный протокол аутентификации бюджетных RFID-меток

Рассматривается усовершенствованный протокол аутентификации RFID-меток (Radio Frequency Identification), в которых отсутствуют ресурсы, необходимые для мощных криптографических преобразований. Подобные метки являются удобным и очень дешевым средством мониторинга, и поэтому получили повсеместное применение. В связи с этим их удобно называть бюджетными RFID-метками. Они используются в логистике при транспортировке грузов, в магазинах для защиты товаров от кражи и т. д. Универсальное применение RFID-меток привлекает внимание злоумышленников в целях промышленного шпионажа, вторжения в частную жизнь, хищения собственности пользователей, что создает новые угрозы безопасности и конфиденциальности информации.

В целях обеспечения безопасности использования подобных RFID-систем приводится базовое описание RFID-модели и предлагается решение на основе алгоритма RSA для обеспечения защищенного процесса аутентификации при обмене информацией между RF-сканером и метками.

Ключевые слова: *протокол аутентификации, RFID-метки, криптография, безопасность, защита данных*

1. Проблема аутентификации RFID-меток

Современный мир товарооборота невозможно представить без применения RFID-технологий. Дешевые, так называемые "бюджетные", радиочастотные метки с низкими функциональными возможностями оказались очень удобным механизмом логистики и мониторинга и поэтому получили необыкновенно широкое распространение. Универсальное применение RFID-меток привлекает внимание злоумышленников в целях промышленного шпионажа, вторжения в частную жизнь, хищения собственности пользователей, что создает новые угрозы безопасности и конфиденциальности информации.

Первые упоминания о необходимости обеспечения безопасности в RFID-системах встречаются в работе [1]. Авторы разработали новые облегченные криптографические протоколы для бюджетных RFID-меток. Они создали механизм защиты от угрозы клонирования меток [1], благодаря которой злоумышленник может подменять товар в магазине меткой-клоном. При этом RFID-система, установленная в магазине, будет считать, что товар по-прежнему находится на полке.

Авторы работы [2] использовали в смарт-картах и сетевых датчиках облегченные криптосистемы с открытым ключом, получившие название NTRU. В то же время, авторы работы [3] предложили использовать электронную цифровую подпись. Несмотря на то, что оба эти варианта по сравнению с ранее известными криптосистемами ведут к очень эффективным механизмам на основе открытых ключей и цифровых подписей, они все равно требуют гораздо больший объем для ресурсов, чем он доступен на бюджетных RFID-метках.

Различные схемы контроля доступа к RFID-меткам предлагают авторы работы [4]. Метка может находиться в одном из двух режимов. В закрытом режиме метка отвечает лишь на запросы, имеющие определенный мета-идентификатор. В открытом режиме она может выполнять операции, связанные с безопасностью и настройками. Цель этой схемы гарантировать, что метка переходит в открытый режим, только если получает соответствующую команду от сопряженного RF-сканера. Отсюда можно сделать вывод, что предложенные протоколы [4] подходят по большей части для аутентификации RF-сканера. К тому же они используют стандартные криптографические хэш-функции и требуют установки генератора псевдослучайных чисел на метки, что также недостижимо в связи с ограничениями ресурсов бюджетных RFID-меток.

2. Условия и ограничения применения бюджетных RFID-меток

В магазине RF-сканер периодически опрашивает метки, прикрепленные к товарам, тем самым осуществляя их идентификацию. В подобной системе необходимо обеспечить защиту от кражи то-

вара путем установки клона метки, при которой RF-сканер будет продолжать опрашивать метки и получать правильный ответ, тем самым не замечая подмены. В связи с этим предлагается усовершенствованный протокол аутентификации бюджетных RFID-меток типа "запрос-подтверждение", который существенно снижает угрозу подобной "атаки клонов".

В рассматриваемой RFID-системе присутствует RF-сканер и набор RFID-меток, которые, в свою очередь, состоят из антенны и соединенного с ней микрочипа. Память микрочипа RFID-метки рассчитана на хранение определенного числа функций, которое может варьироваться в зависимости от сложности исполнения меток. Храниться может как статическая информация, так и перезаписываемые данные, которые с помощью RF-сканера можно считывать и редактировать. Для этого RF-сканер излучает радиочастотные сигналы, после получения которых метка активируется на чтение или запись. Для работы с RFID-системой достаточно, чтобы сканер находился на расстоянии нескольких метров от метки. При этом необязательно, чтобы метка находилась в зоне прямой видимости от сканера. Более того, возможно наличие непроводящих материалов между ними.

RFID-метки могут иметь встроенные средства поддержки криптографических процедур, контроля целостности и других механизмов защиты. При этом их стоимость будет весьма существенной. Поэтому такие решения нельзя отнести к бюджетным. Бюджетные метки, как правило, имеют емкость не более нескольких сотен бит и являются пассивными. Они содержат несколько тысяч вентилей для логических операций, "питаются" от RF-сканера и не могут выполнять фоновые вычисления во время простоя системы. Указанной емкости бюджетных RFID-меток недостаточно даже для применения стандартной криптографической хэш-функции, такой как MD5 или SHA-1 [5]. Поэтому необходимо разработать новую упрощенную процедуру аутентификации, которую смогут поддерживать бюджетные RFID-метки. Эта процедура предусматривает создание общего ключа между RF-сканером и метками перед началом работы системы. Для обмена сообщениями между сканером и метками используется беспроводная односкачковая пересылка. Метки регулярно идентифицирует сканер. При этом он каждый раз опрашивает метки с новым запросом и осуществляет их аутентификацию путем сравнения ответа с правильным значением.

Для злоумышленника задача взлома протокола сводится к созданию правильного ответа метки на запрос сканера. Если злоумышленник собирает информацию из одного или нескольких прогонов протокола, не прерывая при этом связь между сканером и меткой, то такая атака называется *пассивной*. В случае, если злоумышленник имитирует сканер или метку и отвечает на запросы сканера намеренно измененными сообщениями, которые были просмотрены в предыдущих прогонах, то можно говорить об *активной* атаке.

3. Модернизация протокола аутентификации на основе алгоритма RSA

Вначале следует проиллюстрировать некоторые понятия на примере простейшего протокола:

$$R \to T : x \oplus k = a; \quad (1)$$

$$T \to R : f(x) \oplus k = b, \quad (2)$$

где R и T — это RF-сканер и RFID-метка, соответственно; k — секретный ключ между R и T; x — случайный запрос длиной n бит; f — функция отображения последовательности длиной n бит в новую последовательность аналогичной длины. Пусть $I(h, k)$ — функция взаимосвязи между наблюдаемой парой сообщений $h = (a, b)$ и ключом k. Тогда можно считать верным следующее утверждение: $I(h, k) = H(x \oplus f(x))$, где $H(x \oplus f(x))$ — функция энтропии $x \oplus f(x)$. Это может быть доказано так. По определению $I(h, k) = H(h) - H(h|k)$. Вследствие случайного отбора x следует равенство $H(h|k) = n$. Помимо этого,

$$H(h) = H(a, b) = H(a, a \oplus b) = \\ = H(a \oplus b) + H(a|a \oplus b) = H(x \oplus f(x)) + \\ + H(x \oplus k|x \oplus f(x)) = H(x \oplus f(x)) + n.$$

Таким образом, $I(h, k) = H(x \oplus f(x)) + n - n = H(x \oplus f(x))$.

Например, если f — тождественное отображение (то есть $f(x) = x$), то нельзя получить какую-либо информацию о ключе из просмотра прогонов. Тем не менее выбор подобного отображения — это плохой выбор, так как злоумышленник может просто повторить запрос в качестве ответа. Другая крайность, когда f вырождается в константу, например, ответ заменен известным постоянным вектором. В этом случае $I(h, k)$ достигает максимума. Более того, если $x \oplus f(x)$ является взаимно однозначным отображением в m-размерное подпространство n-размерных векторов, тогда $n—m$ биты ключа выбираются независимо.

Выбор линейного двоичного отображения для f опасно. Для доказательства выбираются M и I как две $n \times n$ бинарные матрицы, где M представляет отображение f, а I — единичная матрица. Злоумышленник может установить следующую систему линейных уравнений:

$$(M \oplus I)k = Ma \oplus b. \quad (3)$$

Решение этой системы для неизвестной k не единственное, если ранг матрицы $M \oplus I$ меньше n. Стоит отметить, что злоумышленник может не

знать точного ключа, чтобы создать сообщение успешного ответа, достаточно узнать произвольное решение (3).

Сохраняя основную структуру, можно предложить следующие направления по улучшению протокола:

— **нелинейность:** использование нелинейного f может усложнить злоумышленнику задачу;

— **смешанные операции:** вместо логической операции сложения по модулю *XOR*, которая линейна по отношению к бинарным векторам, могут быть использованы операции целочисленного сложения по модулю или возведения в степень. Это усложнит как определение сообщения, так и его анализ;

— **ключи:** использовать в обоих направлениях разные ключи.

Тем не менее усиление должно быть сделано осторожно и постепенно: возможны только облегченные модификации, и они должны быть сделаны после тщательного анализа.

Одна из модификаций может быть выполнена с введением функции E, которая зашифрована алгоритмом RSA с длиной n, открытой экспонентой e и секретной экспонентой d. Тогда протокол будет выглядеть следующим образом:

$$R \rightarrow T : x;$$

$$T \rightarrow R : E(x\hat{}k),$$

где x — вектор, в котором $0 < m \leqslant n$ битам выставляется значение "1" в различных произвольно выбранных позициях; $x\hat{}k$ обозначает побитовое "И" векторов x и k, маскируя вектор k. Значения битов в векторе k не изменяются на тех позициях, где соответствующие биты в векторе x имеют значение "1", а остальные биты вектора k устанавливаются на значение "0".

Количество операций, когда вычисляется функция E, зависит от двоичного веса экспоненты, а также от бинарного веса открытого (нешифрованного) текста. При этом используется повторение метода возведения в произвольную степень путем многократного возведения в квадрат и умножения: *Square-and-Multiply* [6]), Второй шаг протокола подтверждает, что бинарный вес открытого текста — это максимум m. Кроме того, применяется низкий вес открытой экспоненты (например $2^{16} + 1$).

Пассивная атака. Злоумышленник также может попробовать определить m бит вектора k, просматривая канал и взломав шифрование функции $E(x\hat{}k)$ весьма сложным и изнурительным поиском битов вектора k по координатам, обозначенным вектором x. Таким образом, для этого потребуется не менее 2^m попыток. Объем работы злоумышленника может возрастать при увеличении m, но это также увеличивает и объем работы T. Помимо этого, чем больше прогонов протокола атакуется, тем больше информации о векторе k может быть получено.

Активная атака. Злоумышленник меняет биты на двух позициях просматриваемых запросов x: бит со значением 1 меняется на 0, а бит со значением 0 — на 1. Далее происходит отслеживание: ответное сообщение не изменится, если вектор k имеет значение 0 на обеих позициях, и изменится на оставшиеся три пары значений (01, 10, 11). Очевидно, что возможность нулевой пары равна 1/4. В этом случае злоумышленник имеет небольшой шанс отсканировать биты ключа.

Усиление. Ключ k постоянно смещается со смещением S, которое является подходящим отображением дополнительного секретного значения k' и действующего вектора x: $S = g(k', x)$.

Заключение

В статье представлен усовершенствованный протокол аутентификации, который по всем параметрам подходит под жесткие условия и ограничения применения бюджетных меток. Отдельно подчеркивается, что при разработке протокола аутентификации особое внимание уделено используемому количеству математических расчетов. При этом недопустимо использование сложных вычислений, требующих значительных ресурсов от всех элементов RFID-системы.

Описанный протокол основан исключительно на базовых элементах, которые поддерживаются любой бюджетной RFID-меткой. Стойкость данного протокола рассмотрена по отношению к некоторым характерным видам атак, однако и он может быть скомпрометирован. Поэтому целью дальнейших исследований является расширение исследовательской базы в поисках компромисса между защищенностью RFID-меток и их эксплуатационными качествами.

Список литературы

1. **Sarma S., Weis S., Engels D.** Radio-frequency identification: Security risks and challenges // CryptoBytes. 2003. V. 6, N 1. P. 2—9.

2. **Hoffstein J., Pipher J., Silverman J.** NTRU: A ring based public key cryptosystem // Third International Symposium Algorithmic Number Theory (ANTS III), Portland, Orgeon, USA, June 21—25, 1998. P. 267—288.

3. **Stern J., Stern J. P.** Cryptanalysis of the OTM signature scheme from FC'02 // Proc. of 7th Financial Cryptography Conference, Guadeloupe, French West Indies. January. 2003. P. 138—148.

4. **Weis S., Sarma S., Rivest R., Engels D.** Security and privacy aspects of low-cost radio frequency identification systems // Proc. of First International Conference on Security in Pervasive Computing, Boppard, Germany. March 2003. P. 201—212.

5. **Баричев С. Г., Гончаров В. В., Серов Р. Е.** Основы современной криптографии. М.: Горячая Линия — Телеком, 2002.

6. **Bogusch R. L.** Frequency Selective Propagation Effects on Spread — Spectrum Receiver Tracking // Proc. of the IEEE. 1981. V. 69, N 7. P. 787—796.

УДК 004.932.2: 519.254

Т. Б. Чистякова[1],
д-р техн. наук, проф., зав. каф.,
И. А. Садыков[1], аспирант,
К. Колерт[1, 2], канд. техн. наук,
почетный проф., директор по технологии,
А. Б. Иванов[1],
канд. техн. наук, ст. преподаватель
[1] Санкт-Петербургский технологический институт (ГУ),
e-mail: sapr@ws01.sapr.pu/ru
[2] Компания "Klöckner Pentaplast Europe", Германия

Методы кодирования и идентификации упаковок фармацевтической продукции для защиты от фальсификации

Проанализированы существующие методы защиты упаковок фармацевтической продукции. Предложены методы физической обработки продукции, методы математической обработки результатов сканирования поверхности упаковки. Разработаны алгоритмы кодирования и идентификации, архитектура компьютерной системы и программно-аппаратный комплекс средств защиты полимерных упаковок от подделки для реализации предложенных методов, а также распределенное программное приложение. Работоспособность комплекса проверена на международных промышленных производствах полимерных упаковок.

Ключевые слова: *фальсификация, защита от подделки, распознавание изображений, программно-аппаратный комплекс, кодирование, идентификация*

Актуальность

На сегодняшний день подделка различных видов продукции приобрела индустриальный характер. Объем фальсифицированной продукции в отдельных отраслях сравним с объемом легального производства, а зачастую и превышает его. К сожалению, проблема фальсификации продукции касается практически всех отраслей хозяйственной деятельности, в частности фармацевтической. В общем обороте лекарственных препаратов в России подделки составляют от 10 до 80 %, а ежегодный доход от данного сегмента рынка равен около 7 млрд долл. В целом, в мире ежегодный оборот фальсифицированной продукции составляет около 600 млрд долл. США. За последние 20 лет эта величина выросла на 10 000 %. Приблизительно 5—7 % всей мировой продукции является подделкой.

Для защиты фармацевтической продукции от подделки предлагаются методы кодирования и распознавания упаковок фармацевтической продукции.

Рост преступлений, связанных с подделкой, обусловлен значительным прогрессом в копировальной и лазерной технике, органической химии и т. п. Появилось большое количество относительно недорогой копировальной техники, позволяющей с высокой точностью копировать защитные элементы и создающей реальную угрозу роста фальсификации.

Постановка задачи исследования

В соответствии с изложенным выше целью работы является разработка комплекса методов, моделей и средств защиты полимерных пленок от фальсификации, включающего устройство сканирования (фотографирования) упаковки, программное обеспечение для обработки полученного изображения и вычисления цифровой подписи на основе математических алгоритмов и методов аналитической геометрии, базы данных электронных подписей для каждой полимерной упаковки, а также программного обеспечения, позволяющего выполнить поиск и непосредственную идентификацию полученной электронной подписи на основе алгоритмов комбинаторики.

Для достижения поставленной цели необходимо решение следующих задач:

- анализ существующих проблем и методов защиты продукции от фальсификации;
- разработка общей функциональной архитектуры системы защиты фармацевтической продукции от подделки;
- разработка методов кодирования и идентификации полимерных упаковок, позволяющих с заданной точностью и с учетом объема выпускаемой продукции обеспечить требуемую степень защиты от фальсификации;
- проектирование и разработка компьютерной системы, имеющей клиент-серверную архитектуру и представляющей собой комплекс средств защиты полимерных упаковок от фальсификации;
- разработка программного обеспечения для обработки полученного изображения и вычисления электронной цифровой подписи (ЭЦП) на основе математических алгоритмов и методов аналитической геометрии;
- проектирование и разработка базы данных электронных подписей для каждой полимерной упаковки с учетом особенностей выбранного для каждого типа продукции метода кодирования, а также требований к объему выпускаемой продукции и соответствующему необходимому

объему данных, требуемому для хранения подлинных ЭЦП;

- разработка алгоритма верификации упаковки, позволяющего с заданной точностью и выбранным методом идентификации провести поиск заданной ЭЦП в базе данных и тем самым определить подлинность упаковки;
- апробация разработанной системы на производстве.

Анализ методов защиты продукции от подделки

Анализ существующих методов защиты промышленной продукции показывает, что к ним предъявляются следующие требования:

- сложность и высокая стоимость копирования элементов защиты, т. е. при разработке систем защиты выбираются такие технологии, чтобы фальсификация обходилась в несколько раз дороже оригинала;
- использование нескольких уровней защиты и различных технологий (аппаратных, программных, технологических);
- сохранение механической целостности защитных элементов и элементов упаковки в процессе перехода товара от производителя к потребителю и доступность проверки целостности на любом этапе;
- режимная и правовая защита, доступность контроля за производством самих защитных элементов.

