Моделирование и Анализ Информационных Систем

Том 9 № 2 2002

Ярославский государственный
университет имени П.Г.Демидова

Министерство образования Российской Федерации
Ярославский государственный университет
имени П.Г.Демидова

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ

Том 9 №2 2002

Основан в 1999 г.
Выходит 2 раза в год

Свидетельство о регистрации №019209 от 16.08.99
Государственного Комитета Российской Федерации по печати

Адрес редакции: 150000, Ярославль, ул.Советская, 14
E-mail: mais@uniyar.ac.ru

Научные статьи в журнал принимаются на кафедре ТИ. Статья должна содержать УДК, аннотацию и сопровождаться набором текста в редакторе LaTeX.

СОДЕРЖАНИЕ

Моделирование и анализ информационных систем. Т.9, №2. 2002

Лицензия ЛР №020319 от 30.12.96. Корректор А.А.Аладьева
Подписано в печать 21.09.2002. Формат 60x88$^1/_8$. Печать офсетная.
Усл.печ.л. 9,73. Уч.-изд.л. 5,75. Тираж 100 экз.

Отпечатано на ризографе. Ярославский государственный университет имени П.Г.Демидова, 150 000, Ярославль, ул.Советская, 14

Модел. и анализ информ. систем. Т.**11**, №**2** (2002) 3–11

УДК 519.854.2

Расширение задачи о назначениях

Рублев В.С., Чаплыгина Н.Б.
Ярославский государственный университет
150 000, Ярославль, Советская, 14

Рассматривается естественное соединение классической задачи о назначениях и задачи о равномерном назначении ([1] - [3]).

1. Постановка задачи

В течение планируемого периода из m дней n работников различных профессий должны выполнить определенный объем работ. В каждый k-й день $k \in \overline{1,m}$ они должны произвести s_k работ, задаваемых вектором

$$T^k = (t_1^k, \ldots, t_{s_k}^k) \quad (k \in \overline{1,m}). \tag{1}$$

При этом каждый работник может в этот день выполнить не более одной работы. Возможности выполнения работ работниками на период заданы трехмерной булевой матрицей $R = (R^1, \ldots, R^m)$, где R^k – двумерная булева матрица размерности $n \times s_k$ возможностей на каждый k-й день, элемент которой

$$r_{ij}^k = \begin{cases} 1, \text{ если } i\text{-ый работник может в } k\text{-й день выполнить работу } t_j^k, \\ 0, \text{ если } i\text{-ый работник не имеет такой возможности.} \end{cases}$$

При этом должны быть выполнены условия корректности задачи: назначения на каждый день возможны (классическая задача о назначениях на каждый день разрешима).

Необходимо *равномерно* относительно вектора $H = (h_1, \ldots, h_n)$ *желательного количества работ* распределить работы между работниками, учитывая также возможность задания обязательных назначений при помощи трехмерной булевой матрицы $A = (A^1, \ldots, A^m)$, состоящей из двумерных матриц обязательных назначений на каждый день. Единичный элемент a_{ij}^k матрицы A^k означает обязательное назначение в k-й день i-го работника на выполнение работы t_{ij}^k. Должны выполняться условия:

$$a_{ij}^k \le r_{ij}^k \qquad (i \in \overline{1,n}, k \in \overline{1,m}, j \in \overline{1,s_k}), \tag{2}$$

$$\sum_{i=1}^{n} a_{ij}^k \le 1 \qquad (k \in \overline{1,m}; j \in \overline{1,s_k}) \tag{3}$$

(любая работа не может иметь больше одного обязательного назначения),

$$\sum_{j=1}^{s_k} a_{ij}^k \le 1 \qquad (i \in \overline{1,n}, k \in \overline{1,m}) \tag{4}$$

(любой работник в любой день выполняет не более одной работы).
Заметим, что при отсутствии обязательных назначений матрицу A можно задать нулевой.

Решением задачи без учета требования равномерности распределения работ или *допустимым* решением назовем трехмерную булеву матрицу $X = (X^1, \ldots, X^m)$, состоящую из двумерных матриц назначений на каждый день. При этом для ее элементов x_{ij}^k выполняются условия:

$$0 \le a_{ij}^k \le x_{ij}^k \le r_{ij}^k \le 1 \qquad (i \in \overline{1,n}, k \in \overline{1,m}, j \in \overline{1,s_k}), \tag{5}$$

$$\sum_{i=1}^{n} x_{ij}^k = 1 \qquad (k \in \overline{1,m}; j \in \overline{1,s_k}) \tag{6}$$

(любая работа должна быть выполнена),

$$\sum_{j=1}^{s_k} x_{ij}^k \le 1 \qquad (i \in \overline{1,n}, k \in \overline{1,m}) \tag{7}$$

(любой работник в любой день выполняет не более одной работы).

Допустимому решению X сопоставим вектор $(x_1, \ldots, x_n)$ числа работ каждого работника, где

$$x_i = \sum_{k=1}^{m} \sum_{j=1}^{s_k} x_{ij}^k. \tag{8}$$

Требование равномерного распределения работ относительно вектора H желательного числа работ может быть выражено в виде критерия минимизации среднеквадратичного отклонения вектора $(x_1, \ldots, x_n)$ от вектора H:

$$f_H(X) = \frac{1}{n} \sum_{i=1}^{n} (x_i - h_i)^2 = \frac{1}{n} \sum_{i=1}^{n} (\sum_{k=1}^{m} \sum_{j=1}^{s_k} x_{ij}^k - h_i)^2. \tag{9}$$

Задачу с заданными вектором H, матрицей A и критерием (9) назовем HA-задачей, а решение X, удовлетворяющее этому критерию, назовем *оптимальным* решением этой задачи.

Весьма важным случаем рассматриваемого вектора H желательного числа работ является вектор $H^{\overline{s}}$, все компоненты которого равны среднему числу работ $\overline{s} = \frac{1}{n} \sum_{k=1}^{m} s_k$. В этом случае критерий $f_{H^{\overline{s}}}(X)$ обозначим через $f(X)$, то есть

$$f(X) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{s})^2 = \frac{1}{n} \sum_{i=1}^{n} (\sum_{k=1}^{m} \sum_{j=1}^{s_k} x_{ij}^k - \overline{s})^2. \tag{10}$$

Задачу с этим критерием назовем A-задачей.

Таким образом HA-задачу о назначении работ сформулируем следующим образом:

Заданы план T (1) работ на каждый день, матрица R возможностей каждого работника на каждый день и матрица A обязательных назначений, удовлетворяющие условиям (2)-(4), и вектор H числа желательных назначений. Требуется построить матрицу X, удовлетворяющую условиям (5)-(7) и минимизирующую функционал (9).

Сформулируем также A-задачу о назначении работ:

Заданы план T (1) работ на каждый день, матрица R возможностей каждого работника на каждый день и матрица A обязательных назначений, удовлетворяющие условиям (2)-(4). Требуется построить матрицу X, удовлетворяющую условиям (5)-(7) и минимизирующую функционал (10).

2. Постановка Al-задачи

Для решения общей HA-задачи мы сначала рассмотрим A-задачу и для установления *характеристического* свойства оптимальных решений этой задачи введем вспомогательную *Al-задачу о максимальных назначениях при ограничении на число работ каждого работника*:

Заданы план T (1) работ на каждый день, матрица R возможностей каждого работника и матрица A обязательных назначений, удовлетворяющие условиям (2)-(3), и некоторое ограничение $l > 0$ на число работ каждого работника (в случае обязательных назначений оно может нарушаться). Требуется построить матрицу назначений X, удовлетворяющую условиям (5) и (7), ограничению:

$$\sum_{i=1}^{n} x_{ij}^k \leq 1 \qquad (k \in \overline{1,m}, j \in \overline{1,s_k}) \tag{11}$$

и такую, что $\forall i \in \overline{1,n}$

$$\sum_{k=1}^{m} \sum_{j=1}^{s_k} a_{ij}^k \leq x_i = \sum_{k=1}^{m} \sum_{j=1}^{s_k} x_{ij}^k \leq \max\{\sum_{k=1}^{m} \sum_{j=1}^{s_k} a_{ij}^k, l\} \tag{12}$$

(такое решение будем называть *допустимым*) и которая максимизирует число выполненных работ:

$$\varphi(X) = \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{j=1}^{s_k} x_{ij}^k \to \max. \tag{13}$$

(такое решение будем называть *оптимальным*). Поскольку некоторые назначения являются обязательными, то функционалы

$$\varphi'(X) = \sum_{i=1}^{n} (x_i - a_i) = \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} a_i, \tag{14}$$

где $a_i = \sum_{k=1}^{m} \sum_{j=1}^{s_k} a_{ij}^k$, и $\varphi(X)$ достигают своих максимальных значений на одном и том же решении X.

Al-задача может быть сведена к задаче о наибольшем потоке (см., например, [4]) следующим образом. Построим транспортную (рис. 1) сеть NL, для которой:

- из источника b_0 к вершинам $b_1, \ldots, b_n$, соответствующих работникам, идет n дуг пропускной способности $c_i = max\{l - a_i, 0\}$ $\quad (i \in \overline{1,n})$;
- из каждой вершины $b_i (i \in \overline{1,n})$ идут m дуг пропускной способности 1 к вершинам $b_i^k (k \in \overline{1,m})$, соответствующим i-му работнику в k-й день;

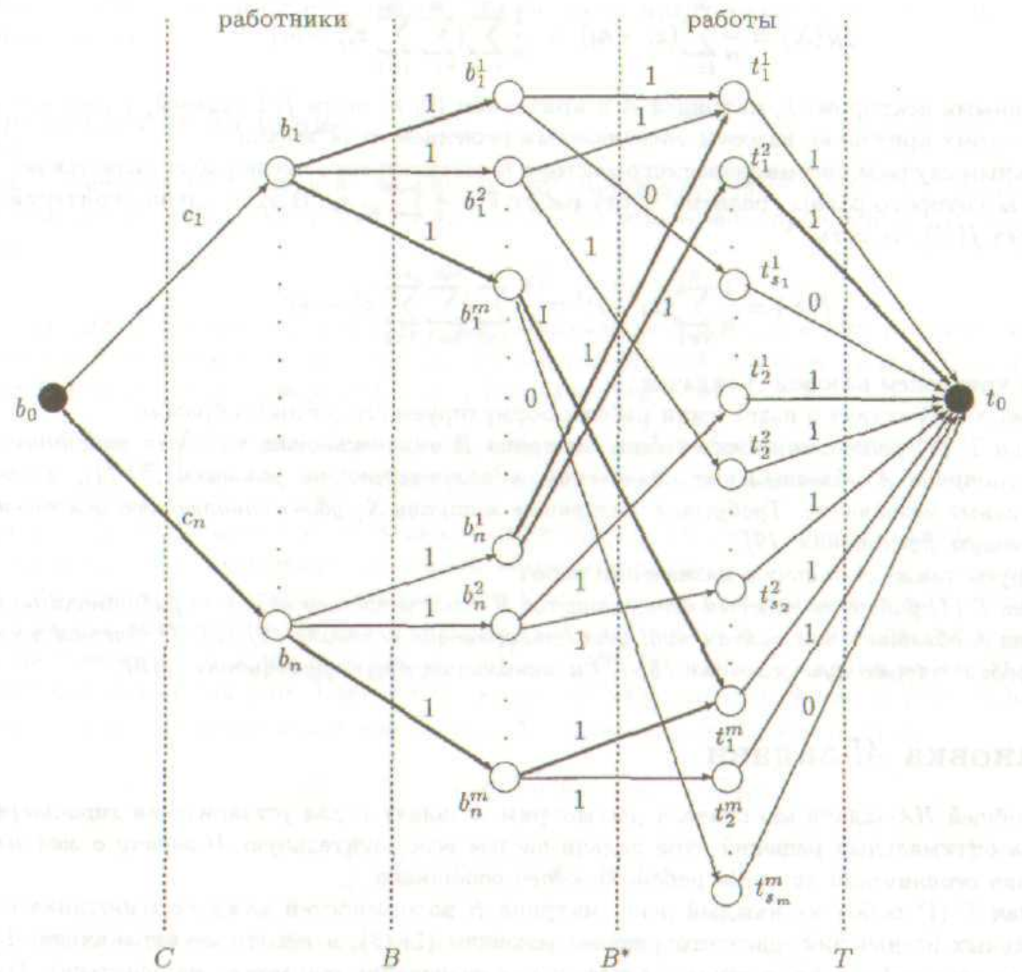

Рис. 1

- из каждой вершины $t_j^k (k \in \overline{1,m}, j \in \overline{1,s_k})$, соответствующей j-й работе в k-й день, в сток t_0 идет дуга пропускной способности 0, если эта работа входит в обязательные назначения, и пропускной способности 1, если она не входит в обязательные назначения;
- из каждой вершины $b_i^k (i \in \overline{1,n}, k \in \overline{1,m})$ в каждую соответствующую по k вершину $t_j^k (k \in \overline{1,m}, j \in \overline{1,s_k})$ идет дуга пропускной способности 0, если $a_{ij}^k = 1$, или пропускной способности 1, если $r_{ij}^k - a_{ij}^k = 1$.

Буквами C, B, B^*, T обозначены определенные разрезы в сети. Любой допустимый поток в заданной сети будет равен по величине потоку, проходящему через каждый из разрезов. Если поток на дугах (b_i^k, t_j^k) разреза B^* обозначить через y_{ij}^k, то в сумме с обязательными назначениями $(X = Y + A)$ он и будет давать допустимое решение задачи и его величина равна значению функционала $\varphi'(X)$. Таким образом надо найти максимальный поток в этой сети. Поставленную задачу можно решить, например, с помощью алгоритма Форда-Фалкерсона.

Построив максимальный поток в данной сети, найдем дополнительные к обязательным назначения работ: это будут те назначения, которым соответствует ненулевой поток на дугах (b_i^k, t_j^k) для $i = \overline{1,n}, k = \overline{1,m}, j = \overline{1,s_k}$, а в сумме с обязательными назначениями они и дадут решение X.

3. Характеристическое свойство оптимального решения A-задачи

Определение: *l-срезкой матрицы X назовем матрицу $X[l]$, полученную следующим образом:*

а) содержит все единичные (равные 1) элементы матрицы A;

б) в каждой i-ой строке матрицы X, содержащей элементов равных 1 более чем $\max\{a_i, l\}$, оставлены равными 1 ровно $\max\{a_i, l\}$ элементов (любых, включая все обязательные назначения), а остальные обнулены;

в) все остальные строки матрицы X оставлены без изменения.

Заметим, что так как

$$a_{ij}^k \le x[l]_{ij}^k \le x_{ij}^k, \tag{15}$$

то

$$x[l]_i = \begin{cases} x_i, & x_i = a_i \quad \text{или} \quad x_i \le l, \\ l, & x_i \ne a_i \quad \text{и} \quad x_i > l. \end{cases} \tag{16}$$

Лемма 1. *Любая l-срезка оптимального решения X A-задачи является оптимальным решением Al-задачи.*

Доказательство. Пусть $X[l]$ – некоторая l-срезка оптимального решения X. Надо показать, что
1) $X[l]$ – допустимое решение Al-задачи, т.е. выполняются условия (5), (7), (11), (12);
2) $X[l]$ – оптимальное решение Al-задачи, т.е. выполняется условие (13).

Из условий (5)-(7) для X и условия (15) следует, что для $X[l]$ выполняются условия (5), (7) и (11). Из (15) и (16) следует выполнение условия (12). Поэтому $X[l]$ допустимое решение Al-задачи. Тогда матрица $Y = X[l] - A$ определяет допустимый поток в сети NL. Решение X может быть представлено в виде выражения над булевыми матрицами, где "+" будем понимать как логическое "или":

$$X = A + Y + (X - X[l]).$$

$X[l]$ оптимально тогда и только тогда, когда поток, определяемый матрицей Y, является максимальным в сети NL.

Предположим противное, поток Y не максимальный, т.е. $X[l]$ не оптимальное решение Al-задачи. Тогда в сети NL согласно алгоритму увеличения потока найдется путь прорыва $P = P(b_0, t_0)$, увеличивающий поток на 1. В общем случае поток P состоит из прямых и обратных дуг и имеет вид, указанный на рис. 2.

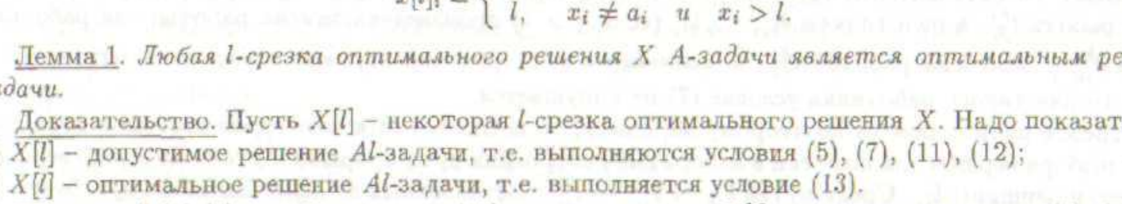

Рис. 2

Например, на рис. 1 этот путь выделен жирными дугами:

$$b_0 \to b_n \to b_n^m \to t_1^m \leftarrow b_1^m \leftarrow b_1 \to b_1^1 \to t_1^1 \leftarrow b_n^1 \to t_2^1 \to t_0.$$

Все дуги такого пути, идущие в направлении от источника к стоку (прямые), имеют 0-поток при единичной пропускной способности, а обратные дуги при той же пропускной способности имеют 1-поток.

Увеличивая поток по данному пути на 1 на прямых дугах и уменьшая его на 1 на обратных дугах, получим новый поток Y^* большей величины.

Рассмотрим матрицу $X^* = A + Y^* + (X - X[l])$. Покажем, что либо она отвечает условиям (5-7) допустимого решения A-задачи, либо ее можно скорректировать так, чтобы она отвечала этим условиям. Затем покажем, что функционал (10) на этом решении имеет значение меньше $f(X)$, что противоречит оптимальности решения X.

Условие (5) – для каждого работника в каждый день не более 1 работы. Если $r_{ij}^k = 0$, то из выполнения условия (5) для X следует $x_{ij}^k = 0, a_{ij}^k = 0$ и из построения сети NL следует $y_{ij}^k = {y^*}_{ij}^k = 0$, что влечет ${x^*}_{ij}^k = 0$. Таким образом условие (5) выполнено.

