ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

7(203)
2013

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

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

УДК 519.6

А. Н. Четырбоцкий,
д-р физ.-мат. наук, вед. науч. сотр.,
Дальневосточный геологический институт
ДВО РАН,
Дальневосточный федеральный университет
e-mail: Chetyrbotsky@yandex.ru

Статистическая интерпретация оценок параметров радиально-базисных функций

Рассмотрены способы оценки статистических свойств параметров радиально-базисных функций. Предложено такое расширение элементов этого семейства, применение которого способствует выявлению рельефа поверхности рассматриваемой функции (функция задается дискретной выборкой своих значений). Выполнено обоснование способа оценки статистической значимости весов радиально-базисных функций. На основании результатов серии вычислительных экспериментов разработана методология формирования набора их центров.

Ключевые слова: *радиально-базисные функции, задача поиска минимума, методы глобальной оптимизации, статистическое оценивание параметров*

Введение

Аппарат радиально-базисных функций (RBF) являет собой эффективное средство построения удобной аналитической формы записи многомерного объекта (функции), который на некотором допустимом дискретном множестве своих аргументов задан набором соответствующих ему дискретных значений. Такое представление применяют в качестве вспомогательного способа решения широкого спектра задач: интерполяции многомерных функций [9], решения уравнений в частных производных [2] и т. д. Для применения этого аппарата следует конкретизировать тип базисной функции RBF, указать координаты центров и выполнить процедуру поиска экстремума соответствующего функционала невязки. Полагается, что каждый представитель семейства RBF радиально изменяется вокруг только соответствующего ему центра и значимо отличается от нуля лишь в некоторой его окрестности. Вопрос же определения числа центров, как правило, решается простым перебором.

Вне рамок такого подхода остаются вопросы оценки статистической значимости весов. Кроме того, требование сферичности окрестности центров существенным образом влияет на достоверность результатов и объем вычислений при выполнении итерационного процесса поиска экстремума. Принятие положения о сферичности указывает на отсутствие различий масштабов у аргументов рассматриваемой функции. Естественно, что более общим случаем является эллиптичность окрестности и различие окрестностей для разных центров.

В представленной здесь работе предпринята попытка решения отмеченных вопросов. Для случая, когда известны точки локальных экстремумов, предлагается методика формирования центров RBF и такая форма записи элементов этого семейства, где учитываются различия масштабов (разномасштабность) изменения их аргументов. Обсуждаются вопросы проведения вычислений, приведено обоснование способа оценки статистической значимости весов нейронных сетей (RBFNN). Для ряда тестовых примеров показано, что предлагаемая методология способствует снижению вычислительных затрат. Практическое использование методики демонстрируется на примере оценивания распределения вязкости в мантии Земли.

1. Постановка задачи

Формальная запись RBF-представления заданной на допустимом множестве $X = \{x_k \in D \subset R^n, k = \overline{1, N}\}$ своих аргументов некоторой функции $F(X, \lambda)$ имеет вид

$$F(X, \lambda) \equiv \sum_{j=1}^{J} a_j \phi(\|X - C_j\|, \lambda) + p(X), \qquad (1)$$

где N — число точек n-мерного пространства, в которых требуется вычислить (1); $\{C_j, j = \overline{1, J}\}$ — набор J центров RBF того же пространства; $\phi(\|X - C_j\|, \lambda)$, $\{a_j, j = \overline{1, J}\}$ — набор однотипных базисных функций RBFNN и соответствующих им весов; $p(X)$ — некоторый полином; $\lambda \equiv \{\lambda_m, m = \overline{1, M}\}$ — настраиваемые при обучении параметры представления (1), M — их общее число; $\|o\|$ — введенная на R^n некоторая метрика (в большинстве случаев — эвклидова метрика) [9]. Везде далее рассматривается ситуация $p(X) \equiv \text{const}$ и тогда полагается $\text{const} \equiv a_{J+1}$

(не исключается также случай $a_{J+1} \equiv 0$, т. е. в выражении (1) может отсутствовать свободный член). Из чего следует увеличение на 1 размерности набора a: $a \equiv \{a_j, j = \overline{1, J+1}\}$, где в зависимости от ситуации на его элементы могут налагаться некоторые ограничения (в частности, сумма элементов полагается равной 0 или 1).

Для построения аппроксимации $F(X, \lambda)$ посредством (1) следует так подобрать зависящее от набора параметров λ соотношение, чтобы оно при $\widetilde{X} = \{x_i \in D \subset R^n, i = \overline{1, I}\}$ в наибольшей степени, в смысле определенного критерия, отвечало бы заданному выборочному дискретному набору значений $\{f_i \equiv f(x_i), x_i \in D \subset R^n, i = \overline{1, I}\}$ рассматриваемой функции. Другими словами, требуется так определить набор $\{\lambda_m, m = \overline{1, M}\}$ и параметры базисных функций $\phi(\|X - C_j\|, \lambda)$, а также так подобрать веса $\{a_j, j = \overline{1, J+1}\}$, чтобы (1) в точках многомерного пространства $\widetilde{X} = \{x_i \in D \subset R^n, i = \overline{1, I}\}$ как можно в большей степени соответствовало бы $\{f_i \equiv f(x_i), x_i \in D \subset R^n, i = \overline{1, I}\}$.

При традиционном подходе полагается сферичность окрестности центров, что обусловливает один и тот же масштаб изменения аргументов функции. Также полагается отсутствие различий значений коэффициентов $\{\lambda_m, m = \overline{1, n}\}$ для разных центров, т. е. для них принимается одна и та же фиксированная окрестность. Простая форма записи базисных функций, где учитывается различие масштабов и различие окрестностей центров — для каждого центра снабдить каждый член записи отдельным коэффициентом и принять различия окрестностей. Тогда записи элементов семейств RBF примут вид [5, 7]: 1) мультиквадрики (MQ) $\phi(\psi_{ij}) \equiv \sqrt{1 + \psi_{ij}}$; 2) обратные мультиквадрики (IMQ) $\phi(\psi_{ij}) \equiv 1/\sqrt{1 + \psi_{ij}}$; 3) обратные квадрики (IQ) $\phi(\psi_{ij}) \equiv 1/(1 + \psi_{ij})$; 4) обобщенные мультиквадрики (GMQ) $\phi(\psi_{ij}, \beta) \equiv (1 + \psi_{ij})^{\beta}$; 5) Гауссианы (GA) $\phi(\psi_{ij}) \equiv \exp(-\psi_{ij})$. Здесь $\psi_{ij} \equiv \sum_{m=1}^{n} \lambda_{jm}(x_{im} - C_{jm})^2$ — мера близости между элементами области $D \subset R^n$ и центрами RBF; $\lambda \equiv \{\lambda_{jm}, j = \overline{1, J}, m = \overline{1, n}\}$ — матрица коэффициентов, определяющих гладкость RBF. Форма представления величины ψ_{ij} показывает, что она есть взвешенная сумма соответствующих квадратов разностей.

Параметры λ, а в случае 4 и β, подлежат определению на основании заданного набора экспериментальных данных. Если не указан набор центров, то элементы матрицы $C = \{C_{jm}, i = \overline{1, J}, m = \overline{1, n}\}$ также следует оценивать по заданной выборке входных данных. Вычислительная процедура оценивания параметров состоит в минимизации меры несоответствия между наборами $\{f_i \equiv f(x_i), x_i \in D \subset R^n, i = \overline{1, I}\}$ и $\{F(x_i, \lambda), x_i \in D \subset R^n, i = \overline{1, I}\}$

$$\Phi(a, P) = \sum_{i=1}^{I} \left[f_i - \sum_{j=1}^{J} a_j \phi(\|x_i - C_j\|, \lambda) - a_{J+1} \right]^2, \quad (2)$$

где $P \equiv \{\{C\}, \{\lambda\}, \{x\}\}$ — совокупность определяющих (2) параметров. Допускаются также случаи, когда аргументами (2) могут быть координаты некоторых точек $\{x_i \in D \subset R^n, i = \overline{1, I}\}$ обучения нейронной сети. Такая ситуация типична тогда, когда требуется заменить ресурсоемкую в плане вычислений сложную аналитическую конструкцию/функцию ее более простой аппроксимацией. В такой ситуации допустимая потеря в точности сопутствует, как правило, существенному снижению объема вычислений.

Задача поиска минимума (2) принимает вид

$$\min_P \min_a \Phi(a, P). \quad (3)$$

Ее решение следует итерационной схеме, каждая итерация которой насчитывает два последовательных этапа: на первом из них следует формирование зависящих от C и λ элементов матрицы $A = A(C, \lambda)$, которая на втором этапе является матрицей коэффициентов линейной системы уравнений для вычисления искомых весов RBFNN:

$$Aa = b, \quad (4)$$

где элементы матрицы $A = \{A_{ij}\}_{i = \overline{1 \cdot I}, j = \overline{1, J+1}}$ и вектора b в рамках принятых здесь положений и обозначений представлены выражениями

$$A_{ij} \equiv \begin{cases} \phi(\|x_i - C_j\|, \lambda) \text{ при } i \leqslant I, j \leqslant J, \\ 1 \text{ при } i \leqslant I, j = J + 1, \end{cases}$$
$$b \equiv \{f_i \equiv f(x_i), x_i \in D \subset R^n, i = \overline{1, I}\}. \quad (5)$$

Согласно (4) и (5) возбуждения x_i генерируют в скрытом слое нейронной сети сигналы $\phi_i \equiv \{\phi(\|x_i - C_1\|, \lambda), \phi(\|x_i - C_2\|, \lambda), ..., \phi(\|x_i - C_J\|, \lambda), 1\}$, где 1 есть единичный сигнал и $i = \overline{1, I}$. Сигналу ϕ_i отвечает выходной сигнал $F_i \equiv \phi_i a^T$.

При поиске решения (3) многопараметричность функционала (2) обусловливает значительные вычислительные трудности использования градиентных методов. Как правило, в задачах такого класса отмечается их многоэкстремальность. Результат нахождения экстремумов поиска и объем соответствующих вычислений непосредственно определяются характером используемой процедуры и выбо-

ром начального приближения. При использовании градиентных методов происходит сходимость поиска к некоторому локальному минимуму. Тогда на каждой итерации для нахождения шага градиентной процедуры требуется расчет элементов соответствующего градиента функционала, что связано с большими объемами вычислений.

В указанной ситуации целесообразным видится применение современных методов решения многоэктремальных задач, а именно методов глобальной оптимизации [11]. Поиск экстремумов на допустимом множестве переменных сводится к определенному перебору локальных решений. Среди них выделяются методы, основанные на поведенческой и эволюционной стратегии коллективного поведения самоорганизующихся живых и неживых систем. В алгоритмах данного класса методов для улучшения глобального поиска минимума применяется локальная оптимизация. Такие алгоритмы обусловливают сходимость генерируемой ими последовательности точек многомерного пространства к глобальному оптимальному решению. Механизм их работы состоит в предварительной генерации совокупности частиц (строк-стрингов или хромосом [10]). Другими словами, сначала формируется выборка первичных (достаточно приблизительных) оценок искомых параметров задачи (3). Далее строится итерационный процесс, на каждой стадии которого определенным образом следует их пересчет. Для t-й итерации простейший алгоритм подобной процедуры (так называемого алгоритма PSO "Particle Swarm Optimization" [10]) записывается в виде

$$\begin{aligned} V_{k,j}^{(t+1)} &= \beta_1 P_{k,j}^{(t)} + \beta_2 \mathrm{rand}(\tilde{P}_{k,j} - P_{k,j}^{(t)}) + \\ &+ \beta_3 \mathrm{rand}(P_j^* - P_{k,j}^{(t)}), \\ P_{k,j}^{(t+1)} &= P_{k,j}^{(t)} + V_{k,j}^{(t+1)}, \end{aligned} \qquad (6)$$

где $\{ V_{k,j}^{(t)},\ k = \overline{1, K}\ ,\ j = \overline{1, J}\ \}$ — сдвиг значения j-го признака частицы (согласно общепринятой терминологии, элементы искомого множества именуются частицами [11]) на t-й итерации; K — заранее заданное общее число частиц (это число определяет размер выборки искомых параметров); $\beta_1, \beta_2, \beta_3$ — эмпирические параметры алгоритма; rand — равномерно распределенное на отрезке [0, 1] случайное число; $\tilde{P}_{k,j}$ — лучшая, согласно (3), позиция k-й частицы за текущие t итераций; $P_k^{(m)} \equiv \{ P_{k,j}^{(m)},\ k = \overline{1, K}, j = \overline{1, J} \}$ — координаты частиц на t-й итерации; $P^* \equiv \{P_j^*, j = \overline{1, J}\}$ координата частицы с наилучшим, в смысле (3), значением целевой функции.

Современные методы решения (3) состоят во введении спектрально близкой к A легко обращаемой матрицы B. Значительное снижение числа обусловленности A достигается умножением обеих частей (4) на матрицу B^{-1}. Для ее построения имеются два основных способа. В первом из них (традиционном) используется стандартное LU разложение матрицы, во втором — построение приближенно обратной (так называемой псевдообратной) к A матрицы (процедуры *lu* и *svd* из работы [8]). Следуя публикациям, второй способ нахождения обратной матрицы является более робастным и более ресурсоемким [3].

2. Статистические оценки параметров

Формирование выборочного набора $\{f_i \equiv f(x_i), x_i \in D \subset R^n, i = \overline{1, I}\}$ и проведение вычислений по схеме (6) определяется фактором случайности, что обусловливает статистический характер исследований. Действительно, сначала из множества значений рассматриваемого объекта (функции) некоторым (в общей ситуации случайным) образом формируется обучающая выборка. Далее на каждой t-й итерации процесса (6) из множества допустимых искомых значений параметров $P^{(t)} \equiv \{ P_{kj}^{(t)},\ k = \overline{1, K},\ j = \overline{1, J}\ \}$ по определенной схеме проводится их целенаправленный отбор. На следующем итерационном шаге вновь следует процедура такого специфического отбора. Сходимость этой вычислительной процедуры обусловлена спецификой алгоритма глобального поиска [10, 11]. Результат решения (3) определяется координатами конкретной уже неслучайной точки многомерного пространства. Таким образом, рассмотренный процесс оценки искомых параметров следует схеме метода наименьших квадратов, где у независимых переменных отсутствуют ошибки. Следуя общей концепции классического метода наименьших квадратов (МНК) [6], несмещенную оценку дисперсии можно определить с помощью следующего выражения:

$$\sigma^2 = e'e/(K - J - 1), \qquad (7)$$

где $e = \{f_i - \sum\limits_{j=1}^{J} a_j \phi(\|x_i - C_j\|, \lambda) - a_{J+1},\ i = \overline{1, I}\ \}$ — вектор выборочных ошибок.

Результаты решения (3) позволяют получить не сами значения искомых параметров, а только их оценки. Поэтому для корректного построения RBFNN и содержательной интерпретации полученных результатов следует не только получить оценки параметров, но также исследовать их статистическую надежность. Количественной мерой такой надежности выступают вычисленные по исходным данным средние квадратичные отклонения (СКО) параметров. Если решение задачи (3) уже найдено, то расчет СКО весов RBFNN и остальных параметров целесообразно выполнять согласно этапам обучения сети. Поскольку веса a являются коэф-

фициентами линейной регрессионной модели, то в рамках положений этой модели матрица их ковариации определяется выражением [1, 10]

$$S^{(a)} = (A'A)^{-1}\Phi(a^*,\ P^*)/(I - J - 1), \qquad (8)$$

где a^* — решение (4) при P^*. Квадратные корни ее диагональных элементов есть СКО весов RBFNN. Элементы $S^{(a)}$ могут оказаться полезными для оценки числа скрытых нейронов RBFNN. Действительно, равенство нулю некоторого элемента a указывает на сокращение числа скрытых нейронов. Проверка гипотезы H_0:$a_j = 0$ выполняется с помощью t-статистики (в системе MATLAB ее численное значение определяется процедурой *tinv*):

$$t = a_j / \sqrt{S_{j,j}^{(a)}}\ . \qquad (9)$$

Для всех примеров здесь уровень значимости равен 0,05. Согласно положениям математической статистики, когда параметр меньше $t\sqrt{S_{j,j}^{(a)}}$, то он полагается равным нулю [6, 4, 1]. Другими словами, этот параметр можно не учитывать, т. е. имеет место избыточность задания числа параметров и их число можно понизить. Подобным образом проверяются и другие частные гипотезы. Например, равенство всех элементов a одному и тому же значению.

3. Вычислительные эксперименты

Для оценки работоспособности численных методов важными являются вопросы выбора соответствующих тестовых примеров. Для этого здесь использованы двумерные варианты функции из работ [8, 10] (см. таблицу).

Тестовые функции

Название функции	Функция	Область определения
Функция Экли	$f(x, y) = 20 + \exp(1) - 20\exp(\varphi) - \exp(\psi)$ $\varphi = -0{,}2\sqrt{(x^2 + y^2)/2}$ $\psi = [\cos(2\pi x) + \cos(2\pi y)]/2$	$x \in [-20, 20]$, $y \in [-20, 20]$
Функция Франке	$f(x, y) = 0{,}75[\exp(-\varphi/4) + \exp(-\psi)] + 0{,}5\exp(-\eta) + 0{,}2\exp(-\varpi)$ $\varphi = (9x - 2)^2 + (9y - 2)^2$ $\psi = (9x + 1)^2/49 - (9y + 1)/10$ $\eta = (9x - 7)^2 + (9y - 3)^2$ $\omega = (9x - 4)^2 - (9y - 7)^2$	$x \in [0, 1]$, $y \in [0, 1]$
Функция Шекеля	$f(x, y) = \sum_{l=1}^{3} [b_l^{(1)} + (x - b_l^{(2)})^2 + (y - b_l^{(3)})^2]^{-1}$ $b^{(1)} = (1, 2, 2)$, $b^{(2)} = (2, 10, 18)$, $b^{(3)} = (10, 15, 4)$	$x \in [0, 20]$, $y \in [0, 20]$

Целесообразность их использования обусловлена достаточно сложным рельефом поверхностей функций, что позволяет оценить применимость и эффективность предлагаемых разработок для представления столь нетривиальных случаев. При расчетах используют семейство гауссовских функций.

При выполнении экспериментов было замечено существенное снижение вычислительных затрат и улучшение сходимости соответствующих итерационных процессов в случае принятия в качестве центров сети RBFNN точек локальных экстремумов используемых тестовых функций. Для оценки эффективности такого подхода было выполнено сравнение двух различных RBF аппроксимаций, которые получены разными способами выбора их центров. В первом случае в качестве центров назначали точки экстремумов функций, а во втором случае — центры RBF были первоначально равномерно распределены в области определения функции. Затем для снижения (2) эти центры назначали искомыми параметрами задачи (3) (наряду с остальными искомыми параметрами они в такой ситуации составляют вектор P) и они участвовали в решении задачи (3) по алгоритму (6). Объем обучающей выборки определялся тем фактом, что коэффициент корреляции между выборочными и модельными значениями был не ниже 0,980. При записи λ точка с запятой отделяет одни координаты от других.

Обучающая выборка для функции Экли строится в узлах 9×9 прямоугольной регулярной сетки. В первом случае центром RBF назначается точка (0,0). Результаты вычислений показывают: $\lambda = (0{,}05, 0{,}05)$, $a = (-17{,}999 \pm 0{,}071;\ 19{,}970 \pm 0{,}001)$, $\sigma^2 = 0{,}230$ и коэффициент корреляции между выборочными значениями функции и RBF-аппроксимацией равен $R = 0{,}984$ (рис. 1, *б*, см. вторую сторону обложки). Совпадение координат центра и элементов λ указывает на симметричность рельефа поверхности данной тестовой функции. Высокое значение R указывает на хорошее совпадение выборочных и модельных значений функции. При записи вектора весов a после знака "±" стоит $t\sqrt{S_{j,j}^{(a)}}$. Поскольку для каждого параметра эта величина меньше его значения, то отвергается гипотеза его равенства нулю. Во втором случае (центры RBF равномерно распределены в области определения функции) при построении RBF аппроксимации (рис. 1, *в*) были использованы семь центров. В этой ситуации оказалось, что $\sigma^2 = 1{,}438$ и $R = 0{,}870$. Сравнение этих показателей показывает, что RBF-аппроксимация в первом случае существенно ближе к оригиналу (рис. 1, *а*, см. вторую сторону обложки), чем во втором случае (в первом случае σ^2 почти в 7 раз меньше этой величины для второго случая и R в первом случае выше этой величины для второго случая).

Обучающая выборка для функции Франке (интересно заметить, что профиль этой функции представляет фирменный знак системы MATLAB) строится в узлах 7 × 7 прямоугольной регулярной сетки. В первом случае центром RBF назначаются точки ее экстремумов: (0,222, 0,222), (0,778, 0,333) и (0,444, 0,778). Результаты вычислений показывают: λ = (8,315, 6,771; 19,704, 19,764; 18,605, 0,042), $a = 0{,}1 \times$ (9,832 ± 0,665, 3,419 ± 0,882, −1,234 ± ± 0,575, 1,511 ± 0,317), $\sigma^2 = 0{,}006$ и коэффициент корреляции между выборочными значениями функции и RBF-аппроксимацией равен $R = 0{,}985$ (рис. 1, *д*, см. вторую сторону обложки). Различие элементов λ указывает на отсутствие симметрии рельефа поверхности данной тестовой функции. Высокое значение R указывает на хорошее совпадение выборочных и модельных значений функции. Поскольку для каждого параметра $t\sqrt{S_{j,j}^{(a)}}$ меньше его значения, то отвергается гипотеза его равенства нулю. Во втором случае (центры RBF равномерно распределены в области определения функции) при построении RBF-аппроксимации (рис. 1, *е*, см. вторую сторону обложки) были использованы сначала шесть, а потом семь центров. Оказалось, что при шести центрах $\sigma^2 = 0{,}006$ и $R = 0{,}970$, а при семи центрах — $\sigma^2 = 0{,}004$ и $R = 0{,}985$. Сравнение по этим показателям всех трех случаев практически показывает их неразличимость между собой. Между тем в случае использования в качестве центров точек экстремумов функции RBF ее аппроксимация более адекватно воспроизводит ее графическое представление. Причем координаты центров функций назначали переменными задачи (3) и оценивали по алгоритму (6).

Обучающая выборка для функции Шекеля строится в узлах 12 × 12 прямоугольной регулярной сетки. В первом случае центром RBF назначаются точки ее экстремумов: (2, 10), (10, 15) и (18, 4). Результаты вычислений показывают: $\lambda = 0{,}1 \times$ (3,246, 2,271; 1,888, 2,228; 2,883, 2,734); $a = 0{,}1 \times$ (6,440 ± 0,225; 2,749 ± 0,149; 4,715 ± 0,242; 0,407 ± 0,026); $\sigma^2 = 2{,}837 \cdot 10^{-4}$ и коэффициент корреляции между выборочными значениями функции и RBF-аппроксимацией равен $R = 0{,}982$ (рис. 1, *з*, см. вторую сторону обложки). Различие элементов λ указывает на отсутствие симметрии рельефа поверхности данной тестовой функции. Высокое значение R указывает на хорошее совпадение выборочных и модельных значений функции. Поскольку для каждого параметра $t\sqrt{S_{j,j}^{(a)}}$ меньше его значения, то отвергается гипотеза его равенства нулю. Во втором случае (центры RBF равномерно распределены в области определения функции) при построении RBF-аппроксимации (рис. 1, *е*) было использовано сначала семь, а потом 36 центров. В обоих случаях координаты центров являлись искомыми параметрами задачи (3). Поскольку для семи центров графическое представление результата оказалось далеким от оригинала, то в данном случае так же как и в работе [7] задавались 36 центров. В этой ситуации $\sigma^2 = 4{,}368 \cdot 10^{-4}$ и коэффициент корреляции между выборочными значениями функции и RBF-аппроксимацией равен $R = 0{,}979$.

Заключение

Выполненные вычислительные эксперименты показывают высокую эффективность предлагаемых здесь разработок. Тот факт, что для их использования требуется знание локальных экстремумов, не является столь существенным ограничением их применения. Действительно, в случае представительной обучающей выборки они могут быть найдены по самой выборке. Зачастую также встречаются задачи, когда сложные в вычислительном плане функции с известными локальными экстремумами требуется заменить более простыми аналитическими функциями или их некоторыми сочетаниями.

Список литературы

1. **Болч Б., Хуань К. Дж.** Многомерные статистические методы для экономики. М.: Финансы и статистика, 1979. 317 с.

2. **Васильев А. Н., Тархов Д. А.** Нейросетевое моделирование. Принципы. Алгоритмы. Приложения. СПб.: Изд-во Политех. ун-та, 2009. 527 с.

3. **Ермаков М. К.** Исследование возможностей матричных методов для решения уравнений Навье—Стокса // Физико-химическая кинетика в газовой динамике. 2010. № 1. С. 12—19.

4. **Кендалл М. Дж., Стьюарт А.** Статистические выводы и связи. М.: Наука, 1976. 736 с.

5. **Осовский С.** Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 344 с.

6. **Рао С. Р.** Линейные статистические методы и их применение. М.: Наука, 1968. 548 с.

7. **Четырбоцкий А. Н.** Параметрическая идентификация радиальных базисных функций нейронных сетей методами глобальной оптимизации //Информационные технологии. 2011. № 11. С. 54—58.

8. **Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., Sorensen D.** LAPACK User's Guide (http://www.netlib.org/lapack/lug/lapack_lug.html), Third Edition, SIAM, Philadelphia, 1999.

9. **Hardy R. L.** Multiqadric equations of topography and other irregular surfaces // J. Geophys. Res. 1971. Vol. 76(8). P. 1905—1915.

10. **Kennedy J.** Particle swarm optimization // Proc. of IEEE International Conference on Neural Networks, 1995. P. 1942—1948.

11. **Tang R., Yao X., Suganthan P. N., MacNish C., Chen J. P., Chen C. M., Yang Z.** Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization // Nature Inspired Computation and Applications Laboratory, USTC, China, 2007.

УДК 004.3.06

Е. В. Гливенко, д-р техн. наук, проф.,
А. С. Фомочкина, аспирант,
e-mail: nastya@gmail.com,
С. А. Прядко, аспирант,
e-mail: sergeypryadko@gmail.com,
РГУ нефти и газа им. И. М. Губкина, г. Москва

Решение системы нелинейных алгебраических уравнений с помощью степени отображения

Описывается возможное применение результатов алгебраической геометрии на примере решения системы нелинейных алгебраических уравнений.

Ключевые слова: *система нелинейных алгебраических уравнений, степень отображения, аффинное преобразование*

Введение

Традиционно для решения систем нелинейных уравнений применяются последовательные методы, такие как метод Ньютона, метод простой итерации и т. д. [1]. Однако у последовательных методов есть ряд недостатков. Во-первых, ввиду того, что каждый следующий шаг зависит от предыдущего, происходит накопление ошибки. Во-вторых, такие методы плохо распараллеливаются, а значит, не позволяют эффективно использовать совершенные вычислительные системы и многопроцессорные комплексы. Поэтому, учитывая постоянное развитие технологий и повышенные требования к точности, набирают обороты параллельные методы [2].

Постановка задачи

Пусть задана система уравнений

$$\begin{cases} F_1(x_1, x_2, ..., x_n) = 0; \\ \vdots \\ F_n(x_1, x_2, ..., x_n) = 0. \end{cases} \quad (1)$$

Требуется найти ее решение, т. е. найти такие наборы $(x_1^i, x_2^i, ..., x_n^i)$, что каждый из них удовлетворяет каждому из уравнений системы (1).

Поступим следующим образом. Зададим в n-мерном пространстве шар K с границей S и постараемся ответить на вопрос: имеются ли внутри шара точки, координаты которых являются решением системы (1).

Если такие точки имеются, то уточнять решение мы будем, уменьшая размер шара.

Для ответа на поставленный вопрос рассмотрим преобразование n-мерного пространства, заданное следующими соотношениями:

$$\begin{cases} x_{1\text{нов}} = x_{1\text{ст}} + F_1(x_{1\text{ст}}, x_{2\text{ст}}, ..., x_{n\text{ст}}) = 0; \\ x_{2\text{нов}} = x_{2\text{ст}} + F_2(x_{1\text{ст}}, x_{2\text{ст}}, ..., x_{n\text{ст}}) = 0; \\ \vdots \\ x_{n\text{нов}} = x_{n\text{ст}} + F_n(x_{1\text{ст}}, x_{2\text{ст}}, ..., x_{n\text{ст}}) = 0. \end{cases} \quad (2)$$

Очевидно, неподвижные точки этого преобразования — это точки, координаты которых являются решениями системы (1).

Таким образом, отыскание решений системы (1) мы сводим к отысканию неподвижных точек преобразования (2). Назовем это преобразованием φ n-мерного пространства в себя. В нашем случае будем предполагать, что преобразование φ непрерывно.

Степень отображения

Остановимся непосредственно на определении степени отображения некоторого преобразования Φ сферы S в единичную сферу Q с центром в начале координат.

Сфера S, как граница n-мерного шара, имеет размерность $(n - 1)$. Рассмотрим триангуляцию сферы S с помощью $(n - 1)$-мерных симплексов, диаметр каждого из которых меньше ε, причем i-й симплекс полученной триангуляции имеет n вершин $p_1^i, p_2^i, ..., p_n^i$. Выбранная нами триангуляция должна обладать тем свойством, что точки каждого набора $p_1^i, p_2^i, ..., p_n^i$ должны находиться в общем положении, т. е. не найдется подпространства размерности меньше, чем $(n - 1)$, содержащего все $p_1^i, p_2^i, ..., p_n^i$.

***Преобразование* Ф.** Рассмотрим точку сферы S с координатами $x_1, x_2, ..., x_n$. Координаты образа этой точки при преобразовании φ обозначим через $y_1, y_2, ..., y_n$, а координаты $\Phi(x_1, x_2, ..., x_n)$ — через $z_1, z_2, ..., z_n$.

$$\begin{cases} z_1 = \dfrac{y_1 - x_1}{\sqrt{(y_1 - x_1)^2 + ... + (y_n - x_n)^2}} - x_1; \\ z_2 = \dfrac{y_2 - x_2}{\sqrt{(y_1 - x_1)^2 + ... + (y_n - x_n)^2}} - x_2; \\ \vdots \\ z_n = \dfrac{y_n - x_n}{\sqrt{(y_1 - x_1)^2 + ... + (y_n - x_n)^2}} - x_n. \end{cases} \quad (3)$$

Таким образом, точки с координатами $z_1, z_2, ..., z_n$ лежат на единичной сфере Q. Применим преобразование Φ к точкам $p_1^i, p_2^i, ..., p_n^i$, где i — это номер

симплекса, натянутого на точки p_1^i, p_2^i, ..., p_n^i. Получим точки $q_1^i = \Phi(p_1^i)$, $q_2^i = \Phi(p_2^i)$, ..., $q_n^i = \Phi(p_n^i)$, лежащие на единичной сфере Q.

Известно, что существует аффинное преобразование U точек p_j^i в точке q_j^i, так как точки p_j^i находятся в общем положении.

Для каждой пары (образа и прообраза) симплексов такое аффинное преобразование свое, т. е. оно зависит от i.

Аффинное преобразование U. Рассмотрим теперь какой-либо определенный симплекс и его вершины p_1, ..., p_n и его образ — симплекс с вершинами q_1, ..., q_n.

Будем обозначать через x_1, x_2, ..., x_n координаты точки прообраза, а через z_1, z_2, ..., z_n — координаты точки образа.

Запишем аффинное преобразование в следующем виде:

$$\begin{cases} z_1 = u_1^{(1)} x_1 + ... + u_1^{(n)} x_n + u_1; \\ z_2 = u_2^{(1)} x_1 + ... + u_2^{(n)} x_n + u_2; \\ \vdots \\ z_n = u_n^{(1)} x_1 + ... + u_n^{(n)} x_n + u_n. \end{cases} \qquad (4)$$

Знак определителя

$$\det = \begin{vmatrix} u_1^{(1)} & \dots & u_n^{(n)} \\ \vdots & & \\ u_n^{(1)} & & u_n^{(n)} \end{vmatrix}$$

будет говорить о том, имеют ли наборы точек p_1, ..., p_n и q_1, ..., q_n одинаковую (если det > 0) или разную (если det < 0) ориентацию.

Известна [1] **теорема**:

Если в пространстве R^{n-1} даны два конечных множества точек: множество p_1, ..., p_n, состоящее из n линейно независимых точек, и множество q_1, ..., q_n, то существует единственное аффинное отображение C пространства R^{n-1}, переводящее точки p_1, p_2, ..., p_n в точки q_1, q_2, ..., q_n. Это отображение C мы получили, поставив в соответствие произвольной точке a с барицентрическими координатами μ_1, ..., μ_n (взятыми по отношению к координатному остову $(p_1, ..., p_n)$) центр тяжести масс μ_1, ..., μ_n, помещенных в точки q_1, ..., q_n.

Постараемся с помощью этой теоремы выразить параметры преобразования U через координаты точек p_1, ..., p_n и q_1, ..., q_n.

Барицентрические координаты [1]. Плоскость $R(a_0, ..., a_r)$ состоит из тех точек пространства R^n, которые могут быть представлены в виде

$$a = \mu_0 a_0 + \mu_1 a_1 + ... + \mu_r a_r \qquad (5)$$

при дополнительном условии, что

$$\mu_0 + \mu_1 + ... + \mu_r = 1 \qquad (6)$$

с другой стороны числа μ_0, μ_1, ..., μ_r однозначно определены точкой a и соотношениями (5) и (6); их называют барицентрическими координатами точки a в координатной системе $(a_0, ..., a_r)$. Это обозначение, введенное Ф. А. Мебиусом, имеет следующее основание: под "материальной точкой" пространства R^n понимают точку с отнесенным к ней действительным числом σ — "массой " материальной точки; если даны материальные точки a_i с массами σ_i и $\mu_i = \dfrac{\sigma_i}{\sigma_0 + \sigma_1 + ... + \sigma_r}$, то точка a из (5) по определению и есть центр тяжести этого распределения масс.

Случай $n = 2$. $R^{n-1} = R^1$ — это прямая. На ней две точки p_1 и p_2, которые преобразование U переводит в точки q_1 и q_2 (рис. 1).

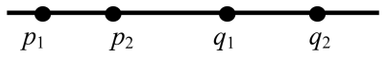


Рис. 1. Переход точек p_i в q_i (i = 1, 2) после преобразования U

Прямая одномерная, поэтому имеем только одну координату.

Обозначим через x_1 координату точки p_1, через x_2 координату точки p_2, через z_1 координату точки q_1, через z_2 координату точки q_2.

Найдем преобразование U. Рассмотрим некоторую произвольную точку x, тогда $z = u_1 x + u_2$. Пусть μ_1 и μ_2 — барицентрические координаты x по отношению к точкам x_1 и x_2. Условие (6) дает $\mu_1 = 1 - \mu_2$. Тогда барицентрические координаты выражаются следующим образом:

$$x = \mu_1 x_1 + \mu_2 x_2 = (1 - \mu_2) x_1 + \mu_2 x_2 = \\ = x_1 + \mu_2 (x_2 - x_1) = x;$$

$$\mu_2 = \frac{x - x_1}{x_2 - x_1};$$

$$\mu_1 = 1 - \frac{x - x_1}{x_2 - x_1} = \frac{x_2 - x}{x_2 - x_1}.$$

Теперь найдем центр тяжести масс μ_1 и μ_2, помещенных в точках q_1 и q_2, координаты которых z_1 и z_2:

$$z = \mu_1 z_1 + \mu_2 z_2 =$$

$$= \frac{x_2 - x}{x_2 - x_1} z_1 + \frac{x - x_1}{x_2 - x_1} z_2 = \frac{x(z_2 - z_1)}{x_2 - x_1} + \frac{z_1 x_2 - z_2 x_1}{x_2 - x_1},$$

т. е. $u_1 = \dfrac{z_2 - z_1}{x_2 - x_1}$, $u_2 = \dfrac{z_1 x_2 - z_2 x_1}{x_2 - x_1}$.

Нас будет интересовать только знак определителя, т. е. знак $u_1 = \dfrac{z_2 - z_1}{x_2 - x_1}$.

Если $x_2 - x_1 > 0$, то знак определителя отрицательный при $z_2 < z_1$ (рис. 2) и положительный в противном случае.

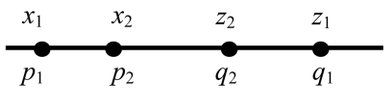


Рис. 2. Расположения точек при $x_2 - x_1 > 0$, $z_2 < z_1$

Если же $x_2 - x_1 < 0$, то знак определителя отрицательный при $z_2 > z_1$ (рис. 3) и положительный в противном случае.

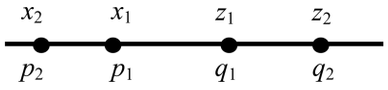


Рис. 3. Расположения точек при $x_2 - x_1 < 0$, $z_2 > z_1$

Случай n = 3. $R^{n-1} = R^2$ — это двумерная плоскость. Заданы три точки на плоскости: точка p_1 с координатами (x_1^1, x_2^1), точка p_2 с координатами (x_1^2, x_2^2), точка p_3 с координатами (x_1^3, x_2^3) и другие три точки: точка q_1 с координатами (x_1^1, x_2^1), точка q_2 с координатами (z_1^2, z_2^2), точка q_3 с координатами (z_1^3, z_2^3). Нужно найти аффинное преобразование плоскости в себя, переводящее точки p_1, p_2, p_3 в точки q_1, q_2, q_3:

$$\begin{cases} x_{1\text{нов}} = u_{11}x_{1\text{ст}} + u_{12}x_{2\text{ст}} + u_1; \\ x_{2\text{нов}} = u_{21}x_{1\text{ст}} + u_{22}x_{2\text{ст}} + u_2, \end{cases} \quad (7)$$

причем

$$\begin{cases} z_1^i = u_{11}x_1^i + u_{12}x_2^i + u_1; \\ z_2^i = u_{21}x_1^i + u_{22}x_2^i + u_2, \end{cases}$$
$$i = 1, 2, 3.$$

Выбираем произвольную точку с координатами (x_1, x_2). Найдем ее барицентрические координаты μ_0, μ_1, μ_2 относительно точек p_1, p_2, p_3:

$$\begin{cases} x_1 = \mu_0 x_1^1 + \mu_1 x_1^2 + \mu_2 x_1^3; \\ x_2 = \mu_0 x_2^1 + \mu_1 x_2^2 + \mu_2 x_2^3; \\ 1 = \mu_0 + \mu_1 + \mu_2, \end{cases}$$

тогда

$$\mu_0 = \frac{\begin{vmatrix} x_1 & x_1^2 & x_1^3 \\ x_2 & x_2^2 & x_2^3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_1^1 & x_1^2 & x_1^3 \\ x_2^1 & x_2^2 & x_2^3 \\ 1 & 1 & 1 \end{vmatrix}}; \mu_1 = \frac{\begin{vmatrix} x_1^1 & x_1 & x_1^3 \\ x_2^1 & x_2 & x_2^3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_1^1 & x_1^2 & x_1^3 \\ x_2^1 & x_2^2 & x_2^3 \\ 1 & 1 & 1 \end{vmatrix}}; \mu_2 = \frac{\begin{vmatrix} x_1^1 & x_1^2 & x_1 \\ x_2^1 & x_2^2 & x_2 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_1^1 & x_1^2 & x_1^3 \\ x_2^1 & x_2^2 & x_2^3 \\ 1 & 1 & 1 \end{vmatrix}}. \quad (8)$$

Теперь найдем точку $(x_{1\text{нов}}, x_{2\text{нов}})$ из выражения (7), которая должна быть центром тяжести точек q_1, q_2, q_3 с весами μ_0, μ_1, μ_2, причем $\mu_0 = 1 - (\mu_1 + \mu_2)$, т. е.

$$\begin{cases} x_{1\text{нов}} = \mu_0 z_1^1 + \mu_1 z_1^2 + \mu_2 z_1^3; \\ x_{2\text{нов}} = \mu_0 z_2^1 + \mu_1 z_2^2 + \mu_2 z_2^3 \end{cases}$$

или

$$\begin{cases} x_{1\text{нов}} = (1 - (\mu_1 + \mu_2)) z_1^1 + \mu_1 z_1^2 + \mu_2 z_1^3; \\ x_{2\text{нов}} = (1 - (\mu_1 + \mu_2)) z_2^1 + \mu_1 z_2^2 + \mu_2 z_2^3. \end{cases} \quad (9)$$

Подставляем теперь в выражение (9) μ_0, μ_1 и μ_2 из (8). Получим систему из двух линейных соотношений, связывающих точку x_1, x_2 и точку $x_{1\text{нов}}x_{2\text{нов}}$:

$$\begin{cases} x_{1\text{нов}} = u_{11}x_1 + u_{12}x_2 + u_1; \\ x_{2\text{нов}} = u_{21}x_1 + u_{22}x_2 + u_2. \end{cases}$$

Теперь нас будет интересовать знак детерминанта $\begin{vmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{vmatrix}$.

Если детерминант положителен, то точки p_1, p_2, p_3 и q_1, q_2, q_3 ориентированы одинаково, в противном случае это не так. Для размерности $n > 3$ рассуждения аналогичны.

Определение степени отображения

Теперь у нас задана триангуляция сферы, т. е. точки $p_1^i, p_2^i, ..., p_n^i$ и точки $q_1^i, q_2^i, ..., q_n^i$ (i — это номер симплекса). Рассмотрим интегральную сумму [2]

$$\frac{1}{\pi} \sum_{i=1}^{N} (\text{площадь симплекса}$$

$$\text{с вершинами } q_1^i, q_2^i, ..., q_n^i)\text{sgn}(u).$$

Предел этой суммы с учетом ориентации каждого слагаемого (ориентацию определяем через аффинное преобразование) при $\varepsilon \to 0$ назовем степенью отображения нашего преобразования Ф n-мерного пространства.

Использование вычисленной степени отображения

Дальнейшие рассуждения будут основаны на известной теореме [1]. Эта теорема говорит, что если наше преобразование Ф $(n-1)$-мерной сферы S (граница шара K) в единичную сферу Q имеет степень отображения плюс или минус единица, то внутри шара K имеется одна и только одна неподвижная точка. Если же внутри шара нет неподвижных точек, то степень отображения Ф сферы S в единичную сферу равна нулю.

Пусть у нас имеется случай нескольких неподвижных точек внутри сферы S (рис. 4).