Обычные методы защиты продукции, такие как радиочастотная идентификация, голограммы и т. д., применимы только для конечной продукции, что неприемлемо для лекарственных препаратов, у которых может быть защищена только упаковка. Кроме того, анализ методов показывает, что для повышения эффективности защиты необходимо использование недетерминированных алгоритмов, основанных на элементе случайности, поскольку это повышает вероятность того, что защитные элементы не будут воспроизведены полностью. Подобные существующие методы защиты, использующие магнитные частицы и металлические наночастицы, решают эту проблему, однако являются крайне дорогостоящими.

Нами предложен более дешевый, по сравнению с описанными выше, двухуровневый метод защиты упаковок, согласно которому в структуре упаковочного материала (полимерной пленки, из которой изготовляется упаковка) случайным образом распределяются частицы люминесцирующего вещества (пигменты), невидимые невооруженным глазом. Для того чтобы активировать данные частицы, необходимо осветить упаковку ультрафиолетовым или инфракрасным излучением с определенной длиной волны с помощью прибора, показанного на рис. 1 [1].

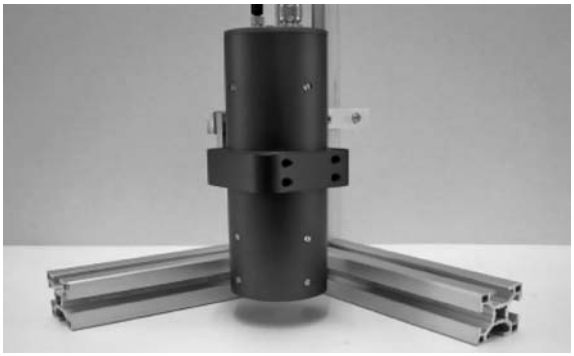

Рис. 1. Измерительный прибор для определения кривой поглощения активированных пигментов

Стоимость производства подобных защитных элементов ниже, чем 0,01 цент за 1 м^2, а содержание люминофоров в общем объеме полимерной пленки составляет 0,001 %, что соответствует приблизительно 1—5 пигментам на 1 см^2 пленки.

Данный метод предполагает наличие защиты двух уровней:

- защиту на уровне определения подлинности состава люминесцирующего вещества посредством фотометрического анализа кривой поглощения активированных пигментов;
- защиту на уровне геометрического анализа взаиморасположения люминесцирующих пигментов относительно друг друга.

Защита упаковки в этом случае предусматривает две стадии: стадию кодирования и стадию идентификации упаковки [2]. Во время кодирования пятна люминофора представляются как точки на поверхности. На основе этих точек строятся геометрические фигуры, элементы которых принимаются за ЭЦП. Цифровые подписи каждой упаковки с данными о типе продукции и времени кодирования сохраняются в базе данных. Для последующей проверки подлинности упаковки происходит повторное кодирование (определение элементов в геометрических фигурах) и последующий поиск вычисленной цифровой подписи в базе данных для определения типа продукции и установления подлинности упаковки.

Функциональная структура комплекса

Общая структура компьютерной системы защиты фармацевтической продукции от фальсификации представлена на рис. 2. Кодирование упаковок выполняется в производственных подразделениях сразу после упаковки продукции. Подсистема кодирования с помощью специального оборудования (инфракрасного сканера, подключенного к ЭВМ) сканирует упаковку, вычисляет ее ЭЦП, которая зависит от конкретного типа продукции и метода

кодирования, после чего заносит ее в базу данных подлинных ЭЦП.

Подсистему верификации, состоящую из устройства считывания, ЭВМ и специального программного обеспечения, устанавливают в пунктах приема и обработки промаркированной продукции, чтобы подтвердить ее подлинность. В подсистеме верификации выполняется сканирование упаковки для получения изображения, содержащего набор точек. После чего формируется множество возможных ЭЦП, которое передается на сервер, где на его основе происходит поиск совпадений в базе данных подлинных ЭЦП.

Клиентское приложение связано через каналы глобальной сети Интернет с серверной частью системы. Серверная часть связана с базой данных ЭЦП, где хранятся подписи для различных типов упаковок и различного рода продукции. Получив ЭЦП, серверная часть идентифицирует ее и определяет подлинность упаковки. Совпадение этой контрольной информации с информацией, хранящейся в базе данных электронных подписей, гарантирует подлинность упаковки. Схематически стадия идентификации представлена на рис. 3.

Рис. 2. Структура программного комплекса

Рис. 3. Стадия идентификации упаковки

Стадия кодирования

Кодирование отсканированного изображения зависит от таких параметров как объект кодирования (полимерная пленка, пластиковая карта и т. д.), элемент кодирования (треугольник, окружность, прямоугольник и т. п.), диаметр нанесенной частицы (микрометр) и точность кодирования. Учитывая все перечисленные выше параметры, можно выбрать подходящий метод кодирования. На данный момент реализовано два метода кодирования: с использованием треугольников и с использованием описанных вокруг треугольников окружностей.

На этапе кодирования сначала проводится распознание изображения (выделение n наиболее ярких центров из набора и представление их в виде точек на плоскости упаковочного материала). При кодировании из этого распознанного набора точек случайным образом выбирается некоторое их число, которое будем называть размером хэша (k). Это число, так же как и метод кодирования в разработанном нами ПО, может быть установлено производителем продукции и зависит от объемов производства, требований к защищенности продукции, а также от разрешающей способности сканирующего устройства.

На рис. 4 представлены точки, распознанные на снимке упаковки, где *1* — общий набор точек, распознанных на упаковке; *2* — единственная распознанная на упаковке точка; *3* — исключенная

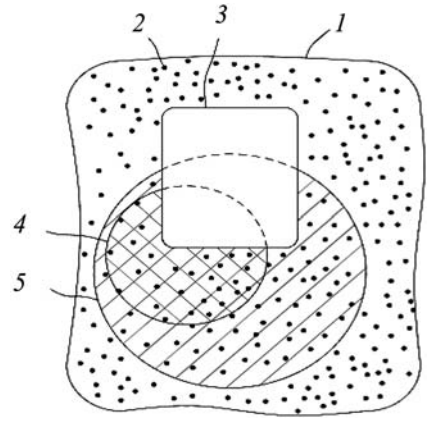

Рис. 4. Наборы точек, найденных на упаковке при распознавании

из распознавания область упаковки (например, деформируемая область фармацевтической упаковки, содержащая таблетку); *4* — набор случайно выбираемых точек k на стадии кодирования (их число равно размеру хэша); *5* — набор наиболее ярких точек (n), распознанных на упаковке. Из данного набора точек случайным образом выбирается набор точек k.

Предметная область (фармацевтическая продукция) подразумевает значительный объем производства (несколько миллионов упаковок одного типа продукции в год). Поэтому для реализации задачи об идентификации подлинности упаковки требуется хранение больших объемов данных, а также возможность быстрого поиска по этим данным. Такая проблема решается методом разделения общего набора данных на меньшие наборы, каждый из которых содержит лишь часть всей общей идентификационной информации. Эти меньшие наборы называют "хранилищами подписей", они представляют собой динамически формируемые таблицы в реляционной базе данных. Таблицы с сохраненными ЭЦП формируются динамически. Их структура зависит от следующих параметров: тип продукции; метод кодирования; число используемых геометрических элементов; диапазон контрольной суммы. Контрольная сумма ЭЦП — это число, которое определяют для каждого набора геометрических элементов, оно зависит от используемого метода кодирования. Для метода кодирования треугольниками это число определяют по формуле

$$S = \sum_{t_1}^{t_n} (a_{\text{ср}} - a_{\text{мин}}), \qquad (1)$$

где $a_{\text{ср}}$ и $a_{\text{мин}}$ — среднее и минимальное значения углов в треугольнике.

Для метода кодирования описанными окружностями контрольная сумма вычисляется по следующей формуле:

$$S = \sum_{t=1}^{n-1} R_i(n - i + 1) - R_{n-i}, \qquad (2)$$

где R_i — радиус i-й окружности; n — общее число окружностей.

Диапазон контрольной суммы — это диапазон значений, в который попадает контрольная сумма, верхняя и нижняя границы диапазона кратны шагу диапазона. Шаг диапазона зависит от требований к защищенности продукции и задается отдельно для каждого метода кодирования. Таким образом, на показатель защищенности продукции влияют следующие факторы: выбранный метод кодирования, точность распознания изображения (разрешающая способность сканирующего устройства), число используемых при кодировании активированных точек, а также диапазон контрольной суммы ЭЦП.

Стадия идентификации

Для проверки подлинности полимерной упаковки осуществляется повторное кодирование (определение элементов в геометрических фигурах) и последующий поиск вычисленной цифровой подписи в базе данных для определения типа продукции и установления подлинности упаковки [3].

Идентификация упаковки происходит в два этапа и представляет собой перебор всех возможных сочетаний точек из общего набора по числу используемых при сохранении (размер хэша) [4]. На стадии идентификации, как видно из рис. 5, пользователь имеет возможность указать максимально допустимое абсолютное отклонение δ (в единицах, используемых при вычислении ЭЦП, для метода кодирования треугольниками — это угловые градусы, для метода с использованием описанных окружностей — микрометры). Также при идентификации задается число наиболее ярких точек, выбираемых на упаковке n, и число случайным образом выбираемых из n точек k, которое используется для вычисления ЭЦП. От этого числа напрямую зависит сложность вычисляемой ЭЦП (число используемых геометрических элементов). Например, для метода кодирования треугольниками при $k = 4$, число треугольников также равно 4, при $k = 5$, число треугольников уже равно 10, а для $k = 6$ точек число треугольников равно 20.

Первый этап идентификации заключается в поиске совпадений ЭЦП, построенных на основе k точек, и называется проверкой полного совпа-

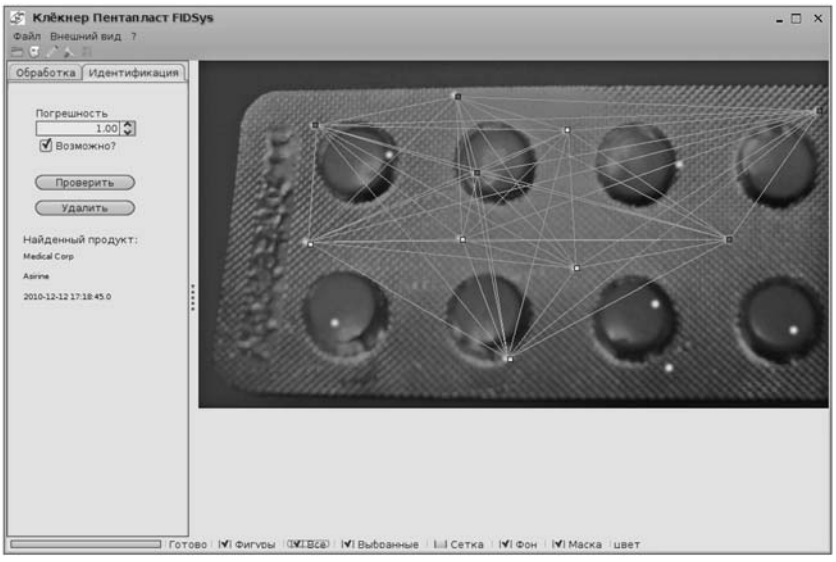

Рис. 5. Внешний вид интерфейса программы

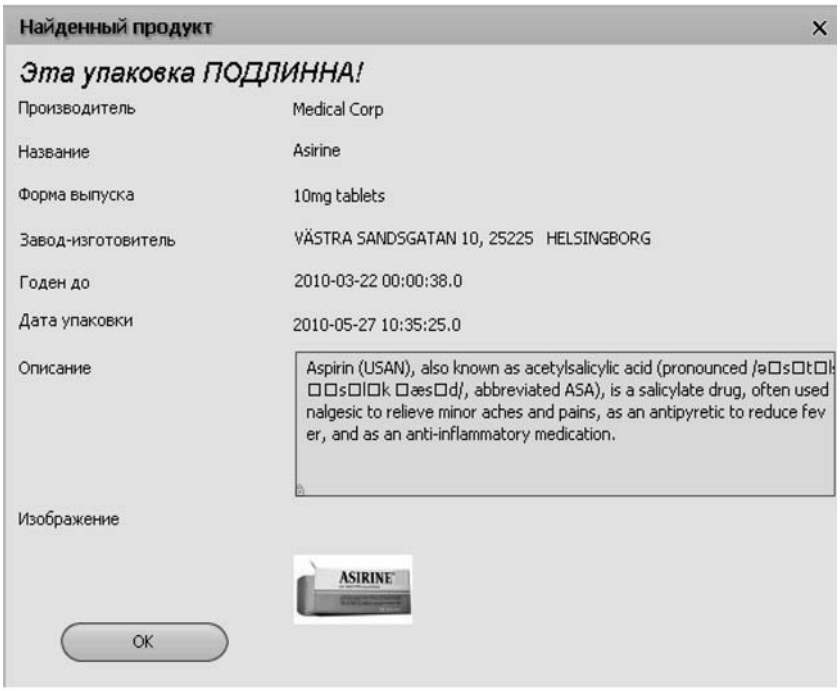

Рис. 6. Сообщение о подлинности упаковки

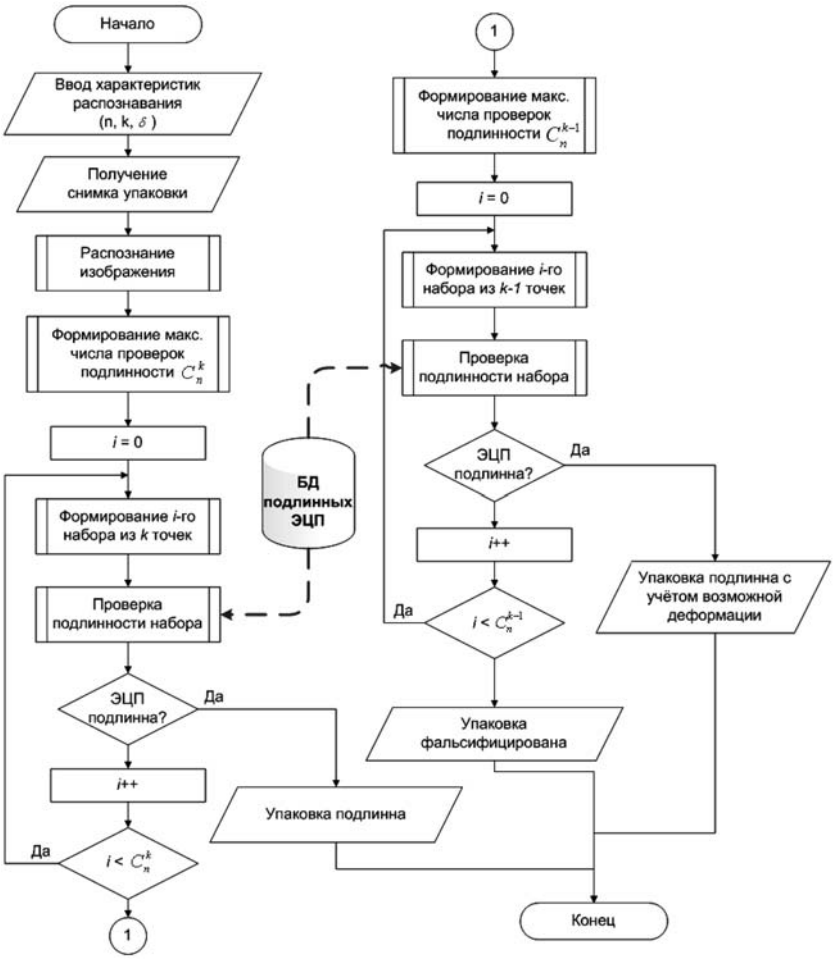

Рис. 7. Обобщенный алгоритм проверки подлинности упаковки

дения. Максимальное число проверок наличия построенной ЭЦП в базе данных, согласно комбинаторике, определяется по формуле

$$C_n^k = \frac{n!}{k!(n-k)!}, \qquad (3)$$

где n — общее число распознанных на упаковке точек; k — используемое для кодирования число точек упаковки.

Если найдено хотя бы одно совпадение построенных подписей и информации из базы данных, упаковка признается подлинной, и пользователь получает сообщение, показанное на рис. 6.

На втором этапе проверяются ЭЦП, построенные на основе $k-1$ точек. Этот этап называется поиском частичного совпадения. Максимальное число проверок при выполнении данного поиска вычисляется по формуле

$$C_n^{k-1} = \frac{n!}{(k-1)!(n-k-1)!}. \qquad (4)$$

Если при выполнении поиска частичного совпадения найдено совпадение хотя бы одной подписи в базе данных, упаковка признается подлинной, но пользователь получает предупреждение о возможной деформации упаковки. В этом случае существует определенная вероятность того, что упаковка подлинна (эта вероятность зависит от параметров кодирования, конкретной упаковки данного типа продукции, типа пленки, используемой при производстве упаковки, а также используемого метода кодирования).

Упаковка признается фальсифицированной в том случае, если ни первый, ни второй этап идентификации не дали результатов. Обобщенный алгоритм идентификации отображен на рис. 7.