Условие (7). Необходимо рассмотреть выполнение данного условия только для тех работников, которых коснулись изменения, т.е. для которых соответствующие вершины участвуют в пути P. В этом случае снятие назначения не нарушает условия (7). Потому рассмотрим только новые назначения. Между двумя вершинами-работами отрезок пути P может иметь 2 варианта, которые в общем виде можно представить седующим образом:

1) $t_{j_{h-1}}^{k_{h-1}} \leftarrow b_{i_h}^{k_h = k_{h-1}} \rightarrow t_{j_h}^{k_h} \quad (h \in \overline{2,q})$,

2) $t_{j_{h-1}}^{k_{h-1}} \leftarrow b_{i_h}^{k_{h-1}} \leftarrow b_{i_h} \rightarrow b_{i_h}^{k_h} \rightarrow t_{j_h}^{k_h} \quad (h \in \overline{2,q})$.

Путь P проходит через каждую вершину-работу и вершину-работника 1 раз и все вершины-работы этого пути не имеют обязательного назначения. Работнику b_{i_1}, не имевшему назначения в $X[l]$ в k_1-й день, назначена работа $t_{j_1}^{k_1}$, а работникам $b_{i_2}, \ldots, b_{i_q}$ (если $q > 1$) произведена замена работы: для работника b_{i_h} работа $t_{j_{h-1}}^{k_{h-1}}$ заменена работой $t_{j_h}^{k_h}$, и, если $k_h = k_{h-1}$ (первый вариант построения отрезка пути прорыва), то для такого работника условие (7) не нарушается.

Если отрезок пути строится по второму варианту, то воспользуемся некоторой свободой последовательности выбора вершин для пометки в алгоритме построения пути прорыва и уточним выбор вершины после пометки вершины b_{i_h}. Среди путей $b_{i_h} \rightarrow b_{i_h}^{k_h} \rightarrow t_{j_h}^{k_h}$, составленных из ненасыщенных дуг, будем выбирать такие, что $x_{i_h j_h}^{k_h} = 1$, т.е. работнику b_{i_h} в решении X была назначена работа $t_{j_h}^{k_h}$ и это назначение не попало в срезку (было срезано, т.е. $x_i > l, y_{i_h j_h}^{k_h} = 0$). Тогда в решении X и, следовательно, в любой его l-срезке эта работа не назначена никакому другому работнику, а в срезке $X[l]$ и, соответственно, в Y вообще не назначена, что означает ненасыщенность дуги $(t_{j_h}^{k_h}, t_0)$ и возможность пометки стока t_0 (в алгоритме Форда-Фалкерсона). Тогда вершина $b_{j_h}^{k_h}$ будет последней среди вершин-работников этого пути. В этом случае ${y^*}_{i_h j_h}^{k_h} = x_{i_h j_h}^{k_h} = 1$ и, следовательно, ${x^*}_{i_h j_h}^{k_h} = 1$. Поэтому других назначений работник b_{i_h} не имел ни в X, ни в Y^*, что означает выполнение и для такого работника условия (7).

Если же такого пути не нашлось, то все пути $b_{i_h} \rightarrow b_{i_h}^{k_h} \rightarrow t_{j_h}^{k_h}$, составленные из ненасыщенных дуг, таковы, что $x_{i_h j_h}^{k_h} = 0$. Это означает, что если работник b_{i_h} свободен от работ в k_h-й день в срезке $X[l]$, то он свободен в этот день и в решении X. Тогда

$$\sum_{j=1}^{s_{k_h}} {x^*}_{i_h j}^{k_h} = \sum_{j=1}^{s_{k_h}} {y^*}_{i_h j}^{k_h} = 1,$$

что также не нарушает условия (7).

Условие (6) – каждая работа должна иметь назначение и притом только одно. Если работа не попала в путь прорыва, то ее не коснулись изменения – она остается назначенной одному работнику и в матрице X^*. Работы $t_{j_h}^{k_h} (h \in \overline{1, q-1})$ переназначены, если $q > 1$ (в противном случае этих работ нет в построенном пути), т.е. для них условие (6) выполнено. Работа $t_{j_q}^{k_q}$ не имела назначения в Y (и в $X[l]$) и назначена в $Y^* ({y^*}_{i_q j_q}^{k_q} = 1)$, но поскольку каждая работа должна иметь назначение в решении X, то найдется $x_{i_0 j_q}^{k_q} = 1$. Тогда в случае равенства $i_q = i_0$ работа $t_{j_q}^{k_q}$ имеет единственное назначение X^*.

В противном случае ($i_q \neq i_0$) нарушается выполнение условия (6): ${x^*}_{i_0 j_q}^{k_q} = 1$ и ${x^*}_{i_q j_q}^{k_q} = 1$. В этом случае скорректируем матрицу X^*, заменив значение элемента ${x^*}_{i_0 j_q}^{k_q}$ на нулевое (т.е. сняв одно из назначений этой работы). При этом выполнение условий (5) и (7) не нарушится. Чтобы не вводить нового обозначения, оставим обозначение этой новой матрицы X^*:

$$X^* = A + Y^* + (X - X[l]) - X0,$$

где $X0$ – матрица таких же размеров, что и остальные входящие в выражение матрицы, у которой все элементы нулевые, кроме элемента $x0^{k_q}_{i_0 j_q}$, определяемого так:

$$x0^{k_q}_{i_0 j_q} = \begin{cases} x^{k_q}_{i_0 j_q} = 1, \text{ если } i_q \neq i_0, \\ 0, \text{ если } i_q = i_0 \text{ (корректировка не требуется)}. \end{cases}$$

Таким образом условие (6) для X^* выполнено (а также (5) и (7)).

Значение функционала f. Сравним $f(X)$ и $f(X^*)$:

$$f(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{s})^2 \quad u \quad f(X^*) = \frac{1}{n}\sum_{i=1}^{n}(x^*{}_i - \overline{s})^2,$$

где $\quad X = A + Y + (X - X[l]), \qquad X^* = A + Y^* + (X - X[l]) - X0.$

Если $i \neq i_0, i_1, \ldots, i_q$, то $x_i = x^*{}_i$, так как все назначения работника b_i сохранились. Рассмотрим 2 случая: $q = 1$ и $q > 1$.

Пусть $q = 1$. Тогда $P = (b_0 \to b_{i_1} \to b^{k_1}_{i_1} \to t^{k_1}_{j_1} \to t_0)$. Работа $t^{k_1}_{j_1}$ не имеет назначения в срезке $X[l]$, но имеет назначение в решении $X : x^{k_1}_{i_0 j_1} = 1$. Следовательно $x_{i_0} > l$ (только в этом случае единичный элемент может не попасть в срезку). Если $i_1 = i_0$, то $x_{i_1} > l$. Тогда в срезку попадают не менее l единиц этой строки X. Число единичных элементов в этой строке срезки $a_{i_1} + y_{i_1} \geq l$ или $y_{i_1} \geq l - a_{i_1}$, и это означает невозможность увеличения потока по дуге (b_0, b_{i_1}), которая входит в путь P. Это противоречие доказывает неравенство $i_1 \neq i_0$. Тогда $x^*{}_{i_0} = x_{i_0} - 1$, и поскольку $x_{i_0} > l$, то $x_{i_0} \geq l + 1$. Поскольку произошло увеличение потока, соответствующего срезке в Al-задаче ($x^*{}_{i_1} = x_{i_1} + 1$), то $x_{i_1} < l$ и $x_{i_1} \leq l - 1$. Рассмотрим разность:

$$f(X^*) - f(X) = \frac{1}{n}\sum_{i=1}^{n}(x^*{}_i - \overline{s})^2 - \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{s})^2 =$$

$$= \frac{1}{n}((x^*{}_{i_1} - \overline{s})^2 - (x_{i_1} - \overline{s})^2 + (x^*{}_{i_0} - \overline{s})^2 - (x_{i_0} - \overline{s})^2) =$$

$$= \frac{1}{n}((x_{i_1} + 1 - \overline{s})^2 - (x_{i_1} - \overline{s})^2 + (x_{i_0} - 1 - \overline{s})^2 - (x_{i_0} - \overline{s})^2) =$$

$$= \frac{1}{n}(2(x_{i_1} - \overline{s}) + 1 - (x_{i_0} - \overline{s}) + 1) = \frac{2}{n}(x_{i_1} - x_{i_0} + 1) < 0,$$

так как $x_{i_0} - x_{i_1} \geq 2$. Это противоречит оптимальности решения X.

Пусть теперь $q > 1$. Поскольку произошло увеличение потока по дуге (b_0, b_{i_1}), то в этой строке в решении X число единиц меньше l и потому в ней нет единичных элементов вне срезки. Наличие единичного элемента $x^{k_q}_{i_0 j_q}$ вне срезки говорит о том, что $i_1 \neq i_0$. И так же, как в случае $q = 1$, $x_{i_1} < l$ и $x_{i_1} \leq l - 1$.

Каждому из работников $b_{i_1}, \ldots, b_{i_q}$ (если $q > 1$) произведено по одной замене работы из $t^{k_1}_{j_1}, \ldots, t^{k_{q-1}}_{j_{q-1}}$ на работы $t^{k_2}_{j_2}, \ldots, t^{k_q}_{j_q}$ соответственно. При этом работникам $b_{i_2}, \ldots, b_{i_{q-1}}$ переназначение произведено внутри работ срезки, а работнику b_{i_q} вместо работы срезки назначена работа вне срезки и суммарное число работ в потоках Y и Y^* одинаково для каждого такого рабочего:

$$y^*{}_{i_2} = y_{i_2}, \ldots, y^*{}_{i_q} = y_{i_q}.$$

В строке же i_0 число единиц уменьшилось, так как в этой строке значение элемента $x^{k_q}_{i_0 j_q}$ (который не попал в срезку) равное 1 заменено на 0.

Заметим, что i_0 может оказаться одним из $i_2, \ldots, i_q$. Возможны случаи: 1) $i_0 \notin \{i_2, \ldots, i_q\}$; 2) $i_0 \in \{i_2, \ldots, i_{q-1}\}$; 3) $i_0 = i_q$. В каждом таком случае для решения X в строке i_0 число единиц больше l, так как $x^{k_q}_{i_0 j_q} = 1$, т.е. есть единичные элементы, не попавшие в срезку и $x^*{}_{i_0} = x_{i_0} - 1$. При этом из $x_{i_0} > l$ следует $x_{i_0} \geq l - 1$. Для остальных же строк с номерами, отличными от i_1, число единичных элементов для X и X^* одинаково. Тогда, рассматривая разность $f(X^*) - f(X)$, получаем то же самое отношение, что и в случае $q = 1 : f(X^*) < f(X)$, т.е. противоречие оптимальности решения X. Полученное противоречие доказывает лемму 1.

Лемма 2. *Если любая срезка $X[l]$ допустимого решения X A-задачи является оптимальным решением соответствующей Al-задачи, то решение X – оптимально для A-задачи.*

Доказательство. Докажем, что если 2 допустимых решения X и Y A-задачи обладают свойством, указанным в условии леммы, то они имеют одинаковое количество строк с определенным k количеством единиц *для любого $k \geq 0$. Обозначим n_k^X и n_k^Y – число* строк, содержащих ровно k единиц в X и Y соответственно. Предположим, что $n_0^X > n_0^Y$. Тогда в любой срезке $X[1]$ число единичных элементов будет меньше, чем в любой срезке $Y[1]$. По условию обе срезки – оптимальные решения одной и той же $A1$-задачи, что означает равное число единиц в обоих решениях. Следовательно, $n_0^X = n_0^Y$. Пусть $n_k^X = n_k^Y$ для $k = 0, \ldots, l-1$ и $n_l^X > n_l^Y$. Тогда в любой срезке $X[l+1]$ число единичных элементов будет меньше, чем в любой срезке $Y[l+1]$, чего также быть не может. Поэтому X и Y имеют одинаковое число строк с любым заданным $k \geq 0$ количеством единиц. Обозначим это число n_k. Тогда

$$f(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{s})^2 = \frac{1}{n}\sum_{k=0,1,2,\ldots} n_k(k - \overline{s})^2 = f(Y).$$

И поскольку любое оптимальное решение Z A-задачи обладает свойством, описанным в условии, то $f(X) = f(Z)$ означает оптимальность решения X.

Леммы 1-2 позволяют сформулировать следующий критерий оптимальности решения A-задачи.

Теорема. *Допустимое решение X A-задачи оптимально тогда и только тогда, когда для любого $l > 0$ каждая l-срезка X является оптимальным решением Al-задачи.*

4. Алгоритм решения A-задачи

Из доказанной теоремы вытекает следующий алгоритм решения A-задачи:

1^o. Полагаем $l = 1$ и $X = A$.

2^o. Решаем Al-задачу (алгоритм Форда-Фалкерсона), полагая в качестве начального потока матрицу X, полученную на предыдущем шаге, и получая в качестве решения новую матрицу X, удовлетворяющую условиям (5) - (7).

3^o. *Если $\varphi(X) = \sum_{j=1}^{m} s_j$, то задача решена — конец алгоритма, иначе выполняем шаг 4^o.*

4^o. Увеличиваем l на 1 и переходим к шагу 2^o.

Доказательство. Заметим, что алгоритм увеличения потока (Форда-Фалкерсона) построением цепочки улучшения не уменьшает количество единиц матрицы, определяющей поток, в любой ее строке. Поэтому p-срезки ($p = 1, \ldots, l-1$) матрицы-решения Al-задачи при решении таким методом будут решениями соответствующих Ap-задач. Таким образом в результате применения алгоритма получается матрица X, удовлетворяющая характеристическому свойству, и по теореме она является решением задачи о равномерном назначении работ, что и требовалось доказать.

5. Учет желательного количества назначений (HA-задача)

Полученный результат для A-задачи можно распространить на HA-задачу. Сначала сведем HA-задачу с планом T, ненулевой матрицей обязательных назначений A, матрицей возможностей R и вектором желательного количества назначений H к другой $H'0$-задаче, для которой:

1) матрица обязательных назначений $A' = 0$;

2) матрица возможностей $R' = R - A$;

3) вектор $H' = (h_1 - a_1, \ldots, h_n - a_n)$ желательного количества назначений;

4) план $T' = \{t'^k_j\}$, где $\forall k \in \overline{1,m}, \forall j \in \overline{1,s_k}$

$$t'^k_j = \begin{cases} t^k_j, & \forall i : a^k_{ij} = 0, \\ 0, & \exists i : a^k_{ij} = 1. \end{cases} \tag{17}$$

Лемма 3. *Решение X НА-задачи оптимально тогда и только тогда, когда $X' = X - A$ является оптимальным решением $H0'$-задачи.*

Доказательство. Пусть Y допустимое решение HA-задачи. Тогда $Y' = Y - A$ является допустимым решением $H0'$-задачи. Так как

$$f_H(Y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - a_i - (h_i - a_i))^2 = \frac{1}{n}\sum_{i=1}^{n}(y'_i - h'_i)^2 = f_{H'}(Y'),$$

то из оптимальности одного решения X для HA-задачи и $X - A$ для $H0'$-задачи следует оптимальность другого, что доказывает утверждение леммы.

Пусть теперь имеется $H0$-задача с планом T, нулевой матрицей обязательных назначений, матрицей возможностей R и вектором H желательного количества работ. Обозначим

$$h_{max} = \max_{1\le i\le n} h_i, \quad m^+ = nh_{max} - \sum_{i=1}^{n} h_i$$

и образуем A-задачу (с выровненным вектором желательного количества работ) следующим образом:
1. Расширим план T до T', добавив m^+ работ по одной в каждый дополнительный день от $m+1$ до $m+m^+$. При этом число работ в k-й день

$$s'_k = \begin{cases} s_k, & k \in \overline{1,m}, \\ 1, & k \in \overline{m+1, m+m^+}. \end{cases} \tag{18}$$

2. Добавим к матрице R m^+ единичных столбцов справа.
3. Определим матрицу A обязательных назначений

$$\forall i \in \overline{1,n} \quad a^k_{ij} = \begin{cases} 1, & j = 1, \quad k \in K_i, \\ 0, & j > 1 \quad \textit{или} \quad k \notin K_i, \end{cases} \tag{19}$$

где через K_i обозначено множество индексов

$$K_i = \{m + (i-1)h_{max} - \sum_{p=1}^{i-1} h_p + 1, \dots, m + ih_{max} - \sum_{p=1}^{i} h_p\},$$

определяющих $h_{max} - h_i$ столбцов, идущих подряд.

Лемма 4. *Решение X $H0$-задачи оптимально тогда и только тогда, когда оптимально решение $X' = X + A$ A-задачи.*

Доказательство. Заметим, что $a_i = h_{max} - h_i$. Пусть Y допустимое решение $H0$-задачи. Тогда $Y' = Y + A$ является допустимым решением A-задачи. Так как

$$f_H(Y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i + a_i - (h_i + a_i))^2 =$$

$$= \frac{1}{n}\sum_{i=1}^{n}(y'_i - h_{max})^2 = \frac{1}{n}\sum_{i=1}^{n}(y'_i - \overline{s'} + \overline{s'} - h_{max})^2 =$$

$$= \frac{1}{n}\sum_{i=1}^{n}(y'_i - \overline{s'})^2 + \frac{1}{n}\sum_{i=1}^{n}(\overline{s'} - h_{max})^2 + \frac{2}{n}\sum_{i=1}^{n}((\overline{s'} - h_{max})(y'_i - \overline{s'})) = f(Y') + (\overline{s'} - h_{max})^2,$$

то соответствующие друг другу решения обеих задач различаются на одну и ту же константу, не зависящую от решения. Поэтому, если решение одной из этих задач оптимально, то и соответствующее решение другой задачи также оптимально, что и доказывает лемму.

Леммы 3 и 4 дают основание сформулировать следующий алгоритм построения оптимального решения HA-задачи:

1°. Переходим к $H0'$-задаче с данными $H' = (h_1 - a_1, \dots, h_n - a_n)$, $\quad R' = R - A$ и планом T', определяемым формулой (17).

2°. Полагаем $h'_{max} = \max_{1 \le i \le n} h'_i$, $m^+ = nh'_{max} - \sum_{i=1}^{n} h'_i$ и добавляем m^+ нулевых столбцов справа в матрицу R'; план T' расширяем, добавив в него m^+ работ по одной в каждый дополнительный день от $m+1$ до $m+m^+$; определяем матрицу A' обязательных назначений по формуле аналогичной (19) и переопределяем матрицу R' возможностей, полагая $R' := R' + A$.

3°. Решаем полученную A'-задачу для данных A', R', T', используя вышеописанный алгоритм, и получаем ее оптимальное решение X'.

4°. Удаляем из решения X' последние m^+ столбцов и определяем для HA-задачи оптимальное решение X, полагая $X = X' + A$.

Литература

[1] Кропанов В. А., Рублев В. С. Равномерное назначение работ минимальной стоимости //Дискретная математика. 2001. Т.13, №4. С.144-156.