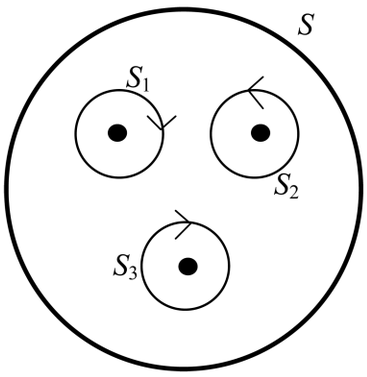


Рис. 4. Случай нескольких неподвижных точек

Расположим внутри сферы S маленькие сферы, каждая из которых окружает только одну неподвижную точку. Известно [1], что степень отображения Ф сферы S в единичную сферу складывается из степеней отображения маленьких S_i.

Отсюда следует, что при степени отображения, равной числу, отличному от нуля, внутри шара K обязательно имеются неподвижные точки. Для их выделения необходимо уменьшить число ε и продолжать процедуру.

Если же степень отображения равна нулю, это может означать отсутствие неподвижных точек или что их четное число (половина со степенью "+1", половина со степенью "−1").

Расчеты показывают, что в последнем случае сфера Q делится на связные области как положительные, так и отрицательные. Трудность представляет тот случай, когда неподвижных точек всего две.

В этом случае, как и в случае отсутствия неподвижных точек, связных областей всего две: одна отрицательная, другая положительная.

Во всех этих случаях требуется повторение процедуры при уменьшенном числе ε.

При компьютерной реализации метод несложно распараллелить (например, с помощью технологии OpenMP [3] или CUDA [4]) за счет разбиения пространства возможных решений на области. Актуальной остается проблема выделения (с помощью компьютерного моделирования) новых признаков отсутствия неподвижных точек.

Список литературы

1. **Александров П. С.** Комбинаторная топология. М., Л.: ОГИЗ, 1947. 660 с.

2. **Дубровин Б. А., Новиков С. П., Фоменко А. Т.** Современная геометрия. М.: Наука, 1979. 760 с.

3. **Антонов А. С.** Параллельное программирование с использованием технологии OpenMP. М.: Изд. МГУ, 2009. 77 с.

4. **Сандерс Дж., Кэндрот Э.** Технология CUDA в примерах: введение в программирование графических процессоров. М.: ДМК Пресс, 2011. 232 с.

УДК 519.857

В. И. Струченков, д-р техн. наук, проф.,
e-mail: str1942@mail.ru,
МГТУ МИРЭА, г. Москва

Математические модели и методы оптимизации в системах проектирования трасс новых железных дорог

Рассматривается задача оптимизации положения трассы новой железной дороги в заданной области варьирования. Предлагаются новые математические модели и алгоритмы проектирования продольного профиля по заданным вариантам плана трассы. Задача решается в несколько этапов во взаимосвязи с другими проектными задачами. Приводится оригинальный алгоритм построения приведенного градиента, основанный на использовании структурных особенностей системы ограничений.

Ключевые слова: *трасса, план, продольный профиль, нелинейное программирование, целевая функция, приведенный градиент*

Введение

Целесообразность оптимизации проектов строительства таких дорогостоящих объектов, как железные и автомобильные дороги, очевидна. В частности, в условиях пересеченного рельефа и сложной геологии затраты на строительство и последующую эксплуатацию могут быть существенно снижены при оптимальном расположении трассы проектируемой дороги на местности.

Задача разработки математических моделей и алгоритмов оптимизации трассы новой дороги впервые рассматривалась лет 50 тому назад. Большие ожидания связывались с новым для того времени методом динамического программирования. Однако и в настоящее время даже в наиболее совершенных современных системах автоматизированного проектирования (САПР), таких как CAD-1 [1] или ее российские аналоги Topomaic Robur [2] и Geonics [3], варианты трассы назначаются вручную. В этих системах компьютер используется для решения сопутствующих рутинных задач, но не как инструмент выработки оптимальных проектных решений.

Известно, что в одних и тех же условиях, располагая одной и той же информацией, различные

специалисты предлагают различные варианты проектных решений. Рассмотрение ограниченного числа интуитивно назначаемых вариантов не гарантирует близость к оптимуму конечного результата такого процесса. В то же время относительно небольшие изменения положения трассы на местности могут приводить к существенным изменениям затрат на строительство и эксплуатацию дороги [4].

Следовательно, проблема разработки адекватных математических моделей и математически корректных алгоритмов оптимизации трасс новых дорог остается актуальной. Это является главным направлением совершенствования САПР.

В этой статье мы постараемся ответить на следующие вопросы:

1. Почему не удается использовать динамическое программирование? Наш ответ на этот вопрос существенно отличается от ответов других авторов [4, 5].

2. В чем заключаются сложности разработки адекватных математических моделей рассматриваемой задачи?

3. Что мешает использовать нелинейное программирование?

В отличие от других авторов [4, 5] мы не отказываемся от поиска математически корректных алгоритмов оптимизации и не будем рассматривать разного рода эвристические алгоритмы, в частности генетические [4, 5].

В качестве первого шага рассмотрим более простую задачу — проектирование продольного профиля по заданным вариантам положения трассы в плане. Ее корректное решение позволит объективно количественно сравнивать варианты плана трассы. Более того, эта задача имеет и самостоятельное значение, когда возможностей варьирования в плане практически нет (например, в обжитых районах план трассы определяется условиями землепользования).

В отличие от других авторов [2, 3] эта задача будет рассматриваться не как геометрическая, а как технико-экономическая во взаимосвязи с другими проектными задачами.

Для ее решения используется нелинейное программирование. Но в отличие от стандартных алгоритмов новый алгоритм не требует решения систем линейных уравнений для поиска направления спуска на каждой итерации. Это позволило проектировать продольный профиль железнодорожного перегона (25...30 км) в приемлемое время на общедоступных компьютерах.

Цель настоящей статьи — проанализировать опыт разработки методов компьютерного проектирования трасс железных дорог и изложить идеи совершенствования математических моделей и алгоритмов оптимизации, которые были использованы при разработке соответствующей подсистемы САПР нового поколения. Детали реализации опущены из-за ограничения по объему статьи.

В последующих статьях будут рассмотрены возможности развития соответствующих алгоритмов для совместной оптимизации плана и профиля новых дорог.

Содержательная постановка задачи

Трасса — это гладкая трехмерная кривая, состоящая из элементов заданного вида и удовлетворяющая целому ряду ограничений.

Традиционно искомая трехмерная кривая представляется в виде двух плоских кривых: плана и продольного профиля. *План* — это проекция трассы на координатную плоскость *XOY*, а *продольный профиль* — это зависимость аппликаты *z* от длины дуги в плане. Задача проектирования оптимальной трассы превращается в две взаимосвязанные задачи: проектирование плана и проектирование продольного профиля.

На положение трассы влияют рельеф земли, геологические, гидрологические, климатические и другие условия.

Оптимальному варианту трассы должен соответствовать минимум взвешенной суммы затрат на строительство и последующую эксплуатацию дороги. Таким образом, трасса — это экстремаль некоторого функционала, а задача поиска оптимальной трассы может рассматриваться как задача вариационного исчисления.

Прежде всего отметим, что в явном виде не удается выразить функционал или записать уравнение трассы, т. е. формализовать задачу.

Требования к трассе железной дороги включают следующее.

В плане. Элементами плана трассы являются отрезки прямых и окружностей, сопрягаемые клотоидами. При этом длины элементов должны быть не менее заданных значений, радиусы кривых и параметры клотоид тоже ограничены. Соответствующие ограничения на план трассы выражаются нелинейными неравенствами относительно переменных, определяющих план [6].

В продольном профиле. Элементы продольного профиля — прямые, так что проектная линия — ломаная, на элементы которой также накладываются ограничения. В углы ломаной вписываются окружности или параболы [4]. Однако возникающими отклонениями от ломаной можно пренебречь в силу их малости [6].

Принципиальная особенность рассматриваемой задачи состоит в том, что в каждом конкретном случае неизвестно и число элементов плана, и число элементов профиля, т. е. размерность задачи, что затрудняет использование теории нелинейного программирования.

Опыт формализации задачи

Попытки игнорировать или как-то усреднить рельеф по отдельным участкам местности, с тем чтобы свести задачу поиска пространственной кривой (трассы) к поиску ее проекции на горизонтальную плоскость (плана) не привели к практическим результатам. В связи с этим приводимые в ряде работ [7] примеры поиска оптимальной трассы как оптимального пути на двумерной сетке следует рассматривать как учебные, не имеющие никакого отношения к реальной проектной задаче.

Проектирование трассы — пространственной кривой — предлагалось рассматривать как поиск оптимального пути на трехмерной сетке с помощью динамического программирования с последовательным сужением "коридора" поиска. Это предложение не привело к положительным результатам по следующим причинам:

- нельзя сравнивать варианты, приходящие в узел трехмерной сетки и отбраковывать все, кроме наилучшего из них, так как множества их продолжений не совпадают из-за ограничений по кривизне. Сравниваемые варианты должны иметь общий последний элемент, а не точку, что принципиально увеличивает вычислительные трудности;
- необходимо использовать мелкую сетку (приблизительно 10 м в плане и 0,01 м в профиле), иначе накапливаются ошибки поиска и получается вариант, далекий от оптимума. Действительно, при проектировании трассы допускается использование отрезков круговых кривых и прямых вставок 20...30 м. Очевидно, размеры сетки должны быть меньше этих значений;
- получаемая в результате такой оптимизации трасса, как ломаная линия, должна быть как-то с малыми отклонениями преобразована в реальную трассу (с окружностями и клотоидами в плане). Такое преобразование представляет отдельную серьезную задачу.

Отметим, что поиск трассы как пространственной кривой сводится к многоэкстремальной задаче нелинейного программирования с нелинейной системой ограничений [6].

В силу отмеченных трудностей и по ряду других причин для поиска оптимальной трассы как пространственной кривой пока не созданы приемлемые для практических целей математические модели и алгоритмы проектирования.

В связи с этим в качестве первого шага была решена задача проектирования продольного профиля по заданному варианту плана трассы.

Эта задача существенно проще, так как фиксируется сразу несколько составляющих строительных и эксплуатационных затрат и появляется возможность формализовать задачу.

Для решения этой задачи также предлагалось использовать динамическое программирование [8]. Однако если при сооружении насыпей предполагается использовать грунты выемок (что обычно и делается), то возникает взаимосвязь положения соответствующих элементов проектной линии, что не позволяет использовать динамическое программирование.

Более успешным было применение нелинейного программирования. Программы, проектирующие продольный профиль, получившие практическое применение еще в 70-х годах [9], были основаны на простых математических моделях:

- поперечные профили земли принимались односкатными;
- конструкции проектных поперечных профилей фиксировались, т. е. не было совместного проектирования продольного и поперечных профилей;
- наличие нескольких слоев грунта не учитывалось;
- продольный профиль земли сглаживался для уменьшения размерности задачи.

Программа для БЭСМ-4 могла проектировать локальные участки (до 5 км) [9].

Эти упрощения были вынужденными из-за ограниченных вычислительных возможностей в то время.

Задача сводилась к следующему. Найти $\min\Phi(\mathbf{x}, \mathbf{c})$ при $\mathbf{Ax} \leqslant \mathbf{b}$, где $\mathbf{x}$ — вектор неизвестных, $\mathbf{c}$ — вектор параметров, матрица $\mathbf{A}$ и вектор $\mathbf{b}$ — задают систему линейных ограничений.

Современные вычислительные возможности позволили создать систему, в которой продольный и поперечный профили проектируются совместно в пределах перегона. При этом формально решается задача того же вида, но используются новые математические модели и алгоритм оптимизации.

Проектирование продольного профиля как задача нелинейного программирования

Если обозначить профиль земли $H(s)$, а проектную линию $Z(s)$, то в первом приближении задача состоит в следующем. По заданной $H(s)$ найти такую ломаную $Z(s)$, чтобы она удовлетворяла всем ограничениям, и был

$$\min \int_0^{S_0} F(Z(s),\ H(s),\ s)ds, \qquad (1)$$

где s — текущая длина от начала; S_0 — длина трассы в плане, а функция F моделирует затраты на элементе длины.

Реальные модели должны учитывать конструкции поперечных профилей земляного полотна, наличие водопропускных и других искусственных сооружений, распределение земляных масс, способы производства земляных работ и др.

Задача вариационного исчисления (1) сводится к задаче нелинейного программирования, обладающей интересными особенностями независимо от конкретного вида функции F.

Поскольку число элементов искомой ломаной неизвестно, то приходится считать, что переломы профиля земли и проектной линии (т. е. профиля трассы) имеют одни и те же абсциссы. Профиль земли всегда представлен в виде ломаной с неравномерным шагом, и такое допущение позволяет фиксировать число элементов n (размерность задачи) и длины s_i элементов (в плане). При этом получится ломаная с бóльшим, чем нужно, числом элементов, но из-за многочисленных ограничений ее отклонения от окончательной $Z(s)$ невелики [9]. Идея в том, чтобы найти эту ломаную путем решения задачи оптимизации, затем преобразовать ее в ломаную с элементами, длины которых не менее допустимой, определив тем самым реальную размерность задачи и начальное приближение, и на последнем этапе выполнить оптимизацию при всех ограничениях и необходимых уточнениях целевой функции. Такой многоэтапный процесс с уточнением математической модели и ее параметров является обычным для решения сложных проектных задач творческого характера.

Система ограничений. Зная число и длины элементов искомой ломаной, можно аналитически выразить все ограничения на $Z(s)$, если принять в качестве неизвестных z_i ($i = 1, 2, ..., n$) ее ординаты в точках перелома. Эти ограничения делятся на три группы.

1. На ординаты в отдельных точках $z_i \leqslant z_i^{\max}$ или $z_i \geqslant z_i^{\min}$.

2. На уклоны элементов профиля

$$a_i \leqslant (z_{i+1} - z_i)/s_i \leqslant b_i \ (i = 1, 2, ..., n - 1),$$

где s_i — длины элементов. Эти ограничения являются дискретным аналогом ограничения на первую производную.

3. На разности уклонов смежных элементов:

$$c_i \leqslant (z_{i+2} - z_{i+1})/s_{i+1} - (z_{i+1} - z_i)/s_i \leqslant d_i.$$

Здесь a_i, b_i, c_i, d_i — константы, вычисляемые по исходным данным.

Эти ограничения являются дискретным аналогом ограничения на кривизну.

В силу малости проектных уклонов длина элемента и его проекции практически совпадают.

Система ограничений имеет четко выраженную структуру.

Ограничениям первой группы соответствует матрица, у которой в каждой строке все элементы равны нулю, кроме одного, который равен 1 или −1, а в каждом столбце не более двух ненулевых элемента. Ограничениям второй группы соответствует двухдиагональная матрица (два блока с разными знаками), а ограничениям третьей группы — трехдиагональная матрица (два блока с разными знаками). Именно эта структура позволяет решить в приемлемое время на общедоступных компьютерах возникающую задачу нелинейного программирования, размерность которой при проектировании реальных объектов достигает 1000 переменных и соответственно более 4000 ограничений.

Оставляя пока вопрос о конкретных моделях целевой функции и соответственно об алгоритме вычисления ее градиента, рассмотрим, как в новом алгоритме используется структура системы ограничений.

Широко известные алгоритмы нелинейного программирования с линейной системой ограничений [10, 11] представляют собой итерационный процесс, состоящий из следующих шагов:

1. Вычисление допустимого начального приближения $\mathbf{z}^0$.

2. Вычисление градиента целевой функции $\mathbf{f}^0$.

3. Определение множества активных ограничений (активного набора).

4. Построение направления спуска $\mathbf{p}^0$ в граничном линейном многообразии.

5. Проверка условий прекращения счета и, если они выполнены, то окончание процесса, иначе следующий шаг.

6. Вычисление шага λ и новой итерационной точки $\mathbf{z}^{k+1} = \mathbf{z}^k + \lambda\mathbf{p}^k$ и переход к п. 2.

Эти алгоритмы отличаются способом построения вектора спуска [10, 11]. Если в качестве направления спуска используется проекция $\mathbf{p}$ антиградиента $-\mathbf{f}$, то стандартный алгоритм для вычисления проекции требует на каждой итерации решать систему линейных уравнений.

Так, по формуле Розена [10]

$$\mathbf{p} = -(\mathbf{E} - \mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}\mathbf{A})\ \mathbf{f}. \qquad (2)$$

Здесь $\mathbf{E}$ — единичная матрица, $\mathbf{A}$ — матрица активных на данной итерации ограничений, а верхний индекс $^{\mathrm{T}}$ означает транспонирование. При большой размерности задачи это приводит к неприемлемым затратам машинного времени.

В использовавшейся ранее программе для вычисления проекции антиградиента на каждой итерации вместо систем с матрицей $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ решались системы линейных уравнений малой размерности [9].

Однако наличие структурных особенностей системы ограничений позволяет при любой комбинации активных ограничений построить базис в соответствующем граничном линейном многообразии и определять направление спуска вообще без решения каких-либо систем линейных уравнений. На этом основан новый алгоритм.

Действительно, пусть мы знаем этот базис, и его столбцы составляют матрицу $\mathbf{C}$.

В качестве направления спуска принят $\mathbf{p}^* = -\mathbf{C}\mathbf{C}^{\mathrm{T}}\mathbf{f}$, т. е. приведенный антиградиент [10, 11]. Для его вычисления достаточно построить только базисные векторы, вычислить $\mathbf{C}^{\mathrm{T}}\mathbf{f}$, а потом умножить $\mathbf{C}$ на результат.

В силу того что $(\mathbf{p}^*, -\mathbf{f}) = (-\mathbf{C}\mathbf{C}^{\mathrm{T}}\mathbf{f}, -\mathbf{f}) = (\mathbf{C}^{\mathrm{T}}\mathbf{f}, \mathbf{C}^{\mathrm{T}}\mathbf{f}) > 0$ при $\mathbf{f} \neq \mathbf{0}$ вектор $\mathbf{p}^*$ — это направление спуска, которое можно использовать вместо проекции антиградиента.

Но для решения задачи надо найти еще и способ проверки возможности исключения ограничений из активного набора. В методе проекции градиента [10, 11] для этого есть формула $\mathbf{u} = -(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}\mathbf{A}\mathbf{f}$, где $\mathbf{u}$ — вектор разложения $\mathbf{f} - \mathbf{p}$ по нормалям, т. е. $-\mathbf{f} - \mathbf{p} = \mathbf{A}^{\mathrm{T}}\mathbf{u}$. Эта формула опять-таки требует вычисления $(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}$ (решения системы линейных уравнений).

Вектор $\mathbf{n}^* = -\mathbf{f} - \mathbf{p}^*$ *не является нормалью к граничному многообразию и не может быть представлен в виде* $\mathbf{A}^{\mathrm{T}}\mathbf{u}^*$.

Один из путей решения вопроса об исключении ограничений из активного набора состоит в построении векторов, каждый из которых нарушает ***одно и только одно*** ограничение. Эти векторы (матрица **B)** образуют базис в ортогональном дополнении к нуль-пространству матрицы **A** и в совокупности с векторами из **C** дают полный базис. В этом базисе матрица активных ограничений диагональная, что упрощает задачу. Например, для i-го ограничения построен такой вектор $\mathbf{d_i}$. Поскольку скалярное произведение $(\mathbf{a_i}^{\mathrm{T}}, \mathbf{d_i}) > 0$, то это ограничение можно исключить из активного набора при $(\mathbf{d_i}, \mathbf{f}) > 0$ [12]. Здесь $\mathbf{a_i}$ — i-я строка матрицы активных ограничений, а $\mathbf{f}$ — градиент целевой функции.

Это правило надо применять ко всем векторам, каждый из которых нарушает одно и только одно ограничение. Как только будет найдено ограничение, которое можно исключить, дальнейший поиск можно прекратить, добавить вектор $\mathbf{b}^i$ в матрицу **C**, перевычислить приведенный антиградиент и продолжить процесс оптимизации. Характерно, что перевычисление приведенного антиградиента не требует работы с матрицами и сводится к вычитанию из каждой j-й компоненты имеющегося приведенного антиградиента величины $d_j^i(\mathbf{d_i}, \mathbf{f})$. Это не сложно, тем более, что $(\mathbf{d_i}, \mathbf{f})$ уже вычислен при анализе возможности исключения ограничения из числа активных.

Важно отметить, что при таком построении дополнительных базисных векторов имеется возможность исключать не одно, а сразу несколько ограничений из активного набора [12]. Если принять меры по предотвращению " зигзагов" [13], это открывает возможность быстрее сформировать нужный набор активных ограничений, ускорить сходимость и сократить время счета.

Вернемся к анализу системы ограничений и покажем, как строить базис и исключать ограничения из активного набора.

Пусть активный набор составлен из ограничений группы 2, т. е. на некотором участке трасса идет предельным уклоном $z_{i+1} - z_i = s_i\, b_i$ ($i = 1, 2, \ldots, r-1$). В данном случае размерность нуль-пространства **M** матрицы активных ограничений равна единице. У базисного вектора $\mathbf{c} \in \mathbf{M}$ все компоненты равны между собой. Принимаем $c_i = 1$ ($i = 1, \ldots, r$). Это участок сдвига. Следовательно, если $\mathbf{p}$ — приведенный градиент, то $p_j = \sum_{i=1}^{r} f_i$, но если $\mathbf{p}$ — проекция градиента, то $p_j = \sum_{i=1}^{r} f_i/r$ ($j = 1, 2, \ldots, r$). И проекция антиградиента, и приведенный антиградиент задают одно и то же направление, хотя и различаются множителем $1/r$. Базисный вектор в ортогональном дополнении к нуль-пространству матрицы **A**, нарушающий только ограничение $\mathbf{z}_{k+1} - \mathbf{z}_k = \mathbf{s}_k\mathbf{b}_k$, если оно было активно, имеет все нулевые компоненты с 1-й до k-й включительно, а остальные равны единице. Его скалярное произведение на антиградиент равно сумме компонент антиградиента с $(k + 1)$-й по последнюю и, если оно положительно, то ограничение можно исключить.

Для ограничений вида 3 (по разности уклонов) размерность нуль-пространства матрицы **A** равна 2. Это участок поворота.

В качестве базисных векторов можно взять векторы, соответствующие повороту проектной линии с центром в начальной и конечной точках соответствующего участка. При этом все уклоны получат равные приращения, а их разности не изменятся. Задав это приращение уклонов δ (например, $\delta = 1$), последовательно вычисляем компоненты базисного вектора **c**:

$$c_1 = 0;\ c_2 = \delta s_1;\ c_3 = \delta(s_1 + s_2);\ \ldots;$$
$$c_r = \delta(s_1 + s_2 + \ldots + s_{r-1}).$$

Здесь нумерация условная, реально первый номер соответствует началу участка предельной кривизны. Аналогично для второго вектора, т. е. поворота с центром в конце участка.

Реально возможны и комбинации активных ограничений из трех групп. Например, на участке, где активны ограничения группы 2 (трасса идет предельным уклоном), одновременно активно одно ограничение группы 1 (если это ограничение остается активным, то нельзя изменить соответствующее z_i). Становится очевидным, что все компоненты вектора спуска для этого участка должны быть равны нулю.

Далее, наличие одного активного ограничения группы 2 на участке, где активны все ограничения группы 3, приводит к тому, что все компоненты вектора спуска на этом участке должны быть равны между собой (участок сдвига). Только при этом условии сохранят активность все ограничения. Именно эти изменения проектной линии при соответствующей комбинации активных ограничений находят отражение в структуре базисных векторов.

Возможны и более сложные комбинации активных ограничений. Однако в любом случае базисные векторы можно построить, анализируя участки сдвига (активны ограничения по уклонам) или поворота (активны ограничения по разности уклонов). Наличие активного высотного ограничения (группа 1)

в некоторой точке на участке поворота оставляет только один базисный вектор, соответствующий повороту с центром в этой точке. Наличие двух таких точек фиксирует весь участок, равно как наличие одной такой точки на участке сдвига.

Начальное приближение для этого процесса строится следующим образом.

1. Берется прямая, соединяющая начальную и конечную точки профиля, которые заданы.

2. Если заданы уклоны примыкания слева (в начале трассы) или справа (в конце трассы), то с использованием предельных разностей уклонов изменяются последовательно уклоны в начале и в конце проектной линии, так что в итоге все ограничения по уклонам и разностям смежных уклонов выполнены.

3. В качестве целевой функции берется сумма квадратов невязок в высотных ограничениях (группа 1), и запускается процесс оптимизации линии, полученной в п. 2.

В итоге имеем или допустимое по всем ограничениям (включая высотные) начальное приближение, или сообщение о невозможности решения задачи.

Остается рассмотреть, как строится модель целевой функции.

Математические модели целевой функции. Если задача решается на минимум объемов земляных работ, то подынтегральная функция в (1) — это площадь поперечного сечения земляного полотна, которая зависит от поперечного профиля земли и конструкции проектного поперечного профиля.

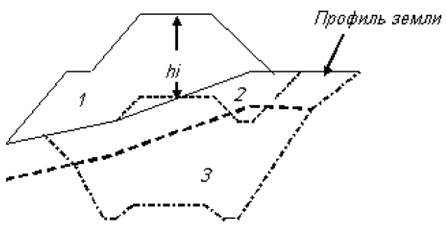
Поскольку в целевой функции (1) $Z(s)$ и $H(s)$ представлены *ломаными с совпадающими абсциссами переломов*, то можно выразить аналитически объемы земляных работ при заданных конструкциях поперечных профилей земляного полотна [9]. В общем случае для реальных поперечников земли и конструкций проектных поперечных профилей (см. рисунок) мы получаем кусочно-квадратические зависимости площади F_i от рабочей отметки по оси h_i:

$$h_i = z_i - H_i. \qquad (3)$$

Объем земляных работ вычисляется через площади по формулам численного интегрирования (в программе по формуле трапеций).

На i-м элементе объем

$$V_i = (F_i(h_i) + F_{i+1}(h_{i+1}))s_i/2.$$



Поперечные профили:
1 — насыпь; *2* — полунасыпь — полувыемка; *3* — выемка

Следовательно, для $i = 2, \ldots, n-1$

$$\frac{\partial V}{\partial h_i} = \frac{\partial F_i}{\partial h_i}(s_{i-1} + s_i)/2. \qquad (4)$$

Если h_1 не фиксирована, то при $i = 1$ принимаем в (4) $s_0 = 0$, аналогично для $i = n$ $s_n = 0$. Формулы (3) и (4) позволяют вычислить градиент.

Расчет выполняется в несколько этапов. На первом этапе поперечные профили земли принимаются односкатными, а проектные поперечные профили в насыпях и в выемках — в виде трапеций.

Полученная проектная линия рассматривается как ось зоны поиска при дальнейших уточнениях.

Реальные конструкции поперечных профилей земляного полотна зависят от геологии. В системе проектирования используются типовые проектные конструкции, для которых на каждом переломе продольного профиля земли в зоне поиска с заданным шагом Δ изменения рабочей отметки вычисляются площади насыпей и выемок. Приближенные зависимости $F_i(h)$ получаются параболической аппроксимацией полученных значений и используются для продолжения оптимизации.

При необходимости расчет повторяется при сужении зоны поиска и уменьшении шага Δ.

Если целевая функция соответствует строительным затратам, то грунты выемок подразделяются на четыре вида в соответствии с возможностями их использования для сооружения насыпей:

1) непригодные;
2) обыкновенные;
3) дренирующие нескальные;
4) скальные.

Площади и объемы вычисляются для каждого вида грунтов отдельно.

Дополнительно для грунтов каждого вида задаются в расчете на 1 м^3: q_1 — затраты на сооружение насыпи из выемки, q_2 — насыпи из грунта карьеров (резервов), q_3 — разработки непригодного или излишнего грунта выемок в отвал. Эти данные могут отличаться на различных участках проектируемого перегона. Затраты на сооружение земляного полотна K вычисляются через объемы насыпей v_f и выемок v_c в зависимости от соотношения объёмов:

$$\text{если } v_f > v_c,$$
$$\text{то } K = q_1 v_c + q_2(v_f - v_c) = q_2 v_f + (q_1 - q_2)v_c;$$
$$\text{если } v_f \leqslant v_c,$$
$$\text{то } K = q_1 v_f + q_3(v_c - v_f) = (q_1 - q_3)v_f + q_3 v_c.$$

Наличие непригодного грунта учитывается отдельно.

Характерно, что коэффициенты при v_f, v_c (приведенные единичные стоимости) меняются от итерации к итерации, если меняется соотношение объемов. Поэтому на каждой итерации вычисляются соответствующие площади и объемы в целом по участку, на котором насыпи и выемки сооружа-

ются совместно, затем определяются приведенные единичные стоимости для грунтов каждого вида и далее градиент целевой функции. Таким образом, учитывается взаимосвязь положения проектной линии в насыпях и выемках, сооружаемых совместно. Именно эта взаимосвязь не позволяет при оптимизации по строительным затратам использовать динамическое программирование, но учитывается в целевой функции при использовании нелинейного программирования. Если задать единичные стоимости в расчете на какое-либо соотношение объемов (например, насыпей больше, чем выемок), то в условиях пересеченного рельефа получается проектная линия с обратным соотношением объемов.

В строительную стоимость включаются и затраты на искусственные сооружения, для чего задаются зависимости их стоимостей от рабочих отметок. Эти зависимости уточняются в процессе проектирования, так же как и типы искусственных сооружений.

Целесообразность включения в целевую функцию при проектировании продольного профиля эксплуатационных затрат по передвижению поездов сомнительна по следующим причинам:

- эти затраты зависят от размеров движения, прогнозировать которые на расчетный период эксплуатации в условиях нестабильной экономики не представляется возможным;
- использование моделей целевой функции с включением в нее данного вида затрат, вычисляемых по равновесным скоростям движения, показало, что они дают изменения проектной линии в пределах 0,2 м. Но при вариациях проектной линии в пределах 0,2 м изменения этого вида затрат находятся в пределах точности их расчета.

Полученная в результате оптимизации проектная линия не удовлетворяет ограничениям по минимальной длине элемента. При ее преобразовании к окончательному виду возможны отклонения, которые при действующих нормах проектирования не превышают 0,4 м [9]. Это преобразование в заданной полосе отклонений с шагом 0,02 м выполняется с помощью алгоритма динамического программирования [14]. Целевая функция при этом соответствует объемам земляных работ.

Полученная в результате линия нужна только как начальное приближение. И все проведенные расчеты были нужны только для установления числа элементов (размерность задачи) и начального приближения для последнего этапа оптимизации с использованием алгоритма нелинейного программирования.

Если речь идет о проектировании продольного профиля для сравнения вариантов плана трассы, то это сравнение может выполняться с использованием уже полученных результатов. Для получения окончательного варианта проектной линии проводится еще один этап расчетов. На этом этапе новыми переменными являются проектные отметки переломов проектной линии, через которые, в силу линейности элементов, легко вычисляются отметки во всех точках перелома профиля земли (старые переменные). Новых переменных примерно на порядок меньше, чем старых.

Практически используются те же программы оптимизации, добавлен только пересчет производных целевой функции по новым переменным через производные по старым переменным.

Заключение

Усовершенствованные математические модели и новый алгоритм оптимизации позволяют решать задачу комплексно, при наличии данных различной полноты и детальности, с использованием различных критериев оптимизации.

Новые модели и алгоритм совместного проектирования продольного и поперечных профилей являются основой соответствующей подсистемы САПР нового поколения. Они могут использоваться как для проектирования реальных объектов, так и в исследовательских целях. Применение упрощенных моделей, в частности, поиск наилучшего среднеквадратического приближения [2], а также использование различного рода эвристических алгоритмов [1] представляется нецелесообразным.

Расчеты на персональном компьютере с тактовой частотой 2 ГГц и ОЗУ 512 Мбайт позволили сделать следующие выводы.

- Время счета при длине перегона до 30 км составляет 2...3 мин, при последующих расчетах существенно меньше, что вполне приемлемо.
- Наибольшее влияние на проектную линию оказывают данные по грунтам и соответствующие единичные стоимости по земляным работам.
- Нельзя пренебречь влиянием затрат на искусственные сооружения.
- Уточнение типовых конструкций поперечных профилей земляного полотна, а также уточнение типов искусственных сооружений не столь существенно, как изменение данных по грунтам и единичным стоимостям.

В следующих статьях будет показано, как при незначительных изменениях рассмотренные модели и алгоритм оптимизации можно применить при проектировании продольного профиля автомобильных дорог.

Список литературы

1. **CARD/1**. URL: http://www.card-1.com/en/home/ Visited: July 14, 2012.

2. **Topomatic** Robur. URL: http:// www.topomatic.ru Visited: July 14, 2012.

3. **Курилко Ю., Чешева В.** Geonics ЖЕЛДОР — САПР // CADmaster. 2007. № 1(36).

4. **Shafahi Y., Shahbazi M. J.** Optimum railway alignment. URL: http://www.uic.org/cdrom/2001/wcrr2001/pdf/sp/2_1_1/210.pdf Visited: July 12, 2012.

5. **Jha M. K., Schonfeld P. M., Yong J.-C. and Kim E.** Intelligent Road Design // WIT Press, Southampton. 2006.
6. **Струченков В. И.** Методы оптимизации в прикладных задачах. М.: Солон-Пресс., 2009.
7. **Вентцель Е. С.** Исследование операций: задачи, принципы, методология. М.: КноРус, 2010.
8. **Михалевич В. С., Быков В. И., Сибирко А. Н.** К вопросу проектирования оптимального продольного профиля дороги // Транспортное строительство. 1975. № 6.
9. **Использование** математических методов оптимизации и ЭВМ при проектировании продольного профиля железных дорог // Тр. Всесоюзного НИИ транспортного строительства. М.: Транспорт, 1977. Вып. 101.
10. **Гилл Ф., Мюррей У, Райт М.** Практическая оптимизация: Пер. с англ. М.: Мир, 1985.
11. **Аоки М.** Введение в методы оптимизации: Пер. с англ. М.: Наука, 1977.
12. **Струченков В. И.** Методы оптимизации в проектировании трасс линейных сооружений // Сб. научных трудов. Искусственный интеллект в технических системах. М.: Гос. ИФТП, 1999. Вып. № 20.
13. **Зойтендейк Дж. Г.** Методы возможных направлений: Пер. с англ. М.: Изд-во иностр. лит., 1963.
14. **Струченков В. И., Козлов А. Н., Егунов А. С.** Кусочно-линейная аппроксимация плоских кривых при наличии ограничений // Информационные технологии. 2010. № 12(172).

УДК 004.023, 519.854.2

В. А. Чеканин, канд. техн. наук, доц.,
e-mail: vladchekanin@rambler.ru,
А. В. Чеканин, д-р техн. наук, проф., зав. каф.,
e-mail: avchekanin@rambler.ru,
ФГБОУ ВПО "Московский государственный технологический университет "СТАНКИН"

Алгоритм решения задач ортогональной упаковки объектов на основе мультиметодной технологии

Рассматривается мультиметодный генетический алгоритм оптимизации решения NP-полных задач ортогональной упаковки объектов. Для мультиметодного генетического алгоритма предлагаются новые эвристики размещения. Эффективность применения мультиметодного генетического алгоритма с разработанными эвристиками исследуется на эталонных задачах двухмерной контейнерной упаковки на листы и на полубесконечную полосу.

Ключевые слова: *задача ортогональной упаковки, мультиметодный генетический алгоритм, эвристики, генетический алгоритм, дискретная оптимизация, вычислительный эксперимент*

Введение

Задача ортогональной упаковки объектов представляет собой важный прикладной раздел комбинаторной оптимизации. Широкий спектр применения решений этой задачи в различных сферах экономической деятельности объясняет повышенный интерес исследователей к совершенствованию методов ее решения.

Сложность решения задачи ортогональной упаковки обусловлена ее принадлежностью к классу NP-полных задач [1]. Практическое применение методов решения задач упаковки, использующих полный перебор вариантов, оказывается неэффективным из-за больших затрат временных ресурсов. В связи с этим одним из перспективных направлений исследований является разработка и совершенствование различных приближенных, а также эвристических методов решения задачи упаковки, в том числе эволюционных алгоритмов.

В общем виде задача ортогональной упаковки объектов размерности D описывается следующим образом: имеются набор N ортогональных контейнеров (D-мерных параллелепипедов) с габаритными размерами $\{W_j^1, W_j^2, ..., W_j^D\}$, $j = 1, ..., N$, и набор n ортогональных объектов (D-мерных параллелепипедов) с габаритными размерами $\{w_i^1, w_i^2, ..., w_i^D\}$, $i = 1, ..., n$. Обозначим положение объекта i в j-м контейнере следующим образом: $(x_{ij}^1, x_{ij}^2, ..., x_{ij}^D)$. Необходимо разместить все объекты в минимальном числе контейнеров при выполнении следующих условий:

- ребра размещенных в контейнере ортогональных объектов параллельны ребрам этого контейнера;
- размещенные объекты не перекрывают друг друга, т. е.

$$\forall j = 1, ..., N, \forall d = 1, ..., D, \forall i, k = 1, ..., n, i \neq k$$
$$(x_{ij}^d \geqslant x_{kj}^d + w_k^d) \vee (x_{kj}^d \geqslant x_{ij}^d + w_i^d);$$

- размещенные объекты не выходят за границы контейнеров, т. е.

$$\forall j = 1, ..., N, \forall d = 1, ..., D, \forall i = 1, ..., n$$
$$(x_{ij}^d \geqslant 0) \wedge (x_{ij}^d + w_i^d \leqslant W_j^d).$$

1. Мультиметодный генетический алгоритм

Использование при решении задач упаковки с помощью генетических методов оптимизации хромосом, содержащих последовательность выбора размещаемых объектов, не всегда оказывается эф-

фективным, особенно при увеличении числа размещаемых объектов. В настоящей работе реализована мультиметодная технология, в основу которой положена идея кодирования решения задачи в виде последовательности алгоритмов решения задачи, называемых эвристиками [2]. Мультиметодный генетический алгоритм (МГА) использует набор хромосом, содержащих последовательность применяемых эвристик. Эвристики при решении задачи упаковки представляют собой алгоритмы выбора размещаемых объектов и областей контейнера для размещения выбранных объектов. Задача поиска оптимального решения задачи упаковки сводится к задаче поиска оптимальной последовательности применяемых эвристик размещения.

Разработанные эвристики размещения применимы при использовании модели "виртуальные объекты" представления ортогональных объектов в контейнерах [3]. В модели "виртуальные объекты" пространство контейнера описывается набором узлов, содержащих ссылки на присоединенные к ним объекты, и набором свободных узлов, содержащих виртуальные объекты. Виртуальный объект представляет собой ортогональный объект наибольшего объема, который может быть присоединен к узлу без перекрытий с размещенными в контейнере объектами.

Для МГА применительно к модели "виртуальные объекты" разработаны следующие эвристики размещения ортогональных объектов:

1) **WF** (Width Fit): присоединение к узлу k текущего контейнера наиболее подходящего объекта i вдоль заданного направления l упаковки:

$$(p_k^l - w_i^l) \to \min,\ w_i^d \leqslant p_k^d \quad \forall d = 1, ..., D,$$

где p_k^l — габаритный размер виртуального объекта узла k, измеренный в направлении l;

2) **SF** (Square Fit): присоединение к узлу k текущего контейнера первого подходящего объекта i с максимальным объемом:

$$\left(\prod_{d=1}^{D} p_k^d - \prod_{d=1}^{D} w_i^d \right) \to \min,\ w_i^d \leqslant p_k^d \quad \forall d = 1, ..., D;$$

3) **NWF** (Next Width Fit): присоединение первого объекта ($i = 1$) из списка упорядоченных по одному из габаритных размеров неразмещенных объектов к ближайшему свободному узлу k текущего контейнера:

$$\forall h < k \exists d: p_h^d < w_i^d,\ w_i^d \leqslant p_k^d \quad \forall d = 1, ..., D; \quad (1)$$

4) **NSW** (Next Square Fit): присоединение первого объекта ($i = 1$) из списка упорядоченных по объему неразмещенных объектов к ближайшему свободному узлу k текущего контейнера при выполнении условия (1).

Первые две эвристики (WF, SF) осуществляют выбор наиболее подходящего объекта для его присоединения к определенному узлу контейнера. Третья и четвертая эвристики (NWF, NSF) работают по обратному принципу — осуществляют поиск такой области контейнера, в которой присоединение заданного объекта наиболее оптимально. Разработанные эвристики обеспечивают как оптимизацию последовательности выбора размещаемых объектов, так и оптимизацию использования свободного пространства контейнера.

2. Вычислительные эксперименты

Анализ эффективности эвристик и МГА проводился при решении эталонных задач двухмерной контейнерной упаковки из библиотеки OR-library [4] для наборов двухмерных прямоугольных объектов, взятых из задач **2DBPP** (2D Bin Packing Problem), сформулированных S. P. Fekete и J. Schepers [5] с известными точными нижними границами решения.

Целевой функцией является число заполненных объектами контейнеров. Значения целевых функций решений сравнивались с теоретически рассчитанными и экспериментально подтвержденными в работе [6] нижними границами, основанными на двойственно выполнимых функциях, рассчитанными по алгоритму S. P. Fekete и J. Schepers.

Задачи решались с использованием разработанного прикладного программного обеспечения **Packer** [7], функционирующего на платформе Microsoft Win32. Эксперименты проводились на персональной ЭВМ (ЦП — AMD 1,79 ГГц, ОЗУ — 896 Мбайт).

В ходе каждого вычислительного эксперимента проводилась серия из 100 экспериментов решения задач ортогональной упаковки с объемом выборки $m = 40, 50, 100, 150, 250, 500$ и 1000 объектов. Диапазоны изменения размеров объектов в тестовых задачах трех различных типов приведены в табл. 1, 2.

Таблица 1

Геометрические параметры размещаемых объектов

Класс объектов	Характеристики объектов	Интервал распределения длины объектов	Интервал распределения ширины объектов
Класс 1	Широкие	[1, 50]	[75, 100]
Класс 2	Длинные	[75, 100]	[1, 50]
Класс 3	Большие	[50, 100]	[50, 100]
Класс 4	Маленькие	[1, 50]	[1, 50]

Таблица 2

Типы решаемых тестовых задач

Тип задач	Соотношение классов размещаемых объектов, %			
	Класс 1	Класс 2	Класс 3	Класс 4
1 (ngcutfs 1)	20	20	20	40
2 (ngcutfs 2)	15	15	15	55
3 (ngcutfs 3)	10	10	10	70

Показателем качества размещения служит относительное отклонение μ от нижней границы, которое рассчитывается по формуле

$$\mu(\%) = \frac{S - \Delta^{(p)}}{n} 100\ \%,$$

где S — целевая функция решения (число заполненных объектами контейнеров); $\Delta^{(p)}$ — нижняя граница задачи (минимально возможное число заполненных контейнеров), n — суммарное число размещенных в контейнерах объектов [8]. Более плотному размещению объектов соответствует меньшее значение μ.