Обеспечение информационной безопасности

Одним из ключевых моментов при разработке и использовании программно-аппаратного комплекса является защита данных. Комплекс имеет клиент-серверную архитекту-

ру и централизованное хранилище данных, что обеспечивает достаточно высокое качество защиты. Проверка подлинности также происходит на сервере идентификации, поэтому данные о подлинных ЭЦП не подвергаются риску быть перехваченными при передаче. Кроме того, для обеспечения дополнительной защиты данных разработан собственный протокол передачи данных. При передаче данные шифруются с использованием симметричного алгоритма блочного шифрования с ключом 256 бит. Каждое предприятие по производству фармацевтической продукции получает свой уникальный код шифрования, который хранится на пунктах верификации и используется для шифрования данных при запросе подлинности упаковок. Для запроса подлинности упаковки выполняется авторизация верификационного пункта, для чего на сервер поступает его идентификационный номер и зашифрованные данные для авторизации (логин-пароль). После успешной авторизации пользователь получает право осуществлять идентификацию упаковок определенных производителей (список разрешенных производителей определяется на стадии регистрации авторизованного верификационного пункта).

Использованные технологии

Программный комплекс разработан с использованием технологий платформы Java2 Enterprise Edition (J2EE) версии 1.6. Данная платформа предоставляет широкий спектр технологий для разработки кросс-платформенных приложений с использованием клиент-серверного взаимодействия.

Серверная часть комплекса реализована в виде Web-приложения, выполняющегося на сервере приложений JBoss AS 5.0. Данное приложение имеет Web-интерфейс для администрирования и конфигурирования системы.

Клиентское приложение, предоставляющее интерфейс для идентификации упаковок, реализовано с использованием библиотеки Swing, а также библиотеки обработки изображений JavaVis.

В качестве централизованной системы управления базами данных комплекс использует MySQL Server 5.1.

Тестирование работоспособности комплекса

Программный комплекс внедрен в эксплуатацию и прошел успешное тестирование на заводах корпорации по производству полимерной пленки (Клёкнер Пентапласт) в Европе и России. Например, для испытаний, которые проводились на фармацевтической продукции, упакованной в полимерную пленку ЭП-73, произведенную по ГОСТ 25250—88 на заводе "ООО Клёкнер Пентапласт Рус", г. Санкт-Петербург, среднее время идентификации одной упаковки составило не более 30 с при наличии подлинных ЭЦП числом до 1 млн в базе данных. Программный комплекс имеет свидетельство Роспатента о государственной регистрации программы для ЭВМ [5]. Предложенный метод защиты продукции имеет немецкий и европейский патенты [6, 7].

Расходы на данную систему очень незначительны вследствие низкой концентрации используемых пигментов и стоимости устройств обработки данных (освещение, камера). Данная технология предлагается всем заказчикам и пользователям полимерной пленки фирмы "Клёкнер Пентапласт" для защиты их ценной продукции от подделки. Использование предложенных физических методов защиты, а также математических методов и алгоритмов кодирования и идентификации позволяет за приемлемое время провести идентификацию фармацевтических упаковок. Данный программно-аппаратный комплекс является гибким инструментом, настраиваемым на различные типы продукции и методы кодирования упаковок фармацевтической продукции для защиты от фальсификации.

Список литературы

1. **Kohlert C., Kohlert M., Chistyakova T., Ivanov A., Sadykov I.** Counterfeit-proofing based on the principle of randomness // Kunststoffe international. 2010. N 7. P. 32—35.

2. **Чистякова Т. Б., Садыков И. А., Колерт К.** Структура системы кодирования и идентификации полимерных изделий // Сб. тр. XXII Междунар. науч. конф. "Математические методы в технике и технологиях — ММТТ-22". Псков, 2009. Т. 9. С. 202—203.

3. **Чистякова Т. Б., Садыков И. А., Колерт К.** Алгоритмы кодирования и идентификации для защиты полимерных изделий от фальсификации // Сб. тр. XXIII Междунар. науч. конф. "Математические методы в технике и технологиях — ММТТ-23". Смоленск, 2010. Т. 12. С. 34—36.

4. **Садыков И. А., Албуткина О. С., Масликов А. И.** Программное обеспечение системы кодирования и идентификации фармацевтических упаковок // Сб. тр. XXIII Междунар. науч. конф. "Математические методы в технике и технологиях — ММТТ-23". Смоленск, 2010. Т. 12. С. 33—34.

5. **Свидетельство** о государственной регистрации программы для ЭВМ 2010614239 Рос. Федерация. Программный комплекс для обеспечения защиты полимерных изделий от подделки / Чистякова Т. Б., Колерт К., Садыков И. А., Албуткина О. С., Масликов А. И. // Программы для ЭВМ. Базы данных. Топологии интегральных микросхем: офиц. бюл. Роспатента. — 2010. — Вып. 3. — С. 448.

6. **Patent** DE 10 2008 032 781 A1. Verpackungsfolie für Produktauthentifizierung, Authentifizierungsverfahren und -system / Kohlert C., Schmidt B., Egenolf W., Chistjakova T.

7. **Patent WO** 2010/003585 A1. Packaging film for product authentication, authentication method and system / C. Kohlert, B. Schmidt, W. Egenolf, T. Chistjakova.

УДК 004.93

В. В. Колюжнов, аспирант,
В. В. Колотов, аспирант,
В. И. Сединин, д-р техн. наук, проф., зав. каф.,
Сибирский государственный университет
телекоммуникаций и информатики,
г. Новосибирск,
e-mail: kvv2@inbox.ru

Новый подход к распознаванию номерных знаков и оценка влияния различных факторов на эффективность распознавания

Предлагается новый подход к распознаванию номерных знаков транспортных средств. Приведены результаты моделирования дорожно-транспортных ситуаций и распознавания номерных знаков. Проанализировано влияние различных факторов на результат распознавания.

Ключевые слова: *компьютерное зрение, обработка изображений, оптическое распознавание текстов, распознавание номерных знаков*

Введение

Идентификация транспортных средств используется в различных областях. Камера видеонаблюдения, установленная на автопарковке, позволяет получить изображение номерного знака из последовательности видеокадров. Эту информацию можно использовать для ускорения процесса регистрации транспортного средства и выписки чека его владельцу. По этому принципу создаются даже полностью автоматизированные парковки, не требующие вмешательства человека. Аналогичная ситуация возникает и при контроле въезда на территорию предприятия. По имеющейся на контрольно-пропускном пункте базе данных разрешенных транспортных средств осуществляется автоматическое открывание ворот. Появляющиеся в России платные дороги также могут оборудоваться системами автоматического контроля проезжающих транспортных средств и автоматического сбора платы за проезд. Во всех крупных городах России, в том числе в Новосибирске, стоит проблема регулярного нарушения водителями правил дорожного движения и возникновения большого числа ДТП. Установив камеры на перекрестках и в наиболее опасных точках дорожного движения, можно анализировать скорость движущихся транспортных средств и выписывать штрафы водителям, превышающим максимально допустимую скорость.

Постановка задачи

Проблему идентификации транспортного средства можно разбить на две части. Первая — распознавание номерного знака транспортного средства, вторая — получение информации из базы данных ГИБДД по распознанному номерному знаку (рис. 1). Вместо базы данных ГИБДД можно использовать любую другую аналогичную базу данных транспортных средств, например, локальную базу данных автопарковки.

Упрощенно процесс распознавания номерного знака можно представить в виде следующих этапов: выделение прямоугольной области, содержащей номерной знак, сегментация символов, распознавание символа (рис. 2).

Для выделения прямоугольной области можно использовать достаточно большое число методов. Например, в [1] описан метод использования второй производной, в [2] — метод горизонтального и вертикального проецирования изображения, в [3] предлагается подход, использующий морфологические операторы. Достоинства и недостатки этих методов отображены в табл. 1. Сегментация символов отражена в [3]. Распознавание символов можно осуществлять с помощью метода, использованного в [4].

Ранее предлагаемые подходы не дают высокого уровня распознавания видеоряда в реальном времени, что необходимо для задач интеллектуальных транспортных систем.

Рис. 1. Проблема идентификации транспортных средств

Рис. 2. Упрощенная схема распознавания номерного знака

Таблица 1

Достоинства и недостатки методов выделения прямоугольной области, содержащей номерной знак

Метод	Достоинства	Недостатки
Использование второй производной	Низкая трудоемкость (дополнительно может быть снижена за счет просчета каждой *n*-й строки)	Низкая эффективность алгоритма
Построение проекций	1. Устойчивость к размеру номерного знака. 2 Высокое качество распознавания за счет сбора большого количества статистики	1. Сложность определения точной области номерного знака. 2. Обнаружение только одного номерного знака
Морфологический подход	Однотипность операций	1. Требование корректной установки глобального порога изображения. 2. Настройка на конкретный размер номерного знака

Рис. 3. Схема предлагаемой системы распознавания

В реальных условиях невозможно выделить из всего множества факторов, влияющих на эффективность распознавания номерных знаков, наиболее значимые, поэтому необходимо провести моделирование, абстрагировавшись от них.

Эта статья направлена на устранение указанных недостатков.

Предлагаемая схема распознавания. На рис. 3 приведена схема предлагаемой системы распознавания.

В качестве предварительной обработки изображение подвергается эквализации гистограммы. Обнаружение области номерного знака происходит в несколько этапов.

Выделение движущихся объектов позволяет в дальнейшем искать в каждой области только один номерной знак. Следующий этап — горизонтальное и вертикальное проецирование — позволяет обнаружить широкую область номерного знака. За счет сбора этим методом большого количества статистических данных достигается высокая эффективность нахождения широкой области номерного знака.

Данный метод может находить несколько ограничивающих прямоугольников, но только один из них является достоверным. Для выбора применяются следующие эвристики:

$$\frac{1}{6} W \leqslant w \leqslant \frac{1}{2} W;$$

$$\frac{1}{36} W^2 k \leqslant S \leqslant \frac{1}{2} W^2 k,$$

где w — ширина ограничивающего прямоугольника; W — ширина выделенного двигающегося объекта; S — площадь ограничивающего прямоуголь-

ника; k — соотношение сторон номерного знака по ГОСТ Р 50577—93 [5].

Далее используется преобразование Хафа для коррекции наклона номерного знака и нахождения его точной границы. Данный метод также может найти несколько потенциальных областей номерного знака. Для выбора лучшей из них применяется следующая эвристика:

$$w_{\text{ГОСТ}} \cdot 0{,}8 \leqslant w_{\text{Хаф}} \leqslant w_{\text{ГОСТ}} \cdot 1{,}2;$$

$$h_{\text{ГОСТ}} \cdot 0{,}8 \leqslant h_{\text{Хаф}} \leqslant h_{\text{ГОСТ}} \cdot 1{,}2,$$

где $w_{\text{Хаф}}$ — ширина ограничивающего прямоугольника после преобразования Хафа; $w_{\text{ГОСТ}}$ — ширина номерного знака по ГОСТ Р 50577—93 [5]; $h_{\text{Хаф}}$ — высота ограничивающего прямоугольника после преобразования Хафа; $h_{\text{ГОСТ}}$ — высота номерного знака по ГОСТ Р 50577—93 [5].

Так как предполагается, что с помощью преобразования Хафа была найдена точная область номерного знака и число символов и их предположительное местонахождение известно из ГОСТ Р 50577—93 [5], то можно использовать эту информацию для сегментации символов.

В качестве подготовки к распознаванию проводится бинаризация изображений символов с помощью метода Otsu [6].

Распознавание отдельных символов осуществляется методом сравнения с шаблоном с помощью следующей метрики [4]:

$$M = \frac{X \circ Y}{\sqrt{(X \circ X)(Y \circ Y)}};$$

$$X \circ Y = \sum_{i=0}^{N} \sum_{j=0}^{P} X_{i,j} Y_{i,j},$$

где N и P — число строк и столбцов матрицы соответственно.

Экспериментальная установка. Для проверки результатов теоретического исследования был создан специальный компьютеризированный макет-стенд для моделирования дорожно-транспортных ситуаций и распознавания номерных знаков (рис. 4).

На рис. 5 представлена структурная схема стенда.

В устройстве сопряжения используется модернизированная схема, описанная в [7]. Подключение к компьютеру осуществляется через COM-порт. С компьютера через устройство сопряжения подаются управляющие сигналы на электрифицированные элементы стенда: светофоры, шлагбаум, устройства радиоуправления.

Радиоуправление осуществляется на частотах 27 и 40 МГц. На эту же частоту настроены и приемники радиоуправляемых моделей автомобилей. К компьютеру также подключена web-камера. Разработанное программное обеспечение позволяет вести съемку с настройкой ее параметров. Снятые изображения стенда обрабатываются разработанным программным обеспечением, реализующим описанную выше схему распознавания номерных знаков. Рассмотрим наиболее характерные случаи применения стенда.

Например, в автоматизированной парковке, оборудованной камерой видеонаблюдения и шлагбаумом, изначально шлагбаум закрыт, web-камера находится в режиме поиска движущихся объектов. Как только объект останавливается, в прямоугольной области, ограничивающей ранее двигавшийся объект, происходит поиск и распознавание номерного знака. Полученный распознанный номерной знак сверяется с записями базы данных автомобилей, разрешенных к проезду. Если такой номер найден, шлагбаум открывается.

Вновь задействуется обнаружение движения. Если движение завершилось вне области действия механизма шлагбаума, то он закрывается. В реальной ситуации, чтобы быть занесенным в такую базу данных, необходимо предварительно оплатить стоянку.

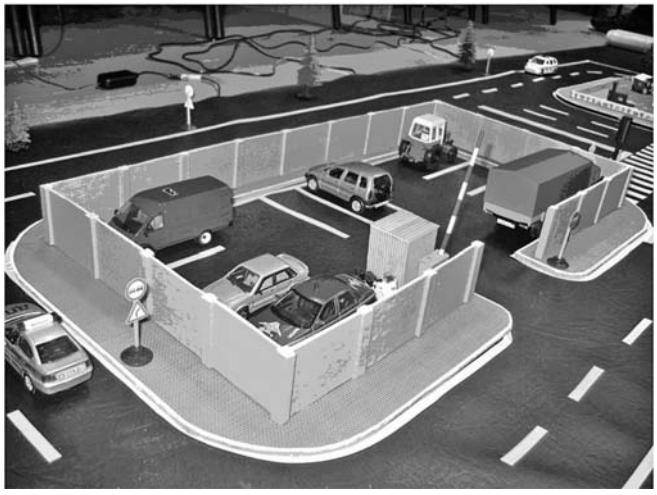

Рис. 4. Стенд для моделирования транспортных ситуаций и распознавания номерных знаков

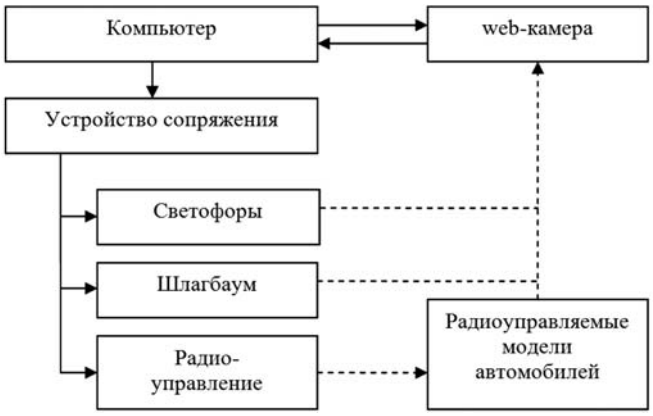

Рис. 5. Структурная схема стенда

Стенд может быть использован и для распознавания номерных знаков нарушителя правил дорожного движения. Распознавание номерных знаков будет полезным и для быстрого составления протоколов в случае дорожно-транспортных происшествий.

Результаты экспериментов

Для проведения экспериментов были подготовлены: база данных реальных фотографий, полученных фотоаппаратом с разрешением 3 Мпикс. Из нее для эксперимента случайным образом были отобраны 100 фотографий. Кроме того, были выбраны 100 фотографий, содержащих чистые номерные знаки. Но эти фотографии, возможно, содержат деформированные номерные знаки, засветку или сложное окружение (большое число объектов, погодные условия, надписи). Также была сформирована база данных фотографий, полученных с web-камеры во время моделирования дорожно-транспортных ситуаций. Для сравнения были отобраны случайным образом 100 сделанных фотографий. Номерные знаки, установленные на моделях автомобилей, сгенерированы программно с учетом требований ГОСТ Р 50577—93 [5] касательно шрифта, местоположения символов, окантовки, расположения креплений. Эти знаки не содержат загрязнений либо деформации, т. е. в модели эти факторы не учтены.

Таблица 2

Экспериментальные результаты

Метод	Процент правильного определения		
	Стенд	Фотографии	Фотографии с чистыми номерами
Использование второй производной	22	12	17
Построение проекций	92	68	88
Морфологический подход	48	32	44
Сегментация номерного знака	92	86	91
Использование метрики для распознавания	74	62	71

Полученные результаты экспериментов сведены в табл. 2.

Обсуждение результатов

Как видно из табл. 2, процент правильного распознавания номерных знаков при работе со стендом гораздо выше, чем при работе с реальными фотографиями. Это происходит по следующим причинам: загрязнение номера, его деформация, сложное окружение, погодные условия. Проценты распознавания фотографий, сделанных на стенде и на фотографиях с чистыми номерами близки по значениям. Это означает, что наиболее значительное снижение эффективности распознавания дает загрязнение номера. Отсюда следует вывод о необходимости разработки алгоритма "очистки" номерных знаков.

Выводы

По результатам проведенной работы можно сделать вывод о необходимости разработки метода для очистки номерных знаков для качественного увеличения коэффициента распознавания.

Предложенный метод нуждается в дальнейшем исследовании и проведении дополнительных экспериментов для сопоставления с другими методами.