[2] Кропанов В. А., Рублев В. С. Задача о равномерном назначении работ и ее обобщения //Моделирование и анализ информационных систем. Ярославль, 2000. Т.7, №2. С.3-12.

[3] Рублев В.С. Задача о равномерном распределении работ /Ярославский госуниверситет. Ярославль. 1986. - Деп.ВИНИТИ №611-В87 26.01.87.

[4] Йенсен П., Барнес Д. Потоковое программирование. М : Радио и связь, 1984.

Модел. и анализ информ. систем. Т.**9**, №**2** (2002) 12–15

УДК 541.1+612.82

Моделирование импульсной активности нейрона с помощью дифференциального уравнения с запаздывающим аргументом с переменной величиной запаздывания

Парамонов И.В.
Ярославский государственный университет
150 000, Ярославль, Советская, 14

В работе описана модель импульсного нейрона, основанная на дифференциальном уравнении с запаздывающим аргументом с переменной величиной запаздывания. Проведен асимптотический анализ модели, его результаты сопоставлены с результатами численного эксперимента, рассмотрен вопрос о существовании периодического решения.

1. Введение

Процессы функционирования нервной системы обусловлены циркуляцией электрических импульсов, генерируемых нейронами. Содержимое тела нейрона заряжено отрицательно относительно внешней среды. На мембране нейрона возникает разность потенциалов, которая меняется во времени в силу ионного обмена через мембрану. В общих чертах процесс генерации импульса (спайка) может быть представлен следующим образом. Первоначально мембрана гиперполяризована и набюдается медленная ее деполяризация. По достижении некоторого значения происходит резкое увеличение скорости деполяризации, разность потенциалов быстро уменьшается, а затем меняет знак. Далее мембрана возвращается в состояние гиперполяризации, и становится возможной генерация следующего импульса нейроном.

Рассматриваемая модель представляет собой модифицированную модель из [1, 2]. Она учитывает изменение разности потенциалов, обусловленное натриевым и калиевым ионным обменом. Так как калиевая проводимость мембраны запаздывает по отношению к натриевой, в уравнение, описывающее нейрон, вводится запаздывание. В отличие от модели в [1, 2] мы считаем это запаздывание равным не константе, а некоторой функции мембранного потенциала.

2. Постановка задачи

Обозначим через $u > 0$ значение потенциала мембраны относительно уровня максимальной ее поляризации и рассмотрим дифференциальное уравнение

$$\dot{u} = \lambda[f_2(u(t - \tau(u))) - f_1(u) - 1] \cdot u. \tag{1}$$

Здесь функции $f_1(u)$ и $f_2(u)$, определяющие соответственно натриевую и калиевую проводимости мембраны, достаточно гладкие, положительные и при $u \to \infty$ стремятся к нулю быстрее, чем $O(u^{-1})$. Пусть $f_{1,2}(0) = R_{1,2} > 0$.

Будем считать, что $\alpha_1 = R_2 - 1 > 0$, $\alpha_2 = R_1 + 1 > 0$, $\alpha = R_2 - R_1 - 1 > 0$. Как мы убедимся в дальнейшем, эти константы будут определять (в экспоненциальной шкале) скорость изменения мембранного потенциала на различных временных промежутках.

Будем считать, что функция $\tau(u)$, определяющая запаздывание калиевой проводимости относительно натриевой, непрерывна. Кроме того, пусть $\tau(0) = 1$ и $\tau(u)$ при $u \to \infty$ монотонно стремится к константе $C > 1$ быстрее, чем $O(u^{-1})$.

Множество начальных функций S определим как множество всех функций $\varphi(s)$, непрерывных на отрезке $[-C, 0]$ и удовлетворяющих условиям: $\varphi(0) = \lambda^{-1}$ и $0 < \varphi(s) \le \lambda^{-1}\exp(\lambda\alpha s/2)$ для всех $s \in [-C, 0]$.

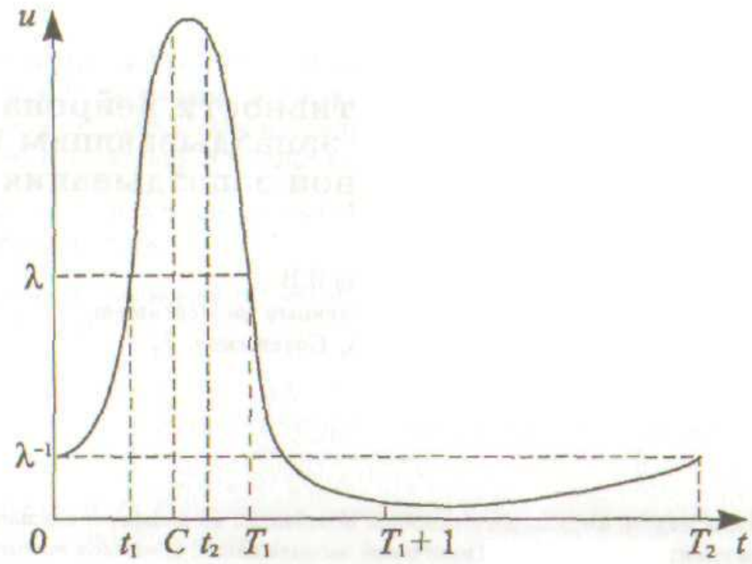

Рис. 1.

3. Аналитическое исследование уравнения нейрона

Аналитическое исследование уравнения (1) заключается в сведении его методом шагов (см., например, [5]) к последовательности обыкновенных дифференциальных уравнений с дальнейшим асимптотическим анализом последних.

Теорема 1. *Если при* $u \in [0, \infty)$ *выполнено условие*

$$R_2 - f_1(u) - 1 > 0, \tag{2}$$

то для любого, сколь угодно малого положительного числа δ*, имеют место следующие асимптотические при* $\lambda \to \infty$ *равенства для решения дифференциального уравнения (1) с начальным условием из множества* S:

$$\begin{aligned}
u(t) &= \exp(\lambda(\alpha_1 t + o(1))), && t \in [\delta, C - \delta],\\
u(t) &= \exp(\lambda(\alpha_1 C - t + C + o(1))), && t \in [C + \delta, T_1 - \delta],\\
u(t) &= \exp(\lambda(-\alpha_2(t - T_1) + o(1))), && t \in [T_1 + \delta, T_1 + 1 - \delta],\\
u(t) &= \exp(\lambda(-\alpha_2 + \alpha(t - T_1 - 1) + o(1))), && t \in [T_1 + 1 + \delta, T_2 - \delta].
\end{aligned}$$

Здесь $T_1 = R_2 C + o(1)$, $T_2 = R_2 C + 1 + \alpha_2/\alpha + o(1)$, *а* $o(1)$ *— величина, стремящаяся к 0 при* $\lambda \to \infty$.

Доказательство.

Пусть $0 \le t \le t_1$, где t_1 находится из условия $u(t_1) = \lambda$. Имеем: $t - \tau(u) \in [-C, 0]$, поэтому $f_2(u(t - \tau(u))) = R_2 + o(1)$. Отсюда

$$f_2(u(t - \tau(u))) - f_1(u) - 1 = R_2 - f_1(u) - 1 + o(1).$$

Так как $f_1(u) \to 0$ при $u \to \infty$ и выполняется условие (2), то существует такое число L, что $f_1(u) \le L < R_2 - 1$ для всех $u > 0$. Получаем:

$$R_2 - f_1(u) - 1 \ge M > 0,$$

где $M = R_2 - L - 1$.

Следовательно, на данном промежутке функция $u(t)$ растет быстрее, чем экспонента с показателем $\lambda[M + o(1)]$. *Из условия* $u(0) = \lambda^{-1}$*, находим оценку:*

$$t_1 < (2\ln\lambda)/(\lambda[M + o(1)]) = o(1).$$

Пусть $t \in [t_1, C]$, тогда $u \ge \lambda$. Поэтому $\tau(u) = C + o(1)$, $f_2(u(t - \tau(u))) = R_2 + o(1)$, $f_1(u) = o(1)$. Уравнение (1) принимает вид:

$$\dot{u} = \lambda[R_2 - 1 + o(1)] \cdot u.$$

Решая данное уравнение с начальным условием $u(t_1) = \lambda$, получаем

$$u(t) = \exp(\lambda(\alpha_1 t + o(1))).$$

Пусть $C \le t \le t_2$, где $t_2 = C + t_1 = C + o(1)$. Заметим, что $\dot u(C) > 0$, а $\dot u(t_2) < 0$. Отсюда следует, что на данном промежутке существует точка максимума $t_{\max} = C + o(1)$. Значение $u(t)$ в этой точке асимптотически равно $u_{\max} = \exp(\lambda(\alpha_1 C + o(1)))$.

Пусть $t_2 \le t \le T_1$, где T_1 определяется из условия $u(T_1) = \lambda$. На этом промежутке уравнение (1) принимает вид:

$$\dot u = \lambda[-1 + o(1)] \cdot u.$$

В качестве начального условия примем соотношение для асимптотического максимума функции $u(t)$, найденное выше. Находя функцию $u(t)$ на данном промежутке:

$$u(t) = \exp(\lambda(\alpha_1 C - t + C + o(1))),$$

и решая уравнение $u(T_1) = \lambda$, получаем $T_1 = R_2 C + o(1)$.

На остальных промежутках асимптотика строится аналогично. Процесс построения асимптотики проиллюстрирован на рис. 1.

В связи со способностью некоторых нейронов периодически генерировать спайки представляет интерес вопрос о периодических решениях уравнения (1).

Теорема 2. *Если при* $u \in [0, \infty)$ *выполнено условие (2) и*

$$\alpha_2/\alpha > C, \tag{3}$$

то уравнение (1) имеет периодическое решение с периодом

$$T_2 = R_2 C + 1 + \alpha_2/\alpha + o(1).$$

Доказательство. Рассмотрим оператор последования Π, введенный по правилу $\Pi(\varphi(t)) = u(T_2 + t)$, $t \in [-C, 0]$. Из условия (3) и теоремы 1 заключаем, что $\Pi(S) \subset S$. Следовательно, в соответствии с принципом Лерэ-Шаудера (см., например, [3]) существует такая начальная функция $\varphi_0(t) \in S$, что $\Pi(\varphi_0) = \varphi_0$, что и свидетельствует о существовании периодического решения $u(t)$ уравнения (1) с периодом T_2.

4. Численное исследование уравнения нейрона

Для численного исследования уравнения (1) был применен метод Дормана и Принса [4] четвертого порядка. При его использовании на каждом шаге интегрирования решение дифференциального уравнения представляется полиномом четвертой степени. Таким образом, находится необходимое непрерывное приближение решения.

В качестве функций $f_{1,2}$ были взяты функции $f_{1,2} = R_{1,2}\exp(-u^2)$. Величина запаздывания $\tau(u) = 1 + (C-1)\exp(-u^2)$. Начальная функция $\varphi(s) = \lambda^{-1}\exp(\lambda\alpha s/2)$. Значения параметров: $R_1 = 1.0$, $R_2 = 2.2$, $\lambda = 2.5$, $C = 4.0$. График численного решения уравнения (1) при данных условиях приведен на рис. 2.

Как легко видеть, результаты численного счета достаточно хорошо согласуются с результатами асимптотического исследования. В частности, значение периода по теореме $1 \approx 19.8$, а на численном счете ≈ 18.

5. Заключение

Построенная модель активности реального нейрона учитывает изменение запаздывания калиевой проводимости относительно натриевой в зависимости от мембранного потенциала. Выбранная зависимость (запаздывание возрастает и стремится к некоторой константе при росте мембранного потенциала) дает весьма содержательную модель, имеющую биологический смысл.

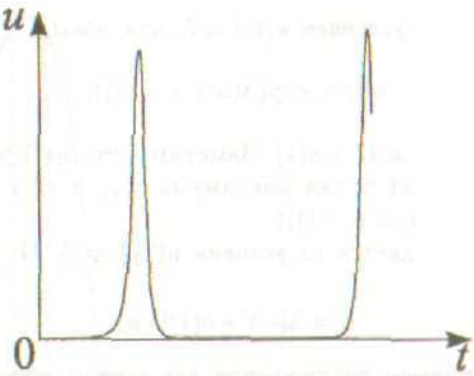

Рис. 2.

Литература

[1] Майоров В. В., Мышкин И. Ю. Об одной модели функционирования нейронной сети // Моделирование динамики популяций. Н. Новгород, 1990. С. 70–78.

[2] Майоров В. В., Мышкин И. Ю. Математическое моделирование нейронов сети на основе уравнений с запаздыванием // Математическое моделирование. 1990. Т. 2, № 11. С. 64–76.

[3] Канторович Л. В., Акилов Г. П. Функциональный анализ. М.: Наука, 1984.

[4] Хайрер Э., Нерсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Т. 1. М.: Мир, 1990.

[5] Эльсгольц Л. Э., Норкин С. Б. Введение в теорию дифференциальных уравнений с отклоняющимся аргументом. М.: Наука, 1971.

Модел. и анализ информ. систем. Т.9, №2 (2002) 16–27

УДК 519.682; 681.324.06

Базовые отношения объектов баз данных и гибкие таблицы

Рублев В.С., Дерябин В.О., Лобачев Д.И., Юсупов А.Р.
Ярославский государственный университет
150 000, Ярославль, Советская, 14

Рассматривается проблема разработки объектной СУБД, лишенной недостатков традиционных объектно-ориентированного и объектно-реляционного подходов. Выделяется система базовых отношений объектов и классов и обосновывается полнота этой системы для описания любых детерминированных динамических систем. Вводится механизм распределенной организации хранения информации классов, объектов, их свойств, позволяющий эффективно адаптировать информацию к динамическим изменениям.

1. Постановка задачи

Описание информации сложных систем при помощи баз данных стало традиционным. Но при этом требуется, во-первых, чтобы любая информация хранилась только в одном месте и при ее коррекции эти изменения могли бы сразу быть использованы везде. Во-вторых, требуется такой структурный подход, при котором информация предоставляется не разрозненными данными, а в виде осмысленных объектов, их свойств и их связей.

Выполнение первого требования стало возможным в результате революционной работы Кодда ([1]) по введению алгебры отношений данных. Однако использование языка реляционных баз данных, который позволяет адекватно реальному миру описывать сложные системы, затруднено практически, так как описание громадного числа мелких объектов и их отношений требует армию разработчиков. Нужен структурный, объектный подход. С возникновением объектно-ориентированного программирования этот подход проник и в организацию баз данных. Но существенное различие реляционной и объектно-ориентированной парадигм не дает возможности безболезненно ввести объектность в реляционные базы данных. В результате выделены общие свойства объектно-ориентированных баз данных (см., например, [2]-[3]), но оставлены для решения и множество проблем, которые в тех или иных объектно-ориентированных реализациях решаются более или менее успешно. Однако большое число статей по объектно-ориентированным базам данных посвящено конфликтам, которые в сильной степени мешают стандартизации единого подхода и получению всех ожидаемых преимуществ объектного проектирования.

Проанализируем некоторые из этих конфликтов. Одним из наиболее важных является конфликт множественного наследования, при котором, если 2 родительских объекта имеют свойство с одинаковым наименованием, то неясно, какое из них следует употреблять для дочернего объекта. Этот конфликт отсутствует, если вместо механизма указания родительских объектов употреблять ссылочное указание родителей. Но в этом случае теряется столь приятное автоматическое наследование свойств родителей. К тому же при проектировании соответствующих отношений в базах данных традиционный способ указания родителей объектно-ориентированного подхода заставляет для каждого дочернего объекта вводить данные родителей, что нарушает основной принцип нормальных форм реляционных баз данных: не повторять информацию. При ссылочном способе такая неприятность исчезает, но вместе с ней исчезает и автоматическое наследование, как указано выше.

В действительности часто бывает необходимо иметь несколько родителей в одном и том же классе. Например, для объекта класса *Товар* необходимо иметь в качестве родительских 2 объекта класса *Поставщик* (основной и запасной). Еще более сложный случай возникает, когда необходимо осуществить *внутреннее наследование*: в одном и том же классе и родительский и дочерний объекты. Например, одна группа продукции может быть родителем для 2 других. Разделение этих групп на разные группы не решает проблему, так как, с одной стороны, неизвестно, сколько же новых классов нам потребуется, а, с другой стороны, ведет к сильной запутанности проектирования.

Заметим, что источник неприятностей связан с тем, что объектно-ориентированный подход явно выделяет только одно из отношений – отношение наследования, предоставляя остальные отношения объектов выражать через ссылки и явно описывать действия, основанные на этих отношениях. Такая искусственность приводит к сильной ограниченности такого подхода. Так, если у нас есть объекты двух разных классов, то для выражения действий, связанных с действиями на отношении таких объектов, мы вынуждены вводить так называемые *дружественные функции* (например, умножение матрицы на вектор) и это явно обедняет объектно-ориентированный подход.

Итак, мы утверждаем, что именно ограниченность объектно-ориентированного подхода и является основным источником неприятностей с проектированием объектно-ориентированных баз данных. Необходимо снять эти ограничения:

1) введением механизмов явного наследования, которые снимают любые ограничения с множественного наследования (в том числе разрешают и внутреннее наследование; нельзя допускать только циклов в графе наследования объектов);

2) введением классов для отношений других классов с описанием действий, связанных с этими отношениями;

3) введением более мощных механизмов наследования, при которых наследуются не только данные и методы родителей, но также свойства отношений родителей с другими объектами.

В результате объектно-ориентированный подход, служащий разработке больших программных комплексов в течение двух десятков лет, получит новую, еще более мощную жизнь (может быть, такой подход следует именовать *системо-объектным*).

В новом объектном подходе среди многообразия отношений объектов нужно выделить основные отношения и механизмы наследования, с помощью которых можно было бы легче проектировать любые отношения объектов детерминированных сложных систем, таких, как автоматизированная система управления большим предприятием, космическим кораблем или сложным физическим, химическим или биологическим процессом. Некоторые принципы организации такого подхода изложены в [4].

2. Основные отношения объектной системы

Отношение наследования, конечно, остается одним из основных отношений. С его помощью частично реализуется одно из главных свойств реляционной базы данных в нормальной форме – неповторяемость информации, так как такая информация наследуется от родительских объектов к дочерним.