2.1. Анализ эвристик мультиметодного генетического алгоритма

В табл. 3 приведены результаты тестирования разработанных эвристик на тестовых задачах S. P. Fekete и J. Schepers двухмерной контейнерной упаковки объектов на листы.

Из построенной на основе табл. 3 диаграммы (см. рисунок) видно, что невозможно выделить эвристику, которая наилучшим образом обеспечивает размещение объектов для всех классов задач ортогональной упаковки. Поэтому начальная популяция решений в МГА содержит хромосомы, состоящие только из определенных эвристик, и хромосомы, содержащие равновероятный набор эвристик, наилучшее соотношение которых выбирается в ходе эволюционного поиска оптимального решения.

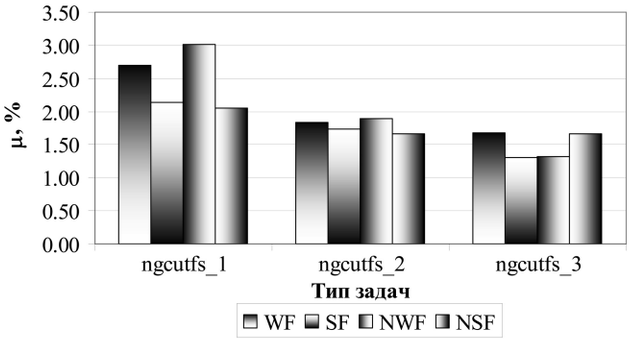


Диаграмма эффективности применения эвристик

2.2. Сравнительный анализ генетического алгоритма и МГА

Эффективность применения генетических методов оптимизации для решения задач упаковки показана в работах [8—10]. Проведенные в работе [11] вычислительные эксперименты показали, что при решении задачи ортогональной упаковки объектов

Таблица 3

Результаты тестирования эвристик на тестовых задачах двухмерной контейнерной упаковки на листы S. P. Fekete, J. Schepers

Тестовые задачи ngcutfs			Эвристика							
			WF		SF		NWF		NSF	
Задача	m	$\Delta^{(p)}$	S	μ	S	μ	S	μ	S	μ
1_11	40	10	12	5,00	12	5,00	12	5,00	12	5,00
1_41	50	14	16	4,00	15	2,00	16	4,00	15	2,00
1_71	100	24	26	2,00	26	2,00	27	3,00	26	2,00
1_101	150	36	40	2,67	39	2,00	40	2,67	38	1,33
1_131	250	58	63	2,00	62	1,60	64	2,40	62	1,60
1_161	500	123	131	1,60	129	1,20	133	2,00	129	1,20
1_191	1000	241	257	1,60	253	1,20	261	2,00	253	1,20
Среднее отклонение, %			—	**2,70**	—	**2,14**	—	**3,01**	—	**2,05**
2_11	40	11	12	2,50	12	2,50	12	2,50	12	2,50
2_41	50	9	10	2,00	10	2,00	10	2,00	10	2,00
2_71	100	23	25	2,00	25	2,00	25	2,00	25	2,00
2_101	150	31	34	2,00	34	2,00	34	2,00	34	2,00
2_131	250	53	59	2,40	58	2,00	59	2,40	57	1,60
2_161	500	99	104	1,00	103	0,80	105	1,20	103	0,80
2_191	1000	205	215	1,00	213	0,80	217	1,20	212	0,70
Среднее отклонение, %			—	**1,84**	—	**1,73**	—	**1,90**	—	**1,66**
3_11	40	7	8	2,50	7	0,00	7	0,00	8	2,50
3_41	50	10	12	4,00	12	4,00	12	4,00	12	4,00
3_71	100	15	17	2,00	17	2,00	17	2,00	17	2,00
3_101	150	21	22	0,67	22	0,67	22	0,67	22	0,67
3_131	250	39	42	1,20	42	1,20	42	1,20	42	1,20
3_161	500	77	81	0,80	81	0,80	81	0,80	81	0,80
3_191	1000	153	159	0,60	158	0,50	159	0,60	158	0,50
Среднее отклонение, %			—	**1,68**	—	**1,31**	—	**1,32**	—	**1,67**

Таблица 4

Результаты тестирования генетических алгоритмов

Тестовые задачи ngcutfs			Алгоритм				
			Packer GA		Packer MGA		
Задача	Число объектов m	$\Delta^{(p)}$	S	μ	S	μ	τ, с
Задачи 1-го типа							
1_11	40	10	12	5,00	12	5,00	1,10
1_41	50	14	15	2,00	**14**	0,00	1,62
1_71	100	24	26	2,00	26	2,00	2,25
1_101	150	36	39	2,00	**38**	1,33	4,62
1_131	250	58	62	1,60	**61**	1,20	25,87
1_161	500	123	130	1,40	**128**	1,00	102,20
1_191	1000	241	254	1,30	**250**	0,90	300,00
Среднее отклонение, %				2,19		2,03	
Задачи 2-го типа							
2_11	40	11	12	2,50	**11**	0,00	0,98
2_41	50	9	11	4,00	**10**	2,00	0,81
2_71	100	23	25	2,00	**24**	1,00	4,40
2_101	150	31	34	2,00	**33**	1,33	9,54
2_131	250	53	57	1,60	57	1,60	17,34
2_161	500	99	103	0,80	103	0,80	48,30
2_191	1000	205	213	0,80	**212**	0,70	210,18
Среднее отклонение, %				1,96		1,56	
Задачи 3-го типа							
3_11	40	7	7	0,00	7	0,00	0,60
3_41	50	10	12	4,00	**11**	2,00	1,48
3_71	100	15	17	2,00	**16**	1,00	4,34
3_101	150	21	22	0,67	22	0,67	4,96
3_131	250	39	42	1,20	**41**	0,80	24,23
3_161	500	77	80	0,60	80	0,60	98,64
3_191	1000	153	158	0,50	**157**	0,40	300,00
Среднее отклонение, %				1,28		1,02	12

эффективными параметрами генетического алгоритма (ГА) являются следующие: вероятность выполнения оператора кроссинговера равна 0,9, инверсии — 0,7, мутации — 0,1. В ГА и МГА размер хромосомы равен числу размещаемых объектов.

В табл. 4 приведены результаты тестирования реализованных ГА **Packer GA** [12] и МГА **Packer MGA**. В ходе вычислительного эксперимента исследовались алгоритмы со следующими параметрами: размер популяции — 100 хромосом, максимальная глубина поиска решений — 1000 поколений, время поиска τ ограничено 300 с.

В табл. 4 приведены результаты ГА и МГА, полученные за время работы алгоритма **Packer MGA**. Проведенные вычислительные эксперименты показали, что среди 21 решенных тестовых задач результаты работы ГА и МГА одинаковы для семи задач, а для 14 тестовых задач МГА обеспечивает получение лучших по сравнению с ГА решений (значения целевой функции для этих задач в табл. 4 выделены жирным шрифтом).

2.3. Вычислительный эксперимент для задач двухмерной контейнерной упаковки на листы

Решения, полученные алгоритмом **Packer MGA**, сравнивались с решениями, полученными на основе следующих известных эволюционных алгоритмов:

- генетический блочный алгоритм Норенкова И. П. **GMA** [8];
- "жадный" генетический алгоритм Genetic Greedy Sub (**GGSub**) [3], реализованный Ширгазиным Р. Р. [13].

Таблица 5

Результаты эксперимента для задач двухмерной контейнерной упаковки на листы

Отклонение от нижней границы μ, %			
Типы задач	GMA	GGSub	Packer MGA
ngcutfs_1	2,39	1,87	**1,63**
ngcutfs_2	1,76	1,37	**1,06**
ngcutfs_3	0,88	0,88	**0,78**

Таблица 6

Результаты эксперимента для задач рулонного раскроя

Класс	Нижняя граница	TS		Packer MGA	
		Решение	η	Решение	η
C1	187,18	187,94	**0,58**	187,86	0,70
C2	60,52	61,04	0,80	60,92	**0,78**
C3	504,12	510,78	0,71	510,06	**1,45**
C4	193,50	199,76	3,33	201,10	4,04
C5	1613,16	1639,00	2,06	1636,50	**2,04**
C6	506,40	527,20	**4,24**	534,86	5,79
C7	1577,82	1591,80	**1,12**	1594,00	1,17
C8	1397,92	1440,00	**3,35**	1477,40	5,65
C9	3343,10	3346,20	**0,07**	3346,80	0,11
C10	909,16	933,22	2,82	936,84	**2,79**
C1—C10			1,98		2,45

Таблица 7

Результаты тестирования алгоритма Packer MGA на тестовых задачах рулонного раскроя

Задача	Число объектов	Ширина полосы	Нижняя граница	Решение (длина)	η	Время решения, с
C11	20	10	60,30	61,60	2,11	2,64
C12	40	10	121,60	122,80	0,98	7,38
C13	60	10	187,40	188,00	0,32	14,28
C14	80	10	262,20	262,40	0,08	23,89
C15	100	10	304,40	304,50	0,03	32,12
Среднее C1	—	—	**187,18**	**187,86**	**0,70**	**16,06**
C21	20	30	19,70	19,90	1,01	2,40
C22	40	30	39,10	39,50	1,01	5,00
C23	60	30	60,10	60,70	0,99	10,42
C24	80	30	83,20	83,70	0,60	15,89
C25	100	30	100,50	100,80	0,30	25,85
Среднее C2	—	—	**60,52**	**60,92**	**0,78**	**11,91**
C31	20	40	157,40	162,70	3,26	3,28
C32	40	40	328,80	332,70	1,17	7,74
C33	60	40	500,00	502,90	0,58	17,42
C34	80	40	701,70	709,40	1,09	27,04
C35	100	40	832,70	842,60	1,17	31,24
Среднее C3	—	—	**504,12**	**510,06**	**1,45**	**17,34**
C41	20	100	61,40	64,60	4,95	2,71
C42	40	100	123,90	129,10	4,03	7,89
C43	60	100	193,00	201,80	4,36	15,47
C44	80	100	267,20	277,70	3,78	20,42
C45	100	100	322,00	332,30	3,10	35,34
Среднее C4	—	—	**193,50**	**201,10**	**4,04**	**16,37**
C51	20	100	512,20	545,70	6,14	3,05
C52	40	100	1053,80	1060,40	0,62	8,99
C53	60	100	1614,00	1638,00	1,47	17,86
C54	80	100	2268,40	2272,40	0,18	26,16
C55	100	100	2617,40	2665,90	1,82	42,44
Среднее C5	—	—	**1613,16**	**1636,48**	**2,04**	**19,70**
C61	20	300	159,90	173,60	7,89	2,87
C62	40	300	323,50	342,20	5,46	8,89
C63	60	300	505,10	536,30	5,82	13,72
C64	80	300	699,70	738,30	5,23	26,87
C65	100	300	843,80	883,90	4,54	41,96
Среднее C6	—	—	**506,40**	**534,86**	**5,79**	**18,86**
Среднее C1—C6	—	—	—	—	**2,47**	—
C71	20	100	490,40	500,90	2,10	3,46
C72	40	100	1049,70	1058,80	0,86	9,34
C73	60	100	1515,90	1531,00	0,99	17,29
C74	80	100	2206,10	2227,80	0,97	36,25
C75	100	100	2627,00	2651,70	0,93	42,50
Среднее C7	—	—	**1577,82**	**1594,04**	**1,17**	**21,77**
C81	20	100	434,60	465,30	6,60	3,98
C82	40	100	922,00	979,00	5,82	8,38
C83	60	100	1360,90	1444,70	5,80	15,88
C84	80	100	1909,30	2011,30	5,07	25,54
C85	100	100	2362,80	2486,50	4,97	36,95
Среднее C8	—	—	**1397,92**	**1477,36**	**5,65**	**18,15**
C91	20	100	1106,80	1108,40	0,14	3,85
C92	40	100	2189,20	2191,20	0,09	13,27
C93	60	100	3410,40	3412,40	0,06	22,57
C94	80	100	4578,60	4588,10	0,21	38,81
C95	100	100	5430,50	5433,80	0,06	53,66
Среднее C9	—	—	**3343,10**	**3346,78**	**0,11**	**26,43**
C101	20	100	337,80	345,60	2,26	3,16
C102	40	100	642,80	659,90	2,59	9,17
C103	60	100	911,10	936,00	2,66	18,67
C104	80	100	1177,60	1217,10	3,25	32,69
C105	100	100	1476,50	1525,60	3,22	38,34
Среднее C10	—	—	**909,16**	**936,84**	**2,79**	**20,41**
Среднее C7—C10	—	—	—	—	**2,43**	—
Среднее C1—C10	—	—	—	—	**2,45**	—

Усредненные результаты проведенного вычислительного эксперимента приведены в табл. 5.

Из табл. 5 видно, что реализованный МГА **Packer MGA** с разработанными эвристиками обеспечивает наиболее плотное размещение объектов для всех трех типов решаемых тестовых задач двухмерной контейнерной упаковки ортогональных объектов.

2.4. Вычислительный эксперимент для задач рулонного раскроя

При решении задачи двухмерной ортогональной упаковки объектов на полубесконечную полосу (1.5 Dimensional Bin Packing Problem, **1.5DBPP** [14], известная также как задача рулонного раскроя, Strip Packing Problem) требуется с максимальным коэффициентом раскроя минимизировать длину занятой части полосы.

Вычислительный эксперимент проводился при решении эталонных задач рулонного раскроя из статей Berkey и Wang [15] (классы задач C1—C6), а также из статей Martello и Vigo [16] (классы задач C7—C10). В ходе эксперимента было решено 500 задач рулонного раскроя (10 классов по 50 задач). Число прямоугольников в каждом классе от 20 до 100. Решение задачи ограничено по времени в 60 с.

Показателем эффективности рулонного раскроя служит отклонение от нижней границы, которое рассчитывается по следующей формуле:

$$\eta(\%) = \frac{L - L_0}{L} 100\ \%,$$

где L — длина занятой части полосы; L_0 — теоретическая нижняя граница задачи. Более плотному размещению объектов соответствует меньшее значение η.

При решении задач упаковки на полубесконечную полосу наименьшее отклонение от нижней границы дает алгоритм поиска с запретами Ермаченко А. И. (**TS**) с декодером блочной структуры [8, 17—18], поэтому алгоритм **Packer MGA** сравнивали с этим алгоритмом.

Усредненные результаты проведенного вычислительного эксперимента приведены в табл. 6, из которой видно, что алгоритм **Packer MGA** обеспечивает для четырех классов задач получение решений с наименьшим отклонением от теоретической нижней границы задачи. Таким образом, алгоритм **Packer MGA** может быть использован и для оптимизации задач рулонного раскроя.

Результаты проведенного вычислительного эксперимента для всех тестовых задач рулонного раскроя из классов C1—C10 приведены в табл. 7.

Заключение

Разработаны и исследованы новые эвристики размещения ортогональных объектов произвольной размерности. Применение разработанных эвристик возможно при использовании модели "виртуальные объекты" представления ортогональных объектов в контейнерах. Реализован мультиметодный генетический алгоритм, использующий разработанные эвристики размещения. Проведены вычислительные эксперименты на эталонных задачах двухмерной ортогональной контейнерной упаковки на листы и рулонного раскроя, подтверждающие высокую эффективность реализованного МГА с разработанными эвристиками.

Перспективным направлением дальнейших исследований является разработка новых эвристик размещения и создание на их основе комбинированных эвристических алгоритмов для быстрого получения субоптимальных решений задач ортогональной упаковки объектов.

Список литературы

1. **Gary M., Johnson D.** Computers intractability: a guide to the theory of NP-completeness. San Francisco: W. H. Freeman, 1979.
2. **Норенков И. П.** Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. 1999. № 1. С. 2—7.
3. **Чеканин В. А., Чеканин А. В.** Эффективные модели представления ортогональных ресурсов при решении задачи упаковки // Информационно-управляющие системы. 2012. № 5. С. 29—32.
4. **Библиотека** OR-library наборов объектов из задач Fekete и Schepers. [Электронный ресурс]. URL: http://people.brunel.ac.uk/~mastjjb/jeb/info.html (дата обращения: 26 декабря 2012).
5. **Fekete S. P., Schepers J.** New classes of fast lower bounds for bin packing problems // Mathematical Programming. Ser. A 91. 2001. P. 11—31.
6. **Мухачева Э. А., Валеева А. Ф., Филиппова А. С., Поляковский С. Ю.** Задача двухмерной контейнерной упаковки: нижние границы и численный эксперимент с алгоритмами локального поиска оптимума // Информационные технологии. 2006. № 4. С. 45—52.
7. **Чеканин В. А., Чеканин А. В.** Оптимизация решения задачи ортогональной упаковки объектов // Прикладная информатика. № 4(40). 2012. С. 55—62.
8. **Валиахметова Ю. И., Филиппова А. С.** Мультиметодный генетический алгоритм для решения задач ортогональной упаковки // Информационные технологии. 2007. № 12. С. 50—56.
9. **Чеканин В. А., Ковшов Е. Е.** Систематизация и анализ структур данных при автоматизации управления склада на основе генетических алгоритмов // Вестник высших учебных заведений. Проблемы полиграфии и издательского дела. 2008. № 5. С. 42—51.
10. **Чеканин В. А., Ковшов Е. Е.** Моделирование и оптимизация технологических операций в промышленном производстве на основе эволюционных алгоритмов // Технология машиностроения. 2010. № 3. С. 53—57.
11. **Чеканин В. А., Ковшов Е. Е., Хуэ Н. Н.** Повышение эффективности эволюционных алгоритмов при решении оптимизационных задач упаковки объектов // Системы управления и информационные технологии. 2009. № 3. С. 63—67.
12. **Чеканин В. А.** Эффективное решение задачи двухмерной контейнерной упаковки прямоугольных объектов // Вестник компьютерных и информационных технологий. 2011. № 6. С. 35—39.
13. **Ширгазин Р. Р.** Эволюционные методы и программное обеспечение для решения задач ортогональной упаковки на базе блочных структур. Автореф. дис. на соиск. уч. степ. канд. техн. наук: 05.13.18. Уфа: УГАТУ, 2006. 15 с.
14. **Dyckhoff H.** A typology of cutting and packing problems // European Journal of Operation Research. 1990. Vol. 44. P. 145—159.
15. **Berkey O., Wang P. Y.** Two-dimensional finite bin-packing algorithms // Oper. Res. Soc. 1987. 38(5). P. 423—429.
16. **Martello S., Vigo D.** Exact solution of the two-dimensional finite bin packing problem // Management Science. 1998. Vol. 44. P. 388—399.
17. **Мухачева Э. А., Ермаченко А. И.** и др. Метод поиска минимума с запретами в задачах двухмерного гильотинного раскроя // Информационные технологии. 2001. № 6. С. 25—31.
18. **Mesyagytov M. A., Smagin M. A., Filippova A. S.** Substitution decoder based on local search algorithm for packing on sheets // CSIT. 2005. Vol. 2. P. 164—166.

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ И СИСТЕМЫ ПРИРОДОПОЛЬЗОВАНИЯ

УДК 004.4, 004.89

С. Л. Беляков, д-р техн. наук, проф.,
М. Л. Белякова, канд. техн. наук, доц.,
М. Н. Савельева, аспирант
e-mail: marina.n.savelyeva@gmail.com
Таганрогский технологический институт
Южного федерального университета

Прецедентный анализ образов в интеллектуальных геоинформационных системах

Рассматривается концептуальная модель образного представления знаний для прецедентного анализа в интеллектуальных геоинформационных системах. Модель включает в себя две образные компоненты — ситуации и решения. Описана структура классов образов ситуаций и решений. Предложено использовать методы расширения ситуаций и решений для повышения достоверности генерируемых решений.

Ключевые слова: *интеллектуальные системы, геоинформационные системы, прецедентный анализ, образные знания*

Интеллектуализация геоинформационных систем, применяемых для принятия решений, является одним из важных направлений повышения качества принимаемых решений. Аналитик и работающая геоинформационная система образуют систему гибридного интеллекта [1]. Визуальные образы карт и планов стимулируют интеллектуальную деятельность аналитиков, решающих трудноформализуемые задачи. В этих ситуациях наиболее ярко проявляется основное достоинство геоинформационных систем — визуализация хранимых в картографическом виде данных. Вместе с тем, являясь универсальным инструментом, геоинформационные системы (ГИС) не одинаково эффективны при решении задач различных классов. Чем сложнее постановка, тем большую роль играют опыт и знания, полученные экспертами при решении аналогичных проблем. Исследование путей использования опыта как "сильного метода" получения решений остается актуальным ввиду специфики представления и использования опыта в информационных системах различных типов [2].

Характерной особенностью организации ГИС является образный характер отображения данных. Картографические объекты ГИС связаны ссылками с семантическими базами данных, мультимедиа-ресурсами и онтологиями, использование которых делает процесс поиска и генерации решений более содержательным. Как результат, растет достоверность решений — их соответствие реальной действительности. В отличие от других систем в ГИС оценка достоверности облегчается картографической визуализацией. Следовательно, образное представление перспективно для манипулирования знаниями в процессах принятия решений.

Существуют объективные трудности накопления и использования знаний в образном виде. Они вызваны, главным образом, отсутствием общеупотребимой модели образного мышления. Традиционные модели представления и использования знаний универсальны и могут использоваться для описания образного мышления как частного случая [2—4]. Однако при переходе к ним неизбежны потери смысловых связей. Вследствие этого даже в случае использования механизмов правдоподобных рассуждений [5] не удается получить ожидаемого эффекта. Наконец, универсальное понимание смысла, заключенного в образе, столь неоднозначно и многообразно, что не дает общего конструктивного пути технической реализации образного мышления [6]. Таким образом, возникает необходимость в особых информационных моделях образного знания.

Цель настоящей статьи — анализ концептуальной модели образного представления и обработки знаний в ГИС, предназначенных для генерации и выбора решений.

Базой процесса генерации и принятия решений с помощью ГИС является картографический анализ [7]. Основными этапами картографического анализа являются следующие:

- подбор картографических материалов;
- построение новых карт, схем и планов на основе отобранных на предыдущем этапе;
- инструментальный анализ и моделирование процессов и явлений прикладной задачи;
- визуальный анализ полученных результатов, картографирование принятого решения.

Процедура картографического анализа естественным образом моделируется механизмом прецедентного анализа [2]. Прецедентный анализ рас-

сматривается как процедура синтеза решений на основе ранее зафиксированных на практике ситуаций — прецедентов. Прикладная задача представляется как проблемная ситуация, для которой должно быть построено решение путем адаптации решений для известных близких по смыслу ситуаций. Оценка близости реализуется исходя из "картины мира", заложенной в интеллектуальной системе [8].

С нашей точки зрения, перспективным является развитие метода прецедентного анализа введением в него концепции образа прецедента. Образом прецедента будем считать обобщение конкретного прецедента, сделанное в целях генерации решения в похожей (близкой) ситуации. Формально, если считать, что прецедент описывается моделью

$$p = <s, d>,$$

где s является описанием ситуации, а d — описанием решения, то образ прецедента I_p является объектом класса, обладающего в терминах объектного подхода [9] методом преобразования

$$M_{sd} : s \to d. \qquad (1)$$

В качестве примера на рис. 1 (см. четвертую сторону обложки) приведена схема парковки транспортного средства, которую привел эксперт, решая задачу доставки груза в заданный пункт назначения, показанный на карте звездочкой. Проблемой является невозможность "вплотную" припарковаться к пункту назначения ввиду отсутствия парковочного места. Эксперт принял решение припарковаться у близлежащего супермаркета. Как видно из схемы, решение — это линия движения от точки въезда в область ситуации до точки парковки. Описанием ситуации s является область на карте, решением d — линия траектории. Ценность зафиксированного в данном случае опыта достаточно мала. Знание состоит в том, что при решении задачи транспортировки в тот же пункт назначения следует использовать ту же траекторию и то же место парковки. Фактически же картина занятости парковочных мест в иное время суток и при въезде с другого направления может значительно отличаться от той, которую наблюдал эксперт. Следовательно, применение полученного опыта даже в "той же" ситуации может оказаться неудовлетворительным.

Чтобы этого избежать, экспертные данные должны нести некоторые обобщения. В первую очередь, эти обобщения выражаются в перечислении экземпляров и классов объектов и отношений, которые специфичны для ситуации в классе I_p.

Например, эксперт может посчитать существенно важным время возникновения прецедента. В качестве временной метки прецедента вместо его точного астрономического времени указать лингвистическое значение "Утро". Чтобы зафиксировать время, эксперт указывает отношения, т. е. объекты связываются ссылками. В примере это следующие ссылки:

- объекта геометрической границы области ситуации на карте на объект класса "Время"; наличие такой ссылки указывает на существенность темпоральной составляющей поведения ситуации;
- объекта класса "Время" на понятие онтологии "Время суток"; данная ссылка отражает интерпретацию времени лингвистическими значениями;
- понятия "Время суток" на лингвистическое значение "Утро" с его привязкой к шкале абсолютного времени.

На рис. 2 (см. четвертую сторону обложки) приведен пример указания возможного расположения пункта назначения при условии, что транспортное средство с грузом размещено в точке прецедента.

Рассматривая данный пример, можно заметить, что образное обобщение даже на уровне существенно важных свойств дает возможность генерировать решения для неединичной ситуации и заметно повышает их достоверность. Однако свойства класса недостаточны для целостного описания образов. Ценность образа определяется тем, насколько широк спектр порождаемых им конкретных ситуаций и решений. Определим методы класса I_p, предназначенные для получения решений из образа. Для этого рассмотрим этапы преобразования (1). Преобразование реализуется в несколько шагов:

1. Генерируется множество вариантов $\{s_j\}$ ситуации s^* в рамках образа I_p. Класс образа должен содержать метод расширения ситуации

$$M_s : s^* \to \{s_j\}, \qquad (2)$$

который порождает только такие ситуации, которые являются модификациями ситуации s^*, не изменяющими принятое решение d.

2. Вызывается метод-индикатор, который определяет, сводится ли обобщенный образ к проблемной ситуации $\tilde{s}$:

$$M_{s^*} = \begin{cases} 0 \text{ если } \tilde{s} \notin \{s_j\}, \\ 1 \text{ если } \tilde{s} \notin \{s_j\}. \end{cases}$$

3. Если $M_{s^*} = 1$, то решением является d.

Концептуально метод (2) определяет то, в какой степени может измениться данная ситуация, сохраняя при этом неизменным решение. Тем самым описывается известная в прецедентном анализе "картина мира" применительно к рассматриваемой ситуации. Для интеллектуальных ГИС подобная концепция полезна тем, что локализует знания. Общая "картина мира" эволюционирует как покрытие пространственно-временных и семантических областей локальными образами прецедентов. Такой подход является альтернативой построению глобальной базы знаний ГИС, которая включает опыт принятия решений в любых ситуациях.

Очевидно, что эксперт способен указать произвольное число методов допустимого изменения ситуации. В рассматриваемом примере такими методами могут быть:

1) нахождение множества объектов, относящихся не только к классу с атрибутом "Супермаркет", но к другим классам с установленным атрибутом "Имеется бесплатная парковка";

2) поскольку ситуация произошла в "центре города", нахождение на карте районов города с атрибутами "деловой центр города", "торговый центр города", "туристический центр города"; здесь предполагается, что в ГИС отражено на одном из слоев зонирование городской территории;

3) нахождение районов на карте, где парковка вдоль улиц запрещена или невозможна вследствие дорожных пробок;

4) нахождение на карте точек доставки товаров в утреннее время.

По аналогии с (2) в образ прецедента вводятся методы, расширяющие решение любым способом так, чтобы сохранить его как решение данного прецедента:

$$M_d : d^* \to \{d_j\}. \qquad (3)$$

Для рассматриваемого примера решения могут быть построены следующими способами:

- как прямая линия, связывающая точку местонахождения транспортного средства и выбранное место парковки;
- как любой путь в транспортной сети, покрывающей проблемную область $\tilde{s}$;
- как кратчайший путь в упомянутой выше области;
- как кратчайший путь через участки в транспортной сети, указанные экспертами (например, это наилучшие пути, показанные таксистами [9]). Такие участки могут в общем случае не входить в область $\tilde{s}$.

Для вариантов решений $\{d_j\}$ в I_p метод-индикатор не является обязательным, так как задача выбора наилучшего решения выходит за рамки инструментария ГИС. Обязательным предполагается метод построения картографической области решения:

$$M_{Rd} : d_j \to R_{d_j}.$$

Область R_{d_j} в контексте образного знания можно рассматривать как визуализацию образа решения.

На рис. 3 (см. четвертую сторону обложки) приведен пример расширения решения двумя возможными траекториями (изображены штриховой линией), построенными в результате двух эвристических алгоритмов маршрутизации.

Обобщая, опишем концептуальную модель образа прецедента как совокупность объектов

$$I = <I_s, I_d>,$$

где I_s — образ ситуации; I_d — образ решения. В свою очередь,

$$I_s = <P_W, P_T, P_C, P_E, L, M_s, M_{s*}>, \text{ где}$$

- P_W — пространственные свойства ситуации; к ним относят пространственные объекты ситуации прецедента на карте;
- P_T — временные свойства ситуации; свойства отражают не только абсолютное время, но и длительность, периодичность, согласованность с другими событиями;
- P_C — семантические свойства ситуации, таковыми являются наименование, оценки по различным классификациям, количественные и качественные значения параметров ситуации;
- P_E — прагматические свойства ситуации, отражающие ее связь с различными прикладными задачами, полезность ее учета в их решении;
- L — список существенно важных для идентификации образа ситуации экземпляров и классов объектов и отношений;
- M_s, M_{s*} — методы расширения ситуации и методы-индикаторы.

Соответственно,

$$I_d = <P_W, P_T, P_C, P_E, M_d, M_{Rd}>,$$

где P_W, P_T, P_C, P_E — аналогично предыдущему, пространственные, временные, семантические и прагматические свойства решения; M_d, M_{Rd} — методы расширения решения и конструирования визуального представления области решения.

Прецедентный анализ при использовании предложенной концептуальной модели реализуется следующим образом.

1. Задается проблемная ситуация путем описания пространственно-временных, семантических и прагматических границ проблемной области $\tilde{s}$.

2. Строится множество образов прецедентов $\{I_{\tilde{s}}\}$, близких проблемной ситуации. В отличие от традиционного отбора прецедентов путем вычисления значений меры близости, отбор образов дополнительно требует оценки результата работы методов расширения ситуации M_s. В множество анализируемых ситуаций могут попасть те, которые непосредственно не имеют отношения к текущей, но близки по сути.

3. Вызовом методов-индикаторов M_{s*} оценивается возможность применения решений из множества $\{I_{\tilde{s}}\}$, в результате чего строится множество возможных решений $\{d_{\tilde{s}}\}$.

4. Для $\{d_{\tilde{s}}\}$ с помощью методов M_d, M_{Rd} создаются варианты альтернатив решения в проблемной ситуации.

5. Если геоинформационной системой не предложено полезных решений, ситуация $\tilde{s}$ фиксируется как ситуация, для которой недостаточно накопленного опыта.

Подводя итог, можно сделать следующие выводы.

Образное представление знаний уместно в системах с обширной информационной основой. Образ как обобщающая структура должен обеспечивать покрытие существующих элементов базы данных. У образного представления знаний должна быть экстенсиональная основа.

Прецедентный анализ при использовании образов прецедентов приобретает существенную особенность — локальность применения знаний. Решения строятся не синтетически из различных образов, а исключительно в рамках конкретного образа прецедента. Это повышает достоверность генерируемых решений.

Повышение достоверности решений в образном представлении обеспечивается методами расширения образа. Они описывают известные эксперту возможности изменений ситуаций и решений, не меняющих суть прецедента.

Работа поддержана грантом РФФИ, проект № 12-01-00032-а.

Список литературы

1. **Венда В. Ф.** Системы гибридного интеллекта: эволюция, психология, информатика. М.: Машиностроение, 1990. 448 с.
2. **Люгер Д. Ф.** Искусственный интеллект: стратегии и методы решения сложных проблем: Пер. с англ. 4-е изд. М.: Вильямс, 2003. 864 с.
3. **Кобринский Б. А.** Образные ряды в интеллектуальных системах // Искусственный интеллект и принятие решений. 2009. № 2. С. 25—33.
4. **Гаврилова Т. А., Хорошевский В. Ф.** Базы знаний интеллектуальных систем. СПб.: Питер, 2000. 384 с.
5. **Вагин В. Н., Головина Е. Ю., Загорянская А. А., Фомина М. В.** Достоверный и правдоподобный вывод в интеллектуальных системах. 2-е изд., испр. и доп. / Под ред. В. Н. Вагина и Д. А. Поспелова. М.: Физматлит, 2008. С. 712.
6. **Валькман Ю. Р.** Целостность образов: о моделировании смысла и понимания // Information Technologies & Knowledg. 2012. V. 6. N 1.
7. **Берлянт А. М.** Картографический метод исследования. М.: Изд-во МГУ, 1988.
8. **Варшавский П. Р., Еремеев А. П.** Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений. 2009. № 1. С. 45—57.
9. **Буч Г.** Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 3-е изд. М.: Вильямс, 1988.
10. **Microsoft** vs Google: битва титанов. Поисковые системы Google уступят идеям компании Microsoft, основывающимся на знаниях таксистов. URL: http://www.globaltaxi.ru/news/875.html

ИНФОРМАЦИЯ

8—10 ноября 2013 в г. Львов пройдет

Четырнадцатая Международная конференция в области обеспечения качества ПО

"SOFTWARE QUALITY ASSURANCE DAYS"

Конференция охватит широкий спектр профессиональных вопросов в области обеспечения качества, ключевыми из которых являются:

- методики и инструменты тестирования ПО;
- автоматизация тестирования ПО;
- подготовка, обучение и управление командами тестировщиков;
- процессы обеспечения качества в компании;
- управление тестированием и аутсорсинг;
- совершенствование процессов тестирования и инновации.

Предыдущая 13-я конференция проходила в Санкт-Петербурге, ее участниками стали более 600 профессионалов. Организатором традиционно выступает компания "Лаборатория тестирования" (http://www.sqalab.ru/).

Обращаем внимание, что 10 ноября пройдет дополнительный день SQA Days English Day в рамках которого пройдут доклады на английском языке. Это отдельное событие в рамках конференции. Число мест на этот день будет ограничено.

Сайт конференции: http://sqadays.com/index-news.sdf/sqadays/sqa_days14

УДК 004.912

А. Ю. Бородащенко,
канд. техн. наук, преподаватель,
Д. С. Гончаров, слушатель,
Академия ФСО России, г. Орел
e-mail: bay55@mail.ru

Алгоритм выявления новых событий

Предложен алгоритм выявления новых событий, позволяющий улучшить качество отбора новостей из сети Интернет путем повышения оперативности, точности и полноты выбора новой информации из массива документов. Алгоритм реализует функцию выделения ключевых слов и словосочетаний, встречающихся в публикациях, и сравнения содержания текстов по ним.

Ключевые слова: *текст, обработка текста, мера Солтона, семантическое сходство, семантическое расстояние, мера близости, алгоритм фильтрации текстов, новая информация*

В настоящее время информационный ресурс сети Интернет состоит из миллионов новостных Web-страниц, формируемых тысячами новостных агентств, к которым возможен свободный доступ любого пользователя. Новостные агентства конкурируют друг с другом на современном рынке информационных технологий, стремясь первыми опубликовать новость с пометкой "срочно", дублируя новости, а часто просто копируя их друг у друга. Для того чтобы найти новую актуальную информацию в этом массиве неструктурированных документов, необходимо использовать мощные информационно-поисковые системы (ИПС), а также осуществлять большую интеллектуальную работу по выборке необходимых данных.

Ведущими источниками новостной информации в Российской Федерации являются такие издания, как: www.vesti.ru — "Вести" — интернет-газета [1]; www.lenta.ru — Lenta.ru — издание Rambler Media Group [2]; www.ria.ru — сетевое издание "РИА Новости" [3]; www.km.ru — сетевое издание KM.ру [4]; http://www.rg.ru — издание Правительства Российской Федерации [5]; http://www.aif.ru — издание "Аргументы и факты" [6] и др.

Каждый новостной ресурс характеризуется определенной степенью достоверности информации, которую он публикует. Новость, появившаяся в региональном новостном источнике, как правило, не привлечет внимания общественности в первое время. В большинстве случаев данное сообщение станет общедоступным, как только его опубликуют федеральные источники. Однако неоспорим тот факт, что каждая новость должна быть своевременной — в этом и заключается ценность информации.

Задачи выявления, отслеживания и группировки событий на основе анализа новостных текстовых документов активно обсуждаются специалистами во всем мире. В работе [7] задачи такого класса описаны как одни из наиболее важных, имеющих большое практическое значение именно сегодня, когда режим доступа к системам интеграции новостей существенно облегчен.

Как правило, обработка поступающего политематического новостного потока требует огромного количества времени и интеллектуальных усилий пользователя. Новые подходы и методы к обработке текста позволили бы существенно снизить нагрузки на человека.

Можно говорить, что технологии семантического анализа текстов до сих пор все еще находятся только в начале своего развития. Они предоставляют решения задач морфологического и синтаксического анализа, реферирования, а также построения семантических сетей для выделенной ими подборки документов [8].

В качестве одной из очевидных тенденций в развитии систем анализа новостного потока можно выделить внедрение методов выявления "новых" событий на основе оценки лексикографического сходства текстов. Данная проблема достаточно активно изучалась с 70-х годов XX-го века, вначале она рассматривалась как "Topic Detection", а позднее как "New Event Detection" [9]. Первые работы связаны с американским ученым Солтоном, векторно-пространственной моделью представления данных и традиционными методами кластеризации [10]. Значительный вклад внес Р. Папка [11], который в своих работах рассматривал записи о новых событиях, как о фрагментах документов, не удовлетворяющих запросам пользователей, построенных с учетом уже известных событий. В настоящее время во многих популярных системах интеграции новостей задача выявления новых событий заменяется выявлением основных новостных сюжетных цепочек.

Противоположная задача (поиск "дублей") решается, как правило, сегодня с использованием алгоритмов антиплагиата. Наиболее популярные из них

(например алгоритм "шинглов" — "чешуек") реализованы сравнением контрольных сумм фрагментов текстов [12]. Как известно, контрольные суммы статических функций очень чувствительны к изменениям, поэтому любая перестановка слов в тексте, а также изменение форм слов (например, замена терминов синонимами) приводит к снижению чувствительности и точности такого алгоритма. Рассматриваемая же задача заключается в выявлении разных в семантическом смысле текстов.

В статье рассматривается один из возможных подходов к решению проблемы выявления новой информации из массива документов, формализованный в виде алгоритма выявления "новых" новостных событий, который может быть применен при поиске новостей в сети Интернет, в ведомственных (корпоративных) сетях и других базах данных, содержащих полнотекстовую информацию.

Задача выявления "новых" новостных событий может быть разделена на четыре этапа (рис. 1).

На первом этапе, в случае большого объема поступившего текста, происходит его автореферирование и формируется поисковый образ документа, состоящий из наиболее весомых ключевых слов и словосочетаний [13, с. 66].

На втором этапе происходит разбиение текста на слова, определение их морфологических основ. Выделяется динамический массив, в который заносится по мере обработки количество вхождений в текст определенной основы.

На следующем этапе происходит расчет весов слов, на основе частоты их встречаемости и синтаксиса предложений (мера Солтона — модель *TF* x *IDF*).

В этой модели для вычисления частоты термина в тексте необходимо воспользоваться формулой

$$TF_i = T_i / N, \qquad (1)$$

где T_i — число i-х основ; N — число слов в тексте.

Однако можно заметить, что при использовании только формулы (1) вес основы будет обратно пропорционально зависеть от длины текста. Также в русском языке присутствуют часто встречаемые слова, которые не будут являться ключевыми для анализируемой новости, однако будут иметь высокую частоту встречаемости в документе. Таким образом, для масштабирования весов основ в документе необходимо воспользоваться формулой обратной частотности документа:

$$IDF_i = \log(D / Td_i), \qquad (2)$$

где D — число документов в базе данных; Td — число документов в базе данных, в которых встречается основа.

На последнем этапе ранжируются веса каждой основы, происходит выборка из них наиболее значимых и поиск в базе данных, содержащей ранее накопленную новостную информацию.

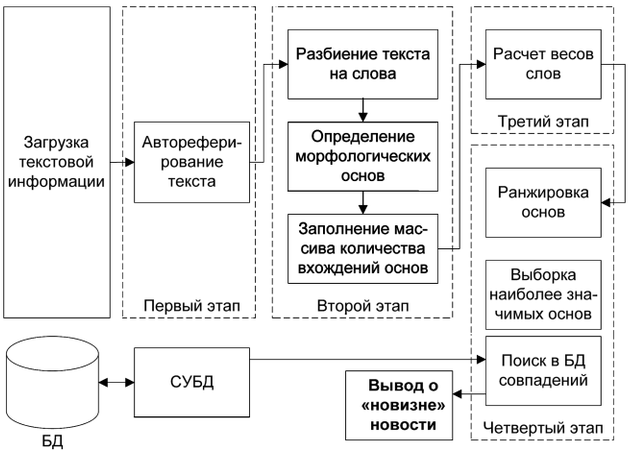


Рис. 1. Схема процесса выявления новой новостной информации

На выходе делается вывод о "новизне" рассматриваемой новости.

На основе представленной на рис. 1 схемы авторами предложен алгоритм.

Сложность алгоритма можно оценить следующим образом: для подсчета T_i необходимо выполнить $(n + n)$ операций: $O(TF) = n$, $O(IDF) = \log(n)$. Тогда сложность алгоритма при вычислении равна $w = 2n + n\log(n)$, $O(w) = n\log(n)$.

Вес каждой основы всегда больше 0 и будет неограниченно расти в зависимости от числа текстовых документов, уже содержащихся в базе данных, в которых отсутствует данная основа.

Рассмотрим работу алгоритма более подробно.

На первом этапе в случае необходимости (новость превышает 250 слов) происходит автореферирование текста, так как алгоритм должен выдавать результат за достаточно короткий промежуток времени (первый этап на рис. 1).

На втором этапе (см. рис. 1) происходит определение числа слов в тексте. Каждое слово обрабатывается программой и по словарю определяется его основа. Это необходимо, так как, например, слова "заяц" и "зайцу" фактически отличаются, однако несут одинаковую смысловую нагрузку.

В выделенный динамический массив заносится каждая основа и происходит расчет ее вхождений в текст.

Для вычисления весов каждой основы воспользуемся следующей формулой (третий этап, рис. 1):

$$W_i = K \cdot TF_i \cdot IDF_i, \qquad (3)$$

где W_i — вес i-го слова; K — весовой коэффициент слова.