Список литературы

1. **Бусыгин Л. А.** Локализация автомобильного номера в потоке видеоданных в режиме реального времени. URL: http://www.inf.tsu.ru/library/DiplomaWorks/CompScience/2006/busigin/diplom.pdf (дата обращения 19.10.2010).
2. **Martinsky O.** Algorithmic and mathematical principles of automatic number plate recognition systems. URL: http://javaanpr.sourceforge.net (дата обращения 19.10.2010) .
3. **Martin F., Gratia M., Alba J. L.** New methods for automatic reading of VLP's (Vehicle License Plates). URL: http://wgpi.tsc.uvigo.es/pub/papers/sppra02.pdf (дата обращения 09.10.2009).
4. **Remus Brad,** License Plate Recognition System. URL: http://remus.ulbsibiu.ro/publications/ (дата обращения 19.10.2010).
5. **ГОСТ Р 50577—93** Знаки государственные регистрационные транспортных средств. Типы и основные размеры. Технические требования. Государственный стандарт РФ.
6. **Otsu N.** A Threshold Selection Method from Gray-Level Histogram // IEEE Transactions on systems, man and cybernetics. 1979. Vol. 9. № 1.
7. **Управление** радиомоделью при помощи компьютера. URL: http://cxem.net/uprav/uprav14.php (дата обращения 15.03.10).

ЖУРНАЛ В ЖУРНАЛЕ

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

№ 7
ИЮЛЬ
2011

УДК 004.93'42

Э. Д. Аведьян[1], д-р техн. наук, зам. нач. лаб., e-mail:avedian@mail.ru,
А. И. Галушкин[1, 2], д-р техн. наук, проф., нач. лаб., e-mail: neurocomputer@yandex.ru,
Д. В. Пантюхин[2], инж., e-mail:dim_beavis@mail.ru
[1] ФГНУ "Центр информационных технологий и систем органов исполнительной власти", г. Москва
[2] Международный центр по информатике и электронике (ИнтерЭВМ), г. Москва

Ассоциативная нейронная сеть CMAC и ее модификации в задаче распознавания образов[1]

Приводятся результаты применения нейронной сети CMAC и ее модификаций в модельной довольно сложной двумерной задаче распознавания образов с невыпуклыми областями решений. Анализ базируется на компьютерном моделировании. Дается краткое описание нейронной сети CMAC и ее модификаций. Исследуется влияние ошибок учителя на точность классификации. Результаты компьютерного моделирования показывают, что модифицированная нейронная сеть CMAC решает задачу классификации с большой точностью.

Ключевые слова: *нейронная сеть CMAC, модификации, распознавание образов, моделирование*

Введение

Формализованная постановка задачи распознавания образов с учителем предполагает, что имеется набор N-мерных данных (образов) $\mathbf{x}[n]$, $n = \overline{1, K}$. Известно число классов $\Re$, из которых состоит набор данных, и принадлежность каждого образа из набора конкретному классу с номером $R = \overline{1, \Re}$. Требуется по данной информации построить решающее правило, которое позволяет определить принадлежность образа конкретному классу, т. е. построить дискриминантную функцию $g(\mathbf{x})$, которая по значению вектора $\mathbf{x}$ определяет номер класса $R = g(\mathbf{x})$.

В такой постановке задача распознавания образов возникла в начале 60-х годов прошлого века, и тогда она имела в основном теоретический характер. Одной из первых работ по распознаванию образов была книга Н. Нильсона [1], в которой было показано, что несложные задачи распознавания образов могут быть решены с помощью обучаемого персептрона Розенблатта [2]. Для решения сложных задач распознавания, где дискриминантная функция представляет нелинейную функцию N переменных, решение может быть достигнуто с помощью многослойного персептрона, т. е. с помощью искусственной многослойной нейронной сети. Заметим, что алгоритм обучения такой сети к моменту опубликования работы [1] не был известен. На Западе алгоритм обучения многослойной нейронной сети, известный теперь под названием алгоритм обратного распространения ошибки, впервые был опубликован в трудах Исследовательской группы по параллельным вычислениям США [3] только в 1986 г. В Советском Союзе алгоритмы обучения многослойной нейронной сети были опубликованы существенно раньше в работах А. И. Галушкина в 1973, 1974 гг. [4], [5]. Эти работы, однако, на Западе оставались довольно долго неизвестными, и только после перевода книги [5] на английский язык [6] с этими результатами познакомились западные специалисты.

Задача распознавания образов нашла широкое применение в создании различных автоматизированных систем, таких как распознавание изображений (символы, человеческие лица, автомобильные номера, картографические изображения), распознавание сигналов, речи, запаха и др. Эта задача по-прежнему продолжает оставаться активной областью исследования как в силу появления большого числа новых приложений, так и поиска новых более эффективных методов и алгоритмов ее решения. Ей посвящаются многочисленные конференции, по этой тематике выпускаются специализированные журналы, в которых освещаются различные стороны задачи. Среди последних публикаций отметим работы [7—12].

Одним из основных инструментов решения задачи распознавания образов служат многослойные нейронные сети (МНС), выступающие в роли универсального аппроксиматора практически любой дискриминантной функции $g(\mathbf{x})$. Литература по МНС представлена довольно широко (см., например, монографии [6], [13]).

Альтернативой [14] МНС служит ассоциативная нейронная сеть CMAC, спектр применения которой очень широк, в том числе и для задач распознавания образов. Здесь также имеется большое число публикаций. Среди появившихся в последнее время работ назовем статьи [7—9]. К сожалению, в русскоязычной научной литературе число публикаций по нейронной сети CMAC незначительно.

[1] Работа выполнена при поддержке ФАНИ, государственный контракт 02.514.12.4003 от 11 июня 2009 г.

1. Постановка задачи

Цель настоящей работы — анализ применения нейронной сети CMAC и ее модификаций в модельной "довольно сложной" задаче распознавания образов, рассмотренной в монографии [13]. Данная задача является двумерной и имеет невыпуклые области решений. В отличие от рассматриваемой в работе [13] задачи, в которой нулевой класс образов состоит из точек, принадлежащих области C_0, а второй класс образов — из точек, принадлежащих области C_1, в настоящей работе задача усложняется введением третьего класса образов, состоящих из точек, принадлежащих области C_2 (рис. 1).

Генерация образов выполняется следующим образом: последовательно появляются случайные точки $\mathbf{v}[n]$, $n = 1, 2, ...$, равномерно распределенные на прямоугольнике размером 5×4, который состоит из объединения всех областей C_0, C_1 и C_2. Точки $\mathbf{v}[n]$, которые оказываются в области C_0, принадлежат нулевому классу и обозначаются точкой. Точки, которые оказываются в области C_1, принадлежат классу 1 и обозначаются крестиком, а в области C_2 — классу 2 и обозначаются звездочкой. На рис. 2 показана реализация последовательности образов, состоящая из 1000 точек, принадлежащая трем классам 0, 1 и 2. Точки из последовательности образов $\mathbf{v}[n]$, $n = 1, 2, ...$ поступают на нейронную сеть вместе с указаниями учителя $y_c(\mathbf{v})$ о принадлежности точки $\mathbf{v}[n]$ конкретному классу: $y_c(\mathbf{v}[n]) = 0$, $\mathbf{v}[n] \in C_0$, $y_c(\mathbf{v}[n]) = 1$, $\mathbf{v}[n] \in C_1$ и $y_c(\mathbf{v}[n]) = 2$, $\mathbf{v}[n] \in C_2$. В настоящей работе исследуется нейронная сеть CMAC с одним выходом. Поскольку нейронная сеть CMAC мало освещена в отечественной литературе, то для понимания существа проблемы в следующем разделе приведено краткое описание этой нейронной сети и ее модификаций.

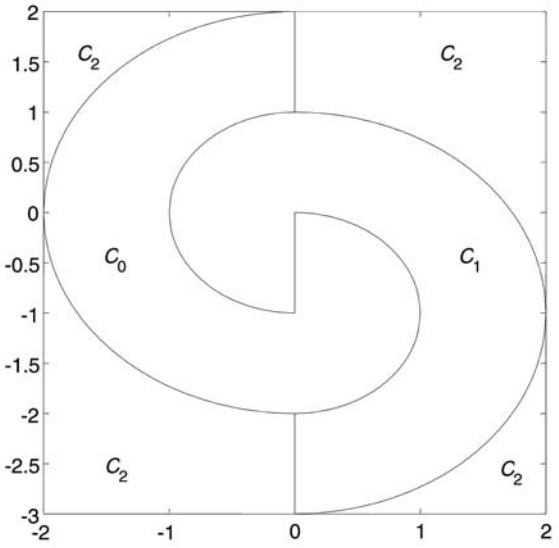

Рис. 1. Области классификации рассматриваемой задачи

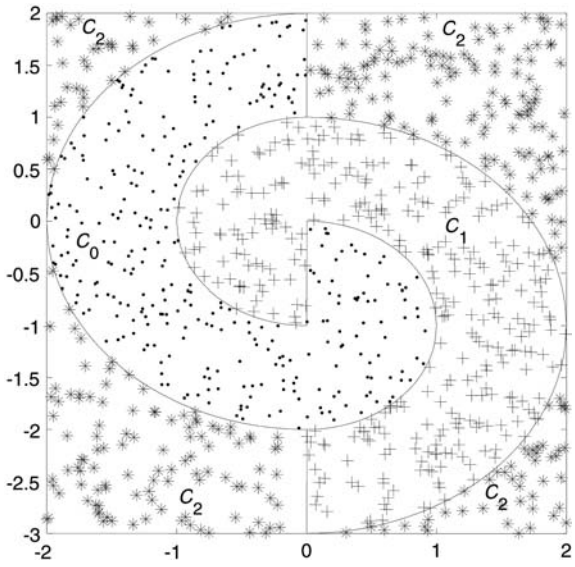

Рис. 2. Реализация тестовой последовательности образов, состоящей из 1000 точек и принадлежащей трем классам 0, 1 и 2

2. Нейронная сеть CMAC: структура

Автором нейронной сети CMAC является американский ученый Дж. Альбус [15, 16]. Название нейронной сети происходит от первых букв ее полного английского названия: *Cerebellar Model Articulation Controller* (мозжечковая модель суставного регулятора). В основу этой нейронной сети положена нейрофизиологическая модель мозжечка для управления роботом-манипулятором.

Исследования, касающиеся различных аспектов CMAC, отражены в монографиях и во многих журнальных статьях. Нейронная сеть CMAC была успешно применена в задачах классификации, управления беспилотным воздушным средством, демпфирования колебаний строительного крана, в задачах подавления и управления вибрацией, построения цифровой модели местности, распознавании образов, в задаче управления движением и во многих других областях. Перечисление различных применений нейронной сети CMAC и библиографических ссылок к ним в силу их большого числа выходит за рамки настоящей работы. Описание CMAC можно найти в статьях [17, 18].

Нейронная сеть CMAC предназначена для запоминания и восстановления функций $y_c(\mathbf{v})$ от N переменных $\mathbf{v}$. Такими функциями могут быть модели нелинейных статических или динамических объектов, модели регуляторов систем управления, разделяющие поверхности в задачах распознавания образов, и другие. Наиболее существенными отличиями CMAC от других нейронных сетей являются следующие.

1. Аргументы запоминаемой и воспроизводимой функции принимают только дискретные значения.

2. Нелинейное преобразование аргументов функции происходит неявно с помощью алгоритма вычисления адресов ячеек ассоциативной памяти, в которых хранятся числа, определяющие значение функции.

Областью определения функции N переменных $y_c(\mathbf{v})$, которую запоминает CMAC, является N-мерный гиперпараллелепипед:

$$\mathbf{V} = \{\mathbf{v}^{(1)} = \overline{v_{\min}^{(1)}, v_{\max}^{(1)}}; \mathbf{v}^{(2)} = \overline{v_{\min}^{(2)}, v_{\max}^{(2)}}; \ldots; \mathbf{v}^{(N)} = \overline{v_{\min}^{(N)}, v_{\max}^{(N)}}\}.$$

Для того чтобы нейронная сеть CMAC запомнила функцию $y_c(\mathbf{v})$, сначала задается число уровней квантования по каждой переменной $x_{\max}^{(i)}$, $i = \overline{1, N}$, затем каждое ребро гиперпараллелепипеда квантуется с постоянным шагом

$$\Delta^{(i)} = \frac{v_{\max}^{(i)} - v_{\min}^{(i)}}{x_{\max}^{(i)}}, \; i = \overline{1, N},$$

и каждому элементу квантования каждой компоненты присваиваются целочисленные номера $x^{(i)} = 1, 2, \ldots, x_{\max}^{(i)}$, $i = \overline{1, N}$. Эти номера связаны со значениями аргументов $v^{(i)}$, $i = \overline{1, N}$, функции $y_c(\mathbf{v})$, задаваемыми в середине каждого интервала, следующим соотношением:

$$v^{(i)} = v_{\min}^{(i)} + \Delta^{(i)}(x^{(i)} - 0{,}5), \; x^{(i)} = 1, 2, \ldots, x_{\max}^{(i)}, \; i = \overline{1, N}.$$

В результате масштабирования, смещения и дискретизации переменных $v^{(i)}$, $i = \overline{1, N}$, вместо функции $y_c(\mathbf{v})$ рассматривается функция $y(\mathbf{x})$ от N переменных $\mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(N)})$, которая определена на целочисленной N-мерной сетке:

$$\mathbf{X} = \{\mathbf{x}^{(1)} = \overline{1, x_{\max}^{(1)}}; x^{(2)} = \overline{1, x_{\max}^{(2)}}, \ldots; \mathbf{x}^{(N)} = \overline{1, x_{\max}^{(N)}}, \quad (1)$$

и которая, естественно, совпадает с исходной функцией в точках дискретизации.

2.1. Алгоритм нелинейного преобразования аргументов, или алгоритм вычисления адресов ассоциативной памяти

В нейронной сети CMAC предполагается, что каждый входной сигнал (аргумент функции) возбуждает, или делает активными, ровно ρ^* ячеек памяти, суммарное содержимое которых равно значению запоминаемой функции. Параметр ρ^* играет очень важную роль, его значение определяет разрешающую способность CMAC и требуемый объем памяти нейронной сети.

Одномерный случай. Каждому значению скалярного аргумента $\mathbf{x}^{(1)} = \overline{1, x_{\max}^{(1)}}$ соответствует ровно ρ^* активных ячеек памяти с номерами $x^{(1)}$, $x^{(1)} + 1$, $x^{(1)} + 2$, ..., $x^{(1)} + \rho^* - 1$, так что максимальному значению аргумента вектора $\mathbf{x}^{(1)} = x_{\max}^{(1)}$ соответствуют ячейки памяти с номерами $x_{\max}^{(1)}$, $x_{\max}^{(1)} + 1$, $x_{\max}^{(1)} + 2$, ..., $x_{\max}^{(1)} + \rho^* - 1$, откуда следует, что число ячеек памяти, необходимых для хранения функции одной переменной в CMAC, равно

$$M^{(1)} = x_{\max}^{(1)} + \rho^* - 1.$$

Номера активных ячеек памяти, соответствующие скалярной переменной $\mathbf{x}^{(1)} = \overline{1, x_{\max}^{(1)}}$, можно представить ρ^*-мерным вектором $\mathbf{m}^{(1)}$, значения компонент которого равны номерам активных ячеек. Алгоритм формирования ρ^* компонент вектора $\mathbf{m}^{(1)}$ номеров активных ячеек для одномерного случая имеет вид

$$m^{(1)}_{[(x_1 \overline{\bmod} \rho^* + i) \overline{\bmod} \rho^*]} = x^{(1)} + i, \; i = \overline{0, \rho^* - 1}, \quad (2)$$

где функция

$$a \overline{\bmod} b = \begin{cases} b, & a \bmod b = 0 \\ a \bmod b, & a \bmod b \neq 0, \end{cases}$$

и $a \bmod b$ — остаток от деления a на b (a и b — целые числа).

Для вычисления компонент вектора $\mathbf{m}^{(1)}$ согласно соотношению (2) достаточно определить номер $x_1 \overline{\bmod} \rho^*$ той компоненты, значение которой равно x_1, т. е вычислить (2) при $i = 0$. Значения следующих компонент равны $x_1 + 1$, $x_1 + 2$ и т. д., при этом оказывается, что после компоненты с номером ρ^* следуют компоненты с номерами 1, 2, ..., $\rho^* - 1$. Например, чтобы запомнить число 22 в восьми ячейках ($\rho^* = 8$), будут активизированы ячейки с номерами $m^{(1)} = (25, 26, 27, 28, 29, 22, 23, 24)$, поскольку $22 \overline{\bmod} 8 = 6$ (число 22 расположено на шестой позиции).