Но для того, чтобы обеспечить множественное наследование с несколькими родителями из одного класса, а также внутреннее наследование, вводится понятие *алиас* класса. Каждый класс может иметь произвольное число алиасов и с каждым из них связано имя алиаса. Только один из алиасов является главным (часто единственным), а его имя дает имя классу. В примере, описанном выше для *товаров* и *поставщиков*, класс *Поставщик* имеет 2 алиаса с именами *Поставщик* (основной алиас, дающий имя классу) и *Запасной поставщик*. Каждый объект класса *Товар* может иметь одного или двух родителей в классе *Поставщик*, в зависимости от потребности. В другом примере с классом *Группа продукции* главный алиас имеет такое же имя, а другой алиас имеет имя *Старшая группа*, и это позволяет объекту класса иметь родителем другую группу через последний алиас. Все свойства Старшей группы (а свойства эти могут быть выражены другими отношениями) наследуются объектом.

Рассмотрим теперь другие отношения для описания сложной системы. Их введение диктуется необходимостью:

- описания структуры сложных объектов из более простых частей;
- описания пространственно-временных характеристик взаимного отношения объектов (в частности, при изменении их структуры);
- описания взаимодействия объектов, в результате которого изменяется их структура или пространственно-временные характеристики.

Для реализации описания структуры сложных объектов в объектно-реляционных базах данных вводится отношение целое-часть. Например, в [5] оно называется агрегированием и композицией. Это отношение явно недостаточно для описания структуры, так как в некоторых случаях необходимо не только указать, что часть входит в целое, но и описать как эта часть входит (с какими характеристиками). Например, объект-смесь состоит из объектов-ингредиентов и необходимо указать количественное вхождение ингредиентов в смесь.

Для такого описания вводится отношение трех объектов, которое мы назвали отношением иерархии: старший объект (куда включается), младший объект (что включается) и объект иерархии (как включается). Отношение объектов задается соответствующим отношением *иерархии* классов и, когда класс иерархии в этом отношении отсутствует (например, задается нулевым объектным идентификатором), то тоже делается и для отношения объектов, и мы получаем частный случай – простое включение одного объекта в другой. В отличие от отношения наследования, при котором дочерний объект наследует свойства его родителей (в частности, свойства иерархического включения других объектов), младший в иерархии объект не наследует свойства старшего объекта. Это основной принцип, по которому решается при проектировании, какое из этих двух отношений выбирать в том или ином случае.

Для описания динамики изменения объекта вводится отношение истории. В этом отношении участвует 2 объекта: предшественник и последователь, а также фиксируется момент изменения и, возможно, пространственные характеристики в момент изменения. С каждым объектом связываются 2 его характеристики: *дата рождения* и *дата смерти* объекта. Таким образом с объектом всегда связан полуинтервал его жизни (от даты рождения до даты смерти, исключая ее) и в отношении *истории* момент изменения является датой смерти предшественника и датой рождения последователя. При помощи такого отношения *многие ко многим* можно описать несколько предшественников у одного последователя (слияние объектов) или, наоборот, несколько последователей у одного предшественника (разбиение объекта), или еще более сложный случай реорганизации объектов. При помощи отношения *истории* объектов можно описывать административные изменения (например, слияние или разделение отделов), изменение агрегатного состояния объекта (например, физическое или химическое превращение) и т.п.

Наконец, в отношении *истории* могут находиться и классы. Методы обработки информации, соответствующие времени жизни объекта, также могут быть разные (и в этом собственно может заключаться причина изменения объекта). Если предшественники и последователи обладают каким-либо единым свойством, то это дает нам возможность проследить за изменением этого свойства, применяя для каждого временного интервала жизни объектов, соответствующие этому интервалу методы обработки указанного свойства.

Кроме того, вводится отношение *истории* параметров класса, что также расширяет возможности описания сложной системы.

Теперь остановимся на методах, связанных с взаимным действием нескольких объектов. В предыдущем разделе мы уже отмечали необходимость ввода классов для таких отношений, которые содержали бы соответствующие методы. Проанализируем различные случаи таких взаимодействий объектов:

1. Действия по передаче некоторого объекта (например, продажа объекта от одного "владельца" к другому) могут быть описаны при помощи 4 объектов: *объект-продавец* (откуда), *объект-покупатель* (куда), *объект-товар* (что) и *объект-как* (сколько, когда и т.п.).

2. Технологическая операция на оборудовании (например, изготовление рабочим на станке детали из сырья или заготовок) может быть описана при помощи 4 объектов: *объект-рабочий* (откуда), *объект-оборудование* (куда), *объект-деталь* (что) и *объект-как* (из чего, сколько, когда и т.п.).

3. Выполнение пользователем действий в системе (например, ввод, коррекция информации или получение отчета) также может быть описано при помощи 4 объектов: *объект-пользователь* (откуда), *объект-система* (куда), *объект-изменения* (что) и *объект-как* (когда, ограничение прав и т.п.).

Теперь видно, что описанные случаи можно обобщить и ввести отношение взаимодействия 4 объектов, описав класс взаимодействий с 4 параметрами, указывающими классы объектов *откуда, куда, что и как*.

Введение взаимодействий позволит создавать корпоративные системы автоматизированного управления, которые основаны на выполнении взаимодействий теми или иными пользователями системы. То есть одним из основных объектов такой системы становится пользователь, который в соответствии с правами на взаимодействия может изменять данные системы и получать отчеты. Более того, при развитии отношения *взаимодействия* можно ввести в *объект-как* условия выполнения взаимодействия в зависимости от завершения других взаимодействий и наличия ресурсов для их выполнения. В этом случае одни взаимодействия могут возбуждать другие (при выполнении условий или при нехватке ресурсов). Если же в *объект-как* ввести также время выполнения взаимодействия, то становится возможным автоматическое моделирование поведения сложной системы с изменением масштаба времени. Но всем этим возможностям и дальнейшей разработке аппарата *взаимодействия* будет посвящена отдельная работа.

Итак, мы определили 4 отношения между объектами, с помощью которых мы можем описать поведение любой сложной детерминированной системы, вводя на основе этих основных отношений любые другие отношения частей системы, их взаимодействия и структурного изменения. Поэтому для целей разработки таких сложных систем описанная система основных отношений объектов *полна*.

Такие же отношения, введенные для классов, являются в свою очередь классами соответствующих отношений для объектов, а цель такого введения будет пояснена в дальнейших разделах.

3. Дополнительные отношения объектной системы

Назовем *врожденными* свойствами объекта как непосредственные его свойства, определяемые параметрами класса объекта, так и свойства непосредственно включенных в него других объектов при помощи отношения иерархии. Врожденные свойства всех родителей объекта (как непосредственных родителей, так и любых других в ациклическом графе родителей) назовем *наследуемыми* свойствами объекта. В некоторых случаях наследуемые свойства могут конфликтовать друг с другом или с врожденными свойствами. Что касается непосредственных свойств родительских объектов, то их конфликтов можно избежать, если параметры разных классов различны, а для родительских объектов одного класса можно сделать различными их алиасы. Но относительно свойств включения других объектов (при помощи отношения иерархии) конфликта не всегда удается избежать и в таких случаях для его разрешения требуется механизм явного выбора этого свойства. Даже в тех случаях, когда конфликт наследуемых свойств не возникает, механизм выбора может оказаться полезным с точки зрения эффективности доступа к данным. К тому же мы покажем, что, используя такой механизм, мы можем добиться большей общности в описании классов.

Для выявления особенностей необходимых конструкций рассмотрим несколько примеров описания *Продукции*. Каждая продукция описывается идентификацией (например, *Обозначение*, *Модель*, *Марка*, *Артикул* и т.п.), стандартом (например, *ГОСТ*, *ОСТ*, *ТУ* и т.п.), физико-химическими показателями, для значений которых стандартами могут быть введены нормативы, ограничивающие допустимые значения показателей. Но так как различная продукция имеет разную идентификацию, то при включении свойств идентификации непосредственно в классы продукции таких классов будет очень много (и, более того, неопределенное число), что нежелательно. Другой способ – поместить все классы идентификации в один обобщенный класс *Идентификация* и включить его в класс *Продукция*, используя отношение иерархии. Но тогда нужно знать, объектами каких классов идентификации определяется тот или иной объект класса *Продукция*. Поэтому сгруппируем продукцию по общности ее идентификации и для этого определим класс *Группа продукции* как один из родителей класса *Продукция*. Далее включим при помощи отношения иерархии новый класс *Классы идентификации*, определяющий указатели на такие классы (идентификаторы классов), в класс *Группа продукции*. Теперь, поскольку это свойство наследуется классом *Продукция*, то для каждого объекта последнего класса среди его родителей в *Группе продукции* (ниже мы покажем, что их может быть несколько) найдется такой объект, который включает по отношению иерархии объекты класса *Классы идентификации*, указывающие идентификаторы соответствующих классов идентификации. Таким образом мы для каждого объекта продукции можем через это наследуемое свойство определить его классы идентификации и тем самым выбрать (или создать) его идентификацию. Но, так как несколько родителей могут содержать такое свойство (*Классы идентификации*), то для разрешения конфликта или ускорения определения таких классов (даже если нет конфликта) мы введем дополнительное отношение *идентификации*. Оно определяется для трех указанных классов: *Продукции* (что идентифицируется), *Группа продукции* (что включает определение идентификации) и *Классы идентификации* (что определяет саму идентификацию). Для объектов также задается отношение *идентификации*, но оно определяется двумя объектами: один из класса *Продукция*, а другой – его родитель – из класса *Группа продукции*. Теперь для определения классов идентификации объекта необходимо по отношению *идентификации* определить объект из класса *Группа продукции*, затем по отношению *идентификации* классов определить класс, задающий классы идентификации (в нашем примере *Классы идентификации*) и по отношению иерархии для объекта класса *Группа продукции* определить объекты этого класса *Классы идентификации*.

Схема на рис. 1 иллюстрирует отношения описанных классов. Основные классы схемы определены обычными прямоугольниками, а классы иерархии определены прямоугольниками со скругленными углами. Шестиугольником показан обобщенный "класс" *Идентификация*, который является объединением всех классов идентификации, имеющих один и тот же параметр *Идентификация*. Сплошными стрелками на них определены отношения наследования, точечными стрелками определены отношения иерархии (в разрыве такой стрелки может определяться класс иерархии). Наконец штриховыми стрелками определено дополнительное отношение идентификации (в других примерах – отношение выбора или отношение ограничения). Светлый конец любой из стрелок указывает на старший класс в отношении, а темный – на младший класс. При этом тупой конец стрелки указывает на отношение *многие...* или *...ко многим*, а острый на отношение *один...* или *...к одному*. На схеме определены также еще 2 класса *Стандарт* и *Комплектующие* и отношение этих классов. Поясним это.

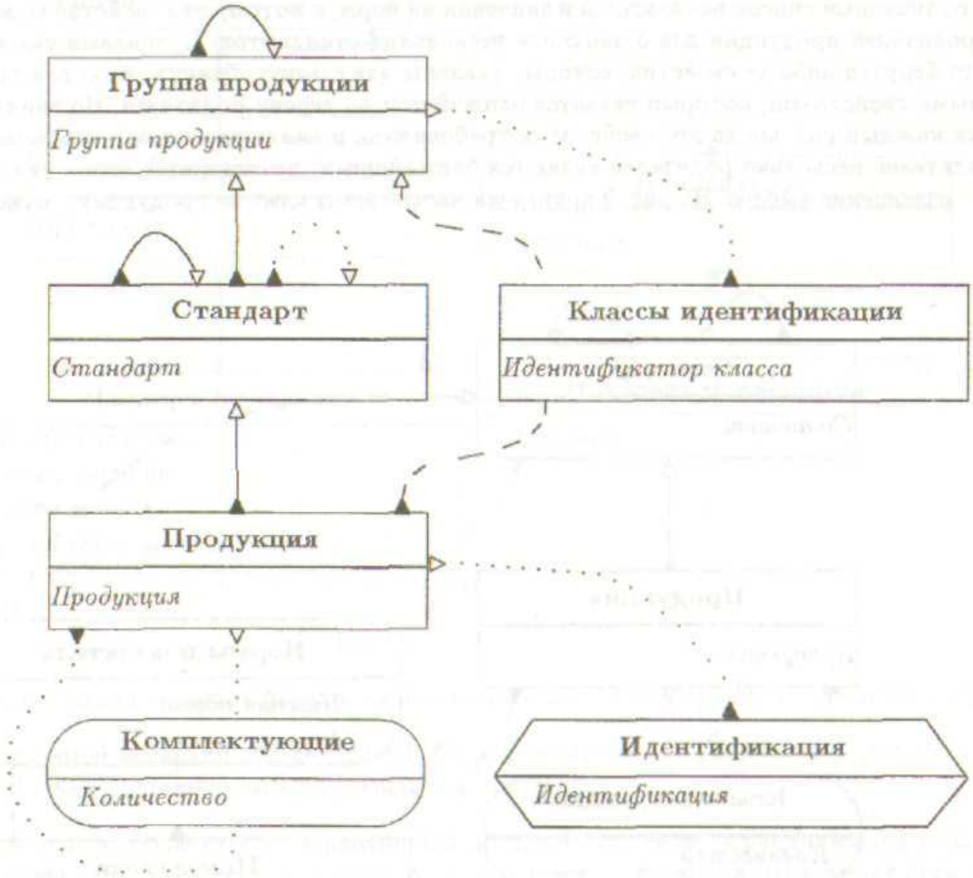

Рис. 1

Каждой группе продукции отвечает один или несколько стандартов. В последнем случае они определяют такое же число подгрупп продукции. Так как эти подгруппы также рационально сделать объектами класса *Группа продукции*, то для отображения связей между группами продукции введем в упомянутый класс отношение внутреннего наследования, добавив к алиасу *Группа продукции* алиас *Старшая группа*. Это отношение необходимо для наследования свойства, определяемого объектами *Классов идентификации* по отношению иерархии со старшей в цепочке подгрупп группой. Для стандартов определим класс *Стандарт* и свяжем соответствующие группы со стандартами отношением наследования, и также отношением наследования свяжем класс (и соответственно объекты класса) *Продукция* с классом (и соответствующими объектами класса) *Стандарт*. Так как стандарты также могут включаться друг в друга (например, ОСТ уточняет ГОСТ, а ТУ уточняет ОСТ и т.д.), но нам необходима передача по наследованию свойств, то в классе *Стандарт* мы вводим внутреннее наследование, добавляя к алиасу *Стандарт* алиас *Старший стандарт*. Таким образом в дерево родителей каждой продукции входят как стандарты, так и группы продукции, и продукция полностью идентифицируется стандартами (дерева родителей) и идентификацией, определяемой по отношениям иерархии и идентификации.

Теперь отметим, что продукция может иметь сложный характер и требуется описание включения в продукцию ее частей, в свою очередь являющихся продукцией. Для этого в класс *Продукция* вводится внутренняя иерархия с классом иерархии *Комплектующие*, определяющим количество комплектующих, включаемых в агрегатированную продукцию. Поскольку стандарты для таких частей-комплектующих могут быть частями стандартов на агрегатированную продукцию, то в классе *Стандарт* также вводится внутренняя иерархия, определяющая включение соответствующих стандартов. На рис. 1 приведена часть схемы классов продукции с отношением идентификации.

Задание списка показателей качества продукции и ограничивающих их норм может быть сделано для каждого объекта класса *Продукция*. Но более рационально указывать это свойство для всей продукции, имеющей одинаковый список показателей и значения их норм, и потому это свойство может быть указано в дереве родителей продукции для одного или нескольких стандартов. Правилами умолчания предполагается, что берутся либо те свойства, которые указаны для самого объекта, либо для того из родителей с указанными свойствами, который является ближайшим по дереву родителей. Но ввиду неэффективности поиска каждый раз, когда это свойство потребовалось, и ввиду возможной неоднозначности (если в дереве родителей несколько родителей являются ближайшими) используется явное указание для выбора свойства – отношение *выбора*. На рис. 2 приведена часть схемы классов продукции с отношением выбора.

Рис. 2

В общем случае отношение *выбора* для классов определяется указанием 4 классов: класса родителя (одновременно старшего по иерархии), класса младшего по иерархии, класса иерархии (эти 3 класса описывают класс наследуемой иерархии) и дочернего класса. Отношение *выбора* для объектов определяется указанием 2 объектов: дочерним и родителем (в дереве или ациклическом графе родителей), для которых по отношению выбора соответствующих классов (дочернего и родительского) определяются классы наследуемой иерархии, а по этим классам и родительскому объекту определяется сама наследуемая иерархия. Помимо этого само отношение выбора как объект снабжается *временным промежутком жизни* (см. предыдущий раздел).

При определении значений показателей продукции (например, для объекта класса *Паспорт продукции* на рис. 3) необходимо ограничить список показателей для выбора. Можно отыскать родительский объект, который имеет свойство *Показатели*, но опять же поиск неэффективен и возможно неоднозначен. Поэтому рационально явное указание родительского объекта, имеющего это свойство, при помощи вновь вводимого отношения *ограничения* (вводится для классов и объектов аналогично отношению выбора). На рис. 3 приведена следующая часть схемы классов рассматриваемого примера для описания продукции с отношением ограничения.

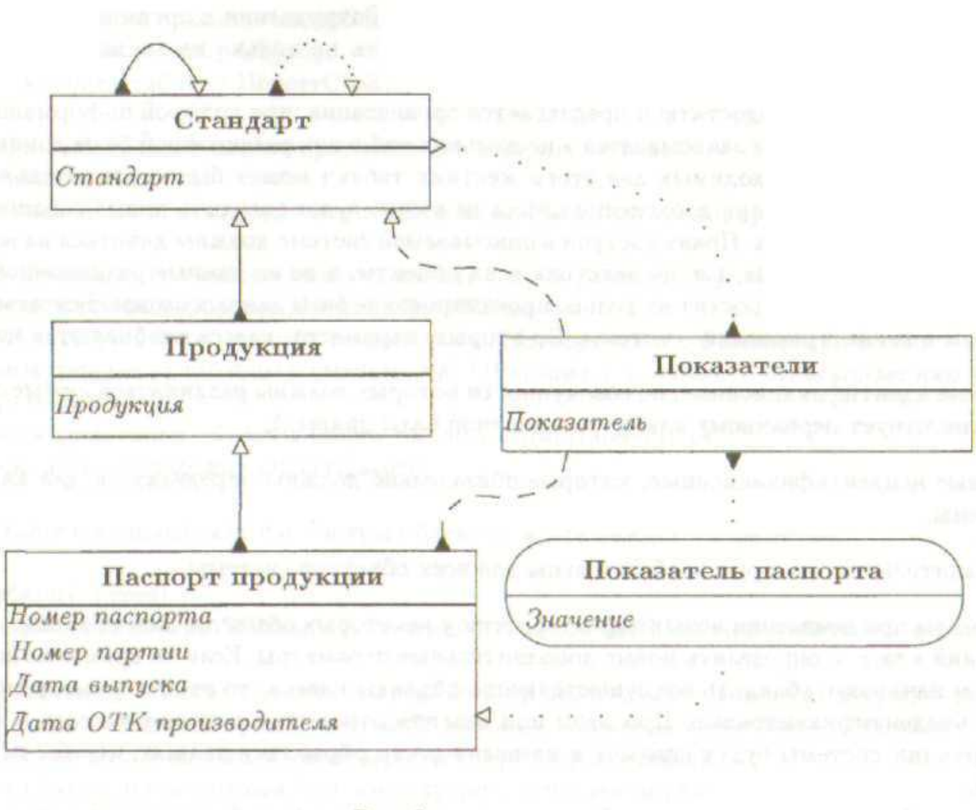

Рис. 3

Итак, при наследовании свойств объекта, включенных в него по иерархии, могут быть 3 случая:

1. Выбор наследуемой иерархии для дочернего объекта определяется указанием родителя, когда имеется неоднозначность такого выбора (отношение *выбора*).