После выполнения указанных операций происходит ранжировка основ по возрастанию частоты повторений. Алгоритм отбирает семь наиболее значимых основ и проводит поиск в БД подобных документов. Поиск осуществляется путем просмотра ключевых слов и словосочетаний новостной инфор-

мации, уже содержащейся в БД, и их попарного сравнения с ключевыми словами и словосочетаниями анализируемой новости.

В случае положительного результата поиск прекращается и пользователю выдается сообщение с уведомлением о неуникальности новости. В противном случае запись заносится в БД.

Коэффициент уникальности содержания текста рассчитывают по формуле

$$K_{i,j} = T_c / T, \quad (4)$$

где T_c — число совпадающих ключевых слов в сравниваемых текстах; T — число слов, участвующих в сравнении текстов.

Документ считается уникальным, если коэффициент уникальности содержания текста удовлетворяет следующему критерию:

$$K_{i,j} \leqslant K_{i,j}^{\text{доп}}, \quad (5)$$

где $K_{i,j}^{\text{доп}}$ — минимальное значение коэффициента, необходимое для принятия положительного решения. Значение этого коэффициента устанавливается пользователем. В ходе экспериментов было установлено, что значение должно быть около 0,56.

Данный критерий справедлив, так как текст считается уникальным, если он несет новую, ранее не встречаемую в публикациях информацию, т. е. в нем содержатся ключевые слова и словосочетания, не встречаемые в других новостных документах. Чем больше совпадений ключевых слов и словосочетаний, тем менее уникальной является анализируемая информация.

Чем ближе $K_{i,j}$ к 0, тем больше уникальность документа.

Результаты испытаний отображены на рис. 2. В экспериментах авторы рассматривали тексты, содержащие информацию о военных конфликтах за конец XX — начало XXI века.

Число слов, участвующих в сравнении, было выбрано эмпирическим методом, путем проведения авторами экспериментов на реализованном программном макете. Испытания проводили на ансамбле из 1000 документов.

Результаты появления ошибок первого рода (ошибки проявляются тогда, когда "новая" новость определяется как уже содержащаяся в БД) и второго

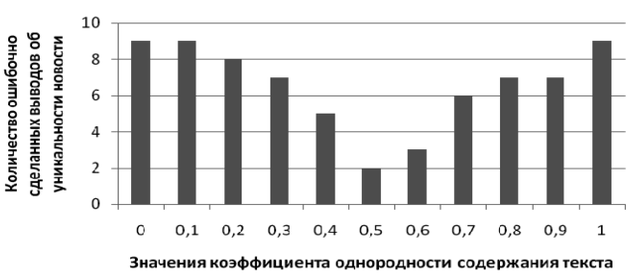


Рис. 2. Результаты испытаний по определению значения коэффициента уникальности содержания текста

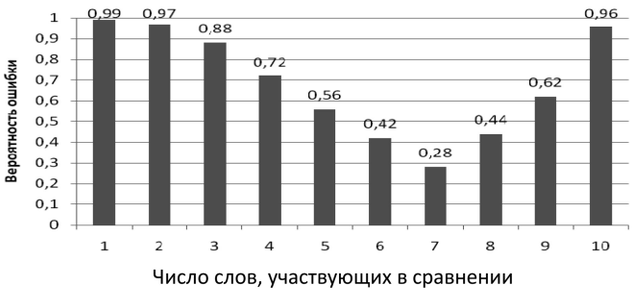


Рис. 3. Вероятность ошибки первого и второго рода в зависимости от числа слов, участвующих в сравнении

Таблица 1

Значения коэффициента *K*

Условия применимости	Значения коэффициента
Слово выделено жирным шрифтом	3
Слово выделено курсивом	3
Слово выделено подчеркиванием	3
Слово присутствует в заголовке	11
Слово встречается в подзаголовках	6

рода — ложное срабатывание (ошибок, появляющихся в случае определения новости, уже содержащейся в БД как "новой") приведены на рис. 3.

Таким образом, можно увидеть, что минимальное число ошибок соответствует семи словам, участвующим в сравнении.

В работе [14] значения коэффициента K, участвующие при расчете весов основ, необходимых для учета семантики текста, приведены в табл. 1.

Данные значения коэффициента при необходимости суммируются. В случае если слово не удовлетворяет ни одному из условий применимости, коэффициент K принимается равным 1.

Результатом работы авторов является созданный алгоритм (рис. 4) и программный продукт, который его реализует. Экранная форма пользовательского интерфейса представлена на рис. 5.

Рассмотрим работу алгоритма на примере. Пусть на вход алгоритма подается ансамбль из четырех текстовых документов.

В блоках 1—5 выполняются предварительные настройки системы для ее нормального функционирования. В блоке 7 определяется основа обрабатываемого слова с помощью морфологического словаря. В блоке 9 происходит поиск вхождения обрабатываемого слова в массив уже обработанных основ. В случае если искомое слово найдено, T_i увеличивается на 1, в противном случае, T_i принимается как 1 и основа слова дописывается в массив основ. В блоке 13 выполняется расчет частоты встречаемости термина, а в блоке 14 — обратной частотности документа. В блоке 15 рассчитывается вес каждой основы и на основании веса в блоке 17 происходит их упорядочение по убыванию. В блоке 18 осуществляется отбор наиболее значащих

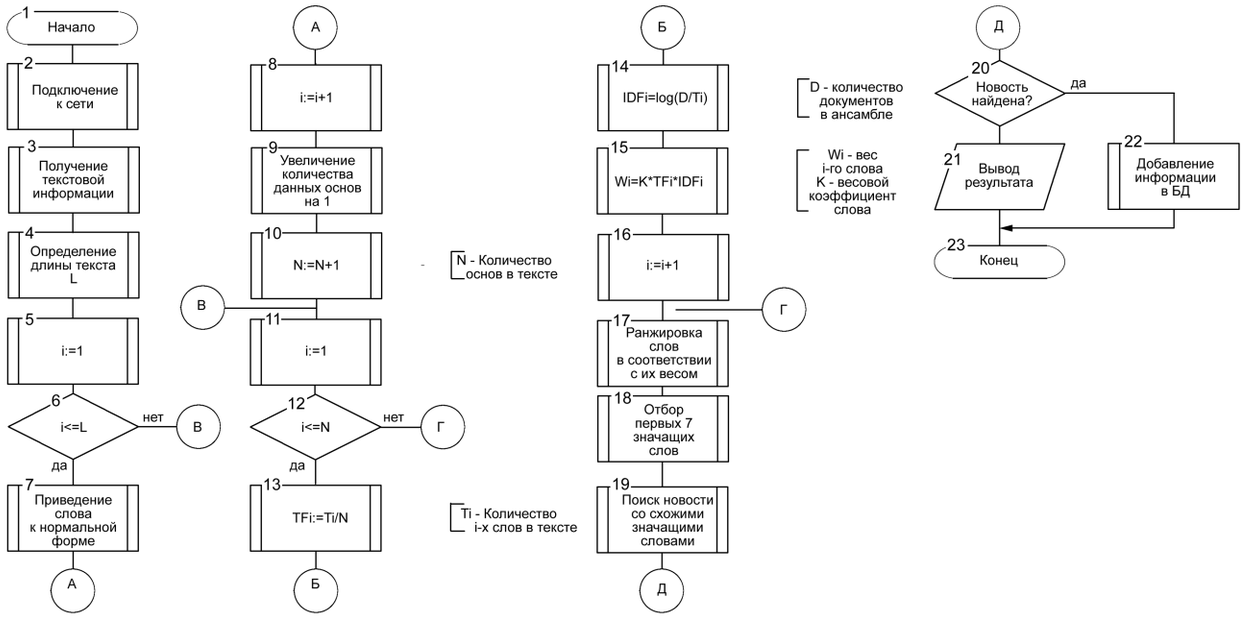


Рис. 4. Алгоритм выявления "новых" новостных событий

основ и в блоке 19 — поиск в базе данных на предмет наличия схожих ранее добавленных новостей. Блок 21 выполняет визуальное оповещение пользователя о дублировании новости, а в блоке 22 происходит добавление текста в базу данных.

Первый текстовый документ

К 22 июня 1941 года у границ СССР было сосредоточено и развернуто 3 группы армий (всего 181 дивизия, в том числе 19 танковых и 14 моторизованных, и 18 бригад). Поддержку с воздуха осуществляли 3 воздушных флота.

Директивой от 31 января 1941 года ставилась задача "уничтожить действующие в Прибалтике силы противника и захватом портов на Балтийском море, включая Ленинград и Кронштадт, лишить русский флот его опорных баз". На Балтике для поддержки группы армий "Север" и действий против Балтийского флота немецким командованием было выделено около 100 кораблей, в том числе 28 торпедных катеров, 10 минных заградителей, 5 подводных лодок, сторожевые корабли и тральщики.

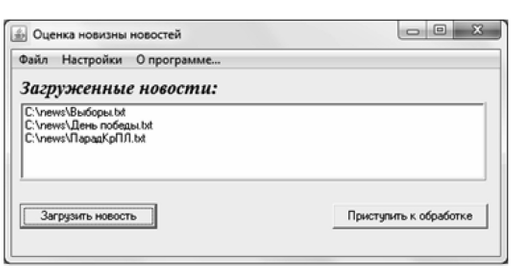


Рис. 5. Экранная форма разработанной программы

Дивизии и бригады были объединены в 9-ю и 4-ю полевые армии, а также 2-ю и 3-ю танковые группы. Задачей группы было: "Наступая крупными силами на флангах, разгромить войска противника в Белоруссии. Затем, сосредоточив подвижные соединения, наступающие южнее и севернее Минска, возможно быстрее выйти в район Смоленска и создать тем самым предпосылки для взаимодействия крупных танковых и моторизованных сил с группой армий "Север" с целью уничтожения войск противника, действующих в Прибалтике и районе Ленинграда".

Ключевыми словами, выделенными разработанным программным продуктом, являются:

ГРУППА; АРМИЯ; ТАНК; ДИВИЗИЯ; СИЛА; ПРОТИВНИК; ИЮНЬ

Второй текстовый документ

На 22 июня 1941 года в приграничных округах и флотах СССР имелось 3 289 850 солдат и офицеров, 59 787 орудий и минометов, 12 782 танка, 10 743 самолета. В составе трех флотов имелось около 220 тысяч человек личного состава, 182 корабля основных классов.

Отражение возможного нападения с запада возлагалось на войска пяти приграничных округов: Ленинградского, Прибалтийского особого, Западного особого, Киевского особого и Одесского. С моря их действия должны были поддерживать три флота: Северный, Краснознаменный Балтийский и Черноморский.

Войска Прибалтийского военного округа под командованием генерала Ф. И. Кузнецова включали в себя 8-ю и 11-ю армии, 27-я армия находилась на формировании западнее Пскова. Эти части держали оборону от Балтийского моря до южной границы Литвы.

Войска Западного особого военного округа под командованием генерала армии Д. Г. Павлова прикрывали минско-смоленское направление от южной границы Литвы до реки Припять на фронте протяженностью 470 км. В состав этого округа входили 3-я, 4-я и 10-я армии. Кроме того, соединения и части 13-й армии формировались в районе Могилев, Минск, Слуцк.

Ключевыми словами, выделенными разработанным программным продуктом являются:

ОКРУГ; ФЛОТ; АРМИЯ; ИЮНЬ; ДИВИЗИЯ; ТАНК; БАЛТИЙСКИЙ

Третий текстовый документ

В середине декабря федеральные войска начали артиллерийские обстрелы пригородов Грозного.

Несмотря на то что Грозный по-прежнему оставался незаблокированным с южной стороны, 31 декабря 1994 года Российской армией начался штурм города. Российские войска были плохо подготовлены, между различными подразделениями не было налажено взаимодействие и координация, у многих солдат не было боевого опыта. Войска имели аэрофотоснимки города, устаревшие планы города в ограниченном количестве. Войскам довели приказ о занятии только промышленных зданий, площадей и недопустимости вторжения в дома гражданского населения.

Западная группировка войск была остановлена, восточная также отступила и не предпринимала никаких действий до 2 января 1995 года. На северном направлении 1-й и 2-й батальоны дошли до железнодорожного вокзала и Президентского дворца. Федеральные силы попали в окружение — потери батальонов Майкопской бригады, по официальным данным, составили 85 человек убитыми и 72 пропавшими без вести, уничтожено 20 танков, командир бригады полковник Савин погиб, более 100 военнослужащих попало в плен.

Ключевыми словами, выделенными разработанным программным продуктом, являются:

ГОРОД; БАТАЛЬОН; БРИГАДА; АРТИЛЛЕРИЙСКИЙ ОБСТРЕЛ; ГРОЗНЫЙ; АРМИЯ; ПОГИБАТЬ

Четвертый текстовый документ

Помещичьих крестьян вначале предполагалось освободить без земли, но потом решено выделить им по 2 десятины. В отличие от "Русской Правды", Конституция Муравьева вводила жесточайший имущественный ценз как на избирательные права, так и на право быть избранным в высшие государственные органы. По Конституции избирательные права получали не более 8 % населения России. Основным способом достижения цели также считалась тактика военной революции. Если сравнивать программы, то программа Пестеля была более демократичной, но программа Муравьева была более реалистична. Представители "Северного общества" понимали: предоставить необразованному, непонимающему народу избирательное право нельзя, так как во главе оказался бы либо новый Пугачев, либо Лжедмитрий. Конституция Муравьева предполагала, что вопрос о государственном устройстве должно было решить Учредительное собрание. Сами заговорщики не брали на себя право избирать судьбу народа.

Ключевыми словами, выделенными разработанным программным продуктом, являются:

КОНСТИТУЦИЯ; ВОЕННЫЙ; НАРОД; ИЗБИРАТЕЛЬНЫЙ; ПРАВО; ПАЛАТА; СТОЛИЦА

Проанализировав результаты работы алгоритма можно рассчитать коэффициенты уникальности содержания текста по формуле (4).

Используя формулу (4), получим коэффициенты $K_{i,\,j}$ (коэффициенты уникальности содержания текстов), которые можно отобразить в виде симметричной матрицы (табл. 2).

Полученные результаты можно интерпретировать следующим образом: четвертый текст полностью отличается от трех оставшихся; первый и второй, третий тексты близки по содержанию, но все же несут разную информацию; первый и второй тексты имеют одинаковый смысл.

Данные результаты аналогичны опытным, полученным в результате экспертной обработки новостей (табл. 3).

В ходе обработки данных результатов методами математической статистики получилось, что при сравнении каждой пары текстов в среднем расхождение результата процедуры формализации текстов и мнения экспертов составляет менее 1 балла (0,4).

Таким образом, в работе предложен алгоритм выявления новых новостных событий, получивший практическую проверку на программном макете и позволяющий существенно повысить качество выделения новой информации из массива документов.

Таблица 2

Результаты работы алгоритма

j	*i*			
	1	2	3	4
1	—	0,5714	0,1428	0
2	0,5714	—	0,1428	0
3	0,1428	0,1428	—	0
4	0	0	0	—

Таблица 3

Результаты работы экспертов

j	*i*			
	1	2	3	4
1	—	0,8	0,1	0
2	0,8	—	0,1	0
3	0,1	0,1	—	0
4	0	0	0	—

Список литературы

1. **"Вести"** — интернет-газета [Офиц. сайт]. URL: http://www.vesti.ru/ (дата обращения 05.10.2012).
2. **Lenta.ru** — издание Rambler Media Group [Офиц. сайт]. URL: http://lenta.ru/ (дата обращения 05.10.2012).
3. **"РИА Новости"** — сетевое издание [Офиц. сайт]. URL: http://ria.ru/ (дата обращения 05.10.2012).
4. **KM.ру** ("KM.ru") — сетевое издание [Офиц. сайт]. URL: http://www.km.ru/ (дата обращения 05.10.2012).
5. **Издание** Правительства Российской Федерации [Интернет-портал. URL: http://www.rg.ru/ (дата обращения 05.10.2012).
6. **Аргументы и Факты** [Офиц. сайт]. URL: http://www.aif.ru/ (дата обращения 05.10.2012).
7. **Ландэ Д. В.** Поиск знаний в Internet. Профессиональная работа: пер. с англ. М.: Вильямс, 2005. 272 с.
8. **Ландэ Д. В.** Ловцы данных. Обзор программ для поиска информации в локальных сетях / Конкурентная разведка в бизнесе [Электронный ресурс]. М.: Агенство A-rsb.RU, 2012. URL: http://www.a-rsb.ru/index.php?go = News&in = view&id = 412, свободный.
9. **Kurt H.** On-line New Event Detection and Tracking in A Multi-Resource Environment — MS. Thesis, Bilkent University, 2001.
10. **Солтон Г.** Автоматическая обработка, хранение и поиск информации: пер. с англ. М.: Сов. радио, 1973. 560 с.
11. **Papka R.** On-line News Event Detection, Clustering, and Tracking. Ph. D. Thesis. University of Massachusetts at Amherst. September 1999.
12. **Зеленков Ю. Г., Сегалович И. В.** Сравнительный анализ методов определения нечетких дубликатов для WEB-документов // Труды 9-й Всероссийской научной конференции "Электронные библиотеки: перспективные методы и технологии, электронные коллекции", RCDL'2007. Сб. работ участников конкурса. Переславль-Залесский, 2007.
13. **Ландэ Д. В., Снарский А. А., Безсуднов А. В.** Интернетика. Навигация в сложных сетях. Модели и алгоритмы. М.: Либроком, 2009. 264 с.
14. **Кукушкин А. А.** Системы искусственного интеллекта: учеб. пособие. В 2 ч. Орел: [б. и.], 2009. Ч. 2.

УДК 004.853

Е. И. Большакова,
канд. физ.-мат. наук, доц.,
e-mail: eibolshakova@gmail.com,
НИУ ВШЭ,
Н. В. Лукашевич,
канд. физ.-мат. наук, вед. науч. сотр.,
e-mail: louk_nat@mail.ru,
НИВЦ МГУ им. М. В. Ломоносова,
М. А. Нокель, аспирант,
e-mail: mnokel@gmail.com,
МГУ им. М. В. Ломоносова

Извлечение однословных терминов из текстовых коллекций на основе методов машинного обучения

Представлены результаты экспериментов по автоматическому извлечению однословных терминов из русскоязычных текстов на основе машинного обучения, позволяющего комбинировать применяемые статистические и лингвистические признаки терминов. Эксперименты показывают, что комбинирование значительно улучшает результаты извлечения терминов, а найденная комбинация признаков может быть использована на расширенной текстовой коллекции без значительной потери качества.

Ключевые слова: *однословные термины, автоматическое извлечение терминов, статистические признаки, лингвистические признаки, машинное обучение*

Введение

Извлечение терминов из текстов определенной предметной области необходимо при решении многих прикладных задач, в первую очередь — для разработки и пополнения различных терминологических ресурсов, таких как терминологические словари, тезаурусы и онтологии [1]. Ручная разработка таких ресурсов экспертами трудоемка, и за последние годы были предложены процедуры, автоматизирующие этот процесс.

Большинство этих процедур для извлечения терминологических слов и словосочетаний используют критерии, базирующиеся на их статистических и лингвистических признаках: частоте встречаемости в текстах, частях речи входящих в термин слов и др. Однако ни один из предложенных признаков не является определяющим [2], и фактически из текстов извлекается довольно большой список слов и словосочетаний, являющихся лишь кандидатами в термины, которые затем должны быть проанализированы и подтверждены экспертом по предметной области текстов. Важно поэтому в ходе извлечения терминологических слов и словосочетаний провести ранжирование терминов-кандидатов, так чтобы в начале итогового списка стояли слова и словосочетания, с наибольшей вероятностью являющиеся терминами.

Для поиска наилучшей комбинации признаков, используемых для извлечения терминов из коллекции текстов определенной предметной области, в исследовательских работах последних лет привлекаются методы машинного обучения [2—6], позволяющие изучить комбинирование большого числа признаков. В частности, в работе [2] показано, что комбинация из более 80 различных статистических признаков, применяемая для извлечения словосочетаний нескольких типов из текстов на чешском языке, дает 20 %-ный выигрыш в точности по сравнению с результатами извлечения на основе одного наилучшего признака.

В целом, комбинирование статистических и лингвистических признаков для извлечения терминов

на основе машинного обучения исследовалось на коллекциях текстов разных языков из разных предметных областей, в том числе на английских биологических текстах и резюме на французском языке [3], на шведских патентных текстах [4], на русскоязычных текстах из естественно-научной и банковской областей [5]. Заметим, что в подавляющем большинстве работ рассматривалось извлечение многословных терминов. Разработанные при этом методы извлечения существенно зависят от предметных областей текстов и размеров текстовых коллекций. Так, результаты работы метода [2] зависят от предметной области коллекции и типов извлекаемых словосочетаний. Предложенный в работе [6] алгоритм голосования демонстрирует хорошее качество извлечения на коллекции текстов общей тематики и более низкое — на коллекции биологических текстов. Поэтому важно не только предложить некоторую комбинацию признаков, но и проверить ее качество в других условиях, например, при увеличении размера текстовой коллекции, что обычно происходит с развитием конкретной терминологии.

В данной работе исследуется применение методов машинного обучения для извлечения однословных терминов из текстовых коллекций. Извлечение однословных терминов — гораздо более сложная задача, поскольку к ним не применимы многие признаки, широко используемые для извлечения терминологических словосочетаний, в частности, такие статистические меры, как *взаимная информация* (MI) и *t-score*.

Среди рассматриваемых нами признаков большинство относится к статистическим, в их числе широко известная в области информационного поиска мера TF-IDF, а также модификации нескольких мер, изначально разработанных для извлечения многословных терминов. Дополнительно нами предложено несколько новых признаков (например, для слов, стоящих рядом с наиболее частотными словами коллекции). Конечная цель наших исследований — нахождение такой комбинации признаков, которая гарантирует наилучшее упорядочивание кандидатов в термины независимо от размеров текстовой коллекции. В проведенных для этого вычислительных экспериментах на основе методов машинного обучения требуемая комбинация признаков определялась сначала на некоторой части текстовой коллекции, а затем тестировалась на всей коллекции.

Процедура извлечения однословных терминов

Процесс извлечения терминов включал два этапа:

1. Отбор по определенным лингвистическим признакам слов-кандидатов из текстов коллекции. В нашем исследовании рассматривались только существительные и прилагательные, поскольку они покрывают большую часть однословных терминов. В экспериментах слова-кандидаты извлекались из коллекции банковских русскоязычных текстов (10 422 документа), взятых из различных электронных банковских журналов: Аудитор, Банки и Технологии, РБК и др.

2. Ранжирование отобранных слов-кандидатов, согласно их статистическим признакам, в целях получения большего числа подтвержденных терминов в начале результирующего списка. Для подтверждения терминологичности слова-кандидата использовался готовый тезаурус, разработанный вручную для Центрального Банка Российской Федерации и включающий более 15 000 терминов из сферы банковской активности, денежной политики и макроэкономики. Слово-кандидат считается термином, если оно содержится в этом тезаурусе.

Для улучшения результатов извлечения терминов на первом этапе кроме обычной нормализации слов проводилась фильтрация возникающих морфологических омонимов. Во-первых, исключались из рассмотрения те варианты нормализации существительных и прилагательных, словоформы которых не согласуются в тексте с соседними словами. В частности, проверялось согласование в падеже, роде и числе существительных с предстоящими прилагательными, в результате чего для словоформы *банке* из текстового словосочетания *в центральном банке* отбиралась только начальная (нормальная) форма *банк* (но не *банка*).

Во-вторых, удалялись слова-кандидаты, начальная форма которых совпадала с начальной формой некоторого слова другой части речи (отличной от существительных и прилагательных), поскольку маловероятно, что они окажутся терминами в данном контексте. Например, исключалось слово *том*, встреченное в словосочетании *в том*, из-за возможной для словоформы *том* начальной формы местоимения *то*.

Признаки для извлечения и ранжирования слов-кандидатов

Для поиска наилучшей комбинации был рассмотрен широкий набор признаков: лингвистических, орфографических и статистических. Большинство из них участвовало в машинном обучении, а несколько лингвистических и орфографических признаков использовались для дополнительной фильтрации извлеченных слов-кандидатов. Для вычисления некоторых признаков использовалась, кроме базовой коллекции текстов предметной области, контрастная коллекция текстов более общей тематики (примерно 1 млн новостных текстов).

Лингвистические и орфографические признаки. В качестве признаков для отбора из всего результирующего множества извлеченных слов-кандидатов тех подмножеств, в которых плотность терминов априори выше (чем в исходном множестве), были взяты падеж слова-существительного и ре-

гистр первой буквы слова. По ним были отобраны соответственно:

- существительные, встречающиеся в тексте в именительном падеже, поскольку такая форма характерна для подлежащих, а подлежащие часто отображают важную информацию для предметной области;
- слова-кандидаты, начинающиеся с заглавной буквы, поскольку они с большой вероятностью представляют именованные сущности рассматриваемой предметной области;
- слова с заглавной буквы, не стоящие первыми в предложении текста, для исключения случаев, когда заглавная буква слова свидетельствует только о начале предложения текста.

Указанные подмножества слов участвовали в экспериментах вместе с исходным множеством слов-кандидатов.

Следующие пять лингвистических признаков с булевским значением были предложены нами и включены в число комбинируемых на основе машинного обучения признаков.

- *неоднозначность* определяет, имеет ли слово-кандидат более одного варианта нормализации;
- *новизна* фиксирует отсутствие слова в морфословаре (т. е. его новизну);
- *специфичность* фиксирует, присутствует ли слово в контрастной коллекции текстов;
- *существительное* и *прилагательное* определяют, соответственно, является ли слово существительным или прилагательным.

Примеры значений этих признаков для слов *"банки"* и *"скоринговый"* представлены в табл. 1.

Таблица 1

Примеры значений лингвистических булевских признаков

Слово	Неоднозначность	Новизна	Специфичность	Существительное	Прилагательное
Банки	1	0	0	1	0
Скоринговый	0	1	1	0	1

Статистические признаки. Статистические признаки разделяются на следующие группы:

1. Признаки, вычисляемые лишь по базовой коллекции (табл. 2), опираются на предположение о том, что термины, как правило, встречаются в коллекции гораздо чаще других слов. Все признаки данной группы были вычислены соответственно для всего множества извлеченных слов-кандидатов, для подмножества существительных в именительном падеже, для подмножества слов-кандидатов с заглавной буквы и для подмножества слов-кандидатов с заглавной буквы, но не начальных в предложении. В табл. 2 используются обозначения:

w — слово из базовой текстовой коллекции;

$TF_t(w)$ — число употреблений слова w в базовой коллекции;

$|W_t|$ — число слов в базовой коллекции;

$DF_t(w)$ — документная частотность слова w в базовой коллекции;

$|D_t|$ — число документов в базовой коллекции;

$freq(w, d_k)$ — нормализованная частотность слова w в документе d_k.

2. Признаки, вычисляемые по базовой и контрастной коллекциям (табл. 3). Основная их идея заключается в том, что частотности терминов в базовой и контрастной коллекциях существенно различаются. В табл. 3 дополнительно используются обозначения:

$TF_r(w)$ — число употреблений слова w в контрастной коллекции;

$|W_r|$ — число слов в контрастной коллекции;

$DF_r(w)$ — документная частотность слова w в контрастной коллекции;

$|D_r|$ — число документов в контрастной коллекции.

3. Признаки, вычисляемые по статистической и контекстной информации из базовой коллекции (табл. 4), соединяют информацию о частотности слов-кандидатов с данными о контексте их упо-

Таблица 2

Признаки, вычисляемые по базовой коллекции

Признак	Формула	Пояснение
Частотность	$TF_t(w)$	Число употреблений слова в базовой коллекции
Документная частотность	$DF_t(w)$	Число документов, где встречается слово
TF-IDF [7]	$TF_t(w)\log\frac{\lvert D_t\rvert}{DF_t(w)}$	Поощряет слова, встречающиеся часто в небольшом числе текстов
TF-RIDF [8]	$TF_t(w)\left(\log\frac{\lvert D_t\rvert}{DF_t(w)} + \log\left(1 - \mathbf{e}^{-\frac{TF_t(w)}{\lvert D_t\rvert}}\right)\right)$	Расширение TF-IDF моделью Пуассона для предсказания терминологичности
Domain Consensus [9]	$-\sum_{d \in D}(freq(w, d_k)\log(freq(w, d_k))$	Признак, основанный на энтропии

Таблица 3

Признаки, вычисляемые по базовой и контрастной коллекциям

Признак	Формула	Пояснение
Относительная частотность [10]	$\frac{TF_t(w)}{\lvert W_t\rvert} \Big/ \frac{TF_r(w)}{\lvert W_r\rvert}$	Поощряет слова, встречающиеся чаще других в базовой коллекции
Релевантность [11]	$1 - \frac{1}{\log_2\left(2 + \frac{TF_t(w)\,DF_t(w)}{TF_r(w)}\right)}$	Штрафует слова, встречающиеся в небольшом числе текстов
Contrastive Weight [12]	$\log TF_t(w) \log \frac{\lvert W_t\rvert + \lvert W_r\rvert}{TF_t(w) + TF_r(w)}$	Предполагает, что обычные слова распределены между коллекциями одинаково
Discriminative Weight [13]	$\log_{10}(TF_t(w) + 10) \log_{10}\left(\frac{\lvert W_t\rvert + \lvert W_r\rvert}{TF_t(w) + TF_r(w)}\right) \log_2\left(\frac{TF_t(w)+1}{TF_r(w)+1} + 1\right)$	Поощряет слова, специфичные для базовой текстовой коллекции
KF-IDF [14]	$3DF_t(w)$, если w есть в контрастной коллекции, и $2DF_t(w)$ иначе	Поощряет слова, отсутствующие в контрастном корпусе
Логарифм правдоподобия [15]	$2\left(TF_t(w)\log\frac{TF_t(w)}{TF_t^e(w)} + TF_r(w)\log\frac{TF_r(w)}{TF_r^e(w)}\right)$, где $TF_x^e(w) = \lvert W_x\rvert \frac{TF_t(w)+TF_r(w)}{\lvert W_t\rvert + \lvert W_r\rvert}$	Модификация для однословных терминов

Таблица 4

Признаки, вычисляемые по статистической и контекстной информации из базовой коллекции

Признак	Формула	Пояснение
MC-value [16]	$TF_t(w) - \frac{\sum_{p \in P_w} CF_t(p)}{\lvert P_w\rvert}$	Модификация для однословных терминов, штрафует слова, являющиеся частями объемлющих фраз
NC-value [17]	$\frac{1}{\lvert W\rvert}$ MC-value(w)$c$weight(w), где $c\text{weight}(w) = \sum_{c \in C_w} \text{weight}(c) + 1$, $\text{weight}(c) = \frac{1}{2}\left(\frac{\lvert W_c\rvert}{\lvert W_t\rvert} + \frac{CF(w)}{TF_t(c)}\right)$	Модификация для однословных терминов, добавляет контекстную информацию в MC-value
MNC-value [18]	0,8MC-value(w) + 0,2CF(w)	Другая модификация NC-value для однословных терминов
Token-LR Type-LR [19]	$\frac{\sqrt{l_{token}(w) r_{token}(w)}}{\sqrt{l_{type}(w) r_{type}(w)}}$	Модификация для однословных терминов заменой слов, являющихся частями, контекстными словами
Token-FLR Type-FLR [19]	$TF(w)$Token-LR(w) $TF(w)$Type-LR(w)	Расширения Token-LR и Type-LR
Insideness [20]	$\frac{F_{\max}}{TF_t(w)}$	Выявляет слова, являющиеся частями объемлющих словосочетаний
Sum3, Sum10, Sum50 [20]	$\frac{\sum_{p \in P_w^N} TF_t(p)}{N}$	Определяет продуктивность слов в создании фраз
NearTermsFreq	NearTermsFreq(w)	Число вхождений слова в контекстное окно 10 наиболее часто встречающихся слов
NearTermsFreq-IDF	$\text{NearTermsFreq}(w) \log \frac{\lvert D_t\rvert}{DF_t(w)}$	Поощряет слова, располагающиеся часто рядом с наиболее часто встречающимися словами в небольшом числе текстов

требления в коллекции. В табл. 4 дополнительно используются обозначения:

P_w — множество всех объемлющих фраз, содержащих слово w;

C_w — множество всех контекстных слов для слова w;

$CF(w) = \sum_{c \in C_w} freq(c)$ — "контекстный фактор" слова w;

$|W_c|$ — число слов, являющихся контекстными для слова w;

$l_{token}(w)$ и $r_{token}(w)$ — суммы частотностей контекстных слов, расположенных в текстах слева и справа от слова w;

$l_{type}(w)$ и $r_{type}(w)$ — число различных контекстных слов, расположенных в текстах слева и справа от слова w;

$F_{\max}$ — максимальная частотность фразы, содержащей слово w;

P_w^N — множество N наиболее часто встречающихся фраз, содержащих слово w.

Последние два признака табл. 4 являются новыми: они были предложены нами исходя из предположения, что термины скорее всего располагаются в текстах рядом с наиболее часто встречающимися словами. Признак NearTermsFreq вычисляется как число вхождений данного слова в контекстное окно для нескольких наиболее часто встречающихся слов, в качестве таковых были взяты первые 10 элементов списка слов-кандидатов, упорядоченных наилучшим отдельным признаком. Еще один признак этой группы NearTermsFreq-IDF соединяет в себе предложенный признак и меру TF-IDF (из первой группы).

Эксперименты по отбору и комбинированию признаков

Для экспериментов были использованы базовая коллекция русскоязычных текстов из банковской сферы (10 422 документов, примерно 15,5 млн слов), а также подсчитанные заранее частотности слов по контрастной коллекции более общей тематики (примерно 1 млн новостных текстов). Все описанные выше признаки вычислялись для списка из 5000 наиболее часто встречающихся слов-кандидатов, извлеченных из базовой коллекции.

Для оценивания результатов извлечения терминов была выбрана мера средней точности AvP [7], определяемая для множества D всех слов-кандидатов и его подмножества $D_q \subseteq D$, представляющего действительно термины (т. е. подтвержденные тезаурусом):

$$\mathrm{AvP}(D) = \frac{1}{|D_q|} \sum_{1 \leqslant k \leqslant |D_q|} \left(r_k \left(\frac{1}{k} \sum_{1 \leqslant i \leqslant k} r_i \right) \right),$$

где $r_i = 1$, если i-е слово-кандидат $\in D_q$, и $r_i = 0$ иначе. Данная формула отражает тот факт, что чем больше терминов сосредоточено в вершине списка слов-кандидатов, тем выше мера средней точности.

Для поиска наилучшей комбинации признаков были опробованы на всем наборе рассмотренных признаков несколько методов машинного обучения, представленных в библиотеке Weka [21], включая линейную и логистическую регрессии, метод J48, LogitBoost, Naive Bayes, деревья решений. При этом проводилась четырехкратная кросспроверка, означающая, что вся исходная выборка разбивалась случайным образом на четыре равные непересекающиеся части, и каждая часть по очереди становилась контрольной подвыборкой, а обучение проводилось по остальным трем. Оказалось, что наилучшее значение средней точности достигается с помощью метода логистической регрессии, которая и была использована в дальнейших экспериментах.

Результаты средней точности представлены в табл. 5. Хотя средняя точность извлечения терминов вычислялась для всех отдельных признаков, в таблицу включены данные только по одному признаку из каждой группы статистических признаков, дающему наилучшую точность. Последняя строка таблицы соответствует комбинации всех признаков с помощью логистической регрессии. Видно, что наилучшим отдельным признаком оказался **TF-RIDF**, и по сравнению с ним логистическая регрессия дает относительный прирост средней точности в 33 %.

В табл. 6 представлены первые 10 элементов из списка извлеченных слов-кандидатов, упорядоченных методом логистической регрессии, а также в соответствии со значениями признаков частотности (как базового способа извлечения) и TF-RIDE (как наилучшего отдельного признака), при этом термины выделены курсивом.

Таблица 5

Средняя точность для отдельных признаков и комбинирования

№	Группа признаков	Лучший признак	AvP, %
1	Использующие только базовую коллекцию		
	Для всех слов-кандидатов	**TF-RIDF**	**41,13**
	Для подлежащих	TF-RIDF$_{subjects}$	35,15
	Для слов с большой буквы	TF-RIDF$_{capital\ words}$	38,85
	Для слов с большой буквы, не стоящих в начале предложения	TF$_{non\text{-}initial\ words}$	39,26
2	Использующие базовую и контрастную коллекции	Логарифм правдоподобия	36,89
3	Использующие статистическую и контекстную информацию	Sum3	37,41
4	Для слов, стоящих рядом с наиболее часто встречающимися словами	NearTermsFreq-IDF$_{ref}$	35,34
	Логистическая регрессия		**54,59**

Таблица 6

Первые десять извлеченных слов-кандидатов

№	Логистическая регрессия	Частотность	TF-RIDF
1	*Банковский*	*Банк*	*Банк*
2	*Банк*	*Банковский*	*Кредитный*
3	Год	Год	*Банковский*
4	*РФ*	*Россия*	*Риск*
5	*Кредитный*	Банка	*Кредит*
6	*Налоговый*	*Система*	*Рынок*
7	*Кредит*	Организация	*Система*
8	*Пенсионный*	*Рынок*	Банка
9	Средство	*Кредитный*	*Страна*
10	Клиент	*Российский*	*Налоговый*

Ясно, что извлечение терминов на основе комбинации сразу всех признаков требует больших вычислительных затрат. Чтобы убрать из рассмотрения избыточные признаки, был применен жадный алгоритм отбора самых важных признаков Add. Алгоритм начинал работу с пустого множества признаков и на каждом шаге добавлял признак, максимизирующий общую среднюю точность. В итоге был найден набор из восьми признаков, дающий 54,26 % средней точности и включающий признаки: *TF-IDF*, *KF-IDF*, *Относительная частотность*, $DF_{non\text{-}initial\ worlds}$, *NC-value*, $TF\text{-}IDF^{reference}_{subjects}$, *Type-FLR* и $TF\text{-}IDF^{reference}_{capital\ words}$. Отметим, что в этом наборе присутствуют представители всех трех групп статистических признаков, включая и признаки, вычисленные только на введенных выше подмножествах слов-кандидатов. Таким образом, число комбинируемых признаков может быть существенно сокращено с потерей менее 1 % в средней точности.

Добавим, что максимальное значение средней точности (55,9 %) достигается на комбинации 30 из 50 признаков. Однако полученные результаты точности извлечения все же хуже аналогичных для многословных терминов [6].

Эксперименты по извлечению при расширении базовой коллекции

Для экспериментального исследования устойчивости работы метода извлечения терминов с помощью комбинации признаков, найденной на основе машинного обучения, базовая текстовая коллекция случайным образом была разбита на две части разного размера. На меньшей части (примерно 1 млн слов) была настроена логистическая регрессия, комбинирующая все признаки, а затем она была протестирована на всей коллекции, и при этом было получено всего лишь 35,93 % средней точности извлечения. Для повышения точности была проведена нормализация числовых значений всех статистических признаков к диапазону [0,1] с нормализующим коэффициентом, равным десятикратному среднему значению признака (число "10" было выбрано экспериментально как наилучшее). В итоге, в ходе эксперимента с нормализованными значениями было получено 47,26 % средней точности для всей текстовой коллекции.

Дополнительно была экспериментально изучена зависимость средней точности от размера меньшей части текстовой коллекции, участвующей в обучении. Нормализованный и ненормализованный варианты комбинирования признаков сравнивались по результатам кросспроверки на меньшей части коллекции и тестирования на всем корпусе текстов (табл. 7).

Видно, что ненормализованный вариант дает лучшие результаты на меньшей части коллекции и худшие на всей коллекции, в то время как для нормализованного варианта ситуация полностью противоположная.

Заключение

В работе описаны эксперименты по автоматическому извлечению однословных терминов из коллекций русскоязычных текстов на основе методов машинного обучения, в ходе которых была найдена комбинация лингвистических и статистических признаков, значительно улучшающая среднюю точность извлечения терминов. Также было установлено, что из примененных методов обучения логистическая регрессия дает лучшие результаты, а число комбинируемых признаков может быть сокращено до восьми без существенной потери точности. Кроме того, найденная на сравнительно небольшой текстовой коллекции комбинация признаков может быть впоследствии использована для извлечения терминов из расширенной текстовой коллекции без значительной потери качества.