Многомерный случай. В многомерном случае каждой компоненте $x^{(i)}$, $i = \overline{1, N}$ вектора $\mathbf{x}$ так же, как и в одномерном случае, соответствует ρ^*-мерный вектор $\mathbf{m}^{(i)}$, $i = \overline{1, N}$, активных ячеек памяти, которые вычисляются, как это было описано выше в одномерном случае, согласно соотношению (2). В результате этих вычислений N-мерному вектору $\mathbf{x}$ ставится в соответствие промежуточная матрица

активных ячеек памяти $\mathbf{M}$ с элементами $m_k^{(i)}$, $i = \overline{1, N}$, $k = \overline{1, \rho^*}$:

$$\mathbf{x} = \begin{vmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ . \\ x^{(N)} \end{vmatrix} \rightarrow \mathbf{M} = \begin{vmatrix} m_1^{(1)} & m_2^{(1)} & m_3^{(1)} & . & m_{\rho^*}^{(1)} \\ m_1^{(2)} & m_2^{(2)} & m_3^{(2)} & . & m_{\rho^*}^{(2)} \\ m_1^{(3)} & m_2^{(3)} & m_3^{(3)} & . & m_{\rho^*}^{(3)} \\ . & . & . & . & . \\ m_1^{(N)} & m_2^{(N)} & m_3^{(N)} & . & m_{\rho^*}^{(N)} \end{vmatrix}.$$

На следующем шаге матрице $\mathbf{M}$ размера $N \times \rho^*$ ставится в соответствие ρ^*-мерный вектор активных ячеек памяти вектора $\mathbf{x}$. Для этого проводится последовательное слияние элементов каждого k-го столбца матрицы $\mathbf{M}$ в один k-й элемент вектора активных ячеек памяти $\mathbf{m}$, так что этот вектор-столбец приобретает следующий вид:

$$\mathbf{m} = (m_1^{(1)}\, m_1^{(2)}\, m_1^{(3)} \ldots m_1^{(N)};\ m_2^{(1)}\, m_2^{(2)}\, m_2^{(3)} \ldots m_2^{(N)}; \ldots; m_{\rho^*}^{(1)}\, m_{\rho^*}^{(2)}\, m_{\rho^*}^{(3)} \ldots m_{\rho^*}^{(N)})^{\mathrm{T}}, \quad (3)$$

каждый элемент которого

$$m_k = m_k^{(1)}\, m_k^{(2)}\, m_k^{(3)} \ldots m_k^{(N)},\ k = \overline{1, \rho^*}, \quad (4)$$

однозначно определяет номер активной ячейки памяти CMAC. Выражение (4) можно трактовать как позиционную запись числа m_k, старший разряд которого равен $m_k^{(1)}$, а младший разряд — $m_k^{(N)}$. К особенностям чисел, образующих символьную запись числа m_k (4), относится тот факт, что остаток от деления каждого из них на ρ^* равен k.

Алгоритм вычисления числа m_k по значениям $m_k^{(i)}$, $i = \overline{1, N}$, принимает наиболее простую форму, когда максимально возможные значения компонент вектора $\mathbf{x}$ удовлетворяют следующему соотношению:

$$x_{\max}^{(i)} = (\mu^{(i)} - 1)\rho^* + 1,\ i = \overline{1, N},$$

где $\mu^{(i)}$, $i = \overline{1, N}$, — целые числа.

В этом случае алгоритм вычисления m_k имеет следующий вид [17, 19]:

$$m_k = (m_k^{(1)} - 1) \prod_{l=2}^{N} \mu^{(l)} + \mathrm{int}((m_k^{(2)} - 1)/\rho^*) \prod_{l=3}^{N} \mu^{(l)} + \ldots + \mathrm{int}((m_k^{(N-1)} - 1)/\rho^*)\mu^{(N)} + \mathrm{int}((m_k^{(N)} - 1)/\rho^*) + 1,\ k = \overline{1, \rho^*}, \quad (5)$$

где $\mathrm{int}(a/b)$ — функция целочисленного деления a на b.

2.2. Объем памяти

Необходимое число ячеек памяти M нейронной сети CMAC следует из формулы (5), когда все числа $m_k^{(i)}$, $i = \overline{1, N}$, принимают свои максимально возможные значения, т. е. когда $m_k^{(i)} = M^{(i)} = \mu^{(i)}\rho^*$, $i = \overline{1, N}$. В этом случае величина M задается выражением

$$M = \rho^* \prod_{i=1}^{N} \mu^{(i)} = \rho^* \prod_{i=1}^{N} \frac{(x_{\max}^{(i)} + \rho^* - 1)}{\rho^*} = \rho^* \prod_{i=1}^{N} \frac{M^{(i)}}{\rho^*}. \quad (6)$$

2.3. Выход нейронной сети CMAC и ее обучение

Значение запомненной в CMAC функции $\tilde{y}(\mathbf{x})$ равно сумме содержимого активных ячеек памяти. Введем следующие переменные:

$\mathbf{w}[n]$ — M-мерный вектор памяти сети, вычисленный на n-м шаге обучения, каждая j-я компонента которого $w_j[n]$, $j = \overline{1, M}$, соответствует содержимому j-й ячейки памяти CMAC;

$\mathbf{a}(x)$ — M-мерный ассоциативный вектор, однозначно связанный с вектором аргументов $\mathbf{x}$ посредством вектора $\mathbf{m}$ номеров активных ячеек памяти по следующему правилу: элементы вектора $a_j, j = \overline{1, M}$, номера которых совпадают с номерами активных ячеек памяти, равны единице, все остальные элементы вектора $\mathbf{a}$ равны нулю. Тогда в соответствии с правилом функционирования нейронной сети CMAC ее выход $\tilde{y}(\mathbf{x}[n])$ при заданном входном векторе $\mathbf{x}[n]$ равен скалярному произведению векторов $\mathbf{a}(\mathbf{x}[n])$ и $\mathbf{w}[n]$:

$$\tilde{y}(\mathbf{x}[n]) = \mathbf{a}^{\mathrm{T}}(\mathbf{x})\mathbf{w}(n), \quad (7)$$

т. е. выход равен сумме содержимого активных ячеек памяти сети.

Алгоритм обучения нейронной сети CMAC функционирует следующим образом. Пусть после $(n-1)$-го измерения значения запоминаемой функции $y(\mathbf{x})$ и соответствующих значений вектора аргументов $\mathbf{x}$, $(y(\mathbf{x}[i]), \mathbf{x}[i])$, $i = \overline{1, n-1}$, был вычислен вектор памяти $\mathbf{w}[n-1]$. Тогда на следующем n-м шаге после измерения значения функции $y[n] \equiv y(\mathbf{x}[n])$ при известном значении аргумента $\mathbf{x}[n]$ сначала с помощью алгоритма нелинейного преобразования аргументов (5) вычисляются номера активных ячеек памяти, далее вычисляется предсказываемое нейронной сетью значение функции $\tilde{y}[n] \equiv \tilde{y}(\mathbf{x}[n])$, равное сумме содержимого актив-

ных ячеек памяти. Вычисляются ошибка предсказания $\varepsilon[n] = y[n] - \tilde{y}[n]$ и значение коррекции $\Delta w[k] = \varepsilon[n]/\rho^*$, которая прибавляется к содержимому активных ячеек памяти. Неактивные ячейки коррекции не подвергаются. Аналитическая форма алгоритма обучения, полученная в работе [20], имеет следующий вид:

$$\mathbf{w}[n] = \mathbf{w}[n-1] + \frac{y[n] - \mathbf{a}^{\mathrm{T}}(\mathbf{x}[n]\mathbf{w}[n-1])}{\mathbf{a}^{\mathrm{T}}(\mathbf{x}[n])\mathbf{a}(\mathbf{x}[n])}\mathbf{a}(\mathbf{x}[n]), \quad (8)$$

$$w[0] = w_0,\ n = 1, 2, \ldots .$$

3. Двухслойная нейронная сеть CMAC как средство устранения влияния помех и ошибок квантования входных сигналов

Нейронная сеть CMAC не подавляет помех измерений. Она только изменяет их спектральные свойства. В этой связи возникает задача придания фильтрующих свойств этой сети, позволяющих устранить влияние помех измерений. Подход к решению этой задачи был предложен в работе [21]. Он основывается на том факте, что оптимальные оценки вектора памяти $\mathbf{w}^*$ могут быть вычислены осреднением по времени оценок $\mathbf{w}[n]$ (8). Для этого к существующей нейронной сети CMAC добавляется второй слой, предназначенный для вычисления и хранения осредненного вектора памяти. На каждом шаге измерений процедуре усреднения подвергаются содержимое только ρ^* активных ячеек памяти вектора $\mathbf{w}[n]$. В случае стационарной помехи с ограниченной дисперсией с увеличением числа наблюдений усредненные оценки вектора $\tilde{\mathbf{w}}[n]$ стремятся к оптимальному значению вектора памяти $\mathbf{w}^*$. Двухслойную нейронную сеть CMAC будем обозначать как CMAC-TL.

4. Дискретная сеть CMAC для построения непрерывных моделей

Точность оценивания с помощью обученной нейронной сети CMAC определяется двумя факторами: точностью, с какой CMAC запомнила предъявленную ему функцию, и ошибками квантования аргументов, по которым следует оценить значение функции $y_c(\mathbf{v})$.

Как отмечалось в разд. 2, для того чтобы нейронная сеть CMAC запомнила функцию $y_c(\mathbf{v})$, сначала каждая компонента вектора $\mathbf{v}$ масштабируется, смещается и квантуется с постоянным шагом. Каждому элементу квантования каждой компоненты присваиваются целочисленные номера, и таким образом исходной функции непрерывных аргументов ставится в соответствие функция $y(\mathbf{x})$ целочисленных аргументов $x^{(i)} = 1, 2, \ldots, x^{(i)}_{\max}$, $i = \overline{1, N}$. Если теперь требуется, чтобы обученная нейронная сеть CMAC воспроизвела значение функции в точке $\hat{x}_{pnt}$, компоненты которой не являются целыми числами, то в этом случае находится ближайшая к ней точка $\hat{x}$ с целыми координатами, в которой и оценивается значение функции $y_c(\mathbf{v})$. Так возникает ошибка квантования по аргументам функции $y_c(\mathbf{v})$. Уменьшения влияния ошибок квантования можно достичь с помощью локального аналогового аппроксиматора [22].

Вычислим значения функции $\tilde{y}(\mathbf{x})$ (7) во всех точках $\mathbf{x} \in GC_k \subset \mathbf{X}$, где GC_k — множество целочисленных точек, принадлежащих гиперкубу с центром в точке $\hat{\mathbf{x}}$: $|x^{(i)} - \hat{x}| \leqslant k$, $i = \overline{1, N}$. Число таких точек равно $L = (2k + 1)^N$. Например, для двухмерного случая, когда $N = 2$, $k = 1$, число точек $L = 9$, а при $N = 2$, $k = 2$ число точек $L = 25$.

Обозначим через $\mathbf{x}[i]$, $i = \overline{1, L}$, все L точек гиперкуба GC_k, и в тех точках $\mathbf{x}[i]$, $i = \overline{1, L}$, которые принадлежат области определения CMAC, вычислим значения $\tilde{y}(\mathbf{x}[i])$. Набор данных $(\tilde{y}(\mathbf{x}[i]), \mathbf{x}[i])$, $i = \overline{1, L}$, позволяет построить непрерывную линейную регрессионную модель функции $y(\mathbf{x}_{pnt})$, с помощью которой можно оценить значение функции $y(\mathbf{x}_{pnt})$ как в целочисленных точках, так и в точке $\hat{\mathbf{x}}_{pnt}$.

Линейная регрессионная модель функции $y(\mathbf{x}_{pnt})$ имеет вид

$$y_{rgr}(\mathbf{x}_{pnt}) = c_0 + x_1 x^{(1)}_{pnt} + c_2 x^{(2)}_{pnt} + \ldots + c_N x^{(N)}_{pnt} = \mathbf{c}^{\mathrm{T}}\mathbf{x}_{pnt}, \quad (9)$$

где $\mathbf{c} = (c_0, c_1, c_2, \ldots, c_N)^{\mathrm{T}}$; $\mathbf{x}_{pnt} = (1, x^{(1)}_{pnt}, x^{(2)}_{pnt}, \ldots, x^{(N)}_{pnt})^{\mathrm{T}}$ — $(N + 1)$-мерные вектор-столбцы, т — знак транспонирования.

Оптимальное значение $\mathbf{c}^*$ вектора параметров $\mathbf{c}$ модели (9) вычисляется из условия минимума суммы квадратов разностей между измеренными значениями $\tilde{y}(\mathbf{x}[i])$ (выходом CMAC в точках $\mathbf{x}[i]$, $i = \overline{1, L}$) и функцией регрессии в этих точках:

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \left(\sum_{J=1}^{L} (\tilde{y}(\mathbf{x}[j]) - \mathbf{c}^{\mathrm{T}}\mathbf{x}[j])^2 \right). \quad (10)$$

Наиболее удобным инструментом решения задачи (10) является алгоритм метода наименьших квадратов в рекуррентной форме, например, при-

веденный в работе [23], поскольку он позволяет обойти процедуру обращения матрицы **K**:

$$\mathbf{c}[i] = \mathbf{c}[i-1] + \mathbf{K}[i](\tilde{y}(\mathbf{x}[i]) - \mathbf{c}^{\mathrm{T}}[i-1]\mathbf{x}[i]), \quad i = \overline{1, L}, \tag{11}$$

$$\mathbf{K}[i] = \mathbf{K}[i-1] - \frac{\mathbf{K}[i-1]\mathbf{x}[i]\mathbf{x}^{\mathrm{T}}[i]\mathbf{K}[i-1]}{1 + \mathbf{x}^{\mathrm{T}}[i]\mathbf{K}[i-1]\mathbf{x}[i]}. \tag{12}$$

При реализации процедуры (11), (12) те точки $\mathbf{x}[i]$, $i = \overline{1, L}$, которые лежат вне области определения аргументов CMAC, в вычислительной процедуре опускаются. Начальное значение вектора $\mathbf{c}[0]$ — нулевой вектор, начальное значение матрицы усиления $\mathbf{K}[0] = \alpha\mathbf{I}$, где $\mathbf{I}$ — единичная матрица, а константа α — достаточно большое число: $\alpha = (10^4 ... 10^6)$. Вычисленное значение $\mathbf{c}^* = \mathbf{c}[L]$ задает функцию регрессии в виде $y_{rgr}(\mathbf{x}_{pnt}) = \mathbf{x}_{pnt}^{\mathrm{T}}\mathbf{c}^*$ и позволяет получить оценку выхода CMAC в аналоговой точке $\hat{x}_{pnt}$ по формуле

$$y_{regr} = \hat{\mathbf{x}}_{pnt}^{\mathrm{T}}\mathbf{c}^*. \tag{13}$$

Нейронную сеть CMAC с локальным аппроксиматором обозначим как CMAC-LA. При этом, если имеется выбор между нейронной сетью CMAC и CMAC-TL, то локальный аппроксиматор будет применяться к двухслойной сети CMAC-TL как к сети, которая обеспечивает меньшую ошибку аппроксимации.

5. CMAC в задаче распознавания образов

При поступлении на вход нейронной сети CMAC последовательности образов с информацией об их принадлежности конкретным классам происходит восстановление дискриминантной функции $g(\mathbf{x})$. В рассматриваемой нами задаче эта функция принимает три значения: 0 в области C_0, 1 в области C_1 и 2 в области C_2. Дискриминантная функция $g(\mathbf{x})$ в CMAC даже при бесконечном числе предъявленных образов не может быть точно восстановлена в силу отмеченных ранее ряда причин. Основная причина невозможности точного восстановления дискриминантной функции заключается в том, что дискриминантная функция $g(\mathbf{x})$ на границах областей C_0, C_1 и C_2 имеет разрывы первого рода. Другими причинами отмеченного факта являются ошибки квантования аргументов и возможные ошибки о принадлежности образов конкретным классам. Следовательно, восстановленная дискриминантная функция $\tilde{g}(\mathbf{x})$ принимает значения, которые в общем случае не совпадают со значениями дискриминантной функции $g(\mathbf{x})$. Поэтому выход $\tilde{y}(\mathbf{x}[n]) = \mathbf{a}^{\mathrm{T}}(\mathbf{x})\mathbf{w}(n)$ (7) нейронной сети CMAC должен быть преобразован с помощью пороговой функции, чтобы принять значения из области значений дискриминантной функции $g(x)$. Пороговая функция, выполняющая преобразование над выходом $\tilde{y}$ нейронной сети CMAC, имеет вид

$$\psi(\tilde{y}) = \begin{cases} 0, & \tilde{y} \leqslant 0{,}5 \\ 1, & 0{,}5 < \tilde{y} < 1{,}5 \\ 2, & \tilde{y} \geqslant 1{,}5 \end{cases}.$$

5.1. Результаты компьютерного моделирования при отсутствии неправильно предъявленных образов

Модели нейронной сети CMAC и ее модификации реализованы в среде программирования Delphi на языке Object Pascal. Программы, реализующие эти модели, позволяют всесторонне исследовать различные свойства нейронных сетей — скорость и точность обучения, влияние помех измерений и квантования на точность обучения и многое другое. В частности, с помощью этих программ проведен анализ результатов применения нейронной сети CMAC и ее модификаций в задаче распознавания образов.

Обучение нейронной сети CMAC и двухслойной сети CMAC-TL проводилось с помощью обучающей выборки, способ генерации которой описан в разд. 1. Аналогичным способом организована генерация тестирующей выборки, по которой оценивается качество распознавания.

Число точек квантования по каждой компоненте входного вектора $\mathbf{v}$ равно $x_{\max}^{(i)} = 257$, $i = \overline{1, 2}$. Отметим, что число точек сетки (1) CMAC, т. е. число возможных различных входных векторов CMAC, равно $257^2 = 66049$.

Оценка качества обученной нейронной сети приводится для трех ее модификаций: классическая нейронная сеть CMAC; двухслойная нейронная сеть CMAC-TL; двухслойная нейронная сеть CMAC с аппроксиматором — CMAC-LA. Критерий качества обученной сети — процент относительной ошибки распознавания как отношение числа неправильно распознанных образов N_{err} к общему числу N_{tst} тестирующих образов, умноженное на 100: $E_{err} = (N_{err}/N_{tst})100\ \%$, для заданной тестирующей выборки.

В таблицах 1—3 приведены значения критерия качества для трех модификаций нейронной сети CMAC и число ячеек памяти M (6) при различных значениях обобщающего параметра ρ^* и длины обу-

чающей выборки N_{lern} = 1000, N_{lern} = 3000 и N_{lern} = 30 000, соответственно.

Анализ данных, приведенных в табл. 1—3, позволяет сделать следующие выводы.