2. Список объектов, включаемых по иерархии для дочернего объекта, ограничивается указанием родительского объекта, для которого этот список определен по иерархии (отношение *ограничения*).

3. Свойства (классы) объектов, включаемых по иерархии для дочернего объекта, указываются при помощи иерархии для родительского объекта (отношение *идентификации*).

Этими 3 случаями исчерпываются возможности наследования свойств включения при помощи отношения иерархии. Действительно, при наследовании необходимо однозначно определить саму иерархию, ограничить выбор ее объектов и определить, наследуются ли сами объекты (прямая ссылка), включенные в иерархию, или они определяют классы других объектов (косвенная ссылка), включаемых по иерархии в дочерний объект.

4. Отображение объектной системы в реляционную базу данных и гибкие таблицы

Полученную объектную систему можно реализовать при помощи реляционной базы данных. Для этого с каждым классом объектов свяжем таблицу и с каждым отношением свяжем дополнительную таблицу, отражающую одно из отношений *"один к одному"*, *"один ко многим"*, *"многие к одному"*, *"многие ко многим"*.

Недостатком такого подхода является чрезвычайно большое число таблиц для сложной системы. С введением нового класса необходимо заводить не только таблицу для его объектов, но и много других таблиц для отношений с другими данными. Если реальность заставила для некоторых объектов добавить параметры, даже не являющиеся идентифицирующими, то требуется организовать дочерний класс для этих объектов, т.е. ввести новую таблицу и много дополнительных таблиц для связей этих объектов. Вдобавок для пользователя такие действия могут быть затруднены, так как на новые таблицы нужно администратору базы данных определить права доступа. Затруднения с организацией могут быть вызваны и тем, что права доступа иногда необходимо давать не только на записи таблицы, но и на ее параметры.

Для преодоления этих недостатков предлагается организация, при которой информация об объектах, классах, параметрах классов записывается в нескольких таблицах реляционной базы данных (мы назовем их *жесткими*). Число необходимых для этого жестких таблиц может быть сравнительно небольшим – порядка 30. Но, во-первых, при добавлении класса не нужно будет заводить новые таблицы и определять права пользователей для них. Права доступа в описываемой системе должны даваться на взаимодействия, классы, объекты, параметры, т.е. на действия и их объекты, а не на данные реляционной базы данных. Такой объектный подход упростит не только проектирование базы данных сложной системы, но и сделает более легким администрирование системы. Во-вторых, параметры класса разбиваются на 3 категории:

1) главные идентификационные, по совокупности которых должны различаться любые объекты класса (соответствует первичному ключу реляционной базы данных);

2) главные неидентификационные, которые обязательно должны определяться для каждого объекта системы;

3) дополнительные, которые необязательны для всех объектов системы.

Таким образом при появлении новых характеристик у некоторых объектов класса можно не заводить новый дочерний класс, а определить новые дополнительные параметры. Если со временем дополнительным параметром начинают обладать все существующие объекты класса, то этот параметр можно перевести в главные неидентификационные. При этом при помощи отношений истории классов и истории объектов конструкции системы будут связаны и позволят вести обработку данных, как бы не замечая таких изменений.

Для значений всех параметров всех объектов и классов можно было бы выделить одну таблицу со строковым типом. Но при этом, во-первых, была бы существенно потеряна эффективность из-за постоянных преобразований к типу параметра и обратно. Во-вторых, усложнилась бы организация прав доступа к отдельным параметрам. В-третьих, возросло бы время доступа к значениям. Поэтому рационально принять способ, при котором значения каждого параметра записываются в отдельную жесткую таблицу. Хотя это и увеличивает во много раз число используемых таблиц, но все же их число на несколько порядков меньше, чем при использовании жестких таблиц. К тому же легко организовать ограничение прав доступа к отдельным параметрам.

Назовем *Динамической информационной моделью (DIM)* подход описания базы данных сложной системы, удовлетворяющий следующим требованиям:
1) объекты системы связаны основными и дополнительными отношениями, описанными в предыдущих разделах;
2) отображение объектов и их связей в реляционную базу данных дает четвертую нормальную ее форму;
3) это отображение строится при помощи описанных *гибких* таблиц.
На рис. 4 приведена схема организации данных DIM. Поясним ее.

Главными в модели являются таблицы Class классов объектов, Object объектов классов, Parameters, определяющая параметры объектов, и значений каждого параметра объектов (через таблицы типа Val_IdParameter). При этом каждый класс, объект, параметр определяется уникальным идентификатором соответственно IdClass, IdObject и IdParameter.

Таблица Class классов объектов определяет для каждого класса:
- имя класса и всех его алиасов (через таблицу Alias, где TypeAlias указывает базовый алиас);
- его родительские и дочерние классы (через таблицу ClassInheritance наследования - граф наследования с дугами типа TypeGroup);
- его старшие и младшие классы в иерархии классов (через таблицу ClassHierarchy - граф иерархии классов); - классы, определяющие классы идентификации его объектов (через таблицу ClassIdentification);
- классы, определяющие классы ограничения иерархии объектов (через таблицу ClassLimitation);
- классы, определяющие классы выбора свойств объектов (через таблицу ClassSelection);
- классы взаимодействия (через таблицу ClassInteraction);
- уровень класса в графе наследования LevelInheritance;
- тип класса TypeClass;

- параметры класса (через таблицу GroupParam);
- объекты класса (через таблицу Object);
- историю класса (через таблицу HistoryClass).

Таблица Object объектов классов определяет для каждого объекта:
- класс (через таблицу Classes);
- иерархически связанные объекты уровней выше и ниже (через таблицу ObjectHierarchy - граф иерархии объектов);
- объекты, определяющие его классы идентификации (через таблицу ObjectIdentification);
- уровень объекта в графе иерархии объектов LevelHierarchy,;
- параметры объекта (через таблицы ObjectInheritance, GroupParam и Parameters);
- значения параметров через таблицы с именами Val_IdParameter, где вместо IdParameter находится идентификатор параметра;
- связанные взаимодействием объекты (через таблицу ObjectInteraction);
- историю объекта (через таблицу ObjectHistory).

Таблица Parameters определяет параметры объектов и для каждого параметра:
- его имя;
- тип (через таблицу Types).
- значения каждого параметра IdParameter каждого объекта IdObject через таблицы с именами типа Val_IdParameter.

Поле LevelInheritance класса (объекта) определяет уровень класса (объекта) в графе наследования (для объектов – ациклическом), представляемого таблицей ClassInheritance (ObjectInheritance):
0 - класс (объект) является изолированной вершиной графа наследования;
1 - класс (объект) является корневым (самым старшим) в наследовании;
2 - класс (объект) не является ни корневым (самым старшим), ни висячим (самым младшим) в графе наследования;
3 - класс (объект) является висячим (самым младшим) в наследовании.

Поле TypeClass класса определяет тип класса:
0 - класс объектов, не являющихся связями других объектов;
1 - класс объектов, являющихся связями в иерархии других объектов;
2 - класс объектов, являющихся связями во взаимодействии других объектов.

Поле LevelHierarchy объекта определяет уровень объекта в ациклическом графе иерархий, представляемого таблицей ObjectHierarchy:
0 - объект является изолированной вершиной графа иерархии;
1 - объект является корневым (самым старшим) в иерархии;
2 - объект не является ни корневым (самым старшим), ни висячим (самым младшим) в графе иерархии;
3 - объект является висячим (самым младшим) в иерархии.

Каждая таблица, кроме таблиц Types, ClassHistory, ObjectHistory, ParameterHistory, содержит поля *BirthDay* и *DeathDay* типа DateTime, определяющие полуинтервал жизни объекта таблицы (класса, объекта, связи, параметра, значения).

Каждый класс может содержать 3 группы параметров с именами:
MIPO - Main Identification Parameters of Objects - TypeGroup=0;
MNPO - Main Nonidentification Parameters of Objects - TypeGroup=1;
APO - Addinional Parameters of Objects - TypeGroup=2.
Эти группы параметров определяют спецификацию объектов класса. При этом параметры из списка MIPO являются идентификационными (любые 2 объекта класса не должны иметь одинаковых наборов значений этих параметров); параметры из списка MNPO являются остальными обязательными параметрами объектов класса, а параметры из списка APO необязательны для того или иного объекта класса. Обязательным является наличие хотя бы одной из групп параметров MIPO или MNPO.

Таблица ClassInheritance описывает связи классов по наследованию (родительские - дочерние) и представляет ориентированный граф с отдельными компонентами для каждого типа классов (не допускается наследование классов разных типов). При этом родительский класс определяется через свои алиасы и таблицу Alias, а тип группы TypeGroup определяет тип параметра дочернего класса.

Схема организации данных ***Динамической информационной модели*** **(DIM)**

Гибкие таблицы

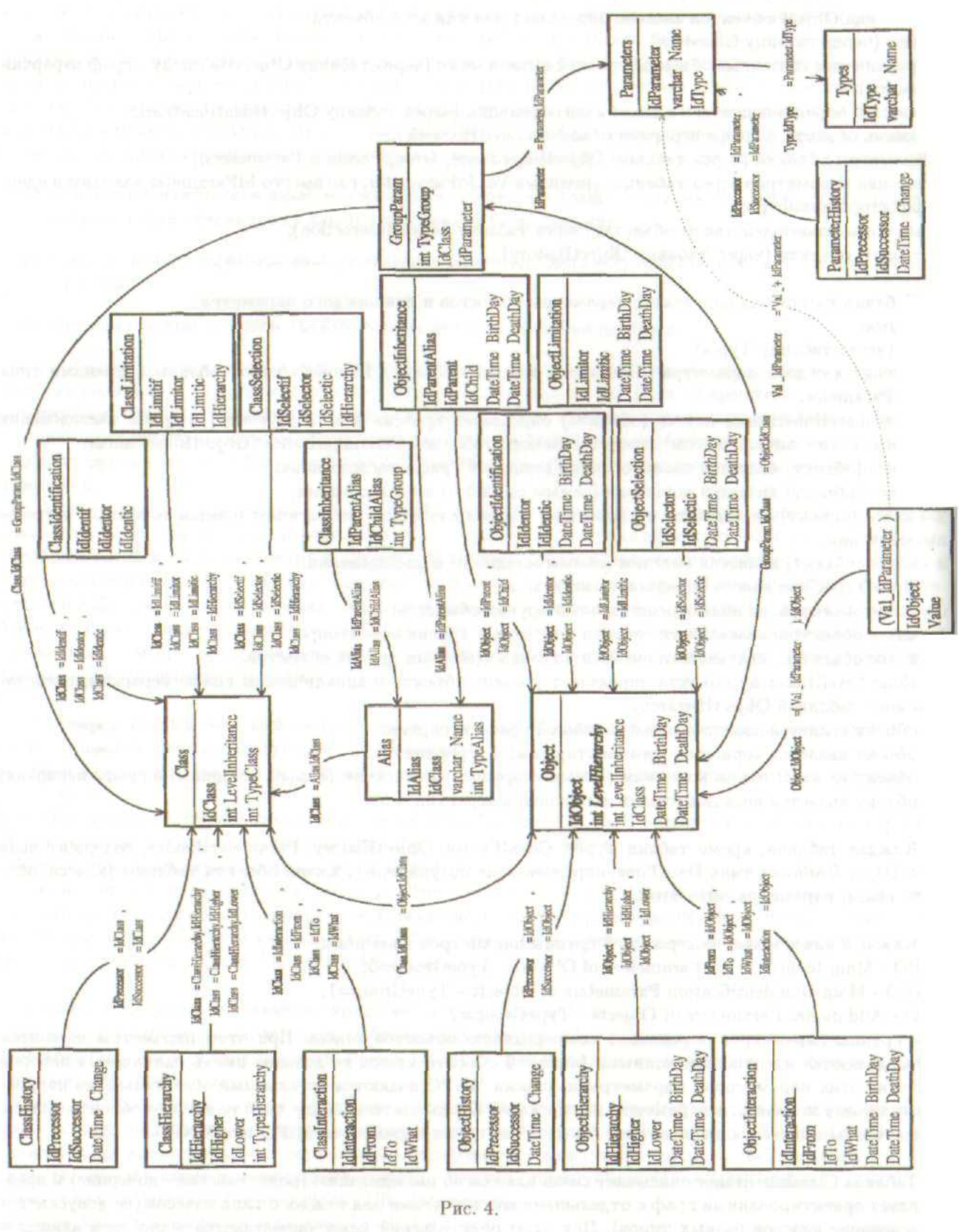

Рис. 4.

Таблица ClassIdentification описывает классы определения идентификации объектов класса IdClassIdentic, указывая на класс IdClassIdentor, содержащий иерархически класс IdClassIdentif, объектами которого являются идентификаторы классов идентификации данного класса IdClassIdentic. Соответствующий объекту класса IdClassIdentic объект класса IdClassIdentor должен принадлежать дереву родительских объектов для первого из них.

Таблица ClassLimitation описывает классы определения ограничения иерархии объектов класса IdClassLimitic, указывая на класс IdClassLimitor, содержащий иерархически класс IdClassLimitif, объекты которого образуют списки для каждого объекта класса IdClassLimitor. Соответствующий объекту класса IdClassLimitic объект класса IdClassLimitor должен принадлежать дереву родительских объектов для первого из них, а указанный выше список объектов IdClassLimitif ограничивает иерархию между этим объектом класса IdClassLimitic и объектами класса IdClassLimitif.

Таблица ClassSelection описывает классы определения выбора свойств объектов класса IdClassSelectic, указывая на класс IdClassSelector, содержащий иерархически класс IdClassSelectif, объекты которого определяют свойства для каждого объекта класса IdClassSelector. Соответствующий объекту класса IdClassSelectic объект класса IdClassSelector должен принадлежать дереву родительских объектов для первого из них, а указанная выше иерархия объектов классов IdClassSelector и IdClassSelectif определяет выбор свойств для объекта класса IdClassSelectic.

Таблица Alias определяет алиасы класса и их имена, а также выделяет параметром TypeAlias базовый алиас класса.

Таблица ClassHierarchy описывает связи классов по иерархии (старшие – младшие). Параметр IdHierarchy определяет идентификатор класса связи между старшим классом IdHigher и младшим классом IdLower (если IdHierarchy=0, то иерархия определяет простое включение объектов младшего класса в объекты старшего класса). Параметр TypeHierarchy определяет тип отношения между старшим и младшим классом иерархии:
0 - многие ко многим;
1 - многие к одному;
2 - один ко многим;
3 - один к одному.

Таблица ClassInteraction описывает классы взаимодействия, определяющие взаимодействие потока объекта класса IdWhat от объекта класса IdFrom к объекту класса IdTo.

Таблица ObjectHierarchy описывает иерархические связи объектов (старшие – младшие) и представляет ациклический ориентированный граф с навешенными на дуги параметрами вхождения младшего объекта в старший. Параметр IdHierarchy задает количественные характеристики связи, определяемые объектом класса IdHierarchy таблицы ClassHierarchy. Параметр IdHigher задает старший объект класса IdHigher таблицы ClassHierarchy. Параметр IdLower задает младший объекта класса IdLower таблицы ClassHierarchy.

Таблица ObjectInheritance для каждого объекта IdChild задает объекты IdParent родительских классов, определяемые алиасами IdParentAlias.

Таблица ObjectIdentification (определения классов идентификации объекта) для объекта IdIdentic из класса IdIdentic (таблицы ClassIdentification) описывает объект IdIdentor класса IdIdentor (таблицы ClassIdentification), которому иерархически подчинены объекты класса IdIdentif (таблицы ClassIdentification). Обязательным условием является то, что объект IdIdentor должен быть предком по наследованию для объекта IdIdentic.

Таблица ObjectLimitation (определения ограничения иерархии объекта) для объекта IdLimitic из класса IdLimitic (таблицы ClassLimitation) описывает объект IdLimitor класса IdLimitor (таблицы ClassLimitation), которому иерархически подчинены объекты класса IdClassLimitif (таблицы ClassLimitation), указывающие ограничение иерархии объектов этого класса для данного объекта IdLimitic. Обязательным условием является то, что объект IdLimitor должен принадлежать дереву предков для объекта IdLimitic.

Таблица ObjectSelection (определения выбора свойств объекта) для объекта IdSelectic из класса IdSelectic (таблицы ClassSelection) описывает объект IdSelector класса IdSelector (таблицы ClassSelection), которому иерархически подчинены объекты класса IdSelectif (таблицы ClassSelection), определяющие выбор свойств данного объекта IdSelectic. Обязательным условием является то, что объект IdSelector должен принадлежать дереву предков для объекта IdSelectic.

Таблицы HistoryClass, HistoryObject, HistoryParameter описывают динамические (во времени) связи классов, объектов, параметров соответственно (предшественники - наследники), определяют время преобразования (классов, объектов, параметров) и представляют ациклические ориентированные графы.

Таблица ObjectInteraction описывает взаимодействия объектов, представляет ориентированный граф с навешенными на дуги из объекта IdFrom в объект IdTo потоками (объектов IdWhat), определяемыми параметрами объекта IdInteraction.

5. Некоторые проблемы реализации

При реализации описанной системы требуется решить ряд проблем[1], направленных на повышение качества ее эксплуатации:

1. Для создания новых классов, параметров, объектов, их отношений, а также для их коррекции необходим удобный инструмент разработчика системы.

2. Для взаимодействия пользователя с системой в *клиент-серверной* технологии требуется разработка современного интерфейса пользователя через аппарат взаимодействий и ограничение прав доступа на них (автоматическое создание и изменение меню пользователя, объектная диагностика ошибок).

3. Создание входных форм, позволяющих пользователю вводить и корректировать данные системы в виде связанных объектов, а не разрозненных таблиц, требует большого предварительного труда разработчика. Трудоемкость можно уменьшить за счет создания средств автоматизированной навигации по объектам системы, основанной на информации о классах системы и их связях. Такой инструмент, дополненный средствами генерации входных форм, должен позволить пользователю в большинстве случаев самому сгенерировать форму.