Таблица 7

Средняя точность при расширении текстовой коллекции

Варинат	Текстовая коллекция	Размер обучающей части коллекции (млн слов)					
		1	2	3	4	5	6
Ненормализованный	Меньшая часть Весь корпус	49,09 % 35,93 %	51,66 % 47,05 %	52,96 % 50,29 %	53,72 % 51,13 %	53,42 % 52,56 %	54,3 % 53,42 %
Нормализованный	Меньшая часть Весь корпус	47,77 % 47,26 %	49,92 % 51,44 %	50,94 % 53,19 %	52,1 % 53,83 %	52,14 % 54,5 %	52,5 % 54,82 %

Список литературы

1. **Лукашевич Н. В.** Тезаурусы в задачах информационного поиска. М.: Изд-во МГУ, 2011.
2. **Pecina P. and Schlesinger P.** Combining Association Measures for Collocation Extraction // Proc. of the COLING/ACL 2006. ACL Press. 2006. P. 651—658.
3. **Azé J., Roche M., Kodratoff Y., Sebag M.** Preference Learning in Terminology Extraction: A ROC-based Approach // Proc. of ASMDA'05. 2005. P. 209—219.
4. **Foo J., Merkel M.** Using Machine Learning to Perform Automatic Term Recognition // Proc. of the LREC2010 Acquisition Workshop. Malta. 2010.
5. **Dobrov B., Loukachevitch N.** Multiple Evidence for Term Extraction in Broad Domains // Proc. of RANLP 2011. Hissar. 2011. P. 710—715.
6. **Zhang Z., Iria J., Brewster C., Ciravegna F.** A Comparative Evaluation of Term Recognition Algorithms // Proc. of LREC 2008. 2008.
7. **Manning C.D. and Schutze H.** Foundations of Statistical Language Processing. Cambridge MA: MIT Press. 1999.
8. **Church K. and Gale W.** Inverse Document Frequency IDF: A Measure of Deviation from Poisson // Proc. of the Third Workshop on Very Large Corpora. Cambridge MA: MIT Press, 1995. P. 121—130.
9. **Naivigli R. and Velardi P.** Semantic Interpretation of Terminological Strings // Proc. of TKE 2002. 2002. P. 95—100.
10. **Ahmad K., Gillam L., Tostevin L.** University of Survey Participation in Trec8: Weirdness indexing for logical document extrapolation and retrieval // Proc. of TREC 1999. 1999.
11. **Peñas A., Verdejo V., Gonzalo J.** Corpus-based Terminology Extraction Applied to Information Access // Proc. of the Corpus Linguistics 2001 Conference. 2001. P. 458—465.
12. **Basili R., Moschitti A., Pazienza M., Zanzotto F.** A Contrastive Approach to Term Extraction // Proc. of the 4th Terminology and Artificial Intelligence Conference. 2001.
13. **Wong W., Liu W., Bennamoun M.** Determining Termhood for Learning Domain Ontologies using Domain Prevalence and Tendency // Proc. of the 6th Australasian Conference on Data Mining. 2007. P. 47—54.
14. **Kurz D. and Xu F.** Text Mining for the Extraction of Domain Retrieval Terms and Term Collocations // Proc. of the Int. Workshop on Computational Approaches to Collocations. 2002.
15. **Gelbukh A., Sidorov G., Lavin-Villa E., Chanona-Hernandez L.** Automatic Term Extraction using Log-likelihood based Comparison with General Reference Corpora // Proc. of the Natural Language Processing and Information Systems. 2010. P. 248—255.
16. **Nakagawa H. and Mori T.** A Simple but Powerful Automatic Term Extraction Method // Proc. of the Second Int. Workshop on Computational Terminology. 2002. P. 29—35.
17. **Frantzi K. and Ananiadou S.** Automatic Term Recognition Using Contextual Cues // Proc. of the IJCAI Workshop on Multilinguality in Software Industry: the AI Contribution. 1997.
18. **Frantzi K. and Ananiadou S.** The C-value/NC-value Domain-Independent Method for Multi-word Term Extraction // Journal of Natural Language Processing. 2000. Vol. 6. N 3. P. 145—179.
19. **Nakagawa H. and Mori T.** Automatic Term Recognition based on Statistics of Compound Nouns and their Components // Terminology. 2003. Vol. 9. N 2. P. 201—219.
20. **Лукашевич Н. и Логачев Ю.** Использование методов машинного обучения для извлечения терминов // Труды КИИ-2010. 2010.
21. **Weka 3.** Data Mining Software in Java. URL: http://www.cs.waikato.ac.nz/ml/weka

УДК 004.85, 519.724, 519.177

Б. Г. Кухаренко, канд. физ.-мат. наук,
ст. науч. сотр., вед. науч. сотр.,
Институт машиноведения РАН, г. Москва,
e-mail: kukharenko@imash.ru,
М. О. Солнцева, аспирант,
Московский физико-технический институт (ГУ),
e-mail: solnceva.chalei@gmail.com

Принцип минимальной длины описания при анализе графов с разреженными матрицами смежности в задачах кластеризации их узлов

В машинном обучении принцип минимальной длины описания (MDL-критерий) определяет порядок модели. При кластеризации узлов графа на основе EM-алгоритма применение MDL-критерия позволяет оценить число кластеров для набора узлов. Для графов с разреженными матрицами смежности MDL-критерий определяет число кластеров в результате анализа этих разреженных матриц по методу перекрестных ассоциаций. Полученная оценка задает начальное число кластеров при кластеризации узлов графов на основе алгоритмов спектральной кластеризации. В качестве примера рассматривается кластеризация узлов транспортной сети.

***Ключевые слова**: принцип минимальной длины описания, графы, разреженные матрицы смежности, EM-алгоритм, метод перекрестных ассоциаций, алгоритмы спектральной кластеризации, транспортные сети*

Введение

Определение размерности модели является одной из основных задач индуктивного статистического вывода [1]. Общее решение этой задачи формулирует *принцип минимальной длины описания данных* (Minimum Description Length (MDL) Principle) [2]. MDL-принцип утверждает, что любая закономерность данных может быть использована для их сжатия. Таким образом, для описания данных требуется гораздо меньше символов, чем число символов, необходимое для посимвольного представления этих данных. MDL-принцип объединяет две концепции: обучение на основе данных и сжатие данных. Он действует как "бритва Оккама", отбирая модель с оптимальным балансом точности описания данных и сложности модели. Сжатие данных формально эквивалентно вероятностному предсказанию, поэтому MDL-методы могут интерпретироваться как методы поиска модели с хорошей предсказательной способностью на скрытых данных [3]. В настоящей работе для идентификации порядка модели используется MDL-критерий, предложенный Риссаненом [4, 5].

В задачах кластеризации вершин графа порядок модели — это определяемое число кластеров. Непосредственный подход к кластеризации узлов графа состоит в использовании в качестве статистической модели для репрезентативной выборки узлов графа смеси Гауссовых распределений в пространстве координат и других параметров, характеризующих узлы графа [6]. Каждый кластер характеризуется набором параметров соответствующего

Гауссова распределения. Конкретный узел графа может быть отнесен к каждому из кластеров с различной вероятностью. Несмотря на то, что при использовании модели смеси Гауссовых распределений MDL-критерий для определения числа кластеров не всегда является лучшим, однако он эффективен с вычислительной точки зрения. Для непосредственной минимизации MDL-критерия Риссанена в работе [6] используется алгоритм ожидания и максимизации правдоподобия (EM-алгоритм) [7, 8].

В настоящей работе при кластеризации узлов графа с разреженной матрицей смежности MDL-критерий используется для определения числа кластеров также при предварительном анализе этой матрицы смежности с помощью алгоритма перекрестных ассоциаций (cross associations) [9]. В результате задается начальное число кластеров при кластеризации узлов графов на основе алгоритма спектральной кластеризации графа без вычисления собственных векторов [10]. Реализация этого подхода привела к разработке программного обеспечения с использованием пакета Graclus [11, 12], которое позволяет быстро разбить граф на кластеры. В настоящей работе представлены численные эксперименты по применению MDL-критерия для определения числа кластеров в наборе узлов графа транспортной сети Омской области.

1. Оценка числа кластеров в модели смеси Гауссовых распределений по MDL-критерию при использовании EM-алгоритма

Пусть $\mathbf{y} = \{\mathbf{y}_n, n = \overline{1, N}\}$ — набор векторов, представляющих узлы графа. Кроме этого, каждый узел $n = \overline{1, N}$ характеризуется случайной переменной x_n, $n = \overline{1, N}$, которая определяет кластер этого узла. Каждый кластер имеет Гауссово распределение, т. е. распределение для вектора $\mathbf{y}_n$ при $x_n = k$ имеет вид

$$N(\mathbf{y}_n|\boldsymbol{\mu}_k, \mathbf{R}_k) =$$

$$= \frac{1}{(2\pi)^{M/2}} |\mathbf{R}_k|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{y}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{y}_n - \boldsymbol{\mu}_k)\right\},$$

где $\boldsymbol{\mu}_k$ — M-мерный вектор среднего кластера; $\mathbf{R}_k$ — матрица ковариации размерности $M \times M$ для кластера k; $|..|$ обозначает детерминант. Поскольку значение x_n, $n = \overline{1, N}$, для каждого вектора $\mathbf{y}_n$, $n = \overline{1, N}$, неизвестно, используется модель смеси Гауссовых распределений

$$p(\mathbf{y}_n|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k N(\mathbf{y}_n|\boldsymbol{\mu}_k, \mathbf{R}_k), \qquad (1)$$

где π_k — вероятность того, что узел графа относится к кластеру k. Полный набор параметров для распределения (1) — это число кластеров K и $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{R}\}$, где $\boldsymbol{\pi} = \{\pi_k, k = \overline{1, K}\}$, $\pi_k \geqslant 0$, $k = \overline{1, K}$, $\sum\limits_k^K \pi_k = 1$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k, k = \overline{1, K}\}$ и $\mathbf{R} = \{\mathbf{R}_k, k = \overline{1, K}\}$, $\det(\mathbf{R}_k) > \varepsilon$, ε — параметр задачи.

Пусть $\boldsymbol{\Omega}^{(K)}$ — набор допустимых значений параметров $\boldsymbol{\theta}$ для модели K-го порядка. Функция правдоподобия определяется как логарифм вероятности для набора векторов $\mathbf{y} = \{\mathbf{y}_n, n = \overline{1, N}\}$:

$$\log p(\mathbf{y}|K, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k N(\mathbf{y}_n|\boldsymbol{\mu}_k, \mathbf{R}_k)\right). \qquad (2)$$

Оценка значения параметров K и $\boldsymbol{\theta} \in \boldsymbol{\Omega}^{(K)}$ по максимуму правдоподобия (2) имеет вид

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta} \in \Omega^{(K)}} \log p(\mathbf{y}|K, \boldsymbol{\theta}).$$

Для приближенной оценки порядка модели используется MDL-критерий

$$\mathrm{MDL}(K, \boldsymbol{\theta}) =$$

$$= -\sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k N(\mathbf{y}_n|\boldsymbol{\mu}_k, \mathbf{R}_k)\right) + \frac{1}{2} L \log(NM), \quad (3)$$

где $L = K\left(1 + M + \frac{(M+1)M}{2}\right) - 1$ — полное число параметров $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{R}\}$ [4—6]. Непосредственная минимизация MDL-критерия (3) трудна, поэтому используется EM-алгоритм [7, 8].

На каждой итерации i EM-алгоритма по правилу Байеса определяется вероятность принадлежности $\mathbf{y}_n$, $n = \overline{1, N}$, к различным кластерам $k = \overline{1, K}$:

$$p(k|\mathbf{y}_n, \boldsymbol{\mu}_k^{(i)}, \mathbf{R}_k^{(i)}) = \frac{\pi_k N(\mathbf{y}_n|\boldsymbol{\mu}_k^{(i)}, \mathbf{R}_k^{(i)})}{\sum\limits_{l=1}^{K} \pi_l N(\mathbf{y}_n|\boldsymbol{\mu}_l^{(i)}, \mathbf{R}_l^{(i)})}.$$

Затем вычисляются оценки параметров распределения (1)

$$N_k^{(i+1)} = \sum_{n=1}^{N} p(k|\mathbf{y}_n, \boldsymbol{\mu}_k^{(i)}, \mathbf{R}_k^{(i)});$$

$$\pi_k^{(i+1)} = \frac{N_k^{(i+1)}}{N};$$

$$\boldsymbol{\mu}_k^{(i+1)} = \frac{1}{N_k^{(i+1)}} \sum_{n=1}^{N} \mathbf{y}_n p(k|\mathbf{y}_n, \boldsymbol{\mu}_k^{(i)}, \mathbf{R}_k^{(i)});$$

$$\mathbf{R}_k^{(i+1)} = \frac{1}{N_k^{(i+1)}} \sum_{n=1}^{N} (\mathbf{y}_n - \boldsymbol{\mu}_k^{(i+1)}) \times$$

$$\times (\mathbf{y}_n - \boldsymbol{\mu}_k^{(i+1)})^{\mathrm{T}} p(k|\mathbf{y}_n, \boldsymbol{\mu}_k^{(i)}, \mathbf{R}_k^{(i)}).$$

При выводе формул для оценок EM-алгоритма используется функция

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = E[\log p(\mathbf{y}, \mathbf{x}|\boldsymbol{\theta})|\mathbf{y}, \boldsymbol{\theta}^{(i)}] - \frac{1}{2} L \log(NM),$$

где $\mathbf{y} = \{\mathbf{y}_n,\ n = \overline{1, N}\}$ и $\mathbf{x} = \{x_n,\ n = \overline{1, N}\}$. Фундаментальный результат для EM-алгоритма [7] состоит в том, что для любого $\boldsymbol{\theta}$

$$\mathrm{MDL}(K, \boldsymbol{\theta}) - \mathrm{MDL}(K, \boldsymbol{\theta}^{(i)}) < Q(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i)}) - Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}).$$

Любое значение $\boldsymbol{\theta}$, которое увеличивает $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})$, гарантирует уменьшение значения MDL-критерия (3).

Можно уменьшить число кластеров от K до $K-1$. Например, два кластера l и m могут быть объединены в один (l, m) при условии

$$\boldsymbol{\mu}_l = \boldsymbol{\mu}_m = \boldsymbol{\mu}_{(l, m)},\ \mathbf{R}_l = \mathbf{R}_m = \mathbf{R}_{(l, m)}, \tag{4}$$

где $\boldsymbol{\mu}_{(l, m)}$ и $\mathbf{R}_{(l, m)}$ обозначают среднюю ковариацию нового кластера, и предполагается, что значения π_l и π_m остаются неизменными и априорная вероятность

$$\pi_{(l, m)} = \pi_l + \pi_m. \tag{5}$$

Обозначаем модифицированный вектор параметров как $\boldsymbol{\theta}_{(l, m)} \in \boldsymbol{\Omega}^{(K)}$ и выделяем в нем вектор параметров $\boldsymbol{\theta}_{(l, m)-} \in \boldsymbol{\Omega}^{(K-1)}$ для $K-1$ отдельных кластеров. При ограничениях (4) и (5) оценки среднего и ковариации кластера (l, m) имеют вид

$$\boldsymbol{\mu}_{(l, m)} = \frac{\pi_l\boldsymbol{\mu}_l - \pi_m\boldsymbol{\mu}_m}{\pi_l + \pi_m}; \tag{6}$$

$$\mathbf{R}_{(l, m)} =$$

$$= \frac{\pi_l(\mathbf{R}_l + (\boldsymbol{\mu}_l - \boldsymbol{\mu}_{(l, m)})(\boldsymbol{\mu}_l - \boldsymbol{\mu}_{(l, m)})^{\mathrm{T}}) + \pi_m(\mathbf{R}_m + (\boldsymbol{\mu}_m - \boldsymbol{\mu}_{(l, m)})(\boldsymbol{\mu}_m - \boldsymbol{\mu}_{(l, m)})^{\mathrm{T}})}{\pi_l + \pi_m}.$$

Используя (6), определяем функцию расстояния

$$d(l, m) = Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) - Q(\boldsymbol{\theta}_{(l, m)}, \boldsymbol{\theta}^{(i)}) =$$

$$= N\pi_l\left\{-\frac{M}{2}(1 - \log(2\pi)) - \frac{1}{2}\log(|\mathbf{R}_l|)\right\} +$$

$$+ N\pi_m\left\{-\frac{M}{2}(1 - \log(2\pi)) - \frac{1}{2}\log(|\mathbf{R}_m|)\right\} -$$

$$- 2N\pi_{(l, m)}\left\{-\frac{M}{2}(1 - \log(2\pi)) - \frac{1}{2}\log(|\mathbf{R}_{(l, m)}|)\right\} =$$

$$= \frac{N\pi_l}{2}\log\left(\frac{|\mathbf{R}_{(l, m)}|}{\mathbf{R}_l}\right) + \frac{N\pi_m}{2}\log\left(\frac{|\mathbf{R}_{(l, m)}|}{\mathbf{R}_m}\right). \tag{7}$$

Функция расстояния (7) является верхней границей для MDL-критерия

$$\mathrm{MDL}(K-1, \boldsymbol{\theta}_{(l, m)-}) - \mathrm{MDL}(K, \boldsymbol{\theta}^{(i)}) \leqslant$$

$$\leqslant d(l, m) - \frac{1}{2}\left(1 + M + \frac{(M+1)M}{2}\right)\log(MN).$$

Для функции $d(l, m)$ (7) можно найти пару, минимизирующую эту верхнюю границу MDL-критерия:

$$(l^*, m^*) = \arg\min_{(l, m)} d(l, m).$$

Такие два кластера можно объединить, и параметры объединенного кластера вычисляются с помощью выражений (4), (5). Результирующий набор параметров $\boldsymbol{\theta}_{(l, m)}$ используется в качестве исходного для дальнейшей EM-оптимизации с $K-1$ кластерами. При задании исходного вектора параметров $\boldsymbol{\theta}^{(1)}$ начальное число кластеров K_0 выбирается таким образом, чтобы для полного числа параметров $\boldsymbol{\theta}^{(1)} = \{\boldsymbol{\pi}^{(1)}, \boldsymbol{\mu}^{(1)}, \mathbf{R}^{(1)}\}$ выполнялось условие $L < \frac{1}{2}MN$. Параметры исходных кластеров выбираются в виде ($[x]$ обозначает целую часть x)

$$\pi_k^{(1)} = \frac{1}{K_0};$$

$$\boldsymbol{\mu}_k^{(1)} = \mathbf{y}_n,$$

$$\text{где } n = [(k-1)(N-1)/(K_0-1)] + 1, \tag{8}$$

$$\mathbf{R}_k^{(1)} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\mathbf{y}_n^{\mathrm{T}}.$$

2. Применение MDL-критерия при анализе разреженных матриц смежности графов

Структура графа отображается его матрицей смежности — бинарной матрицей из нулей и единиц. Фундаментальным при анализе больших разреженных бинарных матриц является поиск в них скрытой структуры. Такой анализ может быть выполнен с помощью алгоритма перекрестных ассоциаций, выделяющего в матрице смежности кластеры с однородной внутренней структурой [9]. Если исходные элементы матрицы смежности составляют $m \cdot n$ "прямоугольников" с "плотностью" либо 0, либо 1, то при завершении алгоритма выявляются прямоугольные блоки с плотностью от 0 до 1, число которых и подлежало определению. Отбрасывание части этих прямоугольников уменьшает сложность описания матрицы смежности. Для оценки сложности описания матрицы смежности в алгоритме перекрестных ассоциаций применяется MDL-критерий, в котором стоимость структуры, выделяемой в матрице, оценивается числом битов, необходимых для передачи всей структуры вместе с данными о каждой выделенной прямоугольной области.

Пусть $\mathbf{D} \in \mathbf{R}^{m \times n}$ — матрица с бинарными элементами $D[i; j] \in \{0,1\}$. Припишем строки матрицы $\mathbf{D}$ к группам строк, а ее столбцы — к группам столбцов:

$$\Psi:\{1, 2, ..., m\} \to \{1, 2, ..., k\},$$
$$\Phi:\{1, 2, ..., n\} \to \{1, 2, ..., l\},$$

где k и l обозначают число независимых групп среди строк и столбцов соответственно. Согласно этому приписыванию (или "перекрестной ассоциации" $\{\Psi, \Phi\}$) элементы матрицы $\mathbf{D}$ перестраиваются таким образом, чтобы строки матрицы $\mathbf{D}$, соответствующие группам 1, 2 и т. д., перечислялись в соответствии с этим порядком, и аналогично — для столбцов матрицы $\mathbf{D}$. Такая перестройка разделяет исходную матрицу $\mathbf{D}$ на прямоугольные блоки мень-

шего размера — подматрицы $\mathbf{D}_{ij}$, $i = \overline{1, k}$, $j = \overline{1, l}$, размерности $a_i \times b_j$, $i = \overline{1, k}$, $j = \overline{1, l}$.

Пусть для матрицы $\mathbf{D}_{ij} \in \mathbf{R}^{a_i \times b_j}$ с бинарными элементами $n_1(\mathbf{D}_{ij})$ — число ненулевых элементов матрицы, $n_0(\mathbf{D}_{ij})$ — число нулевых элементов матрицы, $n(\mathbf{D}_{ij}) = n_1(\mathbf{D}_{ij}) + n_0(\mathbf{D}_{ij}) = a_i \times b_j$, $P_{\mathbf{D}}(k) = \dfrac{n_k(\mathbf{D}_{ij})}{n(\mathbf{D}_{ij})}$, $k = 1,0$. Тогда полная длина кода в битах

$$C(\mathbf{D}_{ij}) = \sum_{i=0}^{1} n_i(\mathbf{D}_{ij}) \log\left(\frac{n(\mathbf{D}_{ij})}{n_i(\mathbf{D}_{ij})}\right) = n(\mathbf{D}_{ij}) H(P_{\mathbf{D}_{ij}}(0)),$$

где $H(..)$ — бинарная энтропия Шэннона. Полная длина кода для матрицы $\mathbf{D}$ с учетом заданных перекрестных ассоциаций имеет вид

$$T(\mathbf{D}, k, l, \Psi, \Phi) = \log k + \log l + \sum_{i=1}^{k-1} \log \overline{a}_i + \sum_{j=1}^{l-1} \log \overline{b}_j + \sum_{i=1}^{k} \sum_{j=1}^{l} \log(a_i b_j + 1) + \sum_{i=1}^{k} \sum_{j=1}^{l} C(\mathbf{D}_{ij}), \quad (9)$$

где $\overline{a}_i = \left(\sum_{t=i}^{k} a_t \right) - k + i$, $i = \overline{1, k-1}$,

$\overline{b}_j = \left(\sum_{t=j}^{l} b_t \right) - l + j$, $j = \overline{1, l-1}$.

Оптимальная перекрестная ассоциация соответствует числу групп строк k^*, числу групп столбцов l^* и перекрестным ассоциациям $\{\Psi^*, \Phi^*\}$ таким, что полная результирующая длина кода $T(\mathbf{D}, k^*, l^*, \Psi^*, \Phi^*)$ (9) минимизируется. Задача определения оптимальных перекрестных ассоциаций является вычислительно сложной, поэтому для ее решения применяется эвристика, состоящая из двух шагов.

1. Внутренний цикл: для заданных чисел k и l найти подходящую перегруппировку (т. е. перекрестную ассоциацию), соответствующую достижению локального минимума функции

$$\sum_{i=1}^{k} \sum_{j=1}^{l} C(\mathbf{D}_{ij}). \quad (10)$$

2. Внешний цикл: поиск наилучших k и l среди тех, которые рассматриваются во внутреннем цикле. Этот этап использует резкое падение функции (10) при малых значениях k и l.

Оценка сложности алгоритма: на каждом шаге алгоритма возрастает либо k, либо l, поэтому сумма $k + l$ всегда возрастает на 1. Следовательно, общая сложность алгоритма оценивается как $O(n_i(\mathbf{D})(k^* + l^*))^2$. На практике для нахождения перекрестных ассоциаций достаточно 20 итераций.

3. Спектральная кластеризация графов со взвешенными дугами

Оценки затрат памяти при спектральной кластеризации узлов графа, обеспечивающей нелинейное разделение этих узлов, порядка $O(KN)$, где K — число спектральных векторов Лапласиана графа

$$\mathbf{L} = \mathbf{D} - \mathbf{A},$$

где $\mathbf{A} = \{A[i; j], i = \overline{1, N}, j = \overline{1, N}\}$ — матрица смежности графа; $\mathbf{D} = \operatorname{diag}\left\{ \sum_{j=1}^{N} A[i; j] \right\}$ — диагональная матрица степеней, N — число узлов графа [10].

В настоящей работе используется алгоритм эквивалентный спектральной кластеризации графов, но без вычисления спектральных векторов и с оценкой затрат памяти порядка $O(N)$ — алгоритм kernel K-means. Этот алгоритм является обобщением стандартного алгоритма K-means [11]. Он использует функции отображения $\boldsymbol{\varphi}(..)$ в пространство более высокой размерности для нелинейного разделения векторов, представляющих узлы графа. В настоящей работе используется программный пакет Graclus, написанный на основе алгоритма kernel K-means [12, 13]. Пусть дан набор векторов $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$. Как алгоритм K-means, алгоритм kernel K-means определяет кластеры $\{\pi_1, \pi_2, ..., \pi_K\}$ ($\{|\pi_1|, |\pi_2|, ..., |\pi_K|\}$ — числа элементов в кластерах), центры которых $\mathbf{m}_k$, $k = \overline{1, K}$, и принадлежности векторов $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ этим кластерам минимизируют целевую функцию

$$D(\{\pi_k, k = \overline{1, K}\}) = \sum_{k=1}^{K} \sum_{\mathbf{y}_i \in \pi_k} \|\boldsymbol{\varphi}(\mathbf{y}_i) - \mathbf{m}_k\|^2, \quad (11)$$

т. е. $\mathbf{m}_k = \dfrac{\sum_{\mathbf{y}_i \in \pi_k} \boldsymbol{\varphi}(\mathbf{y}_i)}{|\pi_k|}$, $k = \overline{1, K}$. Поэтому в формуле (11) квадрат расстояния имеет вид

$$\|\boldsymbol{\varphi}(\mathbf{y}_i) - \mathbf{m}_k\|^2 = \boldsymbol{\varphi}(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_i) - \frac{2 \sum_{\mathbf{y}_j \in \pi_k} \boldsymbol{\varphi}(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_j)}{|\pi_k|} + \frac{\sum_{\mathbf{y}_j, \mathbf{y}_l \in \pi_k} \boldsymbol{\varphi}(\mathbf{y}_j)\boldsymbol{\varphi}(\mathbf{y}_l)}{|\pi_k|^2}. \quad (12)$$

В выражение (12) входят только скалярные произведения функций отображения $\boldsymbol{\varphi}(\mathbf{y}_i)$ и $\boldsymbol{\varphi}(\mathbf{y}_j)$. Если задана матрица ядра $\mathbf{K}$ с элементами $K[i; j] = \boldsymbol{\varphi}(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_j)$, $i = \overline{1, N}$, $j = \overline{1, N}$, то квадраты расстояния (12) между векторами $\mathbf{y}_i$, $i = \overline{1, N}$, и центрами кластеров $\mathbf{m}_k$, $k = \overline{1, K}$, могут быть вычислены без знания значений функции $\boldsymbol{\varphi}(\mathbf{y}_i)$ и $\boldsymbol{\varphi}(\mathbf{y}_j)$. Элементы ядра $K[i; j]$, $i = \overline{1, N}$, $j = \overline{1, N}$, отображают исходные векторы $\mathbf{y}_i$, $i = \overline{1, N}$, в скалярные произведения $\boldsymbol{\varphi}(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_j)$, $i = \overline{1, N}$, $j = \overline{1, N}$. Показано, что любая положи-

тельная полуопределенная матрица **K** может быть ядром в (12) [10].

Целевая функция алгоритма kernel K-means со взвешенными квадратами расстояний имеет вид

$$D(\{\boldsymbol{\pi}_k,\ k = \overline{1, K}\}) = \sum_{k=1}^{K} \sum_{\mathbf{y}_i \in \pi_k} w_i \|\boldsymbol{\varphi}(\mathbf{y}_i) - \mathbf{m}_k\|^2, \quad (13)$$

т. е. $\mathbf{m}_k = \dfrac{\sum\limits_{\mathbf{y}_i \in \pi_i} w_i \boldsymbol{\varphi}(\mathbf{y}_i)}{\sum\limits_{\mathbf{y}_i \in \pi_i} w_i}$, $k = \overline{1, K}$ (веса $w_i \geqslant 0$, $i = \overline{1, N}$).

Отметим, что $\mathbf{m}_k$ представляет "лучший" кластер, поскольку $\mathbf{m}_k = \arg\min\limits_{\mathbf{z}} \sum\limits_{\mathbf{y}_i \in \pi_c} w_i \|\boldsymbol{\varphi}(\mathbf{y}_i) - \mathbf{z}\|^2$. Как и в формуле (12), в выражение для квадрата расстояния (13) входят только скалярные произведения значений функции отображения $\boldsymbol{\varphi}(\mathbf{y}_i)$ и $\boldsymbol{\varphi}(\mathbf{y}_j)$, поскольку

$$\|\boldsymbol{\varphi}(\mathbf{y}_i) - \mathbf{m}_k\|^2 = \boldsymbol{\varphi}(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_i) - \frac{2 \sum\limits_{\mathbf{y}_j \in \pi_k} \boldsymbol{\varphi}(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_j)}{\sum\limits_{\mathbf{y}_j \in \pi_k} w_i} + \frac{\sum\limits_{\mathbf{y}_j, \mathbf{y}_l \in \pi_k} \boldsymbol{\varphi}(\mathbf{y}_j)\boldsymbol{\varphi}(\mathbf{y}_l)}{\left(\sum\limits_{\mathbf{y}_j \in \pi_k} w_j\right)^2}. \quad (14)$$

Используя матрицу ядра $\mathbf{K} = \{K[i; j] = \boldsymbol{\varphi}(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_j)$, $i = \overline{1, N}$, $j = \overline{1, N}\}$, квадрат расстояния (14) представим в виде

$$\|\boldsymbol{\varphi}(\mathbf{y}_i) - \mathbf{m}_k\|^2 = K[i; i] - \frac{2 \sum\limits_{\mathbf{y}_j \in \pi_k} K[i; j]}{\sum\limits_{\mathbf{y}_j \in \pi_k} w_j} + \frac{\sum\limits_{\mathbf{y}_j, \mathbf{y}_l \in \pi_k} K[j; l]}{\left(\sum\limits_{\mathbf{y}_j \in \pi_k} w_j\right)^2}. \quad (15)$$

Определим диагональную матрицу весов

$$\mathbf{W} = \mathrm{diag}\{w_1, \ldots, w_N\}. \quad (16)$$

Как показано в работе [10], целевая функция спектральной кластеризации взвешенного графа получается из формулы (13) с учетом (15) при выборе ядра $\mathbf{K} = \{K[i; j],\ i = \overline{1, N},\ j = \overline{1, N}\}$ в виде

$$\mathbf{K} = \mathbf{W}^{-1}\mathbf{A}\mathbf{W}^{-1}, \quad (17)$$

где **A** — матрица смежности графа; **W** — диагональная матрица весов (16). Однако для произвольной матрицы смежности **A** ядро **K** (17) — не положительно определенное и требуется корректировка (17) в виде

$$\mathbf{K} = \mathbf{W}^{-1}\mathbf{A}\mathbf{W}^{-1} + \sigma\mathbf{W}^{-1},$$

где σ — достаточно большое положительное число [10].

4. Численный эксперимент

На рис. 1 (см. третью сторону обложки) показан граф транспортной сети Омской области, построенный на основе данных проекта OpenStreetmap (*NL*,° — North Latitude — северная широта в градусах, *WL*,° — West Longitude — западная долгота в градусах) [14]. Исходный граф включает 140 000 узлов. Это число сокращается до 38 777 за счет исключения несущественной информации. В описании графа сохраняются узлы, соответствующие исходному пункту каждой дороги, пункту ее окончания и пересечениям с другими дорогами. Объем данных, представляющий эту разреженную матрицу смежности с 100 378 ненулевыми элементами, занимает на диске 3 Гбайт и 1,5 Гбайт в памяти компьютера.

Непосредственный подход к кластеризации узлов графа на рис. 1, описанный в разделе 1, состоит в использовании смеси Гауссовых распределений в качестве статистической модели для репрезентативной выборки узлов графа в пространстве координат и других параметров, характеризующих узлы графа. При использовании EM-алгоритма MDL-критерий определяет восемь кластеров (рис. 2, см. третью сторону обложки) для 1000 заданных узлов.

В общем случае при решении задачи разбиения некоторого графа на кластеры необходимо предварительно выявить скрытую структуру этого графа, которая отражена в его матрице смежности. Как показано в разделе 2, анализ перекрестных ассоциаций матрицы смежности позволяет оценить число кластеров в структуре графа. Такую оценку можно использовать в качестве входного параметра для алгоритмов кластеризации узлов, заданных на графе. Выделенная скрытая структура графа показана на рис. 3. Эта структура представлена блоками из строк и столбцов с различной плотностью нулей и единиц в матрице смежности. Результаты предварительного анализа графа транспортной сети Омской области (см. рис. 1) с помощью алгоритма перекрестных ассоциаций показывают, что в структуре графа на рис. 1 выделяется 15 кластеров, представленных на главной диагонали матрицы смежности этого графа.

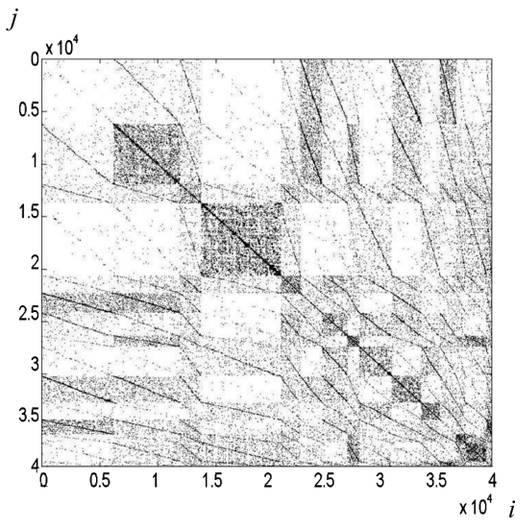


Рис. 3. Скрытая структура графа, представленного на рис. 1

На рис. 4, *а* (по оси абсцисс и ординат — группы узлов) показано абстрактное изображение структуры матрицы смежности, представленной на рис. 3. Насыщенность цвета прямоугольников на рис. 4, *а* зависит от плотности связей в каждом выделенном кластере. Чем темнее прямоугольник, тем больше плотность связей в кластере. Отметим, что полученное на этом этапе число кластеров отражает только общую структуру матрицы смежности. При последующем решении задачи разбиения узлов графа на кластеры следует учитывать только те элементы структуры матрицы смежности, плотность связей в которых выше некоторого среднего уровня. На рис. 4, *б* символами "·" показаны значения плотности связей ρ в выделенных 15 кластерах (*k* — индекс кластера). Как видно на рис. 4, *б*, только восемь из них имеют плотность связей $\rho > \rho_{\text{среднее}} = 3{,}5631 \cdot 10^{-4}$ (показана сплошной линией). Это задает начальное число кластеров $K_0 = 8$ при кластеризации узлов графов на основе алгоритмов спектральной кластеризации (число кластеров на рис. 2). Как показано в разделе 3, при заданном начальном числе кластеров программный пакет Graclus позволяет быстро разбить граф на рис. 1 на кластеры. Результаты кластеризации графа на рис. 1, полученные с использованием программного пакета Graclus, показаны на рис. 5 (см. третью сторону обложки) (оптимальное число кластеров $K = 4$) и рис. 6 (см. третью сторону обложки) (число кластеров $K = 2$).

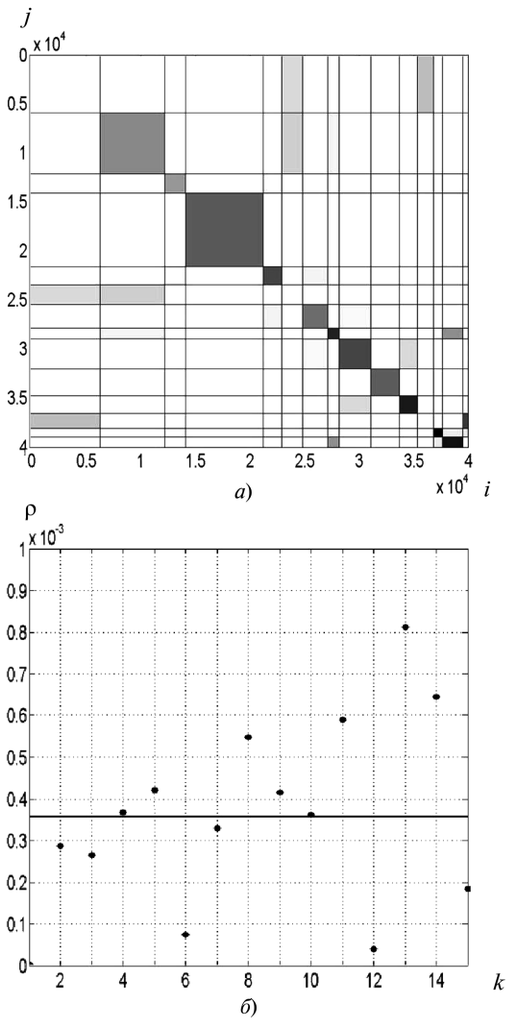


Рис. 4. Оценка числа кластеров в структуре графа, представленного на рис. 1

Заключение

Как показано в настоящей работе, MDL-критерий может успешно применяться для кластеризации заданных узлов на графах транспортных сетей с разреженными матрицами смежности. Алгоритмы, использующие MDL-критерий, дают предварительную оценку числа элементов в скрытой структуре графа, которая может использоваться для задания начальных условий при кластеризации узлов графа и при разбиении существующего графа транспортной сети на подграфы. Рассматриваемые задачи кластеризации актуальны для приложений логистики.

Список литературы

1. **Akaike H.** A new look at the statistical model identification // IEEE Transactions on Automatic Control. 1974. V. AC-19. P. 716—723.

2. **Grünwald P. D.** A tutorial introduction to the minimum description length principle / Grünwald P. D., Myung I. J., Pitt M., eds. Advances in Minimum Description Length: Theory and Applications. Cambridge, MA: MIT Press. 2005.

3. **Grünwald P. D.** The Minimum Description Length Principle. Adaptive Computation and Machine Learning series. Cambridge, MA: MIT Press. 2007.

4. **Rissanen J.** A universal prior for integers and estimation by minimum description length // The Annals of Statistics. September 1983. V. 11. N 2. P. 417—431.

5. **Rissanen J.** Information and Complexity in Statistical Modeling. New York: Springer. 2007. P. 97—103.

6. **Bouman C. A.** CLUSTER: An Unsupervised Algorithm for Modeling Gaussian Mixtures. Purdue University, West Lafayette: School of Electrical and Computer Engineering. 2005. 20 p.

7. **Dempster A., Laird N. M., Rubin D. B.** Maximum likelihood from incomplete data via the EM algorithm // Journal of the Royal Statistical Society B. 1977. V. 39. N 1. P. 1—38.

8. **Кухаренко Б. Г.** Анализ независимых компонент и скрытая Марковская модель для определения доминантных компонент многомерных временных рядов // Информационные технологии. 2010. № 11. Приложение. С. 1—32.

9. **Chakrabarti D., Papadimitriou S., Modha D., Faloutsos C.** Fully automatic cross-associations / Kim W., Kohavi R., Gehrke J., DuMouchel W., eds. // Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004). Seattle, WA: ACM. 2004. P. 79—88.

10. **Dhillon I. S., Guan Y., Kulis B.** Weighted Graph Cuts without eigenvectors: A multilevel approach // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007. V. 29. N 11. P. 1944—1957.

11. **MacQueen J.** Some methods for classification and analysis of multivariate observations // Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Los Angeles: University of California Press. 1967. V. 1. P. 281—296.

12. **Dhillon I. S., Guan Y., Kulis B.** A fast kernel-based multilevel algorithm for graph clustering // Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2005). Chicago, IL: ACM. August 21—24. 2005. P. 629—634.

13. **Dhillon I. S., Guan Y., Kulis B.** Kernel k-means, spectral clustering and Normalized Cuts // Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). Seattle, WA: ACM. August 22—25. 2004. P. 551—556.

14. **Ramm F., Topf J., Chilton S.** OpenStreetMap: Using and Enhancing the Free Map of the World. Cambridge, United Kingdom: UIT Cambridge Ltd. 2010. 386 p.

УДК 004.75

А. А. Петров, аспирант,
e-mail: gtmaster00@gmail.com,
В. Т. Калайда, д-р техн. наук, проф.,
e-mail: kvt@iao.ru,
Национальный исследовательский
Томский государственный университет

Платформа для создания единой вычислительной среды в локальной сети*

Предлагается программная платформа, которая обеспечивает создание распределенных приложений и осуществляет автоматическое управление вычислительным процессом.

Ключевые слова: *распределенная система, система управления, программный комплекс, сервис-ориентированная архитектура*

Введение

В настоящее время задача интеграции вычислительных и информационных ресурсов в единую среду и организация эффективного доступа к ним является одной из основных в развитии современных информационных технологий. Связано это в первую очередь с бурным развитием сетевых технологий и значительным увеличением скорости и надежности передачи данных в сетях. Однако на сегодняшний день компьютеры, объединенные какой-либо сетью, в основном используются как источники информации, а не как источники вычислительных ресурсов. С этим связано возникновение и развитие Grid-технологии и технологии "Облако" [1].

Отдельное место в проблеме интеграции вычислительных ресурсов занимает вопрос об интеграции ресурсов отдельной локальной корпоративной или кампусной сети, так как локальные сети имеют следующие особенности:

- высокая скорость передачи данных по сравнению с глобальными сетями;
- высокая надежность передачи данных по сравнению с глобальными сетями;
- низкие задержки при передаче данных.

Существует множество методов и технологий, позволяющих распределить вычисления по узлам локальной сети, от самых простых (сокеты Беркли [2], удаленный вызов процедур [3]) до высокоразвитых и сложных (Java Enterprise Edition [4], Windows Communication Foundation [5]). Использование данных технологий позволяет разработчику приложения использовать вычислительные ресурсы нескольких компьютеров для решения тех или иных задач.

Однако при использовании данных решений все вопросы по управлению вычислительным процессом — такие как приоритеты операций, выбор узла для вычисления, отслеживание ошибок и другие, ложатся на плечи программиста. Таким образом, увеличивается трудоемкость разработки программного обеспечения и, как следствие, длительность и стоимость разработки.

Модель единой вычислительной среды

Решить указанные проблемы позволит единая платформа, объединяющая все узлы локальной сети в единую вычислительную среду, и предоставляющая вычислительные ресурсы этой среды пользовательским приложениям. Анализ задач, решаемых в корпоративных и производственных сетях, показывает, что эта платформа должна удовлетворять следующему набору требований, продиктованному прежде всего экономичностью решения и простотой его использования:

- не использовать дополнительного дорогостоящего коммутационного оборудования;
- быть достаточно простой в настройке и эксплуатации, не требовать для постоянного обслуживания высококвалифицированных специалистов;
- быть "прозрачной" для конечного пользователя;
- быть "самонастраиваемой", т. е. уметь поддерживать себя в активном состоянии в любых ситуациях (за исключением форс-мажорных) без вмешательства человека;
- поддерживать выполнение задач широкого класса;
- быть надежно защищенной от вторжения извне;
- быть кроссплатформенной и работать в сети, узлы которой управляются различными операционными системами.

Естественно, что эти требования зачастую противоречивы, но именно максимальное удовлетворение им позволит максимально использовать ресурсы имеющихся компьютеров для решения про-

* Работа выполнена при финансовой поддержке Минобрнауки, контракт № 14.515.11.0032.

Рис. 1. Модель единой вычислительной среды

изводственных, технологических и офисных задач организации.

Модель вышеописанной платформы представлена на рис. 1.

Платформа, объединяя вычислительные ресурсы всех узлов локальной сети, создает единую вычислительную среду, в которую помещает вычислительные модули. Каждый модуль реализует отдельный алгоритм программного комплекса (например, модуль вычисления обратной матрицы, модуль нахождения градиента изображения и т. д.). Пользовательские приложения через единый интерфейс отправляют запросы на вычисления тем или иным модулем и получают ответ.