1. Нейронная сеть CMAC обучается быстро. При длине обучающей выборки N_{lern} = 1000 образов наименьшая ошибка составила 9,8 % для CMAC-TL при значении обобщающего параметра ρ* = 32. Заметим еще раз, что число возможных различных входных векторов CMAC равно 257^2 = 66 049, т. е. при обучении было предъявлено только 1,5 % возможных образов.

2. С увеличением длины обучающей выборки растет точность распознавания CMAC при меньших значениях обобщающего параметра ρ* (см. выделенные серым цветом столбцы в таблицах). При N_{lern} = 3000 и ρ* = 16 минимальная ошибка распознавания составила 6,4 % для CMAC-LA, а при N_{lern} = 30 000 (менее половины возможных образов) и ρ* = 8 минимальная ошибка распознавания составила 2,0 % для CMAC-LA.

3. Сравнение CMAC, CMAC-TL и CMAC-LA с точки зрения точности распознавания при отсутствии неправильно предъявленных образов показывает, что CMAC-TL и CMAC-LA более эффективны, чем CMAC, однако значительного преимущества CMAC-LA перед CMAC-TL в задаче распознавания при отсутствии ошибок учителя не наблюдается.

4. Точность распознавания сильно зависит от значения обобщающего параметра ρ*.

Увеличение длины обучающей выборки улучшает точность распознавания при все меньших значениях обобщающего параметра ρ*, однако ошибка распознавания принципиально не может достигнуть нулевого значения. Так, при длине обучающей последовательности N_{lern} = 3 000 000 и ρ* = 2 ошибка распознавания для CMAC-LA равна 0,81 % — значение, близкое к предельно возможному минимальному значению ошибки распознавания.

Для иллюстрации природы ошибок распознавания на рис. 3 показано распознавание нейронной сетью CMAC-LA тестовой последовательности образов, состоящей из 1000 точек. Обучение происходило при предъявлении N_{lern} = 3000 образов, значение обобщающего параметра ρ* = 16. Ошибка распознавания E_{err} = 6,4 %. На этом рисунке хорошо видно, что ошибки распознавания возникают вдоль границ областей C_0, C_1 и C_2.

Таблица 1

Критерий качества распознавания, %. Длина обучающей выборки N_{lern} = 1000, длина тестирующей выборки N_{tst} = 1000

Модификация сети	ρ* = 2 M = 33 282	ρ* = 4 M = 16 900	ρ* = 8 M = 8712	ρ* = 16 M = 4624	ρ* = 32 M = 2592	ρ* = 64 M = 1600
CMAC	65,0	60,4	39,5	12,4	11,7	26,9
CMAC-TL	65,0	60,4	39,8	11,6	9,8	15,7
CMAC-LA	69,7	62,6	39,3	11,6	10,2	15,5

Таблица 2

Критерий качества распознавания, %. Длина обучающей выборки N_{lern} = 3000, длина тестирующей выборки N_{tst} = 1000

Модификация сети	ρ* = 2 M = 33 282	ρ* = 4 M = 16 900	ρ* = 8 M = 8712	ρ* = 16 M = 4624	ρ* = 32 M = 2592	ρ* = 64 M = 1600
CMAC	60,7	44,0	12,5	6,7	9,6	21,4
CMAC-TL	60,7	44,2	13,1	6,7	7,0	15,2
CMAC-LA	67,1	44,4	13,0	6,4	6,4	15,6

Таблица 3

Критерий качества распознавания, %. Длина обучающей выборки N_{lern} = 30 000, длина тестирующей выборки N_{tst} = 1000

Модификация сети	ρ* = 2 M = 33 282	ρ* = 4 M = 16 900	ρ* = 8 M = 8712	ρ* = 16 M = 4624	ρ* = 32 M = 2592	ρ* = 64 M = 1600
CMAC	20,3	2,6	3,1	5,0	9,6	19,7
CMAC-TL	20,1	2,8	2,3	2,9	5,8	13,5
CMAC-LA	14,4	2,2	2,0	2,9	6,2	13,3

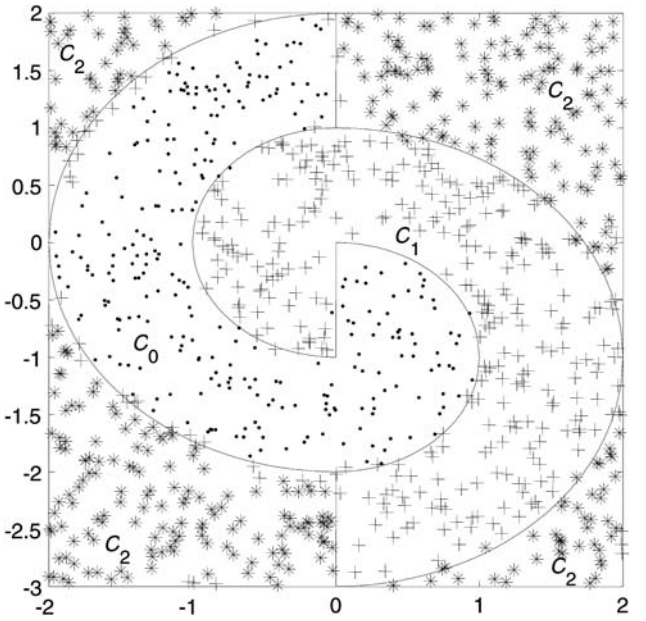

Рис. 3. Распознавание тестовой последовательности образов, состоящей из 1000 точек, нейронной сетью CMAC-LA, обученной по N_{lern} = 3000 образам при ρ* = 16. Ошибка распознавания E_{err} = 6,4 % (данные из табл. 2)

5.2. Результаты компьютерного моделирования при наличии неправильно предъявленных образов

Предположим, что в процессе обучения информация о принадлежности образа тому или иному классу может быть неточной, т. е. учитель может ошибаться, неправильно называя номер класса, к которому относится предъявляемый образ. В описываемых далее экспериментах предполагается, что ошибка учителя вероятностная: он правильно объявляет предъявленный образ с вероятностью p_{in}, а с равной вероятностью $p_{out} = 0{,}5(1 - p_{in})$ ошибочно вместо истинного называет тот или иной образ.

В табл. 4—7 приведены значения критерия качества для трех модификаций нейронной сети CMAC при различных значениях обобщающего параметра ρ^* и длины обучающей выборки $N_{lern} = 1000$, $N_{lern} = 3000$, $N_{lern} = 30\,000$ и $N_{lern} = 300\,000$, соответственно, когда вероятность ошибки учителя равна $p_{err} = (1 - p_{in}) = 0{,}1$. Число ошибочно показанных образов в данном эксперименте при длине обучающей выборки $N_{lern} = 1000$, $N_{lern} = 3000$, $N_{lern} = 30\,000$ и $N_{lern} = 300\,000$, оказалось равным, соответственно, 84, 266, 3058 и 29 370.

Данные, приведенные в табл. 4—7, показывают, что классическая нейронная сеть CMAC не обладает фильтрующими свойствами, т. е. не подавляет влияние помехи измерений, тогда как модифицированные сети CMAC-TL и CMAC-LA явно имеют эти свойства, при этом точность сети CMAC-LA выше точности сети CMAC-TL и, конечно, выше точности классической сети CMAC. Из этих данных также следует, что точность распознавания растет с увеличением длины обучающей выборки. В подтверждение этого вывода приведем следующие данные: ошибки распознавания для CMAC-LA, когда длина обучающей выборки составила $N_{lern} = 3\,000\,000$, $p_{err} = 0{,}1$ и $\rho^* = 2$, составила 1,0 %, т. е. модифицированная сеть CMAC-LA практически полностью исключила ошибки учителя.

Таблица 4

Критерий качества распознавания, %. Длина обучающей выборки $N_{lern} = 1000$, $p_{err} = 0{,}1$; длина тестирующей выборки $N_{tst} = 1000$

Модификация сети	$\rho^* = 2$ $M = 33\,282$	$\rho^* = 4$ $M = 16\,900$	$\rho^* = 8$ $M = 8712$	$\rho^* = 16$ $M = 4624$	$\rho^* = 32$ $M = 2592$	$\rho^* = 64$ $M = 1600$
CMAC	65,8	61,5	43,5	20,1	19,8	28,5
CMAC-TL	65,8	61,5	43,4	17,3	15,6	19,5
CMAC-LA	69,7	63,1	43,0	16,8	15,3	19,8

Таблица 5

Критерий качества распознавания, %. Длина обучающей выборки $N_{lern} = 3000$, $p_{err} = 0{,}1$; длина тестирующей выборки $N_{tst} = 1000$

Модификация сети	$\rho^* = 2$ $M = 33\,282$	$\rho^* = 4$ $M = 16\,900$	$\rho^* = 8$ $M = 8712$	$\rho^* = 16$ $M = 4624$	$\rho^* = 32$ $M = 2592$	$\rho^* = 64$ $M = 1600$
CMAC	62,0	48,5	21,9	18,3	24,4	38,9
CMAC-TL	62,0	48,4	20,4	11,6	11,0	18,2
CMAC-LA	67,6	47,9	19,0	11,3	10,8	18,4

Таблица 6

Критерий качества распознавания, %. Длина обучающей выборки $N_{lern} = 30\,000$, $p_{err} = 0{,}1$; длина тестирующей выборки $N_{tst} = 1000$

Модификация сети	$\rho^* = 2$ $M = 33\,282$	$\rho^* = 4$ $M = 16\,900$	$\rho^* = 8$ $M = 8712$	$\rho^* = 16$ $M = 4624$	$\rho^* = 32$ $M = 2592$	$\rho^* = 64$ $M = 1600$
CMAC	30,4	18,8	20,0	24,2	29,3	31,0
CMAC-TL	29,2	10,8	6,1	3,9	8,0	16,9
CMAC-LA	23,5	6,8	3,8	3,9	7,9	16,5

Таблица 7

Критерий качества распознавания, %. Длина обучающей выборки $N_{lern} = 300\,000$, $p_{err} = 0{,}1$; длина тестирующей выборки $N_{tst} = 1000$

Модификация сети	$\rho^* = 2$ $M = 33\,282$	$\rho^* = 4$ $M = 16\,900$	$\rho^* = 8$ $M = 8712$	$\rho^* = 16$ $M = 4624$	$\rho^* = 32$ $M = 2592$	$\rho^* = 64$ $M = 1600$
CMAC	16,7	17,8	20,1	25,2	27,2	37,4
CMAC-TL	3,8	2,1	2,6	4,6	7,9	14,3
CMAC-LA	1,6	1,8	2,1	3,8	7,6	14,0

Заключение

Результаты машинных экспериментов показывают, что классическая нейронная сеть CMAC, дополненная вторым сглаживающим слоем и локальным аппроксиматором, эффективно решает задачу распознавания образов в случаях, когда в указаниях учителя отсутствуют и присутствуют неверные указания. Эти эксперименты еще раз подтверждают сделанный ранее вывод о том, что классическая нейронная сеть CMAC не обладает фильтрующими свойствами.

Список литературы

1. **Нильсон Н.** Обучающиеся машины. М.: Мир, 1967 (английское издание — 1965 г.).
2. **Розенблатт Ф.** Принципы нейродинамики. М.: Мир, 1967 (английское издание — 1962 г.).
3. **Rumelhart D. E., Hinton G. E., Williams R. J.** Learning internal representations by error propagation // Parallel Distributed Processing. Cambridge. MA: MIT Press. 1986. V. 1, Ch. 8. P. 318—362.
4. **Галушкин А. И.** Об алгоритмах адаптации в многослойных системах распознавания образов // Доклады АН УССР. 1973. № 1, А. С. 15—21, 91.
5. **Галушкин А. И.** Синтез многослойных систем распознавания образов. М.: Энергия, 1974.

6. **Galushkin A. I.** Neural Network Theory. Berlin, Heidelberg: Springer, 2007.
7. **Lin J. S., Huang S. Y., Liu S. H.** Character Recognition Based on the CMAC with An Annealed Chaotic Learning // Asian Journal of Health and Information Sciences. 2007. Vol. 2, N 1—4. P. 66—78.
8. **Li Y., Hong-Li Yuan H. L., Li Y. Z.** A new flatness pattern recognition model based on CA-CMAC network // International Conference on Machine Learning and Cybernetics. 2009. Vol. 1. P. 520—525.
9. **Bucak İ. Ö., Karlik B.** Hazardous Odor Recognition by CMAC Based Neural Networks // Sensors. 2009. N 9. P. 7308—7319.
10. **Персианцев И. Г.** Адаптивное построение иерархических нейросетевых систем для классификации и для сегментации временных рядов // Нейроинформатика-2010. XII Всероссийская научно-техническая конференция. Лекции по нейроинформатаке. Москва, 2010. С. 212—242.
11. **Скругин В. И., Трофимов А. Г., Роик А. О., Наумов Р. А.** Алгоритм классификации сигналов ЭЭГ на основе анализа в частотно-временной области // Нейроинформатика-2010. XII Всероссийская научно-техническая конференция. Сб. научных трудов. Москва, 2010. Часть 1. С. 266—276.
12. **Дудкин А. А.** Нейросетевое распознавание объектов на изображениях топологических слоев интегральных микросхем // Нейроинформатика-2010. XII Всероссийская научно-техническая конференция. Сб. научных трудов. Москва, 2010. Часть 2. С. 143—153.
13. **Хайкин С.** Нейронные сети: полный курс. 2-е изд. / Пер. с англ. М.: Вильямс, 2006. 1103 с. (Параграф 7.5. Компьютерный эксперимент 2. С. 474).
14. **Miller W. T., Glanz F. H., Kraft L. G.** An associative neural network alternative to backpropagation // Proc. of the IEEE. 1990. V. 79. N 10. P. 1561—1567.
15. **Albus J. S.** A new approach to manipulator control: the cerebellar model articulation controller // ASME Trans., J. Dynamic Systems, Measurement and Control. 1975. V. 97, № 3. P. 220—227.
16. **Albus J. S.** Data storage in the cerebellar model articulation controller (CMAC) // ASME Trans., J. Dynamic Systems, Measurement and Control. 1975. V. 97, № 3. P. 228—233.
17. **Аведьян Э. Д.** Ассоциативная нейронная сеть CMAC. Часть I. Структура, объем памяти, обучение и базисные функции // Информационные технологии. 1997. № 5. С. 6—14.
18. **Аведьян Э. Д.** Ассоциативная нейронная сеть CMAC. Часть II. Процессы обучения, ускоренное обучение, влияние помех, устранение влияния помех в двухслойной сети // Информационные технологии. 1997. № 6. С. 16—27.
19. **Аведьян Э. Д.** Алгоритм вычисления номеров ячеек ассоциативной памяти нейронной сети CMAC // Информатизация и связь. Специальный выпуск'2008. Центр информационных технологий и систем органов исполнительной власти — ЦИТиС. 2009. № 1. С. 103—110.
20. **Милицер Ю., Паркс П. С.** Свойства сходимости ассоциативной памяти в обучающихся системах управления // Автоматика и Телемеханика. 1989. № 2. С. 158—184.
21. **Aved'yan E. D.** The cerebellar model articulation controller (CMAC) for identification of stationary plants under random disturbances // Preprints of the DYCOMANS WORKSHOP II, Algarve, Portugal. 1996. P. 97—102.
22. **Аведьян Э. Д.** Дискретная ассоциативная нейронная сеть CMAC для построения непрерывных моделей // Нейрокомпьютеры — разработка применение. 2009. № 1. С. 53—63.
23. **Цыпкин Я. З.** Адаптация и обучение в автоматических системах. М.: Наука, 1968.

УДК 004.8.032.26

В. Н. Вичугов, канд. техн. наук, доц.,
Томский политехнический университет,
e-mail: vlad@aics.ru

Алгоритм настройки радиально-базисной нейронной сети

Приведена структура радиально-базисной нейронной сети. Определены недостатки классического градиентного алгоритма обучения нейронной сети в задачах идентификации объектов управления. Предложен модифицированный градиентный алгоритм обучения, позволяющий устранить недостатки классического алгоритма. Показан пример применения модифицированного алгоритма в задаче аппроксимации двумерной функции.

Ключевые слова: *искусственная нейронная сеть, радиально-базисная нейронная сеть, алгоритм обучения, идентификация*

При использовании нейросетевых методов в задачах автоматического управления часто возникает необходимость построения нейросетевой модели объекта управления на основе полученных входных и выходных сигналов в реальном времени. Использование многослойных перцептронов для построения нейросетевой модели является затруднительным в связи с тем, что дополнительное обучение многослойного перцептрона в некотором участке рабочей области приводит к потере обученного состояния во всей рабочей области нейронной сети, что не позволяет использовать этот тип нейронных сетей в задачах реального времени. Указанный недостаток отсутствует в радиально-базисных нейронных сетях (РБНС), так как каждый их элемент влияет на значение выходного сигнала преимущественно только в ограниченном участке рабочей области, который характеризуется положением центра элемента и параметром σ, называемым шириной радиальной функции. Чем больше значение параметра σ, тем больше размер области, на которую оказывает влияние данный элемент.

РБНС состоит из двух слоев (рис. 1) [1, 2]. Входные сигналы поступают на элементы первого слоя без изменений.