4. Создание форм отчетов в сложной реляционной базе данных требует написания сложных SQL-запросов. Для того, чтобы пользователь мог сам в большинстве случаев построить сложные формы отчетов с включением статистики и других вычислений, необходимо разработать язык отчетов, язык формул и редактор отчетов.

5. Создание аппарата динамики объектов и классов и инструмента отображения динамики в отчетах.

Литература

[1] Codd E.F., A relational model for large shared data banks //*Communication of the ACM.* 1970. V.13, N.6. p.377-387.

[2] Atkinson M., Altaïr F.B., DeWitt D., Dittrich K., Maier D., Zdonik S. The Object-Oriented Database System Manifesto. http://www.cs.cmu.edu/OODBMS/Manifesto. 1995.

[3] Cattell R.G.G. and Barry D. The Object Database Standart: ODMG 3.0. Morgan Kaufmann. January 2000.

[4] Дерябин В.О., Рублев В.С., Принципы организации СУБД-независимого языка ALODS для объектно-динамических систем //Современные проблемы математики и информатики. Ярославль, 1999. Вып.2. С.183-189.

[5] Ambler S.W. Mapping objects to relational databases. http://www-106.ibm.com/developerworks/library. 2000.

[1] по большинству проблем работы ведутся и некоторые уже закончены

Модел. и анализ информ. систем. Т.9, №2 (2002) 28–31

УДК 681.3

О реализации обмена информацией между компонентами процессорного модуля

Васильчиков В.В., Силантьев А.О.
Ярославский государственный университет
150 000, Ярославль, Советская, 14

Описан механизм обмена информацией между арифметическим, управляющим и коммутационным процессорами при программной реализации библиотеки параллельного выполнения рекурсивно-параллельных программ в среде Win32 или Win64. Предполагается, что обмен данными происходит посредством очередей сообщений. Рассмотрены основные проблемы реализации такого подхода при квазипараллельной работе компонентов процессорного модуля и практические способы их решения. Приведены основные типы межкомпонентных сообщений с их приоритетами.

1. Введение

Организация параллельных вычислений на группе отдельных вычислительных машин (рабочих станций или ПК), соединенных каналом связи, по общему признанию является дешевой реализацией массивно-параллельного компьютера. Чаще такие вычисления базируются на модели передачи сообщений и реализуют стандарт MPI (Message Passing Interface), однако существуют и альтернативные подходы.

Одним из них является концепция рекурсивно-параллельных (РП) вычислений. В [1] этот подход подробно рассмотрен, а также описаны программные средства для разработки РП-программ и организации их параллельного выполнения на локальной сети под управлением MS DOS. Для программной реализации аналогичных средств разработки в среде Win32 или Win64, в которые изначально была заложена многозадачность, казалось бы, достаточно воспользоваться соответствующими API-инструментами. Однако сложность механизмов взаимодействия между отдельными компонентами процессорного модуля РП вычислительной системы порождает ряд проблем, рассмотрению которых и посвящена данная публикация.

2. Основные компоненты процессорного модуля

Каждый процессорный модуль системы параллельного выполнения рекурсивно-параллельных программ представляет собой отдельную программную систему, состоящую из следующих функциональных частей или компонентов: арифметический процессор, управляющий процессор, приемная часть коммутационного процессора и его передающая часть. Поскольку каждый компонент процессорного модуля работает, в общем случае, независимо от других или даже параллельно с другими (коммутационный процессор пересылает данные, а арифметический производит некоторые вычислительные действия) то естественным шагом представляется реализация данных компонентов в виде отдельных потоков (thread). Действительно, реализованные таким образом компоненты могут работать псевдопараллельно (а на многопроцессорных машинах - параллельно) друг с другом. Такая организация должна положительно сказываться на общей производительности системы. В любом случае накладные расходы на организацию многопотоковости, по-видимому, не превысят возможного прироста производительности. А поскольку компоненты системы почти не используют общих данных, то при практической реализации не возникнет необходимости разрешения коллизий при одновременном обращении к данным различными компонентами системы.

Однако, хотя компоненты процессорного модуля работают независимо, очевидным образом возникает необходимость обмена некоторыми данными между ними (например, запрос управляющего процессора коммутационному на пересылку данных и т.п.). Для этой цели между компонентами реализован специальный механизм обмена системными сообщениями с использованием нескольких очередей.

3. Системные сообщения и очереди сообщений

Архитектура РП-вычислительной системы и механизмы обмена информацией между отдельными компонентами достаточно подробно описаны в [2], поэтому мы не будем повторять ее описание, а лишь рассмотрим основные моменты ее программной реализации.

Единицей информационного обмена между компонентами системы служит системное сообщение. Сообщение может передаваться лишь от одного компонента другому, компонент, посылающий сообщение называется источником, а принимающий - приемником. В общем случае сообщение представляет собой какой- либо набор данных, структура которого известна только источнику и приемнику. Таким образом, на системном уровне сообщение представляет собой некоторую область данных, первый байт которой идентифицирует тип сообщения (одного байта представляется вполне достаточным для указанных целей, поскольку количество возможных типов сообщений не будет превышать 256), а остальные хранят соответствующую информацию.

Поэтому целесообразно реализовать следующий подход обмена сообщениями: источник выделяет некоторую область памяти под сообщение, заполняет его данными и пересылает приемнику указатель на эту область данных. Приемник в зависимости от типа сообщения каким-либо образом использует данные, находящиеся в сообщении, а затем освобождает занятую сообщением память. Очевидно, что описанный алгоритм наиболее легко реализуем, и не будет вызывать коллизий при обращении к памяти, поскольку приемник сможет работать с сообщением только тогда, когда источник уже выделил память, заполнил сообщение данными и отослал его. После этого источник уже не работает с данным сообщением.

Легко заметить, что в момент пересылки сообщения источником приемник может быть занят какой-либо другой работой. В этом случае возникает необходимость в некотором хранилище для входящих сообщений приемника. Такое хранилище называется очередью сообщений. Если какой-либо компонент может обмениваться сообщениями с другим компонентом, то между ними размещается очередь сообщений. Очередь сообщений служит лишь для односторонней передачи сообщений, то есть каждая очередь сообщений жестко привязана к конкретному компоненту - приемнику, который извлекает сообщения из данной очереди. Компонент, ставящий сообщения в некоторую очередь называется источником для данной очереди. Каждая очередь может иметь либо один, либо несколько источников, то есть компонентов, которые ставят в данную очередь различные сообщения для одного и того же приемника. Если источнику необходимо переслать приемнику какое-либо сообщение, то он, сформировав его и поставив в соответствующую очередь, может продолжать работу, считая, что приемник уже получил данное сообщение.

Функцию потока каждого из компонентов назовем диспетчером сообщений. Фактически основной задачей каждого из компонентов и является обработка входящих сообщений. Функцию потока целесообразно реализовать в виде бесконечного цикла, выбирающего последовательно из всех входящих очередей сообщения и обрабатывающего их. Поскольку сообщения выбираются из очередей последовательно, то между ними естественным путем реализуется дисциплина относительного приоритета. Приоритеты входящих очередей сообщений жестко заданы в виде последовательности вызовов функций чтения из соответствующих очередей в функции потока. С течением времени приоритеты изменяться не могут. Если на очередной итерации цикла диспетчера сообщений все очереди оказались пустыми, то данный поток засыпает до тех пор, пока в какую-либо из входящих очередей данного компонента не придет очередное сообщение.

Все сообщения условно делятся на три группы:

- служебные сообщения и ответы (приоритет 1);
- запросы извне (приоритет 2);
- запросы изнутри (приоритет 3).

Такое разделение необходимо для создания гибкой системы приоритетов сообщений и разделения их по соответствующим входящим очередям. Так, например, арифметический процессор может поставить некоторое сообщение в очереди N 1, 2 или 3 управляющего процессора. Очередь, в которую будет поставлено конкретное сообщение, определяется типом последнего. А поскольку между очередями реализована дисциплина относительного приоритета, то описанный механизм фактически позволяет устанавливать каждому сообщению соответствующий приоритет обработки. Определение приоритета сообщения производится по специальным таблицам. Таблицы приоритетов представляют собой массивы констант - типов сообщений, соответствующих каждому приоритету. Для определения приоритета сообщения имеется специальная функция, возвращающая очередь соответствующего компонента, в которую необходимо положить преданное ей в качестве параметра сообщение.

Основными моментами, могущими вызвать проблемы при реализации данного алгоритма, являются: во-первых, пробуждение потока в нужное время (худшим вариантом было бы просто циклическое отслеживание входящих сообщений), во-вторых, возможность одновременного обращения к одной и той же очереди двух потоков - источника и приемника. Ниже предлагаются основные решения, позволяющие корректно реализовать описанный выше механизм.

Очереди сообщений, по сути очереди указателей, могут быть реализованы стандартными классами очередей из библиотеки STL, входящей в набор стандартных библиотек C++. (Класс-шаблон queue, реализующий очередь объектов, подчиняющуюся стандартной дисциплине FIFO). Достаточно создать наследника данного класса, заменив в нем функции записи и чтения. Таким образом, к каждому объекту-очереди будет привязан поток, для которого она является входящей (посредством некоторого поля типа HANDLE), а функции чтения и записи будут защищены от совместного использования специфической для данного объекта-очереди критической секцией. Кроме того, операция записи в очередь будет пробуждать соответствующий поток. *Таким образом, операции записи или чтения из очередей являются неделимыми во времени.*

Однако при детальном рассмотрении можно обнаружить недостаток данной схемы синхронизации. А именно, рассмотрим следующую ситуацию: допустим, один компонент, проверив все свои очереди сообщений и не найдя в них ни одного сообщения, перевел свой поток в усыпленное состояние. А в момент времени, когда происходила проверка последующих входящих очередей, другой компонент поставил в одну из уже проверенных очередей некоторое сообщение. Легко заметить, что первый поток никак не отреагирует на данное сообщение, поскольку он будет находиться еще в пробужденном состоянии, однако, не найдя сообщение в оставшихся очередях перейдет в усыпленное состояние. Причем, если второй поток будет ожидать какого-либо сообщения от первого, то поскольку первый никогда не проснется, то может произойти общий сбой системы - бесконечное ожидание (dead lock). Для решения описанной проблемы была разработана особая схема синхронизации, основанная на способах усыпления и пробуждения потока, отличных от стандартных функций SuspendThread() и ResumeThread(), предоставляемых Win32 API.

Предлагаемая схема синхронизации описывается ниже.

4. Синхронизация при работе с очередями сообщений

Легко заметить, что поток может быть усыплен (или его работа на время приостановлена) с помощью функций ожидания, к которым относится в частности функция API WaitForSingleObject(). При этом в качестве объекта синхронизации можно рассматривать объект ядра - событие (event). Подробно синхронизация с помощью событий описана в [3]. Необходимо реализовать механизмы усыпления и пробуждения потока таким образом, что попытка пробуждения потока в тот момент, когда он усыплен, ведет к просто пробуждению, а если поток не был усыплен во время попытки пробуждения, то он не будет усыплен при последующей попытке усыпления.

Описанный алгоритм проиллюстрирован на рис.1

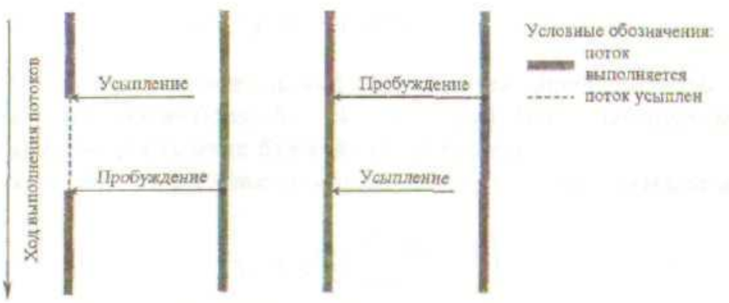

Рис. 1. Механизм специального усыпления и пробуждения потоков

При реализации описанного алгоритма ситуации, описанной в предыдущем разделе, возникнуть, очевидно, не может. Действительно, постановка в очередь любого сообщения приведет либо к пробуждению потока, если поток был усыплен, либо предотвратит его усыпление в конце итерации цикла просмотра очередей сообщений, и данное сообщение будет отработано на следующей итерации.

Идеально для реализации описанной схемы подходит такой объект ядра, как событие с автоматическим сбросом и свободным (non - signaled) начальным состоянием.

Уместно привести здесь краткое описание объекта ядра "событие" (подробно см. [3]). Событие - самая примитивная разновидность синхронизирующих объектов. События обычно применяются не для контроля за доступом к данным, как другие синхронизирующие объекты ядра, а для оповещения о том, что произошло какое-то событие, обычно окончание какой-либо операции. Существует два разных типа объектов "событие": со сбросом вручную (manual-reset events) и с автоматическим сбросом (auto-reset event). Событие может находиться в двух состояниях: свободное (signaled) и занятое (non-signaled). Усыпление потока производится установкой бесконечно долгого ожидания данного объекта ядра (посредством вызова функции WaitForSingleObject() со вторым параметром, равным константе INFINITE). Такой вызов приводит к бесконечно долгому ожиданию установки события в свободное состояние. Таким образом, если событие находилось в занятом состоянии, то произойдет усыпление потока вплоть до освобождения события. Если же событие находилось в свободном состоянии, то усыпления потока не произойдет. В любом случае, поскольку мы рассматриваем событие с автоматически сбросом, после выполнения данной функции событие будет переведено в занятое состояние. Пробуждение потока, усыпленного таким образом, производится установкой события в свободное состояние функцией SetEvent(). Таким образом, *попытка пробуждения потока, находящегося в спящем состоянии*, приведет к его пробуждению, а попытка пробуждения выполняющегося потока приведет к тому, что поток не будет усыплен в конце текущей итерации. Таким образом, любое сообщение, попадающее в очереди сообщений данного компонента системы (потока), будет гарантированно обработано независимо от времени его постановки в очередь.

Что касается реализации такого подхода, то для каждого потока (компонента), имеющего хотя бы одну входящую очередь, необходимо создать указанный объект ядра - событие, дескриптор которого и будет передаваться очередям для пробуждения данного потока. Любая постановка сообщения в одну из входящих очередей некоторого компонента системы будет уведомлять поток, обрабатывающий такие очереди, о том, что следует либо проснуться, либо предотвратить его усыпление, что приведет к гарантированной обработке всех входящих сообщений.

Литература

[1] Васильчиков В.В. Средства параллельного программирования для вычислительных систем с динамической балансировкой загрузки. Ярославль, 2001.

[2] Васильчиков В.В. Алгоритм динамической балансировки загрузки для распределенного выполнения рекурсивно- параллельных программ //Моделирование и анализ информационных систем. Вып. 4. Ярославль. 1997.

[3] Рихтер Дж. Windows для профессионалов: Программирование для Windows 95 и Windows NT 4 на базе Win32 API / Пер. с англ. М.: Издат. отдел "Русская Редакция" ТОО "Channel Trading Ltd.". 1997.

Модел. и анализ информ. систем. Т.9, №2 (2002) 32–44

УДК 519.22

О состоятельном многомерном непараметрическом критерии однородности

Левин А.Ю.
Ярославский государственный университет
150 000, Ярославль, Советская, 14

Изучается многомерный непараметрический критерий однородности, проверяющий гипотезу о том, что несколько выборок в $\mathbf{R}^N$ соответствуют одной и той же генеральной совокупности. Статистика γ критерия связана с понятием близости выборочных элементов. Распределение γ зависит от неизвестных плотностей, т.е. γ-критерий не свободен от распределения. Но он непараметричен в том смысле, что не требует априорных предположений о плотностях. Исследование поведения $\mathbf{E}\gamma$ и $\mathbf{D}\gamma$ (асимптотика, равномерные оценки) показывает, что критерий состоятелен против любых альтернатив.Критерий применим и в бесконечномерном случае.

Так как для $N = 1$ известен ряд классических критериев однородности, основная сфера применимости критерия связана с многомерностью и отсутствием информации о распределениях.

1. Введение. Основной результат

1.1. Статья посвящена классической проблеме однородности для многомерного случая. На распределения не накладывается ограничений (кроме абсолютной непрерывности), так что проблема рассматривается в непараметрической постановке.

Подробнее, рассматриваются $m(\geq 2)$ независимых выборок в $\mathbf{R}^N$

$$S_i = \{x_{i1}, x_{i2}, ..., x_{in_i}\}, \quad i = 1, 2, ..., m,$$

с интегрируемыми по Лебегу плотностями $p_i, \ldots, p_m$ соответственно. Проверке подлежит гипотеза однородности

$$H_0 : p_1(x) = p_2(x) = \ldots = p_m(x) \text{ почти всюду в } \mathbf{R}^N. \tag{1.1}$$

Обозначим через S объединенную выборку объема $n = n_1 + \ldots + n_m$. Как обычно, предполагается, что с ростом n величины $n_1 n^{-1}, \ldots, n_m n^{-1}$ стремятся к каким-либо ненулевым пределам:

$$n_i n^{-1} \to s_i (\neq 0), \quad i = 1, \ldots, m \qquad (n \to \infty). \tag{1.2}$$

Пусть $\theta_k \quad (0 \leq \theta_k \leq n_k)$ - число элементов выборки S_k, ближайший к которым среди остальных $n-1$ элементов S принадлежит той же выборке $S_k \quad (k = 1, \ldots, m)$. Непрерывность распределений обеспечивает почти наверное единственность этих ближайших элементов.

Основной результат. *Критерий, отклоняющий гипозезу H_0 при больших значениях статистики*

$$\gamma_n = \sqrt{n} \left(\sum_1^m \frac{\theta_k}{n_k} - 1 \right), \tag{1.3}$$

состоятелен против любых альтернатив.

Подробнее, имеют место два утверждения:

а) *если гипотеза H_0 верна, то для любого $\epsilon > 0$ можно указать постоянную $C(\epsilon, m)$, зависящую лишь от ϵ и m, такую что при всех n*

$$\mathbf{P}(\gamma_n < C(\epsilon, m)) > 1 - \epsilon; \tag{1.4}$$

б) *если H_0 не верна (т.е. хотя бы две из плотностей $p_1, \ldots, p_m$ различны на множестве положительной меры), то при любом C*

$$\mathbf{P}(\gamma_n < C) \to 0 \quad (n \to \infty). \tag{1.5}$$

Критерий, основанный на этих фактах, будем называть γ-критерием. Полной однозначности здесь, правда, нет, так как имеются различные модификации в связи с пороговыми значениями (по поводу конкретизации $C(\epsilon, m)$ и возможных уточнений см. §§6, 7).