Вычислительная среда в автоматическом режиме планирует процесс выполнения вычислительной задачи таким образом, чтобы процесс максимально соответствовал установленному критерию оптимальности. Примерами таких критериев могут быть минимальное время выполнения задачи, поддержка равномерной загрузки всех узлов сети, минимизация информационного обмена между узлами и т. п. Критерий оптимальности задается пользователем платформы. Планирование происходит на основе данных о параметрах сети, статистической информации, текущей загрузки каждого компьютера и каналов передачи данных. Механизм управления вычислительной средой следит за выполнением задачи: собирает статистическую информацию о времени выполнения той или иной задачи, изменяет план вычислений в критических ситуациях, когда выполнение текущего плана невозможно или нецелесообразно в связи с новыми условиями. Такими условиями могут быть отказ канала передачи данных, резкое повышение загруженности одного узла из-за какой-то задачи и т. п.

Платформа снимает с разработчиков приложений необходимость решать вопросы планирования и управления вычислительным процессом. Создателю приложений достаточно выделить "тяжелые" вычислительные функции программы в отдельные вычислительные модули, заменить вызовы этих функций на обращение к единому интерфейсу платформы и интегрировать эти модули в платформу на этапе установки приложения.

Таким образом, разработчик получает возможность создать распределенное в локальной сети приложение, не увеличивая сроков и стоимости разработки. В свою очередь, конечный пользователь при использовании платформы и распределенных приложений получает увеличение эффективности использования вычислительного оборудования и уменьшение времени выполнения вычислительных задач.

Реализация модели

На основе вышеописанной модели была создана реализация платформы, которая получила название DistributedSystem.

Общая структура системы представлена на рис. 2.

На каждом компьютере вычислительной системы имеется набор модулей, реализующих определенные алгоритмы программного комплекса. Набор модулей на каждом из узлов сети может отличаться.

Система управления распределенными вычислениями включает набор диспетчеров на каждом компьютере сети, выполняющий планирование вычислительного процесса, осуществляющий сборку информации о параметрах и состоянии компьютера, ведущий сбор статистической информации и реагирующий на сообщения от диспетчеров других компьютеров сети.

Таким образом, набор диспетчеров и представляет собой вычислительную среду, описанную выше. К каждому диспетчеру подключены модули, диспетчер предоставляет интерфейс для запроса о выполнении вычислительной задачи пользовательским программам (рис. 3).

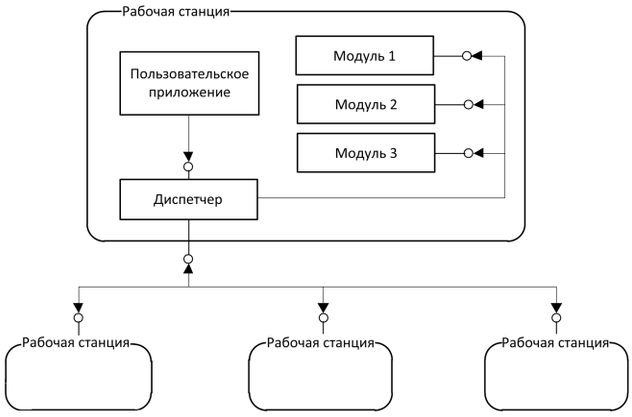


Рис. 2. Общая структура системы

Основной технологией, используемой при создании платформы, является Windows Communication Foundation. Windows Communication Foundation (WCF) — программный фреймворк, используемый для обмена данными между приложениями, входящими в состав .NET Framework [5]. Использование данной технологии позволило создать платформу, основанную на сервис-ориентированной архитектуре. Такая архитектура не требует дополнительного оборудования и значительно увеличивает отказоустойчивость платформы. Кроме того, сервис-ориентированная архитектура позволяет узлам локальной сети входить в вычислительную среду и выходить из нее "на лету", не влияя на работоспособность среды в целом [6].

Одной из главных особенностей технологии является возможность запуска сервиса в простом приложении, без использования специального сервера или контейнера приложений. Данное преимущество позволяет создавать простые в установке и настройке приложения, так как исключает необходимость установки и настройки отдельного сервера. Эта особенность в значительной степени повлияла на выбор технологии для создания платформы.

Между собой диспетчеры обмениваются сообщениями по протоколу SOAP. Это протокол обмена структурированными XML-сообщениями в распределенной вычислительной среде [7]. Использование данного протокола позволит быть системе платформонезависимой и работать в сети, узлы которой управляются разными операционными системами. Основными видами сообщений между диспетчерами являются следующие:

- Calc — используется для запроса на подсчет;
- Hello — используется при появлении нового диспетчера в вычислительной среде. При этом диспетчер получает параметры другого диспетчера (набор модулей, конфигурацию компьютера и т. д.), отправляя в ответ свои параметры;
- State — используется для запроса другого диспетчера о текущем состоянии компьютера (загруженность процессора и т. д.).

Запрос пользовательского приложения на выполнение задачи вычислительным модулем также проводится по протоколу SOAP. Для удобства формирования данного SOAP-вызова с платформой поставляется программный интерфейс (API), который разработчики могут использовать при создании программ.

Модуль представляет собой библиотеку классов .NET. Основное требование — один и только один из классов должен содержать функцию DoJob, которую и вызывает диспетчер при использовании модуля для подсчета. При установке пользовательское приложение сообщает диспетчеру компьютера об устанавливаемых модулях приложения. Диспетчер автоматически помещает указанные модули в вычислительную среду, передавая их другим диспетчерам и таким образом распределяя их по узлам сети.

Пример функционирования платформы представлен на рис. 4.

Цифрами на рисунке обозначены основные операции при выполнении подсчета:

1. Пользовательское приложение отправляет *запрос* программного интерфейса о необходимости совершения подсчета определенным вычислительным модулем.

2. Программный интерфейс формирует SOAP-*запрос* о подсчете и отправляет его диспетчеру узла, на котором выполняется пользовательское приложение.

3. Диспетчер *определяет* в соответствии с выбранным критерием оптимальности, на каком узле будет выполняться подсчет, и *отправляет* сообщение диспетчеру этого узла.

4. Диспетчер удаленного узла *распаковывает* данные из SOAP-сообщения и *вызывает* модуль для совершения подсчета.

5. Совершив подсчет, модуль *возвращает* результат диспетчеру.

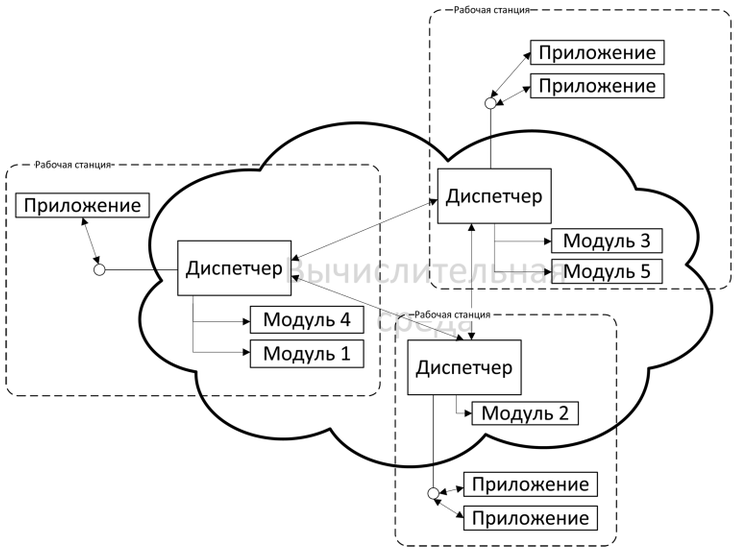


Рис. 3. Реализация вычислительной среды

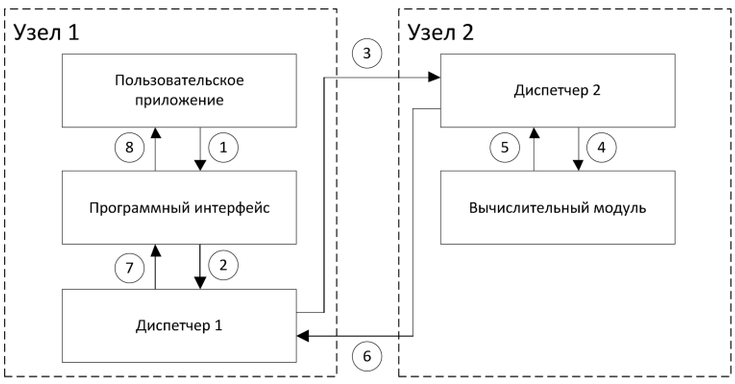


Рис. 4. Пример функционирования платформы

6. Диспетчер удаленного узла формирует SOAP-*ответ* и отправляет его вызвавшему диспетчеру

7. Диспетчер, получив ответ, *отправляет* его программному интерфейсу.

8. Программный интерфейс *распаковывает* ответ и отдает его пользовательскому приложению.

Заключение

Тестирование платформы показало, что ее использование позволяет значительно (до 30 %) уменьшить время выполнения вычислительных задач при работе с большим объемом данных. Также стоит отметить эффективность использования вычислительного оборудования. Распределение программы с помощью вычислительной среды позволило снизить нагрузку на каждый отдельный вычислительный узел.

Таким образом, вышеописанная платформа решает основные проблемы, связанные с созданием и использованием приложений, распределенных в локальной сети. Единая вычислительная среда служит механизмом эффективного управления распределенными вычислениями и позволяет избежать проблем, связанных с конфликтами нескольких не связанных между собой распределенных приложений. Использование платформы значительно упрощает создание распределенных приложений и не требует от разработчика знаний о механизмах распределения и способах управления вычислительными процессами.

Работа платформы практически незаметна для конечного пользователя. Поиск узлов локальной сети, использование узлов и планирование вычислений происходит в автоматическом режиме.

Список литературы

1. **Jefferey K., Neidecker-Lutz B.** The future of cloud computing // Cloud Computing Expert Group Report. 2009. [Электронный ресурс]. URL: http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf, свободный (дата обращения: 15.07.2012).

2. **Таненбаум Э.** Компьютерные сети. 4-е изд. СПб.: Питер, 2003. 992 с.

3. **Олифер Н. А., Олифер В. Г.** Сетевые операционные системы. 2-е изд. СПб.: Питер, 2009. 672 с.

4. **Браун К., Крейг Г., Хестер Г.** Создание корпоративных Java-приложений для IBM WebSphere. М.: Кудиц-образ, 2005. 860 с.

5. **Резник С., Крейн Р., Боуэн К.** Основы Windows Communication Foundation для .NET Framework 3.5. М.: ДМК пресс, 2008. 480 с.

6. **Matthew MacKenzie C.** et al. Reference Model for Service Oriented Architecture 1.0. OASIS Open. 2006.

7. **Ньюкомер Э.** Веб-сервисы. XML, WSDL, SOAP и UDDI СПб.: Питер, 2003. 256 с.

УДК 519.7:004.415.2

Б. А. Соловьев, канд. техн. наук, программист,
e-mail: sol@iao.ru,
ООО "НПП "Стелс", г. Томск,
В. Т. Калайда, д-р техн. наук, проф.,
e-mail: kvt@iao.ru,
Национальный исследовательский
Томский государственный университет

Технология проектирования, создания и администрирования распределенных вычислительных систем, основанная на модели компонентных объектов*

Предлагается программная платформа, которая обеспечивает проектирование, создание из готовых блоков (прикладных объектов) и администрирование распределенных вычислительных систем на основе взаимодействия компонентов.

Ключевые слова: *распределенная система, прикладной объект, шина объектов*

* Работа выполнена при финансовой поддержке Минобрнауки, контракт № 14.515.11.0032.

Введение

При проведении экспериментальных исследований на физико-технических установках зачастую возникает потребность в изменении структуры и прикладных функций программной системы регистрации и управления. Как следствие, продолжительность экспериментальных исследований физического явления существенно увеличивается, в том числе из-за перестройки и повторной сборки программной части установки.

Частично задача быстрой перестройки программной системы решается за счет использования модульной архитектуры с использованием технологий позднего связывания кода. В этом случае ядро программной системы позволяет подключать и отключать (в том числе не прерывая работы всей системы) отдельные модули, реализующие заранее определенные интерфейсы позднего связывания. Это позволяет строить более гибкие программные системы, но степень их гибкости зависит от того, насколько общими являются разработанные программистом интерфейсы и насколько широкий спектр возможных задач был предусмотрен при проектировании. При возникновении условий, выходящих за рамки разработанных интерфейсов, процесс внесения изменений в систему характеризуется теми же недостатками, что и в случае с монолитными системами [1].

Подобные проблемы возникают при проектировании и эксплуатации информационных и управляющих систем, строящихся по иерархической модели:

- датчики и исполнительные механизмы;
- программируемые логические контроллеры (ПЛК), соединенные полевой шиной;
- системы, основанные на компьютерах, собирающие и сохраняющие данные и события с контроллеров;
- рабочие места операторов и диспетчеров [2, 3].

Системы на ПЛК обладают низкой степенью гибкости [4]. Основной причиной широкого применения контроллеров в большинстве случаев является разница в цене контроллера и компьютера. Вместе с тем, для больших систем, содержащих 10 и более контроллеров, замена их одним компьютером и реализация функций контроллеров на нем обойдутся дешевле, отладка проще, процесс разработки и внедрения короче.

Другим недостатком построения систем с "аппаратной решающей частью" является необходимость прокладки полевой шины для контроллеров, задействованных в процессе. Такие сети обладают низкой пропускной способностью, сетевое оборудование сложно согласуется даже при использовании оборудования одного производителя. Кроме того, большинство протоколов такой сети позволяют одновременно обмениваться информацией только паре участников.

В прикладных системах целесообразней использовать сети Ethernet, где почти все проблемы согласования оборудования решены и не возникает проблем при использовании оборудования разных производителей.

Обработка потока данных

В случаях, когда требуется быстрая обработка больших массивов данных, результаты которой могут повлиять на различные участки системы, необходимость в использовании компьютеров и скоростных сетях не вызывает сомнений. Большинство программных систем построены на модели предоставления сервисов. Для обработки данных необходимо сделать запрос к одному серверу, чтобы их получить, затем отправить запрос на обработку другому серверу и на базе полученных данных выработать управляющие запросы для другого сервера. Для систем, конфигурация которых может изменяться в процессе эксплуатации, более приемлема модель, основанная на взаимодействии процессов. Процессы (составные части системы — модули, программы и др.) можно разделить на процессы, являющиеся источниками каких-либо сигналов (объекты-генераторы), процессы, которые изменяют свои свойства под действием этих сигналов (объекты-приемники), и процессы, которые, реагируя на эти сигналы, порождают новые (комплексные объекты). В такой модели преобладают связи источник—приемник с многочисленными обратными связями (связь клиент—сервер — частный случай такой модели).

В настоящее время существует большое число программных систем, построенных по такой модели:

- системы обработки потокового аудио и видео [5];
- системы моделирования;
- системы автоматизации эксперимента [6, 7];
- системы визуального программирования [8, 9].

Первые предназначены для решения небольших бытовых задач в течение короткого времени в рамках одного программного процесса на одном компьютере и не отличаются стабильностью.

Вторая категория программных продуктов ориентирована на итерационную обработку больших массивов данных по созданным моделям и требуют больших вычислительных мощностей. Выполняются, как правило, в кластере или на суперкомпьютере.

Последняя категория не получила распространения из-за неудобств, связанных с необходимостью использования большой рабочей площади для визуального представления программы, и небольшой, в отличие от текстовых языков программирования, гибкостью.

Среди них необходимо отметить среду визуального программирования, получившую широкое распространение среди физиков и инженеров-электронщиков — LabView. В этой системе хорошо сочетаются возможности визуального конструирования программы из готовых блоков и программирования в различных "традиционных" средах. Компонент для LabView может быть написан практически в любой современной среде программирования, большое число выпускаемых сейчас устройств содержат в поставке необходимые компоненты. К недостаткам этой среды стоит отнести только ее громоздкость и сложность поставки "готовой программы" конечному пользователю. Для настройки такой системы конечный пользователь должен хорошо знать LabView [9].

"Базис"

Целью создания "Базиса" была платформа, обеспечивающая построение распределенных систем управления из готовых блоков (прикладных объектов), каждый из которых выполняет одну функцию в рамках предметной области. Объекты имеют набор входов и выходов для обработки и генерации входных и выходных потоков данных. Для разработчика, создающего систему из готовых блоков, она должна выглядеть как граф прикладных объектов, соединенных дугами информационных потоков в соответствии с конкретной предметной областью [10]. Пользователи "Базиса" делятся на три категории:

- программисты, разрабатывающие прикладные объекты;

- администраторы, создающие и настраивающие систему управления из имеющихся прикладных объектов;
- конечные пользователи, которые используют созданную систему и имеют дело только с пользовательскими интерфейсами прикладных объектов.

Для организации общей информационной среды и управления временем жизни прикладных объектов на каждом компьютере, задействованном в процессе, располагается служебный объект "Базиса", называемый "шиной объектов", реализуемый как сервис Windows. Шина создает и уничтожает прикладные объекты, управляет их входами и выходами и организует доставку данных от одного объекта к другому, в том числе посредством другой шины, если взаимодействующие объекты расположены на разных компьютерах. На рис. 1 показаны отношения между объектами "Базиса".

Так как шина объектов реализуется в виде сервиса Windows, это позволяет ей функционировать даже при отсутствии зарегистрированного пользователя. При запуске шина считывает свою конфигурацию из общей базы данных, создает объекты, при необходимости подключает к другим шинам системы для передачи потоков данных подконтрольных прикладных объектов и уведомляет другие шины объектов о своем запуске. Получив уведомление о запуске, другие шины, имеющие связанные с ее объектами генераторы, также создают с ней подключения для передачи "своих" потоков данных [11].

Прикладной объект, его входы и выходы имеют имена для удобства администрирования и проектирования системы. Имя прикладного объекта состоит из имени подсистемы, в которую он входит, и собственного имени, несущего информацию о его функциях в системе, разделенных точкой. Подсистемы могут быть частью системы более высокого уровня, в этом случае ее имя также состоит из имени "старшей" системы и собственного имени, разделенных точкой. Так, прикладной объект, отвечающий за видео-захват, расположенный в помещении 314, может иметь имя "Видеонаблюдение.Блок А.3 этаж.314.Входная дверь.Камера". Имена входов и выходов объекта задаются на этапе разработки объекта и несут информацию о типе и назначении потока данных. Например, вход или выход объекта, работающего с видео, может называться "MJPEG-видео" [12].

Прикладной объект может быть выполнен в виде внутрипроцессного COM-сервера (в этом случае шина взаимодействует с ним через вызовы его COM-интерфейса) или в виде обычного исполняемого модуля Windows, использующего библиотеку позднего связывания, организующую взаимодействие с шиной по протоколу BRMI (Basis Remote Method Invocation) над протоколом TCP. При реализации объекта с использованием первого варианта временем его жизни управляет шина, и он может функционировать даже при отсутствии "входа пользователя в систему". При использовании второго варианта пользователь должен запускать прикладной объект в виде приложения самостоятельно. После запуска такое приложение сообщает своей шине о том, что собирается работать от имени прикладного объекта, и шина проводит его инициализацию вызовом BRMI-методов. Для остановки объекта пользователь также должен остановить его, используя пользовательский интерфейс приложения или диспетчер задач.

Первый вариант построения прикладного объекта больше подходит к объектам, выполняющим вычислительные задачи, пользовательский интерфейс которых используется редко, а требования к производительности жестче. Второй вариант

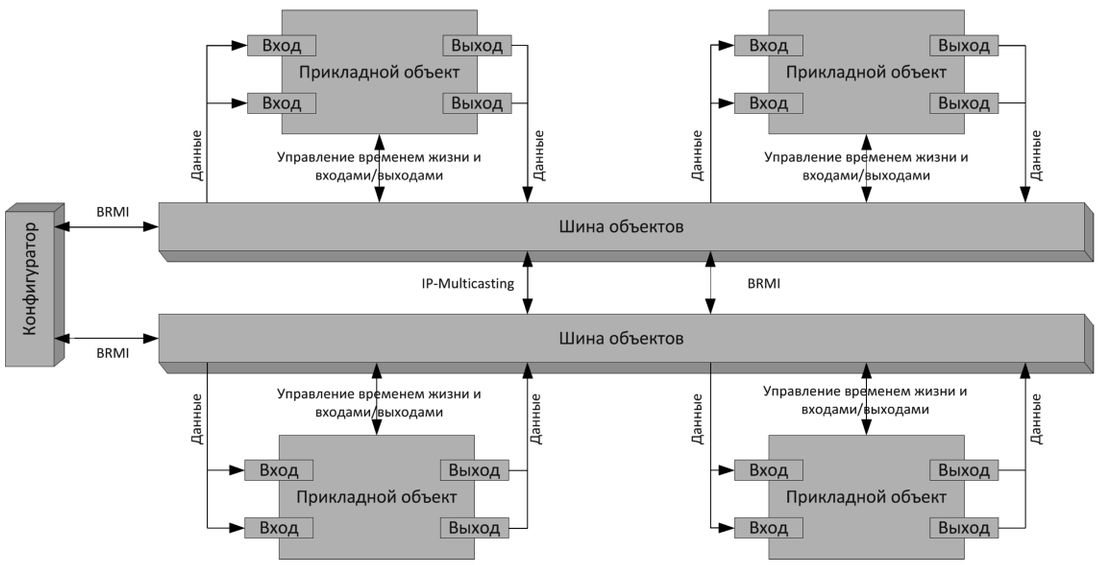


Рис. 1. Объекты "Базиса"

больше подходит для объектов, взаимодействующих с пользователем и для которых внешний вид и удобство играют бóльшую роль, чем скорость обработки и генерации данных.

При запуске и последующей работе прикладного объекта шина проверяет необходимость подключения его входов и выходов (наличие и работу соответствующих объектов-генераторов и объектов-приемников) и по мере поступления данных передает их на соответствующие входы. При генерации данных объект-генератор уведомляет шину о наличии новых данных на выходе. Генерация данных может происходить постоянно и "по требованию". В первом случае объект-генератор поставляет данные на свой выход с заданной интенсивностью, независимо от требований всех его приемников. Во втором случае генератор начинает генерацию необходимого числа пакетов выходных данных только по запросу хотя бы от одного приемника, о необходимом числе пакетов ему сообщает шина объектов.

Каждый выход объекта имеет следующие параметры:

- интенсивность потока данных — указывает на среднее число пакетов данных в секунду. Этот параметр выбирается как максимальная интенсивность, необходимая потребителям. Для потребителей, чьи требования к интенсивности потока ниже предоставляемой, вводится ограничение интенсивности с отбрасыванием "лишних" пакетов данных;
- число пакетов данных, которые необходимо сгенерировать ("замóк" выхода), — предназначен для реализации механизма генерации "по запросу". Если значение "замкá" входа равно 0xffff, данные генерируются постоянно с заданной интенсивностью. Другое значение указывает число пакетов, после которого необходимо остановиться. При генерации каждого последующего пакета значение "замкá" уменьшается на единицу, при поступлении нового запроса его значение принимает максимальное из запрошенного и текущего;
- максимальный размер одного пакета данных;
- максимальное число пакетов данных, одновременно находящихся в выходной очереди.

Последние два параметра служат для оптимизации нагрузки на компьютер. В случае, когда взаимодействующие приемники и генератор находятся на одном компьютере, выходная очередь генератора является входной очередью для всех приемников для исключения "лишнего" копирования данных.

В процессе работы приемник может запросить у своей шины объектов понижение или повышение интенсивности входного потока. В этом случае шина приемника инициирует процесс подстройки интенсивности заданного выхода. Для этого она сообщает генератору (или шине генератора — в случае удаленного взаимодействия) о такой необходимости. Генератор проверяет возможность такого изменения интенсивности (для этого может также понадобиться изменение интенсивности одного из входных потоков самого генератора) и либо принимает новые требования, либо сообщает об их невозможности, завершая процесс подстройки интенсивностей. В результате подстройки интенсивностей приемники продолжат работу на полученной интенсивности, либо для их входов будет введено ограничение интенсивности с отбрасыванием пакетов.

Входы объекта имеют аналогичный набор параметров. При более высокой интенсивности входного потока, чем указано в параметрах входа, происходит отбрасывание пакетов, время прихода которых меньше расчетного.

При установке значения "замкá" входа генератору будет сообщено о необходимости сгенерировать новую порцию данных, а при поступлении указанного числа данных "замóк" входа будет закрыт, и данные на вход перестанут поступать, даже если они все еще генерируются.

В процессе работы приемники и генераторы могут останавливаться и запускаться снова. Для оптимизации трафика шины оценивают необходимость подключения потока и при отсутствии активных приемников отключают выход соответствующего генератора. Шина генератора хранит таблицу подключенных локальных и удаленных приемников и осуществляет доставку данных в указанное место указанным способом.

Доставка данных между объектами осуществляется тремя способами:

- *планируемые в память файлы.* Используется, когда и приемник, и генератор расположены на одном компьютере. В этом случае объекты планируют каждый в свое адресное пространство файл, содержащий очередь выходных данных генератора. Генератор открывает файл с правами на запись, а приемник — только на чтение;
- *IP-Multicasting.* Используется для удаленной доставки потоков с высокой интенсивностью и возможностью потери данных для уменьшения нагрузки на сеть;
- *BRMI.* Используется для гарантированной доставки данных каждой шине. Так как он опирается на протокол TCP, для каждой шины отправляется своя копия данных, что увеличивает нагрузку на сеть.

За счет того, что приемники и генераторы никогда не взаимодействуют между собой напрямую, обеспечивается возможность создания схем взаимодействия объектов, содержащих обратные связи любой сложности. Алгоритм подстройки интенсивностей также содержит механизмы, исключающие "зависание" сеанса подстройки.

Для развертывания и управления системой под управлением "Базиса" служит приложение "Конфигуратор", внешний вид которого показан на рис. 2.

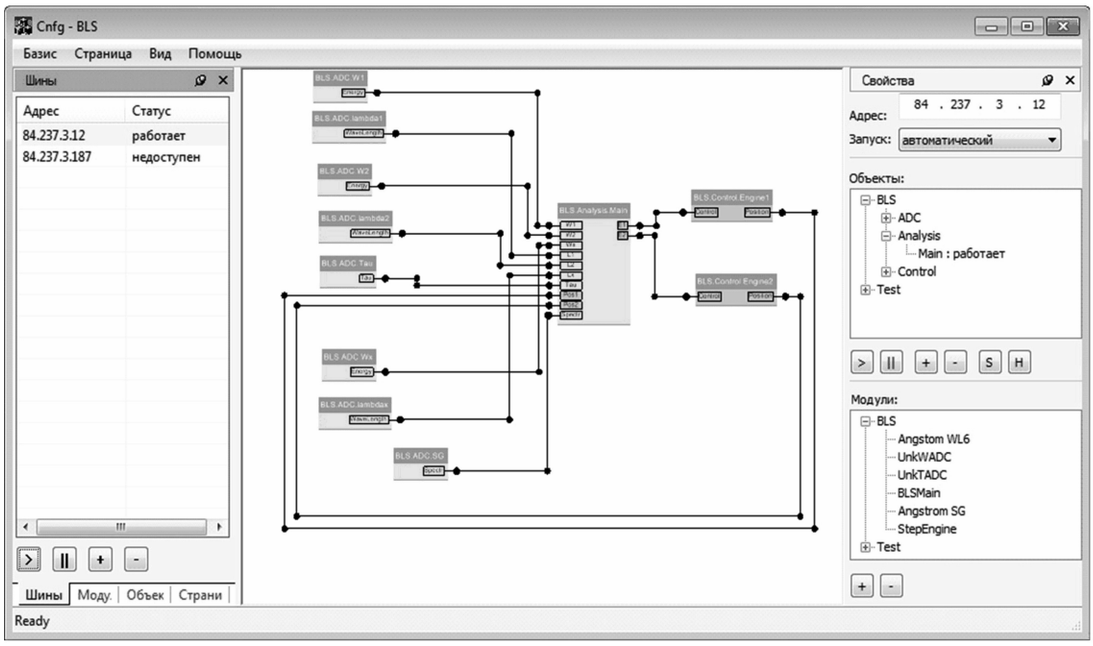


Рис. 2. Конфигуратор "Базиса"

Интерфейсы объектов. Для взаимодействия шины с COM-объектами они должны предоставлять интерфейс IBasisCOMObject50, который содержит следующий набор методов:

- Start — инициализация и запуск объекта, при вызове объекту передается его имя в качестве параметра;
- Stop — остановка и выгрузка объекта;
- InputConfig — конфигурация входа объекта, в качестве параметров передаются характеристики входа, его имя и интерфейс обратного вызова для запроса подстройки интенсивностей и управления "замкóм" IBasisCOMInputCallback50, объект создает экземпляр класса-выхода и возвращает шине указатель на интерфейс IBasisCOMInput50;
- OutputConfig — инициализация выхода объекта, на вход передаются параметры выхода, имя, интерфейс обратного вызова IBasisCOMOutputCallback50, на выход — интерфейс управления выходом IBasisCOMOutput50;
- ShowView — вызов пользовательского интерфейса объекта;
- HideView — закрытие пользовательского интерфейса.

При запуске объекта шина вызывает метод Start. По переданному имени объект определяет свои настройки и инициализируется. Затем шина последовательно вызывает методы настройки входов и выходов объекта, передавая в объект интерфейсы обратного вызова для управления интенсивностью потоков и реализации способа генерации данных "по требованию". Объект предоставляет шине указатели на интерфейсы управления своими входами и выходами.

Интерфейс управления входом объекта имеет определения следующих методов:

- Connect — подключение входа;
- Pause — приостановка работы входа, но не уничтожение объектов, связанных с ним, объект должен быть готов быстро восстановить работу входа;
- Disconnect — отключение входа;
- SetInterval — уведомление об изменении интенсивности входного потока в результате завершения процесса подстройки интенсивностей или изменения значения пользователем;
- SetLock — уведомление об изменении значения "замкá" пользователем;
- Receive — передача новых данных на вход.

Интерфейс управления выходом содержит те же определения, за исключением метода Receive. Метод SetLock сообщает объекту о том, что инициирован процесс подстройки интенсивностей, и ему необходимо подстроиться под новые требования (возможно, передав новый запрос на перенастройку одного из своих входов).

Дублирование методов интерфейса входа и выхода сделано для уменьшения числа запросов на получение интерфейса.

Интерфейс обратного вызова входа объекта содержит следующие методы:

- SetInterval — запрос другой интенсивности входного потока, на вход подается значение желаемого интервала времени между последовательными пакетами;
- SetLock — установка нового значения "замкá" входа.

Интерфейс обратного вызова выхода объекта содержит определение для единственного метода

Receive, который сообщает шине о том, что сгенерирован новый пакет данных.

В случае работы объекта с шиной через протокол BRMI нет возможности передать указатель на интерфейс, поэтому BRMI-интерфейсы содержат те же определения методов, но вместо указателей на интерфейсы используются идентификаторы входов и выходов, уникальные в рамках прикладного объекта. При запуске объекта-приложения, объект сам сообщает шине свое имя при установке соединения с шиной.

Интерфейсы шины. Шина, как сервис, работает с тремя категориями клиентских приложений:

- конфигуратор;
- другие шины;
- прикладные объекты.

При подключении к шине клиент получает доступ к главному (Common) интерфейсу шины, который содержит определения для трех функций. Вызов каждой из функций сообщает шине о типе клиента и переключает главный интерфейс на интерфейс, соответствующий категории клиента:

- Common_InitObject — клиент является прикладным объектом, и работа будет происходить через интерфейс Object;
- Common_InitClient — сообщает о том, что клиент является конфигуратором и дальнейшая работа будет происходить с использованием интерфейса Client;
- Common_InitBus — клиент является шиной и рабочий интерфейс — Bus.

Интерфейс Object реализует методы обратного вызова входов и выходов, которые были описаны выше.

Интерфейс Client содержит большое число методов для управления и конфигурирования системы и передачи уведомлений о процессах в шинах.

Интерфейс Bus предназначен для синхронизации работы подсистем передачи данных шин, имеющих связанные объекты, такие как уведомления о состоянии шины, объектов и их входов и выходов, передачи запросов на изменение "замкóв" и интенсивностей и содержит большое число методов.

Заключение

Построенная с использованием "Базиса" система предоставляет пользователю или администратору широкие возможности по переконфигурированию и распределению задач системы, не прибегая к помощи разработчика. При расширении функций системы достаточно установить на новые компьютеры шину объектов и модули, реализующие новую функциональность. При подключении новых функций остальные части системы могут продолжать работать.

При выходе из строя части системы, выключении питания и потере сети остальные подсистемы продолжают работать в штатном режиме и все связи автоматически восстанавливаются после запуска шин, пострадавших от сбоя.

Возможность организации обратных связей любой сложности между объектами предоставляет возможность использования "Базиса" в моделировании.

В новой версии отказались от использования DCOM в силу того, что многие администраторы, считая, что "DCOM — это уязвимость Windows", запрещают его использование в своих сетях. Это потребовало создания новых механизмов и реализации протокола BRMI, позволяющих существенно упростить разработку прикладных объектов на языках, не имеющих возможности использовать custom-интерфейсы COM (например, объекты, написанные на Delphi, или объекты с богатым пользовательским интерфейсом, созданные под WPF).

Использование BRMI позволяет использовать "Базис" не только в локальных вычислительных сетях, но и в Internet. Однако для такого использования в протокол BRMI необходимо добавить криптографическую защиту, а в "Базис" — контроль доступа.

Разработка и испытание "Базиса" применительно к системам идентификации объектов и видеонаблюдения проводится в Группе обработки и анализа изображений (ГОАИ) Института оптики атмосферы (ИОА СО РАН). В настоящее время на основе "Базиса" реализуются системы контроля и управления экспериментальными установками "Поляризационный лидар" в Национальном исследовательском Томском государственном университете и "Бигармонический лазерный спектрометр" в ИОА СО РАН [13].

Список литературы

1. **Цимбал А. А., Аншина М. Л.** Технологии создания распределенных систем. Для профессионалов. СПб.: Питер, 2003. 576 с.
2. **Нестеров А. Л.** Проектирование АСУТП: Методическое пособие. Кн. 1. СПб.: ДЕАН, 2006. 552 с.
3. **Пескова С. А., Гуров А. И., Кузин А. В.** Центральные и периферийные устройства электронных вычислительных средств. М.: Радио и связь, 1999. 344 с.
4. **Митигин Г. П., Хазанова О. В.** Системы автоматизации с использованием программируемых логических контроллеров. М.: Изд. центр ГОУ МГТУ "СТАНКИН", 2005. 136 с.
5. **DirectShow** [Электронный ресурс]. URL: http://msdn.microsoft.com/en-us/library/windows/desktop/dd375454 (v = vs.85).aspx
6. **Автоматизация** эксперимента: Энциклопедия физики и техники [Электронный ресурс]. URL: http://www.femto.com.ua/articles/part_1/0027.html
7. **Автоматизация** экспериментов с применением методов цифровой обработки сигналов на примере системы определения пространственных координат источника вибрации [Электронный ресурс]. URL: http://www.zetms.ru/support/articles/seismo/dsp.php
8. **Microsoft** Robotics Developer Studio 4 [Электронный ресурс]. URL: http://www.microsoft.com/robotics/
9. **LabVIEW** System Design Software [Электронный ресурс]. URL: http://www.ni.com/labview/
10. **Соловьев Б. А., Калайда В. Т., Елизаров А. И.** Компоненты системы безопасности на базе комплекса "Базис" // Докл. Томского государственного университета систем управления и радиоэлектроники. 2009. № 1(19). Ч. 1. С. 193—200.
11. **Соловьев Б. А., Калайда В. Т.** Базовое программное обеспечение интегрированных распределенных систем безопасности // Информационные технологии. 2006. № 1. С. 43—49.
12. **Соловьев Б. А., Елизаров А. И.** Распределенная система безопасности "ЛИК" // Изв. Томского политехнического университета. 2008. Т. 313, № 5. С. 110—116.
13. **Соловьев Б. А., Калайда В. Т., Лопасов В. П.** Распределенная система управления бигармоническим лазерным спектрометром // Докл. Томского государственного университета систем управления и радиоэлектроники. 2010. № 1 (21). Ч. 2. С. 172—176.

УДК 616-073.96

С. А. Тараканов[1], канд. техн. наук, директор,
e-mail: k.v.tarakanov@gmail.com,
В. И. Кузнецов[2], директор,
e-mail: mail@kbst-itmo.ru,
Н. И. Рыжаков[1], стар. науч. сотр.,
e-mail: mail@kbst-itmo.ru,
А. А. Рассадина[1], канд. техн. наук, стар. науч. сотр.,
e-mail: a.a.rassadina@gmail.com,
В. Н. Когаленок[3], ген. директор.,
e-mail: kvn69@samson-rus.com,
[1]Центр медицинского, экологического приборостроения и биотехнологий НИУ ИТМО, г. Санкт-Петербург,
[2] ООО "Кардиопатруль",
[3]ООО "САМСОН Групп"

Алгоритмы регулирования информационных потоков между врачом и пациентом при дистанционной диагностике в режиме реального времени

Разработанный авторами макет аппаратно-программно-алгоритмического комплекса дистанционного кардиореспираторного мониторинга предназначен для дистанционной онлайн диагностики ЭКГ и кардиореспираторных параметров организма человека. Предложены алгоритмы регулирования информационных потоков, возникающие при передаче диагностируемых данных в процессе мониторинга. Также рассмотрены алгоритмы регулирования информационных потоков, возникающих между участниками дистанционного кардиореспираторного мониторинга.

Ключевые слова: *дистанционная диагностика в режиме реального времени, кардиореспираторный мониторинг*

Введение

Одно из важнейших направлений современной клинической медицины, и телемедицины в частности, — дистанционная диагностика. Преимущественно дистанционное наблюдение осуществляется по методу Холтера. В основе метода — непрерывная регистрация диагностируемых данных постоянно носимым пациентом портативным устройством. Считывание и анализ зарегистрированных данных выполняется в диагностических центрах, которые пациент должен регулярно посещать [1—3]. Другой вариант дистанционной диагностики получил название транстелефонного мониторирования. В этом случае отсутствует непрерывная регистрация диагностируемых данных. Пациент периодически осуществляет диагностику с помощью портативного устройства и далее самостоятельно передает диагностируемые данные в медицинский центр посредством набора телефонного номера центра [1, 4, 5].

Не так давно появились публикации о возможности осуществлять непрерывный онлайн контроль больного на расстоянии с помощью средств мобильной связи [6—17]. Передача непрерывно диагностируемых данных в таких устройствах происходит в автоматическом режиме, без участия диагностируемого пациента, через его сотовый телефон. Пока такие устройства применяют только в США. Устройства выдают специализированные диагностические центры больным, находящимся в реабилитационном периоде, для реализации непрерывной онлайн диагностики в течение 30 календарных дней, и затем пациенты возвращают их. В Европе же и в России таких аналогов нет.

В приказе Минздрава РФ от 30.11.1993 № 283 "О совершенствовании службы функциональной диагностики в учреждениях здравоохранения Российской Федерации", составленном д-ром мед. наук, профессором, Заслуженным врачом Российской Федерации, начальником управления медицинской помощи населению А. Д. Царегородцевым и д-ром мед. наук, профессором, начальником Управления охраны здоровья матери и ребенка Д. И. Зелинской [18] отмечается важное значение дистанционно-диагностических центров как методов функционального исследования. С этим мнением согласен заместитель директора по научной работе Российского научного центра восстановительной медицины и курортологии Росздрава (РНЦ ВМиК), д-р мед. наук, профессор И. П. Бобровницкий [19]. Внедрение дистанционной диагностики в практику российского здравоохранения даст возможность сэкономить значительные средства на стационарном обслуживании, так как позволит перевести многих больных на амбулаторное лечение. Наблюдение больного в непрерывном режиме — это возможность осуществлять оперативное вмешательство при возникновении критических отклонений диагностируемых параметров, сохранить жизнь и здоровье

многим пациентам. Особенно важен дистанционный контроль при наблюдении больных с различными патологиями сердечно-сосудистой системы, так как уровень смертности от этих заболеваний занимает одно из самых высоких мест.

Американский образец и сопутствующее ему программное обеспечение стоят чрезвычайно дорого, требуют дополнительных финансовых вложений в организацию специализированного дистанционного диагностического центра. Вместе с тем прямых аналогов разработке США в других странах не существует. Поэтому предложенный авторами макет аппаратно-программно-алгоритмического комплекса (АПАК) дистанционного кардиореспираторного мониторинга является уникальной отечественной разработкой, апробация, промышленная реализация и внедрение которого в медицинскую практику несомненно станет социально значимым проектом. В создании АПАК участвовали творческие коллективы НИУ ИТМО, ООО "Кардиопатруль" и ООО "САМСОН Групп", г. Санкт-Петербург.

Макет АПАК дистанционного кардиореспираторного мониторинга имеет следующие уникальные особенности:

- одновременная диагностика кардиологических (ЭКГ) и респираторных (ускорение грудной клетки при дыхании) сигналов;
- онлайн диагностика в реальном времени в течение длительного периода (месяц и более) на основе системы мобильного интерфейса.

Основные элементы макета АПАК: портативное диагностическое устройство; устройство для передачи данных — мобильный телефон пациента; устройство для хранения и обработки данных — серверное оборудование диагностического центра.

Особенностью комплекса является использование унифицированного протокола, обеспечивающего передачу диагностируемых данных от наблюдаемого пациента в его электронную карту, хранящуюся на сервере дистанционного диагностического центра (ДДЦ). Реализованный в АПАК унифицированный протокол обеспечивает взаимодействие с любой медицинской информационной системой (МИС). Таким образом, дистанционную диагностику на базе АПАК можно развернуть в любом медицинском учреждении, что соответствует концепции построения отраслевой системы здравоохранения России.

Для практического применения АПАК в медицинской практике требуются четкие алгоритмы регулирования информационных потоков между врачом и пациентом. Задача управления информационными потоками — чрезвычайно важный и интересный раздел дистанционной диагностики в телемедицине, включающий такие аспекты, как диагностика, коммуникации, запись, хранение и обработка данных, альтернативная связь.

В настоящей работе вниманию читателей представлены оригинальные решения алгоритмов регулирования информационных потоков между врачом и пациентом, предложенные командой разработчиков.

Функционирование алгоритмов регулирования информационных потоков будет рассмотрено нами на примере больных сердечно-сосудистыми заболеваниями, выписываемых из стационарных лечебных учреждений на амбулаторное лечение. Предполагается регистрация пациентов в ДДЦ в целях контроля состояния здоровья и осуществляемого лечения.

Алгоритмы регулирования информационных потоков передачи данных в АПАК дистанционного кардиореспираторного мониторинга

Макет АПАК дистанционного кардиореспираторного мониторинга включает:

- носимое пациентом портативное диагностическое устройство — система кардиореспираторного мониторинга (СКМ);
- телефон-трансвитер (сотовый телефон пациента);
- сервисное оборудование дистанционного диагностического центра.