На рисунке использованы обозначения: n — число элементов в первом слое; $x_1, x_2, ..., x_n$ — входные сигналы; m — число элементов во втором слое; $c_{i1}, c_{i2}, ..., c_{in}$ — координаты центра i-го элемента; σ_i — ширина радиальной функции i-го элемента;

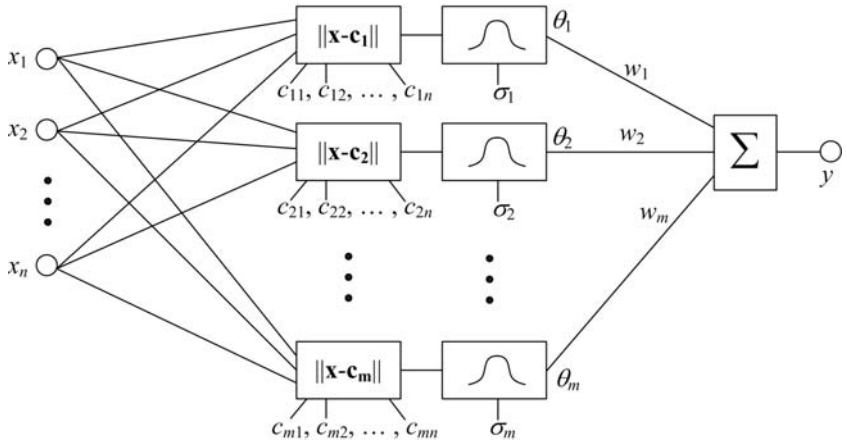

Рис. 1. Структура радиально-базисной нейронной сети

θ_i —выходной сигнал i-го элемента; w_i — весовой коэффициент выходной связи i-го элемента; y — выходной сигнал РБНС.

Выходной сигнал каждого элемента определяется функцией Гаусса [3]

$$\theta_i = \exp\left(-\frac{\sum_{j=1}^{n}(x_j - c_{ij})^2}{2\sigma_i^2}\right).$$

Выходной сигнал РБНС вычисляется как взвешенная сумма сигналов элементов:

$$y = \sum_{i=1}^{m} w_i \theta_i.$$

Для обучения РБНС используется градиентный алгоритм, основанный на минимизации целевой функции ошибки сети. В соответствии с этим алгоритмом для каждого элемента вычисляются величины изменений весового коэффициента Δw_i, ширины элемента $\Delta\sigma_i$ и координат центра элемента Δc_{ij}.

В результате проведенных экспериментов были выявлены некоторые недостатки классического градиентного алгоритма обучения РБНС:

1. В алгоритме обучения РБНС нет правил для первоначального задания числа элементов сети и их параметров, а также нет правил для изменения числа элементов в процессе обучения. Равномерное распределение элементов в рабочей области не всегда является оптимальным. Также может возникнуть ситуация, когда число элементов, заданное первоначально, является недостаточным для достижения требуемого качества обучения.

2. В процессе обучения изменяются параметры всех элементов сети. В результате при увеличении числа элементов вычислительные затраты на обучение также увеличиваются.

3. РБНС не может достичь устойчивого состояния в процессе обучения в случаях, когда существуют элементы с близкими значениями параметров c_{ij} и σ_i. Появление подобных ситуаций во многом зависит от выбранного числа элементов и их начальных параметров. Причина ухудшения качества обучения заключается в том, что в градиентном алгоритме предполагается, что на выходное значение РБНС в каждой точке рабочей области в основном влияет только один элемент. При наличии нескольких элементов в одном участке рабочей области изменение их параметров в соответствии с градиентным алгоритмом не всегда приводит к уменьшению ошибки обучения.

В целях определения ситуаций, когда параметры некоторых элементов становятся близкими друг к другу, было введено понятие коэффициента взаимного пересечения элементов. Для вычисления этого коэффициента для некоторого элемента РБНС необходимо найти второй элемент, центр которого расположен ближе всего к центру рассматриваемого элемента. Значение коэффициента взаимного пересечения определяется как сумма выходной величины текущего элемента в центре второго элемента и выходной величины второго элемента в центре текущего элемента:

$$\rho_i = \exp\left(-\frac{\sum_{j=1}^{n}(c_{ij} - c_{dj})^2}{2\sigma_i^2}\right) + \exp\left(-\frac{\sum_{j=1}^{n}(c_{ij} - c_{dj})^2}{2\sigma_d^2}\right),$$

где i — номер элемента, для которого вычисляется значение коэффициента взаимного пересечения; d — номер элемента, центр которого расположен ближе всего к центру элемента с номером i. Номер элемента d определяется по формуле

$$d = \arg\min_k \sqrt{\sum_{j=1}^{n}(c_{ij} - c_{kj})^2}.$$

Значение коэффициента взаимного пересечения находится в интервале (0; 2]. Коэффициент принимает максимальное значение в том случае, когда центры рассматриваемых элементов совпадают. В ходе экспериментов по аппроксимации различных функций с помощью РБНС было определено, что ошибка РБНС начинает увеличиваться в том случае, когда значение коэффициента взаимного пересечения превышает 1,9. Поэтому для достижения максимального качества обучения РБНС необходимо ограничить максимальное значение коэффициента взаимного пересечения величиной 1,9.

В целях исключения недостатков классического градиентного алгоритма обучения РБНС был разра-

ботан модифицированный градиентный алгоритм, блок-схема которого показана на рис. 2. Блоки, которые отсутствуют в классическом алгоритме, отмечены звездочками. Основные отличия от классического алгоритма заключаются в следующем:

- добавлены правила изменения структуры РБНС в процессе обучения (блок 2). В начале обучения РБНС не содержит элементов. По мере необходимости новые элементы добавляются, а неиспользуемые элементы удаляются;
- уменьшены вычислительные затраты, требуемые для каждого цикла обучения. Это достигается изменением параметров не всех элементов, как в классическом алгоритме, а только элементов, выходная величина которых в рассматриваемой точке больше величины $\theta_{изм}$ (блоки 4 и 5);
- исключена возможность возникновения ситуации, когда параметры некоторых элементов практически совпадают. Для этого вычисленные величины Δc_{ij} и $\Delta\sigma_i$ уменьшаются, если коэффициент взаимного пересечения элементов превышает пороговую величину $\rho_{гр}$, равную 1,9 (блоки 7, 8, 12, 13).

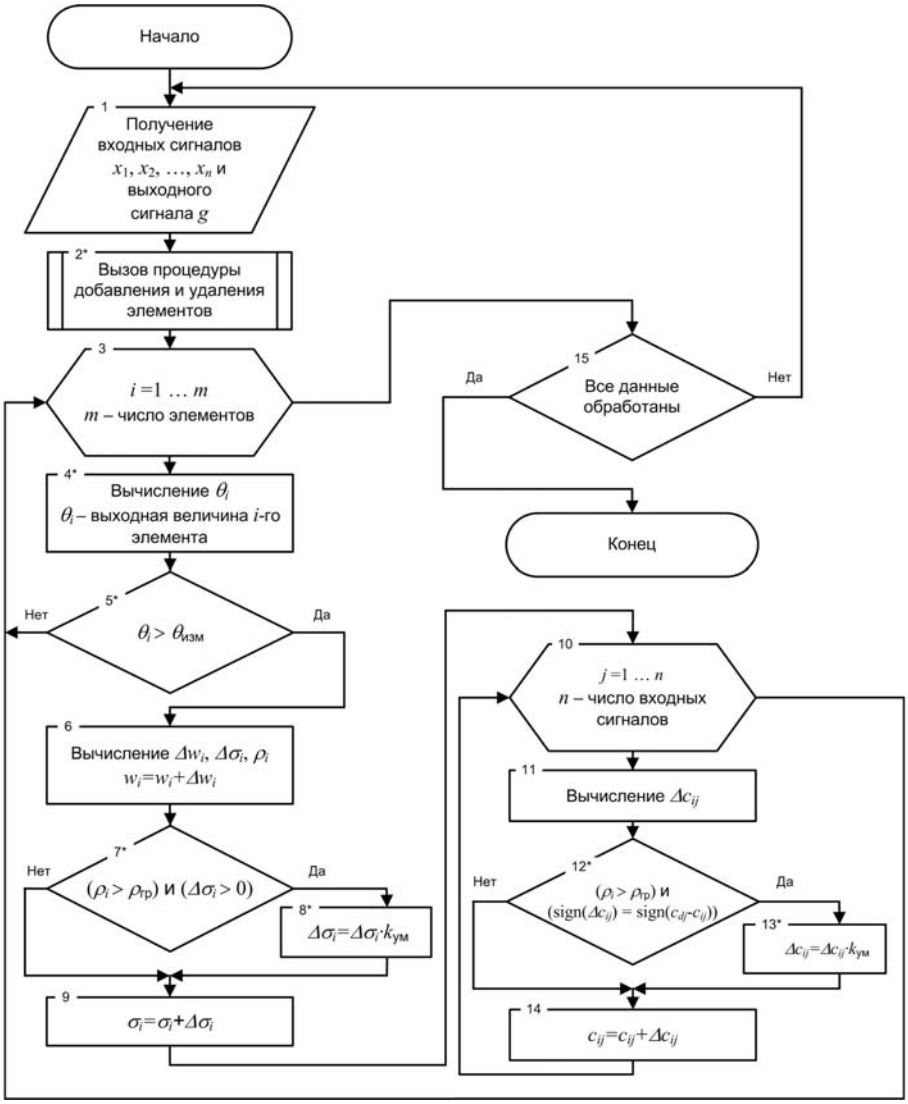

Рис. 2. Блок-схема модифицированного градиентного алгоритма обучения РБНС

Изменение структуры РБНС за счет добавления или удаления элементов приводит к изменению выходного значения РБНС только в окрестности центра добавляемого или удаляемого элемента, а не во всей рабочей области, как в случае с изменением структуры многослойного перцептрона. Поэтому добавление и удаление элементов РБНС возможно осуществлять в процессе обучения без необходимости запуска процесса обучения с самого начала.

Рассмотрим пример аппроксимации двумерной функции

$$f(x_1, x_2) = \sin\left(\frac{x_1^2}{2} - \frac{x_2^2}{4} + 3\right)\cos(2x_1 + 1 - \exp(-x_2))$$

на участке $x_1 \in [-1; 1]$, $x_2 \in [-1; 1]$ с помощью РБНС. Поверхность данной функции показана на рис. 3. При использовании классического градиентного алгоритма перед началом обучения была

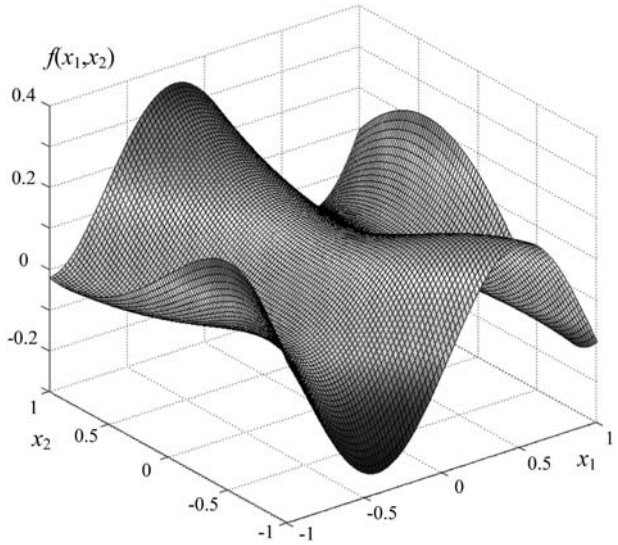

Рис. 3. Поверхность функции $f(x_1, x_2)$

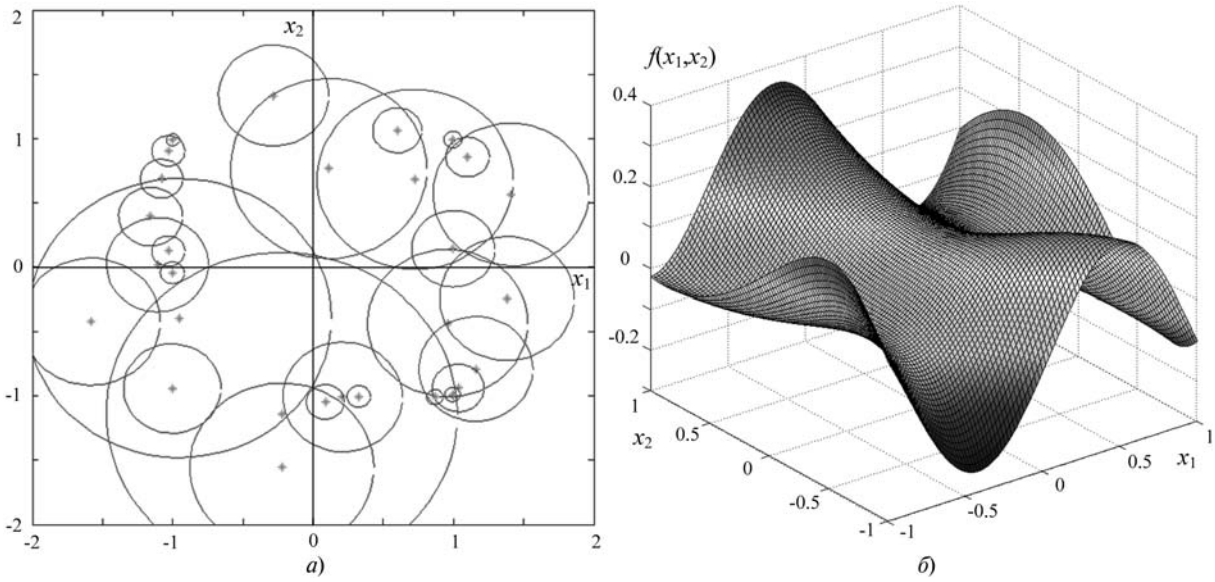

Рис. 4. Результат аппроксимации функции $f(x_1, x_2)$:
а — распределение элементов; *б* — поверхность, показывающая зависимость выхода РБНС от входных значений

задана структура РБНС в виде 36 элементов с начальной шириной $\sigma_0 = 0{,}2$, равномерно распределенных в рабочей области. После приблизительно 10^6 циклов обучения среднеквадратическая ошибка обучения перестала уменьшаться и достигла значения $1{,}554 \cdot 10^{-3}$.

При использовании модифицированного градиентного алгоритма структура РБНС была определена автоматически в процессе обучения. После приблизительно трех миллионов циклов обучения число элементов увеличилось до 30, а среднеквадратическая ошибка обучения составила $1{,}225 \cdot 10^{-3}$. Результаты обучения РБНС показаны на рис. 4.

Отсюда следует, что даже при меньшем числе элементов модифицированный градиентный алгоритм позволяет достичь меньшей ошибки обучения по сравнению с классическим градиентным алгоритмом за счет динамического формирования структуры нейронной сети, но при этом требуется большее количество вычислительных ресурсов. Добавление новых элементов происходит только в те участки, которые характеризуются максимальной ошибкой аппроксимации, что приводит к уменьшению ошибки обучения при меньшем числе элементов по сравнению с классическим алгоритмом обучения.

Выводы

- На основе анализа классического градиентного алгоритма обучения радиально-базисных нейронных сетей разработан модифицированный алгоритм, позволяющий изменять структуру сети в процессе обучения.
- Для исключения ситуаций, когда параметры элементов становятся близкими друг к другу, введен коэффициент взаимного пересечения элементов.
- Экспериментально показано, что модифицированный алгоритм обучения сети позволяет автоматически формировать ее структуру в виде числа элементов внутреннего слоя и их параметров.

Список литературы

1. **Хайкин С.** Нейронные сети: полный курс. 2-е изд. / Под ред. Н. Н. Куссуль. М.: Издат. дом "Вильямс", 2006. 1104 с.

2. **Осовский С.** Нейронные сети для обработки информации: Пер. с польск. М.: Финансы и статистика, 2002. 344 с.

3. **Jianyu L., Luo Siwei, Qi Yingjiana, Huang Yapinga.** Numerical solution of elliptic partial differential equation using radial basis function neural networks // Neural Networks. 2003. № 5/6. P. 729—734.

УДК 004.896, 339.18

В. М. Гриняк, канд. техн. наук, доц.,
e-mail: Viktor.Grinyak@vvsu.ru,
И. С. Можаровский, аспирант,
e-mail: Igor.Mojarovsky@vvsu.ru,
К. И. Дегтярев, аспирант,
e-mail: Konstantin.Degtyarev@vvsu.ru,
Владивостокский государственный
университет экономики и сервиса

Нейросетевая модель планирования сезонных продаж

Статья посвящена проблеме планирования продаж в современных системах управления торговлей. Рассматриваемая в работе модельная интерпретация задачи планирования продаж сезонных товаров основана на методах классификации нейронными сетями Кохонена.

Продемонстрированные результаты применения модели подтверждают ее процедурную разрешимость и эффективность.

Ключевые слова: *планирование продаж, нейронная сеть, классификация, сезонные товары, система 1С: Предприятие 8*

Введение

Эффективная деятельность современных торговых и производственных предприятий возможна только при высокой степени их информатизации. Специализированные программные комплексы — корпоративные информационные системы (КИС) — выступают инструментом такой информатизации. Их основной задачей является создание информационной базы для принятия управленческих решений руководством предприятий, при этом содержание и форма представления информации должны обеспечивать принятие как можно меньшего числа ошибочных решений [1].

Планирование закупок и продаж является важнейшей составляющей работы крупных компаний в условиях, когда номенклатура материалов и товаров исчисляется десятками и даже сотнями тысяч наименований. Современные КИС (например, российские "Галактика", "1С", зарубежные "SAP", "Axapta" и др.) реализуют самые разнообразные подходы к автоматизации планирования: от простейших до весьма продвинутых (ERP, MRP). При этом с алгоритмической точки зрения все современные технологии планирования основаны, по сути, на ретроспективном анализе данных и экстраполяции на предстоящие периоды с использованием методов регрессии и статистики [2].