Ради простоты обозначений ниже обычно будем применять краткие записи типа "γ_n" (иногда опуская и сам индекс n), "$n \to \infty$", игнорируя векторный характер аргумента $\bar{n} = (n_1, \ldots, n_m)$. Так, запись $n \to \infty$ означает, что речь идет о любой последовательности векторов $\bar{n} = \bar{n}\ (n)$, таких что сумма компонент вектора $\bar{n}\ (n)$ равна $n \quad (n = 1, 2, \ldots)$ и выполнено (1.2). Критерии, состоятельные против всех альтернатив, будем часто называть полностью состоятельными или просто ПС-критериями.

Будучи непараметрическим, γ-критерий не свободен от распределения: при выполнении H_0 статистика γ *имеет распределение, зависящее (помимо $n_1, \ldots, n_m$) также и от размерности N и от общей* плотности p_0. Равномерный характер соотношения (1.4) показывает, что эту принципиальную трудность (связанную с многомерностью) удается преодолеть.

1.2. Так как $\quad 0 \le \theta_k \le n_k \quad (k = 1, \ldots, m)$, то выполняются очевидные неравенства $\quad -\sqrt{n} \le \gamma_n \le (m-1)\sqrt{n}$. Ниже (§§2-5) устанавливаются некоторые факты, относящиеся к $\mathbf{E}\gamma_n$, $\mathbf{D}\gamma_n$ - как при выполне*нии, так и при нарушении H_0. Как следствие будут получены (§6) явные значения $C(\epsilon, m)$, позволяющие* отклонить H_0 (с уровнем доверия не ниже требуемого). Способ повышения количественной точности указан в §7.

В заключительной части (§8) обсуждаются возможности обобщений критерия - использование различных метрик, возможность отказа от конечномерности, независимости и др.

1.3. Достоинства γ-критерия - полная состоятельность (и, следовательно, асимптотическая несмещенность), непараметричность, многомерный характер, робастность - ни в коей мере не означают, что применение его всегда целесообразно. Так, при $N = 1$ оно вряд ли оправдано, поскольку здесь имеются классические ПС-критерии однородности (Колмогорова-Смирнова, Крамера - Мизеса - Смирнова, Вальда - Волфовица, Реньи и др.,см.,напр., [1-7]) - также непараметрические и притом эффективно использующие специфику скалярного случая. Далее, по сравнению с многомерными критериями, состоятельными против конкретных типов альтернатив (например, с классическим критерием χ^2 или с критериями, предполагающими нормальность распределений), γ-критерий при таких альтернативах проявляет повышенную "осторожность" при отклонении H_0. Меньшая мощность его в подобных ситуациях (понятие мощности приобретает смысл при конкретизации частных случаев) есть, так сказать, плата за общность.

Применение γ-критерия представляется целесообразным, когда в многомерной задаче при отсутствии надежной информации о распределениях и альтернативах требуется выяснить, имеются ли какие-либо нарушения однородности. Естественно, слабо выраженная неоднородность выявляется лишь при выборках большого объема; с другой стороны, значительное различие распределений обнаруживается, как правило, уже при небольших выборках.

Статистика γ, по-видимому, впервые рассматривалась в [8]. Там же были приведены формулировки большинства результатов настоящей работы.

2. $\mathbf{E}\gamma_n$ в случае однородности

При выполнении гипотезы $H_0 \quad \mathbf{E}\gamma_n$ легко вычисляется. Ближайшим к элементу x_{ki} выборки S_k может равновероятно оказаться любой из $n-1$ остальных элементов S. Поэтому вероятность того, что этот ближайший элемент окажется из S_k, равна $(n_k - 1)(n-1)^{-1}$. Поскольку θ_k есть сумма n_k индикаторов таких событий, то при всех k

$$\mathbf{E}\theta_k = n_k \frac{n_k - 1}{n-1}, \quad \mathbf{E}\gamma_n = \sqrt{n}\left(\sum_{k=1}^{m} \frac{n_k - 1}{n-1} - 1\right) = -\sqrt{n}\frac{m-1}{n-1}.$$

Итак, величина $\mathbf{E}\gamma_n$ отрицательна и стремится к 0 при $n \to \infty$.

3. Асимптотика $\mathbf{E}\gamma_n$ в общем случае

3.1. Ниже будет показано, что в случае неоднородности предел $\mathbf{E}\gamma_n/\sqrt{n} \quad (n \to \infty)$, по контрасту с однородным случаем, положителен.

Через $\mathbf{R}^N_+$ здесь и далее обозначается множество таких точек Лебега (см.,напр.[9]) для функций $p_1, \ldots, p_m$, в которых по крайней мере одна из этих функций отлична от 0. Пусть, далее, $A_{kn}(z) = A_k(z, \overline{n})$ - событие, состоящее в том, что ближайший к $z(\in \mathbf{R}^N)$ элемент S принадлежит выборке S_k $(1 \le k \le m)$.

Теорема 1. *Для любой точки* $z \in \mathbf{R}^N_+$ и любого k $(k = 1, \ldots, m)$

$$\mathbf{P}\Big(A_{kn}(z)\Big) \to g_k(z) = \frac{s_k p_k(z)}{s_1 p_1(z) + \ldots + s_m p_m(z)}, \quad (n \to \infty). \tag{3.1}$$

Доказательство теоремы 1 начнем с частного случая.

3.2. **Лемма 1.** *Пусть $p_1, \ldots, p_m$ постоянны в некоторой окрестности $U = U(z)$ точки z. Тогда имеет место соотношение (3.1).*

Обозначим через α_n (α_{in}) - число элементов выборки S (S_i), которые попадут в U $(i = 1, \ldots, m)$. Ясно,что $\alpha_n = \alpha_{1n} + \ldots + \alpha_{mn}$.

Пусть $\alpha_n > 0$. Все α_n элементов, которые попадут в U, имеют одинаковое условное распределение - а именно, равномерное в U, в силу условия леммы. Так как они, очевидно, независимы, то любой из них равновероятно с остальными может оказаться ближайшим к z. Поэтому

$$\mathbf{P}\Big(A_{kn}(z)|\alpha_{1n}, \ldots, \alpha_{mn}\Big) = \alpha_{kn}\alpha_n^{-1} \quad (\alpha_n > 0); \tag{3.2}$$

при $\alpha_n = 0$ правая часть заменяется величиной $\mathbf{P}\Big(A_{kn}(z)|\alpha_n = 0\Big)(k = 1, \ldots, m)$, несущественной для дальнейшего.

Пусть $|U|$ - объем U. В силу (1.2) и закона (усиленного) больших чисел при любых $i, k (1 \le i, k \le m)$

$$\frac{\alpha_{in}}{n_i} \overset{\text{п.н.}}{\to} |U| p_i(z), \qquad (n \to \infty),$$

$$\frac{\alpha_{kn}}{\alpha_n} = \frac{n_k}{n} \cdot \frac{\alpha_{kn}}{n_k} \left(\frac{\alpha_n}{n}\right)^{-1} = \frac{n_k}{n} \cdot \frac{\alpha_{kn}}{n_k} \left(\frac{n_1}{n} \cdot \frac{\alpha_{1n}}{n_1} + \ldots + \frac{n_m}{n} \cdot \frac{\alpha_{mn}}{n_m}\right)^{-1} \overset{\text{п.н.}}{\to}$$

$$s_k |U| p_k(z) \Big(s_1 |U| p_1(z) + \ldots + s_m |U| p_m(z)\Big)^{-1} = g_k(z) \quad (n \to \infty).$$

Поскольку $z \in \mathbf{R}^N_+$, то $\mathbf{P}(\alpha_n = 0) \to 0$ при $n \to \infty$, так что случаем $\alpha_n = 0$ можно пренебречь; итак,

$$\mathbf{P}\left(A_{kn}(z)|\alpha_{1n}, \ldots, \alpha_{mn}\right) \overset{\text{п.н.}}{\to} g_k(z) \quad (n \to \infty), k = 1, \ldots, m.$$

Случайные величины в левой части равномерно ограничены, поэтому тот же предел должны иметь их математические ожидания:

$$\mathbf{P}\Big(A_{kn}(z)\Big) = \mathbf{EP}\left(A_{kn}(z)|\alpha_{1n}, \ldots, \alpha_{mn}\right) \to g_k(z) \quad (n \to \infty), k = 1, \ldots, m.$$

Лемма 1 доказана.

3.3. В общем случае воспользуемся леммой 1 в сочетании с мажорированием и последующим предельным переходом.

Пусть v_N - объем единичного шара в $\mathbf{R}^N$ (явное выражение здесь не требуется), т.е. $v_N r^N$ - объем шара радиуса r. Зафиксируем произвольную точку $z \in \mathbf{R}^N_+$. Поскольку она является точкой Лебега любой из плотностей p_i, для шаров U_r радиуса r с центром в z

$$\frac{1}{v_N r^N} \int\limits_{U_r} p_i(x) dx \to p_i(z), \quad i = 1, \ldots, m \ (r \to 0). \tag{3.3}$$

Если ξ_i обладает плотностью распределения p_i, то для функции распределения F_i скалярной величины $\| \xi_i - z \|$ имеем поэтому (при $t > 0$)

$$F_i(t) = \int\limits_{U_t} p_i(x) dx = p_i(z) v_N t^N + o(t^N) \qquad (t \to 0). \tag{3.4}$$

Выберем произвольное ϵ $(0 < \epsilon < 1)$ и положим

$$a_{i1} = (1 - \epsilon) p_i(z), \quad a_{i2} = p_i(z) + \epsilon, \quad i = 1, \ldots, m. \tag{3.5}$$

Ясно, что $0 \le a_{i1} \le p_i(z) < a_{i2}$, причем среднее неравенство является строгим при $p_i(z) \ne 0$. Поэтому, с учетом (3.4), на некотором промежутке $[0, r], r = r(\epsilon) > 0$, справедливы неравенства

$$a_{i1} v_N t^N \le F_i(t) \le a_{i2} v_N t^N \le 1, \quad i = 1, \ldots, m \quad (0 \le t \le r). \tag{3.6}$$

Для $i = 1, \ldots, m$ введем функции F_{i1}, F_{i2}, полагая

$$F_{i1}(t) = a_{i1} v_N t^N, \quad F_{i2}(t) = a_{i2} v_N t^N, \qquad i = 1, \ldots, m \quad (0 \le t \le r) \tag{3.7}$$

и продолжая их на всю числовую прямую с сохранением неравенств

$$F_{i1}(t) \le F_i(t) \le F_{i2}(t), \qquad i = 1, \ldots, m \quad (-\infty < t < \infty) \tag{3.8}$$

и свойств функций распределения (абсолютно непрерывных); это, очевидно, возможно. Функции распределения F_{i1}, F_{i2} соответствуют (в том же смысле, в каком F_i соответствуют p_i), в частности, N-мерным плотностям

$$p_{ij}(x) = \frac{F'_{ij}(\| x - z \|)}{N v_n \| x - z \|^{N-1}}, \qquad i = 1, \ldots, m, \quad j = 1, 2 \quad (x \in \mathbf{R}^N). \tag{3.9}$$

Если $x \in U_z$, то $0 \le \| x - z \| \le r$; поэтому из (3.7), (3.9) следует, что

$$p_{ij}(x) \equiv a_{ij}, \qquad i = 1, \ldots, m, \quad j = 1, 2 \quad (x \in U_r). \tag{3.10}$$

3.4. Пусть $k \quad (1 \le k \le m)$ фиксировано. Наряду с задачей о предельном значении $\mathbf{P}\Big(A_{kn}(z)\Big)$, рассмотрим две вспомогательные задачи о предельных значениях $\mathbf{P}\Big(A'_{kn}(z)\Big), \mathbf{P}\Big(A''_{kn}(z)\Big)$. Здесь оба события $A'_{kn}(z)$, $A''_{kn}(z)$, как и $A_{kn}(z)$, означают, что ближайшим к z элементом S будет элемент выборки S_k; разница лишь в исходных распределениях. Именно, для события $A'_{kn}(z)$ $(A''_{kn}(z))$ плотности $p_1, \ldots, p_m$ заменяются соответственно вспомогательными плотностями

$$p_{11}, \ldots, p_{k-11}, p_{k2}, p_{k+11}, \ldots, p_{m1} \tag{3.11}$$

$$(p_{12}, \ldots, p_{k-12}, p_{k1}, p_{k+12}, \ldots, p_{m2}). \tag{3.12}$$

Так как $p_{i2}(z) = a_{i2} > 0 \quad (i = 1, \ldots, m)$, не все функции (3.11) равны 0 в точке z; то же относится и к функциям (3.12). В силу (3.10) применима лемма 1, откуда

$$\mathbf{P}\Big(A'_{kn}(z)\Big) \to \frac{s_k\Big(p_k(z) + \epsilon\Big)}{(1-\epsilon)\sum\limits_{i \ne k} s_i p_i(z) + s_k\Big(p_k(z) + \epsilon\Big)} \qquad (n \to \infty), \tag{3.13}$$

$$\mathbf{P}\Big(A''_{kn}(z)\Big) \to \frac{(1-\epsilon) s_k p_k(z)}{\sum\limits_{i \ne k} s_i\Big(p_i(z) + \epsilon\Big) + (1-\epsilon) s_k p_k(z)} \qquad (n \to \infty). \tag{3.14}$$

3.5.**Лемма 2.**_При любых k, n справедлива двусторонняя оценка_

$$\mathbf{P}(A''_{kn}) \le \mathbf{P}(A_{kn}) \le \mathbf{P}(A'_{kn}). \tag{3.15}$$

Ввиду аналогии ограничимся доказательством второго неравенства. Пусть $\xi_{i1}, \ldots, \xi_{in_i}$ - расстояния от z до соответствующих элементов выборки S_i. Эти n_i случайных величин имеют введенную выше функцию распределения F_i и независимы. Положим еще

$$\xi_i = min\{\xi_{i1}, \ldots, \xi_{in_i}\}, i = 1, \ldots, m; \quad \xi = min\{\xi_1, \ldots, \xi_{k-1}, \xi_{k+1}, \ldots, \xi_m\},$$

а $\xi'_{i1}, \ldots, \xi'_{in_i}, \xi'_i, \xi'$ имеют аналогичный смысл, но при плотностях (3.11) (вместо $p_1, \ldots, p_m$).

Любая из величин $\xi'_{k1}, \ldots, \xi'_{kn_k}$ имеет функцию распределения $F_{k2}(t) \ge F_k(t)$ $(-\infty < t < \infty)$ и поэтому ξ'_k стохастически не больше ξ_k (что сразу следует из выражения для функции распределения минимума). Повторяя то же соображение, заключаем, что при $i \ne k \quad \xi'_{il}$ стохастически не меньше ξ_{il} (с учетом неравенства $F_i \ge F_{i1}$) и, следовательно, ξ' стохастически не меньше ξ. Величины, входящие в неравенство

$$\mathbf{P}(\xi_k < \xi) \le \mathbf{P}(\xi'_k < \xi') \tag{3.16}$$

независимы, причем $\xi'_k(\xi)$ стохастически не больше $\xi_k(\xi')$. Итак, (3.16), т.е. второе из неравенств (3.15), справедливо. Лемма 2 доказана.

Осталось заметить, что при $\epsilon \to 0$ правые части (3.15) и (3.16) стремятся к правой части (3.1). Переход к пределу при $n \to \infty, \epsilon \to 0$ с учетом леммы 2, завершает доказательство теоремы 1.

Она допускает прозрачную интерпретацию: для выборок S_k предельные вероятности содержать ближайший (среди всех выборочных) элемент к фиксированной "хорошей" точке z, пропорциональны ожидаемым количествам элементов S_k в малой окрестности $z \quad (k = 1, \ldots, m)$.

3.6. Пусть B_{kjn} - событие, состоящее в том, что ближайший к x_{kj} среди $n-1$ остальных элементов S окажется в выборке S_k.

Следствие 1. *При любых k, j и любом $x \in \mathbf{R}_+^N$*

$$\mathbf{P}(B_{kjn}/x_{kj} = x) \to g_k(x) \quad (n \to \infty). \tag{3.17}$$

В самом деле, единственное различие по сравнению с теоремой 1 состоит в том, что здесь из выборки S_k исключается элемент x_{kj}, так что объем n_k этой выборки должен быть теперь заменен на $n_k - 1$. Эта деталь несущественна для асимптотики левой части (3.17) при $n \to \infty$.

3.7. Откажемся теперь от фиксации x_{kj} и вернемся к априорной интерпретации x_{kj} как случайной величины.

Следствие 2. *При любых k, j (dx - элемент объема в $\mathbf{R}^N$)*

$$\mathbf{P}\Big(B_{kjn}\Big) \to \int\limits_{\mathbf{R}_+^N} g_k(x)p_k(x)dx \qquad (n \to \infty). \tag{3.18}$$

Действительно, по формуле полной вероятности:

$$\mathbf{P}\Big(B_{kjn}\Big) = \int\limits_{\mathbf{R}_+^N} \mathbf{P}\Big(B_{kjn}/x_{kj} = x\Big)p_k(x)dx, \quad j = 1, \ldots, n_k. \tag{3.19}$$

Интегрирование по $\mathbf{R}_+^N$ вместо $\mathbf{R}^N$ оправдано: можно ограничиться интегрированием лишь по тем x, где $p_k(x) > 0$, и, кроме того, почти все точки $\mathbf{R}^N$ являются точками Лебега суммируемых функций $p_1, \ldots, p_m$ (см. [9]). Далее, подинтегральные функции в (3.19) имеют общую суммируемую в $\mathbf{R}_+^N$ мажоранту $p_k(x)$. Поэтому возможен предельный переход под знаком интеграла, и (3.19) с учетом (3.17) дает (3.18).