Функции СКМ заключаются в регистрации ЭКГ и кардиореспираторных сигналов, фильтрации регистрируемых сигналов от посторонних шумов и помех, первичной обработке и записи сигнала, порционной передаче записи с заданной периодичностью на телефон-трансивер. В устройстве предусмотрена экстренная передача сигнала при критических отклонениях диагностируемых данных от нормы. Передача сигналов СКМ осуществляется через стандартные порты, позволяющие передавать до 10 тысяч измерений в секунду и беспроводную технологию BlueTooth.

Телефон-трансивер осуществляет ретрансляцию полученных данных на серверное оборудование ДДЦ, посредством технологии беспроводной пакетной передачи данных в сотовых сетях (GPRS, EDGE, 3G — в зависимости от зоны покрытия сотового оператора). Так как большой объем передаваемой посредством мобильных сотовых сетей информации связан с большим расходом электрической энергии на прием-передачу, и при непрерывной передаче сигналов телефон-трансивер быстро разряжается, в АПАК предусмотрена порционная передача записанных СКМ данных.

Серверное оборудование ДДЦ предназначено для записи и хранения диагностируемых данных в электронной карте больного, обработки данных, оповещения о диагностических отклонениях специалистов центра. Использование унифицированных протоколов передачи, хранения и обработки данных позволяют интегрировать специализированное программное обеспечение серверного оборудо-

вания ДДЦ в любую медицинскую информационную систему, обеспечивая универсальность АПАК.

Предполагается, что контроль принимаемых сигналов будет осуществляться круглосуточно дежурным оператором ДДЦ, а связь с дежурным врачом — посредством Интернет и телефона.

Алгоритмы регулирования информационных потоков между врачом и пациентом

В регулировании информационных потоков между врачом и пациентом участвуют пациент, дежурные оператор ДДЦ и врач. При контроле больных, выписываемых из стационара, желательно амбулаторное сопровождение сиделкой или, при ее отсутствии, родными и близкими.

Алгоритмы регулирования информационных потоков представлены на рис. 1.

Передача сигналов от СКМ в ДДЦ выполняется автоматически без участия пациента. В случае, если пациент ощущает дискомфорт, испытывает нарушения дыхания или сердцебиения, чувствует боль в сердце или груди и др., он может отправить информацию о своем текущем состоянии на сервер ДДЦ в приоритетном режиме, нажав специальную кнопку СКМ.

Основная задача оператора ДДЦ осуществлять контроль входных данных и обеспечивать передачу сигналов приоритетного и экстренного порядка. Также в задачи оператора входит оперативное реагирование при отсутствии сигналов от СКМ.

Дежурный врач осуществляет ежедневный контроль наблюдаемых пациентов, оперативный анализ данных, поступивших в приоритетном и экстренном режиме.

При отсутствии сигналов от носимого пациентом СКМ предусмотрена альтернативная обратная связь с пациентом (рис. 2). Оператор ДДЦ в этом случае действует согласно инструкции. В частности, в его задачи входит оперативное реагирование при исчезновении сигналов от СКМ, информирование и передача врачу результатов диагностики за последний промежуток времени. Врач проводит анализ полученных результатов и осуществляет связь с пациентом или наблюдающим за ним персоналом. Пациент по возможности должен проинформировать персонал ДДЦ о причинах возникших сбоев в передаче данных.

В случае экстренной госпитализации желательно наличие у больного медицинской карты, хранящей основные данные о пациенте и ДДЦ.

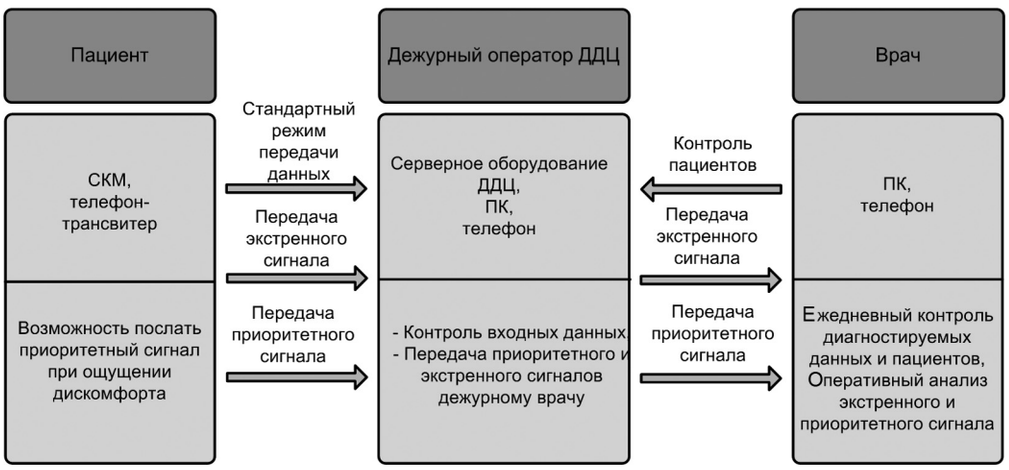


Рис. 1. Алгоритмы регулирования информационных потоков

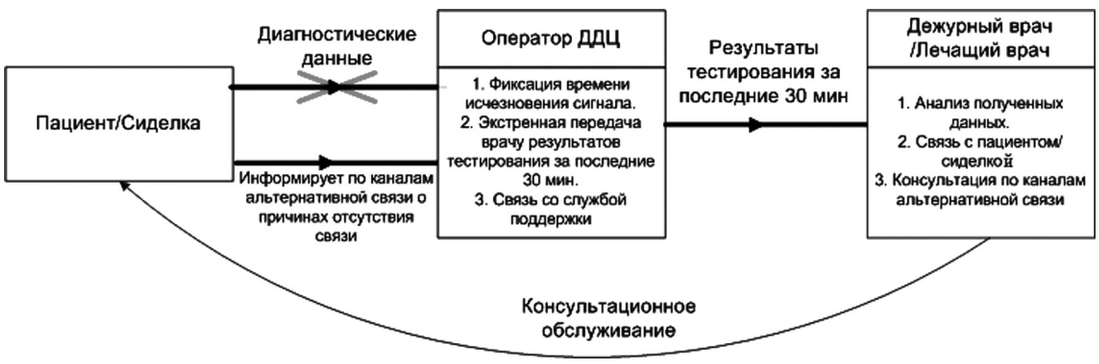


Рис. 2. Альтернативная обратная связь с пациентом

Обсуждение

Алгоритмы регулирования информационных потоков определяются, с одной стороны, особенностями функционирования АПАК дистанционного кардиореспираторного мониторинга, а с другой стороны, зависят от протоколов взаимодействия медицинского персонала ДДЦ и пациента.

Для реализации онлайн диагностики АПАК применены технологии беспроводной пакетной передачи данных (GPRS, EDGE, 3G), обеспечивающие доступ к услугам сети Интернет через сотового оператора. При такой связи оплачивается только объем посланной/полученной информации, а не эфирное время. Так как объем передаваемых с СКМ данных достаточно большой, эта услуга может оказаться дорогой для пациентов. Возможен и другой вариант передачи сигнала — пакетная передача *sms*. Этот вариант является более дорогим решением. Тем не менее, в будущем, на этапе практического внедрения АПАК, возможно согласование с операторами сотовой связи специализированных медицинских тарифов, рассчитанных на пакетную передачу данных как посредством технологий беспроводной пакетной передачи данных, так и *sms*-сигналов.

Заключение

Дистанционная диагностика в реальном времени откроет уже в ближайшем будущем новые возможности амбулаторного наблюдения за больными реабилитационно-востановительного периода, или нуждающимися в стационарном обследовании.

АПАК дистанционного кардиореспираторного мониторинга позволит обслуживать такого пациента амбулаторно, а не содержать в круглосуточном стационаре, что сэкономит бюджетные средства и одновременно повысит качество жизни населения. При наличии четких алгоритмов регулирования информационных потоков передачи данных и взаимодействия врача и пациента врач сможет осуществлять постоянное наблюдение за больными, включая оперативное вмешательство при возникновении критических отклонений в самочувствии или при бессимптомных отклонениях.

Список литературы

1. **Ослопов В. Н., Богоявленская О. В., Милославский Я. М.** и др. Инструментальные методы исследования сердечно-сосудистой системы. ГЭОТАР-Медиа. 2012. 624 с.

2. **Fujiki A., Yoshioka R., Sakabe M., Kusuzaki S.** QT/RR relation during atrial fibrillation based on a single beat analysis in 24-h Holter ECG: The role of the second and further preceding RR intervals in QT modification // Journal of Cardiology. 2011. Vol. 57, N 3. P. 269—274.

3. **Segura-Juárez J., Cuesta-Frau D., Samblas-Pena L., Aboy M.** A microcontroller-based portable electrocardiograph recorder // IEEE Trans. Biomed. Eng. 2004. Vol. 51, N 9. P. 1686—1690.

4. **Mundt W., Montgomery K., Udoh U.** et al. A multiparameter wearable physiologic monitoring system for space and terrestrial applications // IEEE Transactions on Information Technology in Biomedicine. 2005. Vol. 9, N 3. P. 382—391.

5. **Kouidi E., Farmakiotis A., Kouidis N., Deligiannis A.** Transtelephonic electrocardiographic monitoring of an outpatient cardiac rehabilitation programme // Clin. Rehabil. 2006. Vol. 20. P. 1100—1104.

6. **Wen C., Yeh M., Chang K., Lee R.** Real-time ECG telemonitoring system design with mobile phone platform // Measurement. 2008. Vol. 41, N 4. P. 463—470.

7. **Dorn R., Völker M., Neubauer H., Hauer J., Johansson J.** A 3-channel ECG measuring system for wireless applications // Proc. of MeMeA 2006 — International Workshop on Medical Measurement and Applications, Benevento, Italy. 2006, 20—21 April. P. 49—52.

8. **Väisänen O., Mäkijärvi M., Silfvast T.** Prehospital ECG transmission: comparison of advanced mobile phone and facsimile devices in an urban Emergency Medical Service System // Resuscitation. 2003. Vol. 57, N 2. P. 179—185.

9. **Engin M., Yamaner Y., Engin E.** A biotelemetric system for human ECG measurements // Measurement. 2005. Vol. 38, N 2. P. 148—153.

10. **Mamaghanian H., Khaled N., Atienza D., Vandergheynst P.** Compressed Sensing for Real-Time Energy-Efficient ECG Compression on Wireless Body Sensor Nodes // IEEE Transactions on Biomedical Engineering. 2011. Vol. 58, N 9.

11. **Lee H. J., Lee S. H., Ha K. S.** et al. Ubiquitous healthcare service using Zigbee and mobile phone for elderly patients // International Journal of Medical Informatics. 2009. Vol. 78, N 3. P. 193—198.

12. **Figueredo M. V. M., Dias J. S.** Mobile Telemedicine System for Home Care and Patient Monitoring // Proc. of the 26th Annual International Conference of the IEEE EMBS, San Francisco, CA, USA. 2004, September 1—5. P. 3387—3390.

13. **Goñi A., Burgos A., Dranca L.** et al. Architecture, cost-model and customization of real-time monitoring systems based on mobile biological sensor data-streams // Computer Methods and Programs in Biomedicine. 2009. Vol. 96, N 2. P. 141—157.

14. **Warren I., Weerasinghe T., Maddison R., Wang Odin Y.** Telehealth: A Mobile Service Platform for Telehealth // Procedia Computer Science. 2011. Vol. 5. P. 681—688.

15. **Picard R. W., Liu K. K.** Relative subjective count and assessment of interruptive technologies applied to mobile monitoring of stress // International Journal of Human-Computer Studies. 2007. Vol. 65, N 4. P. 361—375.

16. **Su C. J.** Mobile multi-agent based, distributed information platform (MADIP) for wide-area e-health monitoring // Computers in Industry. 2008. Vol. 59, N 1. P. 55—68.

17. **Winkler S., Schieber M., Lücke S.** et al. A new telemonitoring system intended for chronic heart failure patients using mobile telephone technology — Feasibility study // International Journal of Cardiology. 2011. Vol. 153, N 1. P. 55—58.

18. **Царегородцев А. Д., Зелинская Д. И.** Приказ Минздрава РФ от 30.11.1993 № 283 "О совершенствовании службы функциональной диагностики в учреждениях здравоохранения Российской Федерации". URL: http://www.lawmix.ru/med/17101 (Дата обращения: 28.08.2012).

19. **Бобровницкий И. П.** Разработка и внедрение инновационных технологий восстановительной медицины в практику здравоохранения Российской Федерации // Восстановительная медицина и реабилитация. Т. 1. 2010. URL: http://www.rosmedportal.com/index.php?option = com_content&view = article&id = 775 (дата обращения: 28.08.2012).

УДК 004.8.032.26

И. И. Мельников, аспирант,
К. А. Демиденков, аспирант,
И. А. Емельянов, магистрант,
И. А. Евсеенко, канд. техн. наук, доц.
ГУВПО "Белорусско-российский университет",
e-mail: mel_igor@mail.ru

Детектор движения на основе импульсных нейронных сетей

Предложены модель нейронной сети на базе импульсных нейронов, позволяющая выделять движущиеся объекты на видеоизображении, и детектор движения на ее основе. Предлагаемый детектор представляет собой альтернативу детекторам на базе детерминистских методов, так как требует меньше вычислительных ресурсов при той же скорости обработки видеоизображения.

Ключевые слова: *модель импульсной нейронной сети, модель нейрона "обобщение—отклик", возбуждающий синапс, ингибирующий синапс, выделение переднего плана, детектор движения*

Введение

Автоматизированное выделение и распознавание движущихся объектов — весьма перспективное направление исследований в области цифровой обработки видеоизображения и распознавания образов. Оно позволяет без участия человека контролировать технологический процесс, фиксировать нарушения правил дорожного движения, проводить автоматизированный сбор и анализ информации и др. Актуальность исследований связана еще и с тем, что во многих странах все более широкое применение находят системы видеонаблюдения.

Так, в Республике Беларусь все больше систем видеонаблюдения появляется на дорогах в целях слежения за движением транспорта. Одновременно с этим существует необходимость в автоматизированном сборе и анализе информации о транспортных потоках в крупных городах для оперативного реагирования на изменения дорожно-транспортной обстановки (например, изменение режимов работы светофоров вследствие роста плотности транспортного потока). Другими словами, существует необходимость в автоматизированных системах определения плотности транспортного потока и его состава [9, 10]. В таких системах могут быть использованы камеры видеонаблюдения в качестве источника информации, однако необходимо реализовать методы, которые бы позволяли, во-первых, выделять движущиеся объекты на видеоизображении, во-вторых, распознавать их как транспортные средства определенного класса, причем делать это достаточно быстро, не требуя большого количества вычислительных ресурсов.

Прежде чем непосредственно распознавать движущиеся объекты, их нужно выделить из статического фона. Для этого существует ряд детерминистских методов выделения переднего плана: методы вычитания фона, методы временной разности, методы оптического потока и др. Выбор метода сильно влияет на эффективность работы всей системы распознавания. И чем эффективнее метод, тем, как правило, он сложнее и требует больше ресурсов.

Цель данной работы — представить принципиально иной подход к цифровой обработке видеоизображений, в частности, к выделению движущихся объектов, основанный на нейронных сетях. Это позволит создать эффективный детектор движения, который сможет стать альтернативой детекторам, разработанным на базе детерминистских методов, в том числе и улучшенным, путем применения параллельной обработки сегментов видеоизображения [9, 10].

Выделение движущихся объектов с помощью нейронных сетей

Человеку не составляет труда быстро выделять движущиеся объекты. Однако за этим умением скрывается довольно сложная система обработки визуальной информации — сетчатка глаза [4, 6]. Последняя состоит из сложных цепочек нейронов, на переднем фронте которых находятся фоторецепторы (светочувствительные клетки), непосредственно воспринимающие оптические сигналы и преобразующие их в физиологические возбуждения. Возбуждения от фоторецепторов передаются по интернейронам, или вставочным нейронам, которые синаптически контактируют друг с другом и связывают фоторецепторы с ганглиозными клетками сетчатки, посылающими через зрительный нерв сигналы дальше в мозг. Различные типы клеток отвечают за обработку различных характеристик изображения: яркость, цвет, движение объектов. Под воздействием внешнего раздражителя, будь то световой сигнал или сигнал от соседнего нейрона, текущий нейрон начинает испускать импульсы одинаковой амплитуды. И чем сильнее внешнее воздействие, тем чаще испускаются сигналы. В работе [5] показано, как сеть ганглиозных клеток может мгновенно обнаруживать движущийся объект и даже выделять несколько таких объектов. Результаты исследований, представленные в работах [4—6], раскрывают принципы работы сетчатки при выделении движущихся объектов с точки зрения физиологии. Но можно ли использовать данную информацию для создания искусственных нейронных сетей,

способных выделять движущиеся объекты так же быстро и точно?

В работах [7, 11, 12] для этих целей предлагается использовать искусственные импульсные (спайковые) нейронные сети, которые являются наиболее близкими аналогами биологических нейронных сетей. В отличие от традиционных нейронных сетей они передают информацию не в форме усредненного значения нейронной активности, а через последовательность импульсных сигналов [3, 8]. Выходной сигнал нейрона состоит из коротких электрических импульсов (также называемых действующими потенциалами или спайками). Форма импульсов не изменяется при передаче по аксону. Цепь действующих потенциалов, вызванных одним нейроном, называется импульсной последовательностью — чередой одинаковых событий, возникающих в определенные или случайные моменты времени. Поскольку все генерируемые импульсы имеют примерно одинаковую форму, то информация содержится не в форме импульсов, а в их числе и в точном времени их возникновения.

Модель импульсной нейронной сети для выделения движущихся объектов

Общая структура импульсной нейронной сети, применяемая для выделения движущихся объектов и используемая в разрабатываемом детекторе, представлена на рис. 1. Входной слой нейронов является аналогом слоя фоторецепторов сетчатки глаза, поэтому далее будем называть нейроны первого слоя рецепторами. Каждому пикселю (x, y) входного кадра видеоизображения соответствует свой рецептор $N_r(x, y)$. Скрытый слой представляет собой аналог слоя интернейронов сетчатки глаза. Он состоит из двух независимых друг от друга массивов нейронов N_1 и N_2. Они имеют такие же размеры, как и первый слой, и соединены синаптическими связями как с входным слоем нейронов (рецепторов) N_r, так и с выходным слоем нейронов N_{out}.

Каждый рецептор $N_r(x, y)$ соединен с интернейроном $N_1(x, y)$ посредством возбуждающего синапса, передающего сигналы без задержек, и ингибирующего синапса, передающего сигналы с синаптической задержкой Δt. Также каждый рецептор $N_r(x, y)$ соединен с интернейроном $N_2(x, y)$ посредством возбуждающего синапса, передающего сигналы с синаптической задержкой Δt, и ингибирующего синапса, передающего сигналы без задержек. Допустим, что $I_{nr}(x, y, t)$ — сила тока, текущего от рецептора $N_r(x, y)$ в момент времени t. Если ток, идущий от рецептора $N_r(x, y)$ стабилен, т. е. $I_{nr}(x, y, t) = I_{nr}(x, y, t - \Delta t)$, а возбуждающий и ингибирующий входы интернейрона $N_1(x, y)$ сбалансированы путем настройки весов $w^{ex}_{N_1(x, y)}$ и $w^{ih}_{N_1(x, y)}$ соответственно, то интернейрон $N_1(x, y)$ будет находиться в состоянии покоя. Аналогичная ситуация складывается и с интернейроном $N_2(x, y)$. Если же сила тока рецептора $N_r(x, y)$ растет, т. е. $I_{nr}(x, y, t) > I_{nr}(x, y, t - \Delta t)$, то баланс нарушается, поскольку сигнал, приходящий с возбуждающего синапса, сильнее, чем задержанный на период Δt сигнал, приходящий с ингибирующего синапса. Интернейрон $N_1(x, y)$ начинает генерировать импульсы (спайки). Если же сила тока рецептора снижается, т. е. $I_{nr}(x, y, t) < I_{nr}(x, y, t - \Delta t)$, то интернейрон $N_1(x, y)$ не реагирует, но зато начинает генерировать импульсы интернейрон $N_2(x, y)$, поскольку задержанный на период Δt сигнал, приходящий с возбуждающего синапса, сильнее, чем сигнал, приходящий с ингибирующего синапса. Другими словами, нейронная сеть начинает реагировать на изменение яркости пикселей, которая может быть вызвана прохождением движущегося объекта по статическому фону.

Выходной слой нейронной сети N_{out} имеет такие же размеры, как входной и скрытый слои. Каждый нейрон данного слоя $N_{out}(x, y)$ соответствует каждому пикселю (x, y) выходного кадра видеоизображения. Интернейроны $N_1(x, y)$ и $N_2(x, y)$ соединены возбуждающими синапсами без задержек с выходным нейроном $N_{out}(x, y)$. Он продуцирует сигналы только в том случае, когда получает им-

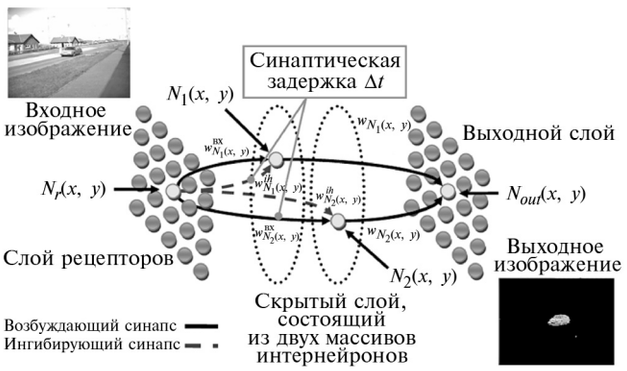


Рис. 1. Модель импульсной нейронной сети для детектора движения

Рис. 2. Выделение движущегося объекта с помощью импульсной нейронной сети

пульсы от интернейрона $N_1(x, y)$ или $N_2(x, y)$, иначе он находится в состоянии покоя. Значение яркости серой шкалы каждого пикселя (x, y) выходного кадра видеоизображения пропорционально частоте генерации импульсов выходным нейроном $N_{out}(x, y)$ и имеет значение 0 (черный), если выходной нейрон $N_{out}(x, y)$ не генерирует никаких сигналов с течением определенного периода времени T. В противном случае яркость пикселя (x, y) будет выше 0 (рис. 2).

Модель импульсного нейрона

Существуют различные модели [2] импульсных нейронов: Ходкина-Хаксли, "обобщение—отклик" (*Integrate-and-Fire*), импульсного отклика (*Spike Response Model*) и т. д. Наиболее детализированной и сложной является модель Ходкина-Хаксли [8]. Она основана на экспериментальном исследовании большого числа нейронов кальмара. Система дифференциальных уравнений данной модели описывает точную реакцию потенциала мембраны нейрона в ответ на различные входные воздействия. Однако такая реалистичность приводит к большим вычислительным затратам, в результате чего модель не очень подходит для экспериментов с нейронными сетями, состоящими из большого числа нейронов, как в данном случае.

Основываясь на работе [7], в рамках разрабатываемого детектора движения используется модель нейрона "обобщение—отклик" (*integrate-and-fire, IaF*), которая более проста при математическом описании и достаточно эффективна.

В модели *IaF* импульсы рассматриваются как короткие импульсные токи. После того как импульс приходит на синапс, мгновенно заряжаются все связанные с ним постсинаптические нейроны. Это изменение напряжения называется постсинаптический потенциал. По достижении потенциала мембраны нейрона порогового значения, он сбрасывается и генерируется новый импульс.

Пусть $G_{x, y}(t)$ — яркость серой шкалы для отдельно взятого пикселя (x, y) входного изображения в момент времени t; $q^{ex}_{x, y}(t)$ — проводимость возбуждающего синапса, идущего от рецептора $N_r(x, y)$, $q^{ih}_{x, y}(t)$ — проводимость ингибирующего синапса, идущего от рецептора $N_r(x, y)$, тогда формулы трансформации яркости серой шкалы примут вид:

$$q^{ex}_{x, y}(t) = \alpha G_{x, y}(t),\ q^{ih}_{x, y}(t) = \beta G_{x, y}(t),$$

где α и β — некоторые коэффициенты преобразования. В соответствии с работой [7] импульсный нейрон $N_1(x, y)$ можно описать следующими уравнениями:

$$\frac{dg_{N_1^{ex}(x, y)}(t)}{dt} = -\frac{1}{\tau_{ex}} g_{N_1^{ex}(x, y)}(t) + \alpha G_{x, y}(t);$$

$$\frac{dg_{N_1^{ih}(x, y)}(t)}{dt} = -\frac{1}{\tau_{ih}} g_{N_1^{ih}(x, y)}(t) + \beta G_{x, y}(t);$$

$$c_m \frac{d\nu_{N_1(x, y)}(t)}{dt} = g_m(E_m - \nu_{N_1(x, y)}(t)) +$$

$$+ \frac{w_{N_1^{ex}(x, y)} g_{N_1^{ex}(x, y)}(t)}{A_{ex}} (E_{ex} - \nu_{N_1(x, y)}(t)) +$$

$$+ \frac{w_{N_1^{ih}(x, y)} g_{N_1^{ih}(x, y)}(t - \Delta t)}{A_{ih}} (E_{ih} - \nu_{N_1(x, y)}(t - \Delta t)),$$

где $g_{N_1^{ex}(x, y)}(t)$ и $g_{N_1^{ih}(x, y)}(t)$ — проводимости мембраны, соответственно, возбуждающего и ингибирующего синапсов, соединяющих нейрон $N_r(x, y)$ и $N_1(x, y)$; τ_{ex} и τ_{ih} — характеристическое синаптическое время возбуждающего и ингибирующего синапсов соответственно (обычно равно 2 мс); Δt — синаптическая задержка при передаче импульса от нейрона $N_r(x, y)$ нейрону $N_1(x, y)$; $\nu_{N_1(x, y)}$ — потенциал мембраны нейрона $N_1(x, y)$; E_m — равновесный потенциал мембраны нейрона; g_m — проводимость мембраны нейрона; E_{ex} и E_{ih} — значения равновесных потенциалов возбуждающего и ингибирующего синапсов соответственно; A_{ex} — площадь поверхности мембраны нейрона $N_1(x, y)$, соединенной с возбуждающим синапсом; A_{ih} — площадь поверхности мембраны нейрона $N_1(x, y)$, соединенной с ингибирующим синапсом; c_m — удельная емкость мембраны нейрона; $w_{N_1^{ex}(x, y)}$ — сила возбуждающей синаптической связи между рецептором $N_r(x, y)$ и интернейроном $N_1(x, y)$; $w_{N_1^{ih}(x, y)}$ — сила ингибирующей синаптической связи между рецептором $N_r(x, y)$ и интернейроном $N_1(x, y)$. Коэффициенты $w_{N_1^{ex}(x, y)}$ и $w_{N_1^{ih}(x, y)}$ подобраны таким образом, чтобы интернейрон $N_1(x, y)$ оставался в состоянии покоя при $G_{x, y}(t) = G_{x, y}(t - \Delta t)$. Аналогичные уравнения строят и для интернейрона $N_2(x, y)$.

Когда потенциал мембраны интернейрона $N_1(x, y)$ или $N_2(x, y)$ достигает порогового значения v_{th}, он генерирует импульс, передаваемый на выходной нейрон $N_{out}(x, y)$. Пусть $S_{N_1(x, y)}(t)$ отражает после-

довательность импульсов, генерируемых нейроном $N_1(x, y)$, тогда

$$S_{N_1(x,y)}(t) =$$

$$= \begin{cases} 1, \textit{ если интернейрон } N_1(x, y) \textit{ сгенерировал} \\ \textit{импульс в момент времени } t, \\ 0, \textit{ если интернейрон } N_1(x, y) \textit{ не сгенерировал} \\ \textit{импульс в момент времени } t. \end{cases}$$

Последовательность импульсов, генерируемых нейроном $N_2(x, y)$, обозначим $S_{N_2(x,y)}(t)$. Выходной нейрон $N_{out}(x, y)$ будет описываться следующими уравнениями:

$$\frac{dg_{N_{out}(x,y)}}{dt} = -\frac{1}{\tau_{ex}} g_{N_{out}(x,y)}(t) +$$

$$+ (w_{N_1(x,y)} S_{N_1(x,y)}(t) + w_{N_2(x,y)} S_{N_2(x,y)}(t));$$

$$c_m \frac{d\nu_{N_{out}(x,y)}}{dt} = g_m(E_m - \nu_{N_{out}(x,y)}(t)) +$$

$$+ \frac{g_{N_{out}(x,y)}(t)}{A_{ex}} (E_{ex} - \nu_{N_{out}(x,y)}(t)),$$

где $g_{N_{out}(x,y)}$ — значение проводимости каждого из синапсов, соединяющих интернейроны $N_1(x, y)$ и $N_2(x, y)$ с выходным нейроном $N_{out}(x, y)$; $\nu_{N_{out}(x,y)}$ — потенциал мембраны нейрона $N_{out}(x, y)$; $w_{N_1(x,y)}$ — сила возбуждающей синаптической связи между интернейроном $N_1(x, y)$ и выходным нейроном $N_{out}(x, y)$; $w_{N_2(x,y)}$ — сила возбуждающей синаптической связи между интернейроном $N_2(x, y)$ и выходным нейроном $N_{out}(x, y)$.

Пусть $S_{N_{out}(x,y)}(t)$ — последовательность импульсов, генерируемых выходным нейроном $N_{out}(x, y)$. Тогда частота импульсов, генерируемых выходным нейроном $N_{out}(x, y)$, рассчитывается по следующей формуле:

$$r(x, y, t) = \frac{1}{T} \sum_{t}^{t+T} S_{N_{out}(x,y)}(t),$$

где T — период измерения импульсов, генерируемых выходным нейроном $N_{out}(x, y)$.

Преобразуя величину $r(x, y, t)$ для каждого пикселя выходного изображения в значение яркости серой шкалы, можно получить более светлую область, описывающую движущийся объект (см. рис. 2).

Заключение

Предложенный выше подход для обнаружения и выделения движущихся объектов является попыткой сымитировать способности человеческого глаза достаточно быстро выделять движущиеся объекты и превзойти существующие детерминистские методы по скорости выделения движущихся объектов и экономии вычислительных ресурсов. А разрабатываемый на базе данного подхода детектор движения, как программный модуль, сможет найти достойное применение в области цифровой обработки видеоизображения. Так, предполагается применение данного детектора в автоматизированных системах управления дорожным движением как альтернативу существующим детекторам, даже с учетом возможного улучшения последних путем использования параллельных вычислений для одновременной обработки сегментов видеоизображения и выделения движущихся объектов в пределах каждого из них [9, 10]. Подобный подход уже показал свою эффективность, позволяя быстро и точно выделять движущиеся объекты на видеоизображении [7].

Также не следует забывать, что элементы импульсной нейронной сети могут быть реализованы аппаратно [11] или программно с применением современных технологий параллельных вычислений на базе графических процессоров [1]. Это может значительно ускорить процесс выделения движущихся объектов на видеоизображении, хотя возможно потребует больших затрат для реализации и предварительной настройки детектора движения.

Список литературы

1. **Gallbraith B.** Computational Modeling of Biological Neural Networks on GPUs: Strategies and Performance / Marquette University: Master's Theses (2009). URL: http://epublications.marquette.edu/theses_open/61/ (дата обращения 30.09.2012).
2. **Gerstner W., Kistler W. M.** Spiking Neuron Models. Single Neurons, Populations, Plasticity. Cambridge University Press, 2002. 496 p.
3. **Maass W.** Networks of Spiking Neurons: The Third Generation of Neural Network Models // Neural Networks. — Elsevier Science Ltd., 1997. Vol. 10, N 9. P. 1659—1671.
4. **Masland R. H.** The Fundamental Plan of the Retina // Nature Neuroscienc. 2001. N 4. P. 877—886.
5. **Olveczky B. P., Baccus S. A., Meister M.** Segregation of Object and Background Motion in the Retina // Nature. 2003. N 423. P. 401—408.
6. **Wassle H.** Parallel Processing in the Mammalian Retina // Nature Reviews Neuroscience. 2004. N 5. P. 747—757.
7. **Wu Q., McGinnity T. M., Maguire L., Cai J.** [and etc.]. Motion Detection Using Spiking Neural Network Model // ICIC'08 Proc. of the 4th international conference on Intelligent Computing. — Berlin: Springer-Verlag, 2008. 8 p.
8. **Бендерская Е. Н., Никитин К. В.** Рекуррентные нейронные сети в задачах распознавания образов // Материалы IX Международной конференции "Интеллектуальные системы и компьютерные науки" (23—27 октября 2006 г.). М.: МГУ им. М. В. Ломоносова, 2006. Т. 1. Ч. 1. С. 60—69.
9. **Демиденков К. А., Мельников И. И.** Разработка автоматизированной системы обнаружения и идентификации транспортных средств для измерения плотности транспортного потока // Технические науки: теория и практика: материалы междунар. заоч. науч. конф. Чита: Молодой ученый, 2012. С. 11—16.
10. **Демиденков К. А., Мельников И. И.** Разработка автоматизированной системы определения плотности транспортного потока // Інформатика, математика, автоматика (IMA :: 2012): матеріали та програма навуково-технічної конференції. Сумы: Сумский гос. университет, 2012. С. 150.
11. **Колесницкий О. К., Бокоцей И. В., Яремчук С. С.** Аппаратная реализация элементов импульсных нейронных сетей с использованием биспин-приборов // Научная сессия МИФИ — 2010. Сборник научных трудов. М.: МИФИ, 2010. Ч. 1: XII Всероссийская научно-техническая конференция. Нейроинформатика — 2010. С. 121—132.
12. **Лукьяница А. А., Шишкин А. Г.** Цифровая обработка видеоизображений. М.: Ай-Эс-Эс Пресс, 2009. 518 с.

УДК 004.032.26

В. Ю. Осипов, д-р техн. наук, проф.,
Федеральное государственное
бюджетное учреждение науки
Санкт-Петербургский институт информатики
и автоматизации Российской академии наук,
e-mail: osipov_vasiliy@mail.ru

Метод управления синапсами в рекуррентной нейронной сети

Рассматривается подход к наделению рекуррентной нейронной сети новыми свойствами. Предлагается путем управления синапсами нейронов изменять направления ассоциативного взаимодействия сигналов в сети. Показано, что за счет такого управления осуществимо селективное запоминание сигналов и извлечение их из памяти сети с изменением порядка вызова. Достижимы узконаправленные ассоциативные обращения к различным областям памяти и переходы от одних ассоциаций к другим, что существенно расширяет функциональные возможности сети по обработке информации.

Ключевые слова: *нейронная сеть, синапсы, сигналы, ассоциации, управление*

Введение

Одной из актуальных научных задач в области искусственных нейронных сетей выступает расширение их функциональных возможностей по интеллектуальной обработке информации.

Анализ известных нейронных сетей [1—6] свидетельствует, что пока они далеки от совершенства. Ни одна из них не способна решать широкий спектр творческих задач. Эти сети не обладают гибкостью, свойственной биологическим нейронным сетям.

Среди известных нейронных сетей одними из наиболее гибких выступают рекуррентные многослойные сети с управляемыми синапсами [7—11]. Под управлением синапсами понимается изменение их характеристик в зависимости от текущих состояний слоев нейронной сети в целях достижения максимума ассоциативного взаимодействия в ней сигналов и расширения функциональных возможностей. За счет управления синапсами удается осуществлять пространственные сдвиги совокупностей единичных образов (ЕО), передаваемых от слоя к слою, и продвигать эти совокупности вдоль слоев сети, наделить ее логической структурой, улучшить ассоциативные свойства и память. Однако при этом не обращено внимание на возможность изменения ориентации распределения плотности мощности сигналов в поперечных сечениях расходящихся ЕО при передаче их от слоя к слою в зависимости от текущих состояний слоев.

В интересах расширения функциональных возможностей рекуррентной нейронной сети (РНС) по обработке информации предлагается метод управления синапсами, предусматривающий изменение направлений ассоциативного взаимодействия обрабатываемых сигналов.

Постановка задачи

Известна рекуррентная нейронная сеть с управляемыми синапсами [7, 9]. Структура этой сети может иметь вид, приведенный на рис. 1, где 1.1...1.N, 2.1...2.N — нейроны, соответственно, первого и второго слоя; N — число нейронов в каждом слое; ЕЗ — единичная задержка. Синапсы нейронов отображены в виде овалов. В сеть подаются сигналы, разложенные, в общем случае, на пространственно-частотные составляющие в базисе, согласованном с входным слоем сети. Причем каждая составляющая преобразована в последовательность единичных образов с частотой повторения как функцией от амплитуды составляющей. Эти последовательности можно рассматривать как последовательность совокупностей ЕО, поступающих на вход сети. Входными сигналами могут выступать, например, акустические, оптические и другие информационные воздействия. В качестве результатов обработки сигналов в сети используют последовательные совокупности ЕО на выходном слое сети после преобразования их в соответствующие им исходные сигналы. Нейроны этой сети могут находиться в трех состояниях: ожидания, возбуждения и невосприимчивости. Любой нейрон сети переходит в состояние возбуждения, только если суммарный потенциал на его входе превысит порог возбуждения. Каждый единичный образ из текущей совокупности подается в блоках динамических синапсов на совокупность своих синапсов, обеспечивающих связь нейрона, породившего ЕО, в общем случае, со всеми нейронами взаимодействующего слоя (рис. 1). За счет такого разветвления каждый ЕО преобра-

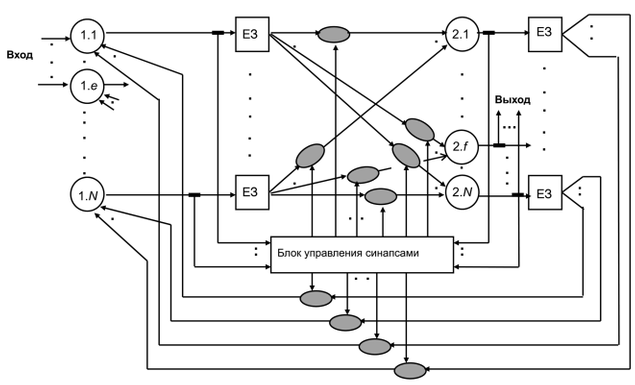


Рис. 1. Рекуррентная нейронная сеть с управляемыми синапсами

зуется в расходящийся пучок сигналов. Время задержки единичных образов в образуемых контурах известной сети меньше времени невосприимчивости нейронов после возбуждения. При передаче совокупностей ЕО от слоя к слою путем изменения весов синапсов осуществляются сдвиги этих совокупностей вдоль слоев. Веса $w_{ij}(t)$ синапсов, с электрической точки зрения проводимости, определяются через произведение их весовых коэффициентов $k_{ij}(t)$ и функций ослабления $\beta(r_{ij}(t))$, зависящих от удаленности r_{ij} связываемых синапсами нейронов: $w_{ij}(t) = k_{ij}(t)\cdot\beta(r_{ij}(t))$. Весовые коэффициенты можно рассчитывать как $k_{ij}(t) = 1 - \exp(-\gamma \cdot g_{ij}(t))$, где γ — постоянный коэффициент; $g_{ij}(t)$ — текущее число запомненных на синапсе ЕО с учетом их стирания [10]. Функция ослабления $\beta(r_{ij}(t))$ синапсов задается в следующем виде [7]:

$$\beta(r_{ij}(t)) = \frac{1}{1 + \alpha\sqrt[h]{r_{ij}(t)}},\ i = \overline{1, N};\ j = \overline{1, N}, \qquad (1)$$

h — степень корня; α — положительный постоянный коэффициент; N — число нейронов в каждом слое РНС. Входящая в выражение (1) величина r_{ij}, измеряемая в единицах нейронов, в зависимости от реализуемых пространственных сдвигов совокупностей единичных образов вдоль слоев, при условии, что расстояние между слоями стремится к нулю, определяется как [8]

$$r_{ij} = \sqrt{(\Delta x_{ij} + n_{ij}d)^2 + (\Delta y_{ij} + m_{ij}q)^2}, \qquad (2)$$

где $n_{ij} = \pm 0, 1, ..., L - 1$; $m_{ij} = \pm 0, 1, ..., M - 1$; Δx_{ij}, Δy_{ij} — проекции связи j-го нейрона с i-м на оси X, Y в плоскости принимающего слоя без учета пространственных сдвигов; d, q — единичные сдвиги, соответственно, по координатам X, Y; L, M — число, соответственно, столбцов и строк, на которые разбивается каждый слой нейронной сети за счет сдвигов. Произведение $d \times q$ определяет площадь рабочего поля каждого слоя сети. Она равна числу входящих в поле нейронов.

С формальной точки зрения сдвиги совокупностей ЕО по координатам X, Y с учетом (1), (2) осуществляются путем одновременного изменения Δx_{ij} и Δy_{ij} на величины $n_{ij} \cdot d$, $m_{ij} \cdot q$ для всех синапсов одного из слоев.

На рис. 2, *а*, *б* приведен пример логической структуры рекуррентной нейронной сети с управляемыми синапсами со сдвигами совокупностей ЕО вдоль слоев. Рис. 2, *а* соответствует виду спереди на сеть, а рис. 2, *б* — виду сверху на первый ее слой. На фоне первого слоя (рис. 2, *б*) условно показаны сглаженные уменьшенные формы *1* поперечных сечений расходящихся ЕО в совокупностях, передаваемых от одного слоя к другому и направления *2* распределения плотности мощности в поперечных сечениях этих образов. Направления продвижения совокупностей ЕО вдоль слоев на рис. 2, *б* обозначены цифрой *3*. В соответствии с рис. 2 входные сигналы в виде последовательных совокупностей ЕО подаются на первое поле первого слоя, продвигаются по сети, ассоциируясь друг с другом. Результаты обработки информации снимаются с последнего поля второго слоя. Однозначное соответствие между составляющими входных и выходных сигналов достигается за счет приоритетности коротких связей между нейронами.