Планирование продаж сезонных товаров представляет собой особенно важную и сложную задачу. При работе с такими товарами информационная система должна автоматически их идентифицировать и применять к ним специальные алгоритмы анализа с учетом специфических параметров сезонности — периодичности продаж и графика распределения продаж по периодам. Эти параметры должны быть выбраны таким образом, чтобы обеспечивать максимальную достоверность планирования, однако все современные КИС (в их стандартной конфигурации) если и реализуют возможности работы с сезонными товарами, то предоставляют пользователю выбирать параметры сезонности исключительно интуитивно, "вручную"; при большом списке номенклатуры их корректный выбор, таким образом, может быть неосуществим.

В настоящей статье рассматривается модель задачи планирования продаж сезонных товаров, в основе которой лежат представления нейронных сетей Кохонена. Модель позволяет автоматизированно идентифицировать сезонные товары и качественно оценивать их параметры сезонности.

Основные модельные представления

В задачах экономического анализа минимальным периодом оценивания какого-либо показателя являются, как правило, сутки, соотнесенные с конкретной датой. Однако при планировании чаще всего используются не сутки, а "укрупненные" периоды: неделя, декада, месяц, квартал и т. д.

Пусть X_k — значение выбранного показателя (им может быть, например, число продаж, выручка от продаж и т. п.) в период с номером k. Модель изменения значения показателя X во времени может быть выражена формулой

$$X_k = G(k) + \eta(k), \qquad (1)$$

где $G(k)$ — функция, выражающая детерминированный закон эволюции величины X (тренд); $\eta(k)$ — случайная величина, характеризующая отклонение фактического значения показателя от его тренда (здесь и далее будем считать, что $\eta(k)$ — некоррелированная случайная величина с нулевым математическим ожиданием). При решении задачи планирования известная функция $G(k)$ используется для экстраполяции значений показателя X, а свойства $\eta(k)$ ложатся в основу оценки достоверности планирования (например, выраженной в форме доверительных интервалов) [3].

Примем, что тренд $G(k)$ есть функция, представляемая как

$$G(k) = Ag(k), \qquad (2)$$

где $g(k)$ есть периодическая функция с периодом J, так что $g(k) = g(k + J)$, $\sum_{k=1}^{J} g(k) = 1$; A — величина,

характеризующая суммированное значение показателя X за период. Функция $g(k)$, таким образом, задает график распределения показателя X в течение периода. (*Пояснение:* авторы надеются, что употребление термина "период" как для обозначения номера интервала времени k в формуле (1), так и для обозначения свойства периодической функции $g(k)$ в формуле (2) не затруднит читателю понимание текста).

Из практики известно, что период, свойственный продажам сезонных товаров, составляет один год, что однозначно идентифицирует соответствующую этому отрезку времени величину J. В обсуждаемом контексте основной проблемой планирования продаж становится, таким образом, идентификация тех номенклатурных позиций, для которых $g(k)$ является периодической функцией и определение вида функции $g(k)$ в пределах годового периода.

Примем гипотезу о том, что среди множества обрабатываемых системой номенклатурных позиций имеются такие, которые характеризуются сходным "типом сезонности"; у таких товаров функции $g(k)$ будут схожими. Таким образом, можно говорить о возможности классификации товаров по признаку схожести их функций $g(k)$ — близости этих функций к некоему эталонному графику распределения.

На рис. 1 показана модель нейронной сети со слоем нейронов Кохонена. Здесь $x_1, ..., x_J$ (вход) — нормированные значения величины X_k, так что $\sum_{k=1}^{J} x_k = 1$; $I_1, ..., I_{J+1}$ — точки разветвлений; w_{kj} — весовые коэффициенты, отождествляемые с эталонными графиками распределения $g(k)$ того или иного типа сезонности, причем k — номер периода, j — номер "типа сезонности"; $K_1, ..., K_{n+1}$ — нейроны Кохонена, причем n — число типов сезонных товаров, а товары, отнесенные к типу с номером $n + 1$ считаются не имеющими выраженной сезонности; $y_1, ..., y_{n+1}$ (выход) — двоичные значения, характеризующие тот "тип сезонности", к которому была отнесена обрабатываемая сетью номенклатурная позиция (считается, что значение 1 принимает только один из y_k, соответствующий нейрону с максимальным входом); p — входное пороговое значение, характеризующее максимальное возможное расстояние от входного вектора до эталонного графика распределения; если входы нейронов $K_1, ..., K_n$ меньше p, то товар не является сезонным — активируется нейрон K_{n+1}.

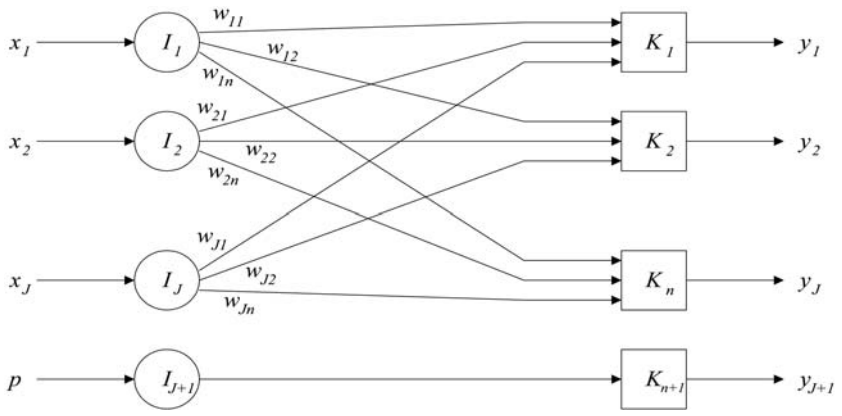

Рис. 1. Модель нейронной сети

Обучение сети (рис. 1) состоит в выборе коэффициентов w_{kj}. При известных свойствах списка номенклатуры их число и начальные значения могут быть заданы пользователем-экспертом интуитивно, а затем уточнены на ретроспективных данных о продажах в процессе реализации стандартного алгоритма обучения сети Кохонена без учителя [4]. В тех же случаях, когда сделать это не представляется возможным (очень большая номенклатура разнородных товаров), число классов n и начальные значения w_{kj} могут быть определены с помощью известных алгоритмов кластеризации. Особенно перспективным для рассматриваемой задачи представляется алгоритм горной кластеризации [5].

Таким образом, с учетом изложенного, алгоритм планирования сезонных продаж представляется следующей последовательностью операций:

1. *Конфигурирование и обучение нейронной сети* (рис. 1):

- выбор из базы данных о продажах значений X_k по каждой номенклатурной позиции с определенной периодичностью;
- задание конфигурации нейронной сети — числа классов n и числа периодов для анализа J; J задается экспертом (исходя из специфики конкретного предприятия), n может как задаваться экспертом, так и определяться решением задачи кластеризации;
- задание начальных значений w_{kj} — эталонных графиков распределения продаж по периодам для каждого типа сезонности; w_{kj} могут как задаваться экспертом, так и определяться решением задачи кластеризации;
- обучение сети — уточнение значений w_{kj} алгоритмом обучения сети Кохонена без учителя; в дальнейшем обучение сети представляет собой динамический процесс, проводимый по мере поступления новых данных и обработки новых номенклатурных позиций.

2. *Планирование продаж:*

- выбор из базы данных о продажах значений X_k по нужной номенклатурной позиции с определенной периодичностью;
- нормирование данных и подача x_k на вход нейронной сети, определение "типа сезонности";

- если товар отнесен к "сезонным", то планирование (экстраполяция) продаж с учетом его графика распределения по периодам $g(k)$, идентифицируемого по значениям соответствующих весов w_{kj};
- если товар отнесен к "несезонным", то планирование продаж традиционными способами (например, встроенными средствами КИС).

При работе над задачей авторы имели в виду, что базовым средством расширения функциональности современных КИС является, как правило, встроенный скриптовый язык системы. Такой подход существенно ограничивает вычислительную мощность системы, делает ее чувствительной к емким математическим процедурам. Побудительным мотивом обращения авторов к нейросетевой модели явилась сравнительно низкая требовательность моделей такого типа к вычислительным ресурсам КИС.

Результаты решения задачи

Эксперимент по планированию продаж с использованием предлагаемой методики проводился на реальных данных о продажах крупной компании, занимающейся торговлей автозапчастями, номенклатура товаров которой включает более 13 тыс. наименований, при этом продажи более чем 2,5 тыс. наименований носят регулярный характер.

С учетом специфики товаров и учетной политики предприятия были приняты следующие значения параметров задачи: показатель X_k равен количеству проданных товаров; интервал времени между X_k и X_{k+1} равен одному месяцу; период J равен 12 месяцам, причем $k = 1$ соответствует январю, а $k = 12$ — декабрю. Число "типов сезонности" n и начальные значения w_{kj} задавались путем решения задачи кластеризации с использованием горного алгоритма [5].

На рис. 2, *а* показаны упорядоченные по возрастанию значения потенциалов первого (сплошная линия), второю и третьего (штриховые линии) и остальных (пунктирные линии) кластеров, найденных горным алгоритмом. Здесь P — потенциал кластера в точке; i — номер точки (номер номенклатурной позиции). Видно, что вес кластеров, начиная с четвертого, становится очень незначительным (очень небольшая разница между пунктирными графиками на рисунке). Это означает, что на обрабатываемых данных можно выделить лишь три выраженных типа сезонности.

На рис. 2, *б—г* показаны найденные значения $g(k)$ для товаров первого, второго и третьего типа сезонности соответственно (жирная сплошная линия) и значения x_k тех товаров, что были отнесены к тому или иному типу (точки) по результатам работы алгоритма горной кластеризации. Пороговое значение p (см. рис. 1) было принято равным 0,2, при этом к первому типу сезонности было отнесено 152 товара, ко второму — 31, к третьему — 45. Несмотря на то, что число этих товаров не превышает 8 % от регулярно продаваемых, сумма выручки по этим товарам составляет около 37 %, что подтверждает актуальность рассматриваемой задачи.

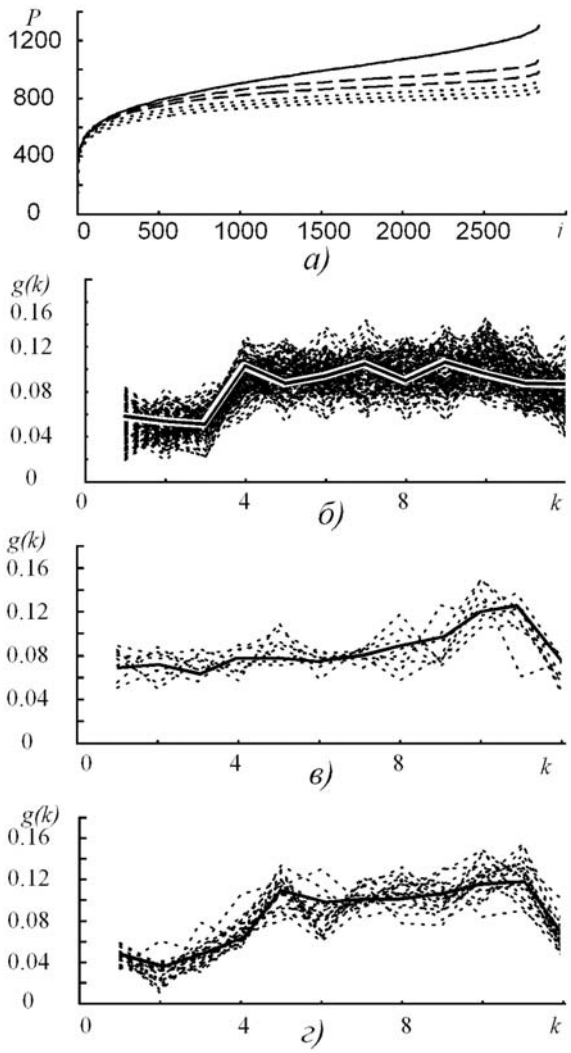

Рис. 2. Результат конфигурирования нейронной сети

На рис. 3 показан результат обучения нейронной сети на ретроспективных данных (за "прошлый" год) и реальные продажи товаров, идентифицированных сетью как сезонные, в "текущем" году (с января по май). Здесь сплошная линия — значения весовых коэффициентов w_{kj} для первого (рис. 3, *а*), второго (рис. 3, *б*) и третьего (рис. 3, *в*) типа сезонности, точки — значения x_k реальных продаж тех товаров, что были распознаны сетью как сезонные того или иного типа. Выделенные сетью товары характеризуются:

- низким уровнем продаж в январе—марте с резким ростом в апреле и стабильным уровнем до конца года (первый тип сезонности);
- стабильным уровнем продаж с января по июль с ростом осенью и падением к концу года (второй тип сезонности);
- низким уровнем в зимний период и высоким с мая по октябрь (третий тип сезонности).

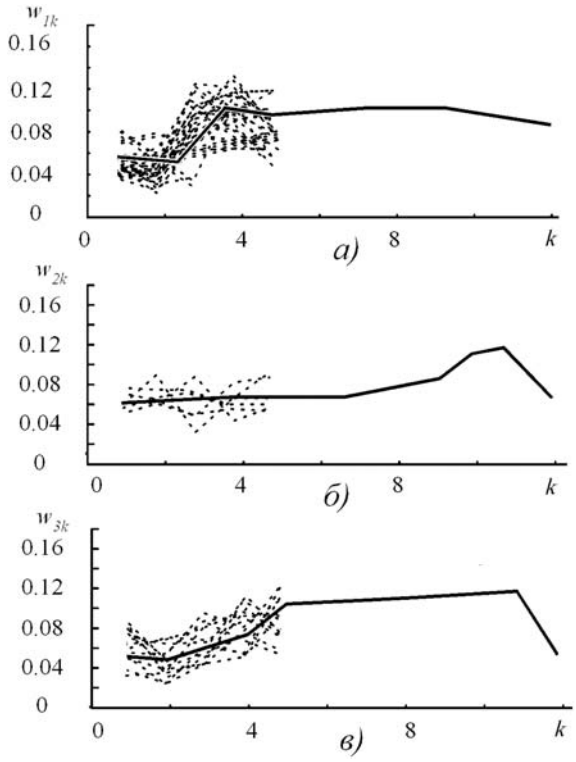

Рис. 3. Результат обучения нейронной сети и текущие продаж

Такие типы сезонности действительно имеют место для некоторых групп автотоваров.

Предлагаемая в работе модель планирования сезонных продаж была адаптирована к данным корпоративной информационной системы "1С: Управление торговлей 8" и "1С: Управление производственным предприятием 8" и реализована в виде обработки платформы "1С: 8", основное назначение которой — предоставить менеджеру выборку сезонных номенклатурных позиций и рекомендации по планированию их продаж. Программа была апробирована на реальном предприятии и подтвердила конструктивность предложенной методики: число отказов клиентам по причине отсутствия товаров сократилось на 15 %; вместе с тем, загрузка склада уменьшилась на 12 %; на 21 % уменьшился срок нахождения товаров на складах: производительность системы оказалась в приемлемых рамках.

Работа ориентирована на расширение функциональности современных корпоративных информационных систем.

Список литературы

1. **Девятов Д. Х., Каплан Д. С., Иванов К. А.** Информационная система руководителя металлургического предприятия // Информационные технологии. 2008. № 9.

2. **Turban E., McLean E., Wetherbe J.** Information Technology for Management: Transforming Business in the Digital Economy / John Wiley & Sons, 2002.

3. **Гриняк В. М., Семенов С. М.** Модель планирования продаж в современных корпоративных информационных системах // Естественные и технические науки. 2009. № 1.

4. **Горбань А. Н., Дунин-Барковский В. Л., Кирдин А. Н.** и др. Нейроинформатика. Новосибирск: Наука, 1998.

5. **Штовба С. Л.** Введение в теорию нечетких множеств и нечеткую логику. URL: http://matlab.exponenta.ru/fuzzylogic/book1/index.php

CONTENTS

Existing methods of production protection had been analyzed. The methods of physical treatment of production, as well as methods of mathematical processing of the scanned package had been proposed. The algorithms of coding and identification, the software-hardware set of tools for protection of polymer packages against forgery, the architecture of a computer system for the implementation of the proposed methods, as well as distributed software application had been developed. Working capacity of a complex is checked up on the international industrial productions of polymeric packages.

Keywords: falsification, protection against forgery, recognition of images, hardware-software complex, coding, identification

In this paper we propose a new approach to vehicle license plates recognition. Results of road situations modeling and vehicle license plates recognition are presented. Influence of different causes on recognition results was analyzed.

Keywords: computer vision, image processing, OCR, license plate recognition

In this paper, we give outcomes of the CMAC neural network with its modifications in fairly difficult pattern classification two-dimensional problem, involving nonconvex decisional region. Comparative analysis is based on the digital simulation. Short description of the CMAC neural network and its modifications is given. The influence of supervisor error on the classification accuracy is analyzed. Digital simulation shows that modified CMAC neural network solves the classification problem with high accuracy.

Keywords: CMAC neural network, modification, pattern recognition, computer simulation

Structure of radial-basis neural network is described. Imperfections of classical gradient learning algorithm of neural networks in tasks of identifying control object are determined. Modified gradient learning algorithm allowing removing imperfections of classical one is proposed. The example of applying the modified algorithm in the task of two-dimensional function approximation is shown.

Keywords: artificial neural network, radial-basis neural networks, learning algorithm, identification

Sales planning in modern ERP systems is considered in this paper. Season sales is watched as a main problem. Model of season sales planning is based on Kohonen network. Results of model applied are demonstrated.

Keywords: sales planning, neural network, classification, season sales, 1S: Enterprise 8

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 *М. Г. Джавадян*.
Сдано в набор 06.05.2011. Подписано в печать 22.06.2011. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 9,8. Уч.-изд. л. 11.39. Заказ 464. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.
Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"
142110, Московская обл., г. Подольск, ул. Кирова, 15