На рис. 3 раскрывается понятие расходящегося единичного образа, своеобразного пучка, где *1* — нейрон передающего слоя; *2* — нейрон принимающего слоя, в направлении которого передается расходящийся единичный образ (этот нейрон связан с нейроном *1* синапсом с функцией ослабления $\beta(r_{ij}(t)) = 1$, так как для них $r_{ij}(t) = 0$); *3* — нейроны принимающего слоя, в направлении которых расходится этот образ через синапсы с функциями

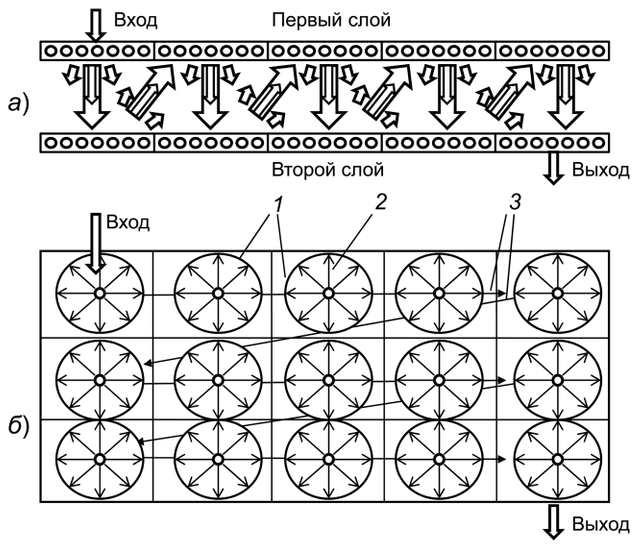


Рис. 2. Структура сети

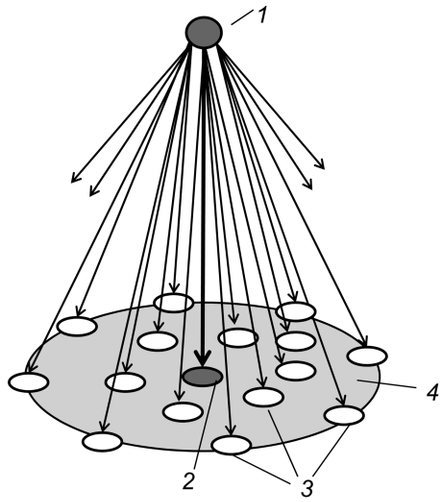


Рис. 3. Поперечное сечение расходящегося единичного образа

$\beta(r_{ij}(t)) < 1$ и $r_{ij}(t) > 0$; *4* — область принимающего слоя, содержащая нейроны, в направлении которых от нейрона *1* передается единичный образ с ослаблением не больше заданного значения. Эта область является поперечным сечением расходящегося единичного образа в плоскости принимающего слоя.

Согласно (1) распределение плотности мощности в поперечных сечениях расходящихся ЕО не имеет выраженной пространственной ориентации. Мощность единичных образов на входе нейронов тем выше, чем меньше ее потери на синапсах. Мощность, рассеиваемая на каждом синапсе, определяется квадратом тока, протекающего через него, умноженного на сопротивление синапса, равное обратной величине его проводимости — весу $w_{ij}(t) = k_{ij}(t)\beta(r_{ij}(t))$. При постоянном значении α в функции (1) этому распределению плотности мощности соответствует круговая форма поперечных сечений ЕО без смещения энергетических центров. Такая форма обеспечивает всенаправленное ассоциативное взаимодействие ЕО в сети. Однако это не всегда оправдано. Желательно иметь возможность оперативного изменения формы поперечных сечений расходящихся ЕО, придания ей и, соответственно, распределению плотности мощности вдоль слоев сети направленных свойств и управления этой направленностью. Это позволит варьировать направлениями ассоциативного взаимодействия сигналов в сети с учетом текущих состояний слоев и наделять искусственные нейронные сети новыми свойствами по интеллектуальной обработке информации.

Необходимо разработать метод управления синапсами сети, предоставляющий такие возможности.

Метод управления синапсами

С формальной точки зрения изменение направлений ассоциативного взаимодействия сигналов в сети достижимо путем изменения коэффициента α, входящего в функцию (1) ослабления синапсов. Для изменения этого коэффициента предлагается учитывать направления от нейронов передающего слоя к нейронам принимающего слоя через соответствующие углы. Например, зная координаты *i*-го нейрона передающего слоя и пространственный сдвиг совокупностей ЕО, можно найти координаты нейрона принимающего слоя, в направлении которого передается расходящийся единичный образ, а также координаты нейронов этого слоя, по которым данный образ расходится. По этим координатам в плоскости *X*, *Y* можно найти углы, соответствующие направлениям на нейроны принимающего слоя от нейрона этого слоя, в направлении которого передается расходящийся ЕО. Обозначим этот угол через φ_{ij}. Принимая во внимание, что сдвиг не влияет на значение φ_{ij}, этот угол с учетом (2) равен $\varphi_{ij} = \text{arctg}(\Delta x_{ij}/\Delta y_{ij})$, где Δx_{ij}, Δy_{ij} — расстояния между соответствующими нейронами в плоскости *X*, *Y* принимающего слоя.

Для изменения направленности ассоциативного взаимодействия сигналов в сети можно предварительно задаться одной или несколькими видами форм поперечных сечений расходящихся ЕО с явно выраженными направленными свойствами. Пусть для примера это будет эллиптическая форма, изначально ориентированная по оси *Y*. Для этой формы каждому значению угла χ от оси *Y* соответствует свое значение коэффициента направленности $G(\chi)$. Эти данные по $G(\chi)$ могут выступать в качестве исходных для определения значений коэффициента α в выражении (1).

Если все поперечные сечения расходящихся ЕО должны быть ориентированы по оси *Y*, то $\alpha = \vartheta/G(\chi)$, где угол χ принимается равным φ_{ij}; ϑ — положительная постоянная величина. С учетом этого обозначим α как α_{ij}, а угол χ через χ_{ij}.

При необходимости осуществления поворота вектора ориентации распределения плотности мощности в поперечных сечениях расходящихся ЕО на угол $\psi(x_{1t}, x_{2t})$ с учетом текущих состояний x_{1t}, x_{2t}, соответственно, первого и второго слоев сети, углы χ_{ij} должны рассчитываться как $\chi_{ij} = \varphi_{ij} + \psi(x_{1t}, x_{2t})$. В зависимости от знака угла $\psi(x_{1t}, x_{2t})$ повороты могут реализовываться как по часовой, так и против часовой стрелки. С учетом этого функцию ослабления синапсов можно записать в виде

$$\beta(r_{ij}(t), \varphi_{ij}, \psi(t)) = \frac{1}{1 + \vartheta \sqrt[h]{r_{ij}(t)}/G(\varphi_{ij} + \psi(x_{1t}, x_{2t}))}, \quad (3)$$

где $G(\varphi_{ij} + \psi(x_{1t}, x_{2t}))$ — коэффициент направленности формы поперечного сечения расходящегося ЕО в направлении от *i*-го к *j*-му нейрону в плоскости *X*, *Y*.

На рис. 4 приведены примеры, поясняющие изменение распределения плотности мощности в поперечных сечениях расходящихся ЕО и, соответственно, изменение ориентации ассоциативного взаимодействия единичных образов друг с другом и элементами памяти сети, где *1*, *2* — нейроны, формирующие ЕО; *3* — направления передачи ЕО по синапсам с наименьшим ослаблением, $\beta(r_{ij}(t)) = 1$; *4* — векторы ориентации распределения плотности

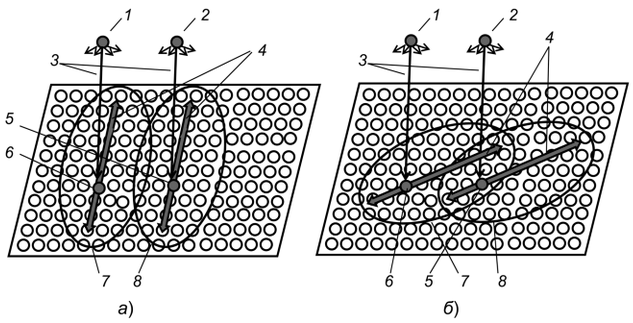


Рис. 4. Распределение плотности мощности в поперечных сечениях расходящихся единичных образов:
а — ориентация векторов распределения по оси *Y*; *б* — поворот векторов на заданный угол

мощности в поперечных сечениях расходящихся ЕО; *5*, *6* — нейроны принимающего слоя, в направлении которых от соответствующих нейронов передающего слоя передаются расходящиеся ЕО; *7*, *8* — сглаженные эллиптические формы поперечных сечений расходящихся ЕО, формируемых, соответственно, нейронами *1* и *2*.

Полагается, что каждый нейрон одного слоя связан со всеми нейронами другого слоя. Значение этой связи с увеличением расстояния между нейронами согласно (3) убывает. Рис. 4, *а* отражает случай ориентации распределения плотности мощности в поперечных сечениях расходящихся ЕО вдоль оси *Y*, а рис. 4, *б* — поворот вектора ориентации на угол $\psi(x_{1t}, x_{2t})$. Из анализа рис. 4 следует, что ассоциативное взаимодействие единичных образов друг с другом и с элементами памяти сети можно характеризовать областями, накрываемыми поперечными сечениями расходящихся ЕО, и их пересечениями. Заметим, что согласно рис. 4, *а* единичные образы от нейронов *1*, *2* не могут связаться друг с другом с заданным уровнем. Поперечное сечение *7* расходящегося ЕО от нейрона *1* не покрывает нейрон *5*, в направлении которого от нейрона *2* передается ЕО. Аналогично можно сказать и о поперечном сечении *8*, которое не покрывает нейрон *6*. Однако в соответствии с рис. 4, *б* расходящиеся единичные образы от нейронов *1*, *2* ассоциируются (связываются) успешно и в дальнейшем это позволяет извлекать их из памяти друг другом.

Изменять ориентацию распределения плотности мощности в поперечных сечениях расходящихся единичных образов (поворачивать эти образы) вокруг направлений их передачи с учетом текущих состояний слоев можно в зависимости от числа ассоциативно вызываемых из памяти сети ЕО.

Задача управления в этом случае может быть сформулирована в следующем виде. Требуется на момент t_k найти целесообразное значение угла ψ_o^k поворота расходящихся единичных образов вокруг направлений их передачи в сети, при котором будет достигаться $F_o^k(\psi_o^k)$ — максимум числа ассоциативно вызываемых из ее долговременной памяти ЕО на интервале *T*:

$$F_o^k(\psi_o^k) = \max_{z \in Z} \sum_{r=1}^{T} F_{zr}^k(\psi_{zr}^k),$$

где $F_{zr}^k(\psi_{zr}^k)$ — число единичных образов, которые могут быть ассоциативно вызваны из памяти сети на *r*-м такте при *z*-м значении угла ψ_{zr}^k поворота.

Если учесть инерционность процесса ассоциативного вызова информации из памяти сети, то решение этой задачи можно свести к заблаговременному поиску угла $\psi_o^{k-z_0}$, при котором достигается максимум числа $F_o^{k-z_0}(\psi_o^{k-z_0})$ ассоциативно вызываемых из долговременной памяти сети единичных образов на одном из $k - z$ шагов:

$$F_o^{k-z_0}(\psi_o^{k-z_0}) = \max_{z \in Z} F_{zr}^{k-z}(\psi_{zr}^{k-z}). \qquad (4)$$

Поиск такого угла $\psi_o^{k-z_0}$ согласно (4) предусматривает варьирование возможными значениями углов для оптимальной настройки на ассоциативный вызов информации из памяти сети.

Примером предварительно заданных жестких правил выступают повороты единичных образов на заданные углы тогда, когда единичные образы формируются нейронами заданных полей, строк или столбцов нейронов в слоях сети.

Один из примеров поворота расходящихся ЕО (векторов ориентации распределения плотности мощности в их поперечных сечениях) с использованием жестких правил приведен на рис. 5, где *1* — уменьшенные формы поперечных сечений расходящихся единичных образов, формируемых нейронами, входящими в логические поля, на которые разбиваются слои сети (неуменьшенные формы этих сечений накрывают сразу несколько логических полей слоев сети); *2* — направления ориентации распределений плотности мощности в поперечных сечениях расходящихся ЕО. Для каждой строки слоя задается свое направление этой ориентации.

При ориентировании распределения плотности мощности в поперечных сечениях расходящихся ЕО, которые в ближайшее время могут покинуть сеть, в направлении центра сети, можно усилить циклический вызов одних сигналов другими из ее памяти. Если ориентировать это распределение на выход сети, циклический вызов будет сведен к минимуму, однако при этом расширяются возможности по извлечению из памяти сети ЕО, которые должны предшествовать обрабатываемым сигналам.

В случае изменения рассматриваемой ориентации расходящихся ЕО, относящихся к одному типу сигналов, в отношении других сигналов, предоставляется возможность осуществлять переходы от решения одной задачи к другой. Например, один и

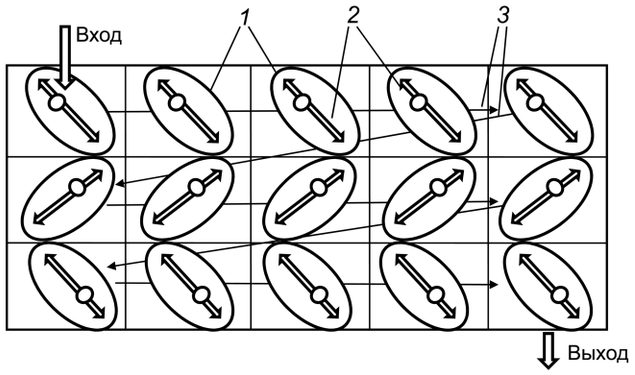


Рис. 5. Пример направлений ориентации распределений плотности мощности в поперечных сечениях расходящихся единичных образов, привязанных к логическим полям слоев сети

тот же сигнал, обрабатываемый в сети, несущий информацию об оптическом поле наблюдаемого объекта, может быть ассоциирован в разное время с акустическими сигналами от него, а также с сигналами о других объектах, связанных с первым. Рассмотрим случай, когда отсутствует возможность управления направлением ассоциативного взаимодействия сигналов в сети. Для него при ассоциативном вызове информации наблюдаются "распыление" энергии вызова и попытки одновременного извлечения из памяти сети всех сигналов, связанных с обрабатываемым воздействием. Это влечет за собой существенно ограниченные возможности памяти сети и может приводить к "зашумлению" сигналов друг другом. В другом случае, при наличии рассматриваемого управления достижимы сосредоточение энергии вызова на конкретных направлениях и областях слоев и последовательный вызов из памяти различных ассоциированных сигналов. Например, сначала вызывается запомненная акустическая информация об интересующем объекте, а затем информация о связанных с ним объектах или наоборот.

Результаты моделирования и их обсуждение

В интересах подтверждения наличия таких возможностей осуществлялось математическое моделирование. Была разработана программная модель двухслойной рекуррентной нейронной сети с изменяемой направленностью и направлениями ассоциативного взаимодействия обрабатываемых сигналов за счет управления синапсами. Каждый слой сети состоял из 1050 нейронов и разбивался за счет пространственных сдвигов передаваемых совокупностей единичных образов на одну строку, содержащую 25 одинаковых полей, размером 6 × 7 нейронов. Сигналы в сеть вводились через первое поле, а снимались с последнего поля. Совокупности единичных образов продвигались вдоль слоев слева направо.

Приведем описание одного из простых экспериментов с использованием этой сети. На вход сети, реализующей изначально круговую ориентацию распределения плотности мощности в поперечных сечениях расходящихся единичных образов, подавались обучающие выборки. Перед началом обучения сеть всегда обнулялась. Эти выборки состояли из последовательных совокупностей единичных образов, несущих информацию о буквах, составляющих слова "work", "worker", "network", записанных в обратном порядке. Каждый раз после обучения сети словам "work", "worker", "network" в нее вводилось слово "work" и анализировались результаты его обработки при различных направлениях ориентации распределения плотности мощности в поперечных сечениях расходящихся ЕО.

На рис. 6 приведены результаты обработки. Рис. 6, *а* соответствует случаю завершения последовательного ввода в сеть совокупностей, содержащих буквы, составляющие слово "work" в обратном порядке записи. При неизмененной круговой ориентации рассматриваемого распределения наблюдались попытки вызова словом "work" из памяти сети недостающих частей "net" и "er" до слова "networker". Когда ориентация эллиптического распределения составляла 90°, вызывалось слово "network" в виде "krowten" (рис. 6, *б*). При ориентировании этого распределения на 180° слово "work" наблюдалось на выходе сети неизмененным (рис. 6, *в*). В случае ориентации эллиптического распределения на 270° из памяти сети вызывалось слово "worker" в виде "rekrow" (рис. 6, *г*). Из этого примера видно, что при одном и том же обучении сети в зависимости от поворота вектора ориентации распределения плотности мощности в поперечных сечениях расходящихся единичных образов из памяти вызываются различные сигналы.

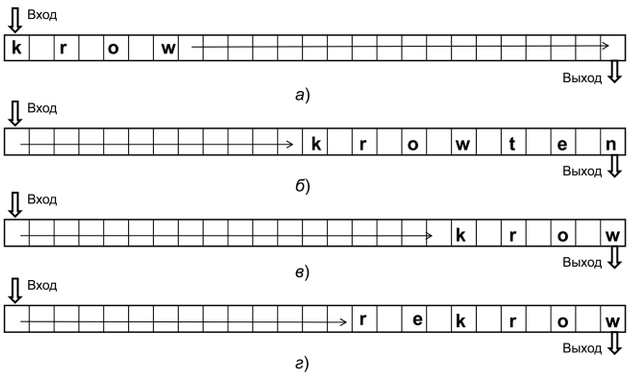


Рис. 6. Состояния первого слоя сети:
а — после обучения при завершении ввода в сеть слова "work"; *б* — перед "work" из долговременной памяти сети вызвано "net", сформировано слово "network"; *в* — на выходе сети только введенное в нее слово "work"; *г* — за "work" вызвано "er", сформировано слово "worker"

В результате установлено, что осуществление предлагаемых поворотов влечет изменение не только направлений ассоциативного запоминания и вызова сигналов из памяти сети, но и вида, порядка вызываемых сигналов, их объема. При этом структура сигналов в нейронной сети при осуществлении рассматриваемых поворотов не разрушается.

Заключение

Предложенный метод управления синапсами в рекуррентной нейронной сети позволяет существенно расширить ее возможности по обработке информации. За счет этого метода предоставляются возможности избирательного ассоциативного запоминания сигналов на элементах сети и избирательного вызова их из памяти в зависимости от текущих состояний слоев. Если в традиционных цифровых процессорах предусматривается обращение к памяти по адресу, а в известных нейронных сетях — по содержанию сигналов, то в предлагаемой сети

в какой-то мере присутствует комбинация этих обращений. Только вместо конкретного адреса выступает направление и ширина формы ассоциативного взаимодействия сигналов. Наличие таких возможностей у нейронной сети позволяет в широких пределах изменять порядок вызываемых из памяти связанных сигналов, переходить от одних ассоциаций к другим, исключить одновременный вызов альтернативных сигналов. При этом расширяются возможности самой памяти.

Метод может быть применен при создании перспективных ассоциативных интеллектуальных машин и систем.

Список литературы

1. **Galushkin A. I.** Neural Networks Theory. Berlin—Heidelberg: Springer—Verlag, 2007.
2. **Dreyfus G.** Neural Networks. Methodology and Applications. Berlin: Springer — Verlag Berlin Heidelberg, 2005.
3. **Franco L., Elizondo D., Jeres J.** (Eds). Constructive Neural Networks. Berlin: Springer — Verlag, 2009.
4. **Huajin Tang, Kay Chen Tan, Zhang Yi.** Neural Networks: Computational Models and Applications. Berlin: Springer — Verlag, 2007.
5. **Хайкин С.** Нейронные сети: полный курс, 2-е изд.: пер. с англ. М.: Вильямс, 2006. 1103 с.
6. **Осовский С.** Нейронные сети для обработки информации: пер. с пол. М.: Финансы и статистика, 2002. 344 с.
7. **Осипов В. Ю.** Рекуррентная нейронная сеть с управляемыми синапсами // Информационные технологии. 2010. № 7. С. 43—47.
8. **Осипов В. Ю.** Оптимизация ассоциативных интеллектуальных систем // Мехатроника, автоматизация, управление. 2011. № 3. С. 35—39.
9. **Осипов В. Ю.** Устойчивость рекуррентных нейронных сетей с управляемыми синапсами // Информационные технологии. 2011. № 9. С. 69—73.
10. **Осипов В. Ю.** Стирание устаревшей информации в ассоциативных интеллектуальных системах // Мехатроника, автоматизация, управление. 2012. № 3. С. 16—20.
11. **Осипов В. Ю.** Метод настройки ассоциативной интеллектуальной системы на входные сигналы // Информационные технологии. 2012. № 9. С. 54—59.

УДК 004.891

А. А. Рындин, д-р техн. наук, проф.,
e-mail: sapris@mail.ru,
В. П. Ульев, соискатель,
e-mail: u_vitalii@mail.ru,
Воронежский государственный
технический университет

Принципы функционирования и оптимизации нейронных сетей прямого распространения большой размерности

Рассматривается задача оптимизации скорости обучения и функционирования нейронных сетей прямого распространения большой размерности. Предложена методика расчета реакции нейронной сети с применением принципов параллелизма, а также рассмотрены способы оптимизации алгоритма обучения обратного распространения ошибки Румельхарта—Хинтона—Вильямса.

Ключевые слова: *нейронная сеть, адаптивная коррекция веса, параллелизм в нейросетях*

Исследование аппроксимационной способности технологии нейронных сетей проводится в рамках диссертационной работы по изучению возможностей управления процессами кредитования юридических лиц на основе разработки скоринговых моделей оценки. Инструментом управления процессом банковского кредитования выступает скоринговая система, позволяющая на основе моделей оценки, накопленной статистики кредитования юридических лиц и других факторов финансовой эффективности организации и экономического состояния региона прогнозировать кредитоспособность потенциальных заемщиков. Использование технологии нейронных сетей в составе гибридной скоринговой системы, сочетающей методики нечетких множеств и генетических алгоритмов, позволяет эффективно решать основную задачу категорирования и прогнозирования кредитоспособности потенциальных заемщиков.

Предметом исследования данной статьи является оптимизация скорости обучения и реакции многослойной нейронной сети прямого распространения. Выбор вида нейронной сети осуществляется исходя из условия задачи исследования: разработки скоринговой модели оценки кредитоспособности юридических лиц, где итоговая оценка является интегральным показателем от совокупности ряда финансовых и экономических критериев функционирования организации. Другими словами, скоринговая модель есть не что иное, как попытка описания сложной нелинейной зависимости между финансовыми и экономическими показателями предприятия и его уровнем кредитоспособности, являясь при этом инструментом прогнозирования финансовой стабильности организации на ближайшее будущее. Эксперименты с размерностью нейронной сети прямого распространения сигнала в части числа скрытых слоев и числа нейронов в них позволяют предположить, что с увеличением размерности сети появляется возможность более точной аппроксимации исходной нелинейной функции зависимости финансовых показателей и уровня кредитоспособности. Проведение подобных экспе-

риментов затруднено временным фактором обучения и реакции нейронной сети большой размерности в силу ограниченности вычислительных мощностей современной техники при применении классических алгоритмов функционирования и обучения нейронных сетей прямого распространения. В статье предложены принципы применения технологии параллельных вычислений и методики оптимизации алгоритма обучения Румельхарта—Хинтона—Вильямса.

Рассмотрим классический пример многослойной нейронной сети прямого распространения с обучением по методике Румельхарта—Хинтона—Вильямса — алгоритм обратного распространения ошибки (Back propagation).

Алгоритм Румельхарта—Хинтона—Вильямса относится к алгоритмам обучения с учителем. Для обучения сети, так же как и для однослойного персептрона, необходимо иметь множество пар векторов $\{xs, ds\}$, $s = 1...S$, где $\{xs\} = \{x1, ..., xs\}$ — множество входных векторов x; $\{ds\} = \{d1, ..., ds\}$ — множество эталонов выходных векторов. Совокупность пар $\{xs, ds\}$ образуют обучающее множество. Число элементов S в обучающем множестве должно быть достаточным для обучения сети, чтобы под управлением алгоритма сформировать набор параметров сети, дающий нужное отображение $x \to y$. Ошибкой сети можно считать $Es = \|ds - ys\|$ для каждой пары (xs, ds). Суть алгоритма обучения сводится к минимизации суммарной квадратичной ошибки, которая имеет следующий вид:

$$E = \frac{1}{2}\sum_{j}\sum_{S}(y_j^s - d_j^s)^2, \qquad (1)$$

где j — число нейронов в выходном слое.

Таким образом, если считать обучающее множество S заданным, то ошибка сети зависит только от вектора параметров: $E = E(P)$. При обучении на каждой итерации корректируются параметры сети в направлении антиградиента E:

$$\Delta P = -\varepsilon \nabla E(p). \qquad (2)$$

В теории оптимизации доказано, что данный подход обеспечивает сходимость к одному из локальных минимумов функции ошибки при условии правильного выбора $\varepsilon > 0$ на каждой итерации. Такой метод оптимизации называется методом наискорейшего спуска. Коррекции параметров сети необходимо рассчитывать на каждой итерации. Поэтому каждая итерация требует расчета компонент градиента и выбора оптимального шага. Алгоритм обратного распространения ошибки — способ расчета компонент градиента. Идея метода в том, чтобы представить E в виде сложной функции и последовательно рассчитать частные производные по формуле для сложной функции. Алгоритм обратного распространения разбивается на два этапа. На первом этапе на вход сети подается некоторый входной вектор из обучающего множества, выполняется расчет выходов нейронной сети. На втором этапе подсчитывается ошибка δ для каждого выхода сети и начинается ее обратное распространение от выходного слоя к входному, учитывая предположение, что связь с большим весовым коэффициентом вносит большую долю ошибки:

$$\Delta W_{ij} = -\varepsilon\left(\frac{\partial E}{\partial W_{ij}}\right), \qquad (3)$$

$$W'_{ij} = W_{ij} + \Delta W_{ij}, \qquad (4)$$

где W_{ij} — весовой коэффициент связи между i-м и j-м нейронами.

Если в процессе обучения наступает момент, когда ошибка в сети попадает в рамки допустимых значений, говорят, что наблюдается сходимость алгоритма обучения.

К наиболее значимым недостаткам алгоритма обратного распространения ошибки можно отнести следующие:

- наличие локальных минимумов функции или гиперплоскости ошибки, в которых возможно зацикливание алгоритма;
- большое число итераций обучения, требуемое для достижения приемлемых значений ошибки сети.

Общий порядок алгоритма обучения с учителем следующий: получить реакцию сети, провести коррекцию сети, получить реакцию сети.

Расчет реакции сети, коррекции ошибки классически выполняется последовательным образом, переходя от слоя к слою. Так, например, при расчете выхода нейронной сети необходимо последовательно от слоя к слою вычислить выходные значения каждого нейрона. Скорость вычисления выхода конкретного нейрона зависит от времени сбора информации о числе входных сигналов и их значениях. Если пренебречь этим моментом, то можно определить время расчета реакции нейрона как постоянную величину τ, зависящую только от вычислительной мощности процессора компьютера. Тогда при последовательном расчете реакции нейронной сети, состоящей из n нейронов (рис. 1), потребуется время

$$T = \sum_{1}^{n}\tau, \qquad (5)$$

где T — общее время реакции сети; n — число нейронов сети; τ — время реакции одного нейрона.

Очевидно, что увеличение масштаба сети приводит к существенному увеличению времени ее реакции.

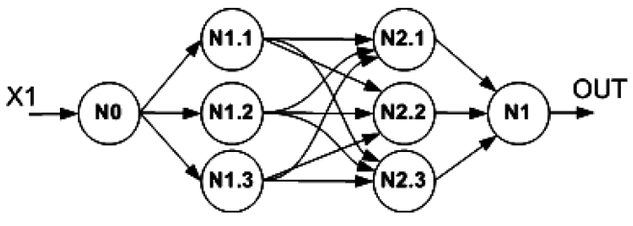


Рис. 1. Многослойная нейронная сеть прямого распространения

Биологический прототип — нервная клетка, функционирует непрерывно, формируя выходной сигнал по факту имеющихся входных сигналов, при условии достижения некоторого порога срабатывания. Применение параллельных вычислений позволяет более качественно реализовать природную модель, а значит получить лучшие результаты. При этом параллельные вычисления, в отличие от последовательных, более сложны в понимании и требуют механизмов синхронизации.

При реализации алгоритмов параллельных вычислений в нейронных сетях можно выделить две основные концепции:

- каждый нейрон сети имеет свой, независимый от других, цикл жизни;
- нейроны объединены в группы, внутри которых имеют независимый цикл жизни; переход между группами осуществляется последовательно.

Для многослойных нейронных сетей прямого распространения очевидно использовать концепцию параллельных вычислений в группе, где под группой нейронов можно понимать слой. Изменение входного сигнала в нейросети должно вызвать каскады реакций связанных нейронов последовательно по слоям, где внутри слоя реакция каждого нейрона рассчитывается параллельно. Учитывая данный подход и современные достижения в микропроцессорной инженерии, общее время реакции слоя нейронов, очевидно, будет несоизмеримо меньшим, чем при последовательных вычислениях. Многоядерность и многопоточность современных микропроцессоров позволяют получить многократно большее процессорное время за единицу времени.

Расчет времени реакции нейросети в общем виде при параллельных вычислениях и последовательном переборе слоев будет при условии округления в большую сторону $round\left(\frac{N_i}{P}\right)$ следующим:

$$T = abs\left(round\left(\frac{N_1}{P}\right)\right)\tau + abs\left(round\left(\frac{N_2}{P}\right)\right)\tau + \ldots$$

$$\ldots + abs\left(round\left(\frac{N_k}{P}\right)\right)\tau, \qquad (6)$$

где T — общее время реакции сети; k — число слоев нейросети; N_i — число нейронов в слое i; τ — время реакции одного нейрона; P — число параллельных потоков микропроцессора ЭВМ.

Ориентировочный анализ времени расчета реакции нейросети, представленной на рис. 1, с использованием технологии параллельных вычислений будет следующим. При расчете реакции на микропроцессоре Intel Core *i*-7 с четырьмя ядрами, где каждое ядро реализует по два параллельных вычислительных потока, может быть одновременно использовано до восьми параллельных потоков. Представленная нейросеть имеет два скрытых слоя и один выходной, общее число нейронов — семь, не считая входного вектора. Общее время реакции сети с использованием параллельных вычислений составит 3τ. При однопоточном последовательном расчете время реакции нейросети — 7τ. Время реакции нейросети будет сокращено в 2,3 раза.

При масштабировании нейросети в 2 раза, т. е. до четырех скрытых слоев, где в каждом слое по 6 нейронов, время реакции нейросети с использованием параллельных вычислений составит 5τ, с использованием последовательного вычисления — 25τ. Время реакции нейросети будет сокращено в 5 раз.

На рис. 2 приведены оценочные расчеты времени реакции многослойных нейросетей разной размерности.

Далее рассмотрим основные подходы обеспечения и ускорения сходимости для нейронных сетей прямого распространения с обратным распространением ошибки.

1. *Оптимизация выбора начальных весов.* Цель состоит в определении таких начальных значений весов, при которых начальное значение ошибки минимально. Классический подход состоит в случайном выборе малых значений для всех весов.

2. *Упорядочение данных.* Чтобы обучение не двигалось в ложном направлении при обработке задачи классификации или распознавания, но не задачи аппроксимирования временных рядов, данные нужно перемешивать случайным образом. Иначе есть вероятность, что нейросеть "выучит" последовательность случайно оказавшихся рядом значений как истинное правило, и потом будет делать ошибку.

3. *Управление значением шага коррекции веса.* По сути, шаг коррекции веса — мера точности и скорости обучения сети. Показатель точности обратно пропорционален показателю скорости обучения. При этом следует отметить следующий важный момент: при увеличении шага коррекции увеличивается скорость обучения, но ошибка может не снизиться до требуемого уровня за счет того, что сеть просто пропустит убывание функции ошибки. При снижении шага коррекции повышается точность подстройки сети, тем не менее алгоритм спуска по поверхности ошибки может попасть в локальный минимум и низкого значения шага коррекции просто не хватит, чтобы его преодолеть.

4. *Оптимизация топологии сети.* Цель данного подхода — определение оптимальной топологии сети, обеспечивающей лучшую сходимость. При

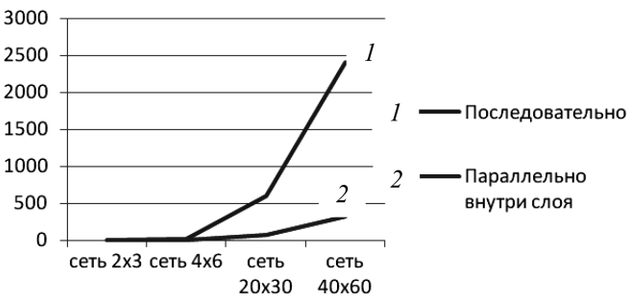


Рис. 2. Время реакции многослойных нейросетей прямого распространения

этом следует отметить два основных подхода: деструктивный и конструктивные методы, реализующие удаление или добавление элементов нейросети соответственно.

В результате исследования сходимости обучения многослойной нейронной сети прямого распространения определено, что наиболее существенное влияние на процесс обучения имеет коэффициент коррекции веса. В настоящее время известны такие оптимизационные методики, как обучение по расписанию (увеличение шага коррекции с ростом итераций обучения) и применение импульса, определяющего вектор смещения веса с предыдущего шага итерации. При этом можно отметить следующие недостатки: в первом случае изменение коэффициента коррекции по времени не учитывает реального значения функции ошибки, во втором случае использование импульса позволяет подстраивать шаг коррекции под конкретный вес в зависимости от его предыдущего состояния, что в целом может привести к параличу нейросети.

Таким образом, методика адаптивной коррекции веса с обратной связью является попыткой компенсирования описанных выше недостатков. Суть методики заключается в том, что изменение шага коррекции сети учитывает состояние ошибки ее выхода, тем самым реализуя обратную связь, обеспечивающую увеличение точности подстройки сети при снижении ошибки выхода нейросети и наоборот. При этом обеспечивается успешное прохождение локальных минимумов функции ошибки в диапазоне неприемлемых, высоких значений ошибки вывода нейронной сети за счет большого шага коррекции.

На каждой итерации рассчитывается новое значение шага коррекции весовых коэффициентов как максимальная доля абсолютного отклонения выхода нейросети от его эталонного значения:

$$ERR_k = \frac{abs(d_k - y_k)}{d_k}, \qquad (7)$$

$$\varepsilon = ERR_{\max} = \max(ERR_1 ... ERR_k). \qquad (8)$$

В результате сравнения алгоритмов обратного распространения ошибки со статическим шагом коррекции и адаптивной коррекцией веса с обратной связью получены следующие практические результаты. Смоделирована нейронная сеть прямого распространения с одним нейроном во входном слое, десятью нейронами в скрытом слое и одним нейроном в выходном слое. Сформирована обучающая выборка на основе функции синуса четверти периода. Адаптивный метод коррекции веса с обратной связью показал существенное увеличение скорости обучения нейронной сети более чем в 30 раз до достижения заданного порога приемлемой ошибки.

Обеспечение оптимального выбора начальных весов позволяет еще до запуска процедуры обучения нейросети предопределить время настройки. Классический подход состоит в случайном выборе малых значений для всех весов. Данный подход в полной мере оправдан для связей входных векторов или нейронов, их реализующих, с первым скрытым слоем. При проведении практических испытаний нейронной сети прямого распространения с обратным распространением ошибки и сигмоидальной функцией активации отмечена тенденция возрастания масштаба изменения весовых коэффициентов от первого слоя к последнему.

Суть предлагаемой методики первичной оптимизации весовых коэффициентов заключается в их последовательном усилении от слоя к слою на основе начальной случайной инициализации, тем самым реализуя некоторое приближение начального распределения весовых коэффициентов к обученному состоянию нейросети:

$$w_{ij} = w_{ij} \times k \times laynum, \qquad (9)$$

где w_{ij} — весовой коэффициент связи i-го и j-го нейрона; k — коэффициент масштабирования; $laynum$ — порядковый номер слоя j-го нейрона.

Таким образом, значения весовых коэффициентов, изначально инициализированные случайным образом малыми величинами, увеличиваются в соответствии с порядковым номером слоя. В качестве универсализации данного правила, в качестве параметра *laynum* может выступать порядковый номер связи в топологической модели прохождения сигнала от входа к выходу сети.

Полученные практические результаты показывают увеличение скорости обучения по методике обратного распространения ошибки от 2 до 10 раз, что может быть объяснимо уменьшением числа итераций корректировки масштабированных начальных весовых коэффициентов относительно их базовых малых значений. В качестве недостатка данного подхода следует отметить преднамеренную ориентацию нейросети, при которой алгоритм градиентного спуска окажется в не очень удачной области поверхности ошибки и не сможет достичь приемлемых результатов.

Таким образом, применение предложенных методик оптимизации скорости обучения нейронной сети с использованием концепции параллельных вычислений позволит существенным образом сократить не только время обучения нейронной сети, но и время ее реакции, что дает возможность перейти к изучению свойств нейронных сетей большой размерности.

Список литературы

1. **Уоссермен Ф.** Нейрокомпьютерная техника: теория и практика. М.: Мир, 1992. 240 с.

2. **Горбань А. Н.** Обучение нейронных сетей. М.: Изд. СССР—США СП "Параграф", 1990. 160 с.

3. **Барцев С. И., Охонин В. А.** Адаптивные сети обработки информации. Красноярск: Ин-т физики СО АН СССР, 1986. Препринт № 59Б. 20 с.

4. **Хайкин С.** Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. М.: Вильямс, 2006. 1104 с.

5. **Rumelhart D. E., McClelland J. L.** Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Cambridge, MA: MIT Press, 1986.

ПИСЬМО В РЕДАКЦИЮ

В последнее время поистине глобальный характер приобрела проблема плагиата в научных исследованиях. С ней все чаще сталкиваются научные журналы. В редакцию нашего журнала поступило письмо, касающееся этой проблемы. Приглашаем заинтересованных специалистов выразить свое мнение по поводу данного письма.

Редакция журнала

Российскую науку в настоящее время сотрясает полоса скандалов по поводу плагиата и мошенничества при выполнении НИР, защите кандидатских и докторских диссертаций. Отрадно, что вновь назначенный руководитель ВАК В. М. Филиппов в своих публичных выступлениях сделал акцент на усиление борьбы с плагиатом в научной сфере на государственном уровне, заметив, что технические науки наименее подвержены этому злу. Однако в данном письме речь пойдет об использовании авторитетного технического журнала из перечня ВАК для изложения результатов исследований, авторство на которые принадлежит другим.

В журнале "Информационные технологии" № 2, 2012 г. была опубликована статья "Кодирование информации методом пропуска полупериодов сетевого напряжения в системах освещения", авторы Т. А. Барбасова, Е. В. Вставская, В. И. Константинов из ФГБОУ ВПО Южно-Уральский государственный университет, г. Челябинск.

В данной статье авторами предложен метод кодирования информации пропуском полупериодов напряжения сети. Пропуск полупериодов предлагается организовать с помощью силовых оптоэлектронных реле. При этом авторы отмечают: "Суть метода заключается в том, что **информацию несет количество полупериодов сетевого напряжения, расположенных между двумя пропущенными полупериодами"**.

Уведомляю Вас о том, что содержание данной статьи в части сути и описания предложенного авторами метода нарушает авторские права, охраняемые патентом RU2390933, МПК 7 H04B 3/54, H02J 13/00 "Способ адресной передачи информации по линиям электроснабжения переменного тока", авторы Сапронов А. А., Никуличев А. Ю., Лещенко А. Г. [и др.]; опубл. 27.05.2010, бюл. № 15 (копия патента прилагается), в котором изложено:

"1. Способ адресной передачи информации по линии электроснабжения переменного тока, заключающийся в том, что ... передаваемую информацию кодируют последовательностью символов из заранее заданного алфавита, причем каждому символу соответствует его порядковый номер в алфавите (код символа), ... затем пакет передают по линии электроснабжения, **отличающийся тем,** что в начале пакета выполняют первое прерывание питающего напряжения (маркер начала), затем **значение поля идентификатора приемника кодируют подачей в линию соответствующего ему количества полуволн питающего напряжения,** затем выполняют второе прерывание питающего напряжения (разделитель полей), после которого значение второго поля пакета кодируют подачей в линию соответствующего коду символа количества полуволн питающего напряжения, после которого выполняют следующее прерывание — разделитель полей и далее аналогично передают все остальные поля пакета, в каждом приемнике определяют моменты прерываний питающего напряжения по отсутствию импульсов, поступающих от датчика моментов перехода через ноль..."

Мало того, что авторы, нарушая профессиональную научную этику, в своих публикациях не сочли нужным упомянуть работы других ученых и специалистов в данной научной области, но и открыто стали использовать и присваивать результаты чужого интеллектуального труда. Авторы статьи из Южно-Уральского госуниверситета не могли не знать о трудах своих коллег, так как ранее при подаче заявки на патент 99913, указанный авторами статьи в списке литературы под номером 6, в качестве ближайшего аналога ими был указан патент RU 2338317, МПК 7 H04B3/00 "Способ и устройство передачи и приема информации по линиям распределительных электрических сетей переменного тока", авторы Сапронов А. А., Старченко И. Е., Никуличев А. Ю., опубл. 10.10.2007, бюл. № 28.

Об отсутствии элементарной профессиональной научной этики говорит и тот факт, что в другой работе "Управление выходной мощностью источников света с передачей информации по проводам питающей сети посредством широтной манипуляции", авторы Е. Вставская, В. Константинов, М. Пожидай (журнал "Полупроводниковая светотехника", № 4, 2012) также предлагается метод передачи информации и говорится о том, что "Предложенный метод передачи информации защищен патентом РФ № 99913". В то же время следует обратить внимание, что данный патент (см. литературу к статье) является **лишь патентом на полезную модель.** Как известно, технические решения, относящиеся к способам (методам), не могут быть объектом полезной модели.

Материал, изложенный в настоящем письме, обсуждался научной общественностью ФГБОУ ВПО "Южно-Российский государственный университет экономики и сервиса" и специалистами малого инновационного предприятия "Электронные информационные системы" (патентообладатель). Поступок коллег из Южно-Уральского государственного университета получил публичное осуждение, а патентообладатель (ООО НП "Электронные информационные системы") выразил готовность в судебном порядке отстаивать свои авторские права.

Учитывая вышеизложенное, прошу Вас на страницах журнала опубликовать данное письмо.

С уважением, от имени научно-исследовательского коллектива, проректор по инновационной работе ФГБОУ ВПО "Южно-Российский государственный университет экономики и сервиса" (г. Шахты Ростовской обл.), заведующий кафедрой "Энергетика и безопасность жизнедеятельности", доктор технических наук, профессор

Сапронов Андрей Анатольевич

CONTENTS

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова.* Технический редактор *Е. В. Конова.*

Корректор *Е. В. Комиссарова.*

Сдано в набор 29.04.2013. Подписано в печать 27.06.2013. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT713. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".

105120, г. Москва, ул. Нижняя Сыромятническая, д. 5/7, стр. 2, офис 2.