3.8. *Вернемся к статистике γ_n. Пусть I_{kj} - индикатор события $B_{kj} = B_{kjn}$. Тогда при $n \to \infty$*

$$\mathbf{E}\theta_k = \mathbf{E}\Big(\sum_{j=1}^{n_k} I_{kj}\Big) = \sum_{j=1}^{n_k} \mathbf{P}(B_{kj}) = n_k \mathbf{P}(B_{k1}) \to n_k \int\limits_{\mathbf{R}_+^N} g_k(x)p_k(x)dx,$$

$$\mathbf{E}\Big(\frac{\gamma_n}{\sqrt{n}}\Big) = \sum_{k=1}^{m} \frac{\mathbf{E}\theta_k}{n_k} - 1 \to \int\limits_{\mathbf{R}_+^N} \frac{s_1 p_1^2(x) + \ldots + s_m p_m^2(x)}{s_1 p_1(x) + \ldots + s_m p_m(x)} dx - 1. \tag{3.20}$$

3.9. **Следствие 3.** *При невыполнении гипотезы H_0*

$$\lim_{n \to \infty} \mathbf{E}\Big(\frac{\gamma_n}{\sqrt{n}}\Big) > 0. \tag{3.21}$$

Для доказательства воспользуемся неравенством Коши:

$$\Big(\sum_{i=1}^{m} s_i p_i\Big)^2 = \Big(\sum_{i=1}^{m} \sqrt{s_i} \cdot \sqrt{s_i p_i^2}\Big)^2 \le \sum_{i=1}^{m} s_i p_i^2, \tag{3.22}$$

поскольку сумма всех s_i равна 1. При этом на множестве положительной меры знак неравенства в (3.22) должен быть строгим. Иначе оказалось бы, что почти всюду пропорциональны векторы

$$(\sqrt{s_1}, \ldots, \sqrt{s_m}), (\sqrt{s_1 p_1^2(x)}, \ldots, \sqrt{s_m p_m^2(x)}),$$

т.е. $p_1(x) = \ldots = p_m(x)$, что противоречит предположению.

С учетом (3.20), (3.22) получаем требуемое неравенство:

$$\lim_{n\to\infty} \mathbf{E}\Big(\frac{\gamma_n}{\sqrt{n}}\Big) > \int\limits_{\mathbf{R}_+^N} \frac{\Big(s_1p_1(x)+\ldots+s_mp_m(x)\Big)^2}{s_1p_1(x)+\ldots+s_mp_m(x)}dx - 1 =$$

$$= \int\limits_{\mathbf{R}^N} \Big(s_1p_1(x)+\ldots+s_mp_m(x)\Big)dx - 1 = s_1+\ldots+s_m - 1 = 0.$$

4. Оценка $\mathbf{D}\gamma_n$ при гипотезе H_0

4.1. В §3 изучались вопросы асимптотики (при $n\to\infty$), тогда как ниже рассматриваются значения $\mathbf{D}\gamma_n$ при произвольных n.

Теорема 2. *Если справедлива гипотеза H_0, то при любом $n(\geq 4)$*

$$\mathbf{D}\gamma_n \leq \frac{2(n-m)(m-1)}{n-3} \quad (n\geq 4). \tag{4.1}$$

Правая часть здесь не превосходит $4(m-1)$ и с ростом n сходится к $2(m-1)$. Неравенство (4.1) будет получено как следствие более точной (но и более громоздкой) оценки, куда входят $n_1,\ldots,n_m$.

Из определения γ_n непосредственно следует, что

$$\frac{1}{n}\mathbf{D}\gamma_n = \sum_{k=1}^m \frac{1}{n_k^2}\mathbf{D}\theta_k + \sum_{r\neq s}\frac{1}{n_rn_s}cov(\theta_r,\theta_s). \tag{4.2}$$

Переход от (4.2) к (4.1) требует достаточно точных оценок для слагаемых в правой части (4.2). Этому посвящены последующие пункты.

4.2. Итак, пусть выполнена гипотеза H_0, т.е. все n элементов различных выборок S_i имеют одну и ту же плотность $p(x)$. Введем ряд обозначений.Прежде всего удобно избавиться от двойной индексации элементов x_{ik} и обозначать их просто как $x_1,x_2,\ldots,x_n$. При этом множество индексов $H_1=\{1,\ldots,n_1\}$ отвечает элементам выборки S_1 и т.д., вплоть до множества $H_m=\{n-n_m+1,\ldots,n\}$, отвечающего выборке S_m. Ближайший к x_i среди остальных $n-1$ элементов S обозначим через $f(x_i)$, а индикатор события $f(x_i)=x_j$ - через I_{ij}. (Отметим, что эти индикаторы, вообще говоря, зависимы).

Очевидно, $I_{ii}=0,\quad \mathbf{E}I_{ij}=(n-1)^{-1}\quad$ при $i\neq j\qquad (1\leq i,j\leq n)$.

Положим также

$$a=\mathbf{E}\Big(I_{12}I_{34}\Big), b=\mathbf{E}\Big(I_{12}I_{23}\Big), c=\mathbf{E}\Big(I_{12}I_{32}\Big), d=\mathbf{E}\Big(I_{12}I_{21}\Big). \tag{4.3}$$

Поскольку все x_i одинаково распределены и независимы, их совместное распределение перестановочно (т.е. симметрично). Поэтому $\mathbf{E}\Big(I_{ij}I_{kl}\Big)=a$ при любых различных индексах i,j,k,l, $\mathbf{E}\Big(I_{ij}I_{jk}\Big)=b$ при любых различных индексах i,j,k и т.д.

4.3. Величины (4.3) зависят от плотности p; число их можно сократить до двух, поскольку

$$(n-2)b+d=(n-3)a+b+c=(n-1)^{-1}. \tag{4.4}$$

В самом деле, индикаторы $I_{12},I_{13},\ldots,I_{1n}$ отвечают событиям, образующим разбиение. Следовательно (так как $\mathbf{E}(I_{23}I_{12})=b$)

$$(n-1)^{-1}=\mathbf{E}I_{21}=\mathbf{E}(I_{21}I_{12})+\ldots+\mathbf{E}(I_{21}I_{1n})=d+(n-2)b,$$

$$(n-1)^{-1}=\mathbf{E}I_{23}=\mathbf{E}(I_{23}I_{12})+\mathbf{E}(I_{23}I_{13})+\ldots+\mathbf{E}(I_{23}I_{1n})=b+c+(n-3)a.$$

4.4. Как уже отмечалось в §2, при любом k

$$\theta_k=\sum_{i,j\in H_k} I_{ij},\quad \mathbf{E}\theta_k=\sum_{i,j\in H_k}\mathbf{E}I_{ij}=\frac{n_k(n_k-1)}{n-1} \tag{4.5}$$

Переходя к ковариациям величин θ_k, заметим, что, в силу (4.5)

$$\mathbf{E}(\theta_r\theta_s) = \sum_{i,j\in H_r} \sum_{k,l\in H_s} \mathbf{E}(I_{ij}I_{kl}). \tag{4.6}$$

При $r \neq s$ множества H_r и H_s не пересекаются, так что все ненулевые слагаемые (отвечающие неравенствам $i \neq j, k \neq l$) равны a. Число их, очевидно, есть $n_r(n_r-1)n_s(n_s-1)$. С учетом (4.5) имеем при $r \neq s$

$$cov\Big(\theta_r,\theta_s\Big) = n_r(n_r-1)n_s(n_s-1)a - \frac{n_r(n_r-1)n_s(n_s-1)}{(n-1)^2} =$$

$$n_rn_s(n_r-1)(n_s-1)\Big(a - \frac{1}{(n-1)^2}\Big) \quad (1 \le r,s \le m, \quad r \neq s).$$

Отсюда находим второе слагаемое в (4.2):

$$\sum_{r\neq s}\frac{1}{n_rn_s}cov\Big(\theta_r,\theta_s\Big) = \Big(a - \frac{1}{(n-1)^2}\Big)\sum_{r\neq s}(n_r-1)(n_s-1) =$$

$$\Big(a - \frac{1}{(n-1)^2}\Big)\Big(\sum_{r=1}^{m}(n_r-1)(n-n_r) - (m-1)(n-m)\Big). \tag{4.7}$$

4.5. Переходим к первому слагаемому. При $r = s$ (4.6) принимает вид

$$\mathbf{E}\theta_r^2 = \sum_{i,j,k,l\in H_r} \mathbf{E}(I_{ij}I_{kl}). \tag{4.8}$$

Здесь счет усложняется, т.к. ненулевые слагаемые уже не совпадают.

Отбросим нулевые слагаемые, возникающие, если $i = j$ или $k = l$ или $i = k, j \neq l$. Слагаемые с $i = k, j = l$ дают вклад

$$\sum_{i,j\in H_r} \mathbf{E}I_{ij}^2 = \sum_{i,j\in H_r} \mathbf{E}I_{ij} = \frac{n_r(n_r-1)}{n-1}\Big(= \mathbf{E}\theta_r\Big).$$

Остальные слагаемые имеют один из типов a, b, c, d и количества их равны соответственно

$$n_r(n_r-1)(n_r-2)(n_r-3); 2n_r(n_r-1)(n_r-2); n_r(n_r-1)(n_r-2); n_r(n_r-1).$$

Суммируя, находим $\mathbf{E}\theta_r^2$, откуда с учетом (4.5)

$$\mathbf{D}\theta_r = n_r(n_r-1)\Big(\frac{1}{n-1} + (n_r-2)(n_r-3)a + (n_r-2)(2b+c) + d\Big) - \frac{n_r^2(n_r-1)^2}{(n-1)^2}.$$

Ввиду (4.4) можно выразить $\mathbf{D}\theta_r \quad (r = 1,\ldots,m)$ лишь через a и b:

$$\mathbf{D}\theta_r = n_r(n_r-1)\Big((n_r-2)(n_r-3)a + (n_r-2)\Big(b - (n-3)a + \frac{1}{n-1}\Big) - (n-2)b +$$

$$\frac{2}{n-1} - \frac{n_r(n_r-1)}{(n-1)^2}\Big) = n_r(n_r-1)(n-n_r)\Big(\frac{n_r}{(n-1)^2} - (n_r-2)a - b\Big). \tag{4.9}$$

4.6. Объединяя (4.2),(4.7) и (4.9), находим

$$\frac{1}{n}\mathbf{D}\gamma_n = \sum_{k=1}^{m}\frac{(n_k-1)(n-n_k)}{n_k}\Big(\frac{n_k}{(n-1)^2} - (n_k-2)a - b\Big) +$$

$$\Big(a - \frac{1}{(n-1)^2}\Big)\Big(\sum_{k=1}^{m}(n_k-1)(n-n_k) - (m-1)(n-m)\Big) =$$

$$\frac{(m-1)(n-m)}{(n-1)^2} + \lambda_1 a - \lambda_2 b \tag{4.10}$$

с положительными коэффициентами

$$\lambda_1 = mn - n + m^2 + m - 2n\sum_{k=1}^{m}\frac{1}{n_k}, \quad \lambda_2 = \sum_{k=1}^{m}\frac{(n_k-1)(n-n_k)}{n_k}.$$

Положительность λ_1 усматривается из неравенства

$$n\sum_{k=1}^{m}\frac{1}{n_k} = (n_1+\ldots+n_m)(\frac{1}{n_1}+\ldots+\frac{1}{n_m}) = 1 + \frac{n_1}{n_2}+\ldots+\frac{n_m}{n_{m-1}} + 1 \le$$

$$1 + \frac{n_1}{2}+\ldots+\frac{n_m}{2}+1 = m + \frac{1}{2}(m-1)n.$$

4.7. В (4.10) $\mathbf{D}\gamma_n$ выражена через величины a, b (как правило, неизвестные). Избавимся от них, переходя к неравенству. В силу (4.4) $a \le (n-1)^{-1}(n-3)^{-1}$. Поскольку $\lambda_1, \lambda_2, b \ge 0$, имеем оценку

$$\frac{1}{n}\mathbf{D}\gamma_n \le \frac{(m-1)(n-m)}{(n-1)^2} + \frac{1}{(n-1)(n-3)}\Big(mn - n + m^2 + m - 2n\sum_{k=1}^{m}\frac{1}{n_k}\Big).$$

Так как $n_1+\ldots+n_m = n$ и все n_i положительны, сумма величин, обратных к $n_1,\ldots,n_m$, не меньше, чем m^2n^{-1}. Отсюда непосредственно следует теорема 2:

$$\mathbf{D}\gamma_n \le n\Big(\frac{(m-1)(n-m)}{(n-1)^2} + \frac{(m-1)(n-m)}{(n-1)(n-3)}\Big) \le \frac{2(m-1)(n-m)}{n-3}.$$

5. Асимптотическая оценка сверху $\mathbf{D}\gamma_n$ в общем случае

5.1. Рассмотрим теперь поведение $\mathbf{D}\gamma_n$ без предположения об однородности. Ограничимся следующим предложением.

Теорема 3. *Имеет место оценка*

$$\mathbf{D}\gamma_n = o(n) \quad (n\to\infty). \tag{5.1}$$

В связи с перенумерацией элементов в п.4.2 сейчас удобно ввести дополнительную индексацию также для выборок и их объемов. Положим для любого $k (1\le k\le n)$ $\quad S^k = S_i, n^k = n_i$, если $x_k \in S_i$ (напр.,$S^1 = \cdots = S^{n_1} = S_1, n^1 = \cdots = n^{n_1} = n_1$ и т.д.). Пусть I_k есть индикатор события $f(x_k)\in S^k$, где, как и ранее, $f(x_k)$ - ближайший к x_k среди $n-1$ остальных элементов S $\quad (k=1,\cdots,n)$. В этих обозначениях

$$\gamma_n = \sqrt{n}\Big(\sum_{k=1}^{n}\frac{1}{n^k}I_k - 1\Big).$$

5.2. Теорема 3 будет доказана, если мы проверим, что

$$\frac{1}{n}\mathbf{D}\gamma_n = \sum_{k=1}^{n}\frac{1}{(n^k)^2}\mathbf{D}I_k + \sum_{i\ne j}\frac{1}{n^in^j}cov(I_i,I_j) = \Sigma_1 + \Sigma_2 = o(1) \quad (n\to\infty).$$

Можно ограничиться слагаемым Σ_2, так как $\Sigma_1 = O(n^{-1}) \quad (n\to\infty)$.

Пусть $c_{qr} = cov(I_i,I_j)$ при всех i,j таких, что $i\ne j, x_i\in S_q, x_j\in S_r$. Каждой неупорядоченной паре выборок S_q, S_r при $q\ne r$ в сумме Σ_2 отвечает $2n_qn_r$ одинаковых слагаемых, равных $n_q^{-1}n_r^{-1}c_{qr}$, а при $q=r$ - $n_r(n_r-1)$ одинаковых слагаемых, равных $n_r^{-2}c_{rr}$. Отсюда

$$|\Sigma_2| \le \sum_{q,r=1}^{m}|c_{qr}|.$$

При любом n в правую часть в качестве слагаемых входит фиксированное число (а именно, m^2) ковариаций. Поэтому бесконечная малость Σ_2 при $n\to\infty$ вытекает из следующего факта.

Лемма 3. *При всех q,r имеет место соотношение*

$$c_{qr}\to 0 \quad (n\to\infty; \quad 1\le q,r\le m).$$

Данное утверждение достаточно прозрачно: вряд ли может вызвать сомнение тот факт, что с ростом n зависимость между различными индикаторами сходит на нет. Итак, на "интуитивном" уровне теорему 3 можно считать подтвержденной.

5.3. Для полноты приведем подробное доказательство леммы 3.

Зафиксируем индексы q, r, а также точки $z', z'' \in \mathbf{R}^N$ такие, что

$$z' \neq z'', \quad z', z'' \in \mathbf{R}_+^N, \quad p_q(z') > 0, p_r(z'') > 0. \tag{5.1}$$

Выберем разбиение $\mathbf{R}^N$ на непересекающиеся борелевские множества H', H'' такие, что $z'(z'')$ - внутренняя точка $H'(H'')$ и

$$b_k' = \int\limits_{H'} p_k(x)dx \neq 0, \quad b_k'' = \int\limits_{H''} p_k(x)dx (= 1 - b_k') \neq 0, \quad k = 1, \cdots, m$$

(что, очевидно, возможно). Положим далее,

$$b' = s_1 b_1' + \cdots + s_m b_m', \quad b'' = s_1 b_1'' + \cdots + s_m b_m'' (= 1 - b'),$$

$$s_k' = s_k b_k' (b')^{-1}, \quad s_k'' = s_k b_k'' (b'')^{-1}, \quad k = 1, \cdots, m. \tag{5.2}$$

В соответствии с разбиением $\mathbf{R}^N$, каждая из выборок $S_1, \cdots, S_m, S$ также разбивается на две "подвыборки" S_1' и S_1'', $\cdots, S_m'$ и S_m'', S' и S'' из элементов, которые попадут в H' и H'' соответственно. Объемы этих подвыборок обозначим через n_1' и n_1'', $\cdots, n_m'$ и n_m'', n' и n''. Эти объемы - случайные величины и связаны очевидными соотношениями

$$n_k' + n_k'' = n_k \quad (k = 1, \cdots, m), \quad n' + n'' = n. \tag{5.3}$$

При этом, согласно закону больших чисел, почти наверное

$$\frac{n_k'}{n_k} \to b_k', \quad \frac{n_k''}{n_k} \to b_k'', \quad \frac{n'}{n} \to b', \quad \frac{n''}{n} \to b'' \quad (n \to \infty). \tag{5.4}$$

Элементы выборок S_k', S_k'' обладают сосредоточенными в H', H'' соответственно плотностями $(k = 1, \ldots, m)$

$$p_k'(z) = (b_k')^{-1} p_k(z) \quad (z \in H'), \quad p_k''(z) = (b_k'')^{-1} p_k(z) \quad (z \in H''). \tag{5.5}$$

Далее, элементы выборок $S_1', \ldots, S_m' (S_1'', \ldots, S_m'')$, очевидно, независимы. Роль величин s_k играют здесь определенные в (5.2) величины $s_k'(s_k'')$. В самом деле, в силу (5.4) (для s_k'' выкладка аналогична)

$$\frac{n_k'}{n'} = \frac{n_k}{n} \frac{n_k'}{n_k} \frac{n}{n'} \quad \xrightarrow{\text{п.н.}} \quad \frac{s_k b_k'}{b'} = s_k' \quad (n \to \infty), \quad k = 1, \cdots, m. \tag{5.6}$$

Итак, к "просеянным" выборкам $S_1', \cdots, S_m'$ $(S_1'', \cdots, S_m'')$ применимы результаты §3 с заменой p_k, n, n_k, s_k на p_k', n', n_k', s_k' $(p_k'', n'', n_k'', s_k'')$. Различие состоит в том, что объемы выборок теперь случайны и обычная сходимость $n_k n^{-1}$ к s_k заменена п.-н. -сходимостью в (5.6). Это, однако, не играет роли: для соответствующих разделов §3 имели значение лишь количества - заведомо случайные - элементов различных выборок $S_1, \cdots, S_m$, попавших в окрестность некоторой точки z.

Пусть событие $A_k'(z)$ $(A_k''(z))$ состоит в том, что ближайший к z элемент S' (S'') принадлежит S_k' (S_k''). Теорема 1 с учетом (5.2),(5.5) дает

$$\mathbf{P}(A_q'(z')) \to s_q' p_q'(z') \left(s_1' p_1'(z') + \cdots + s_m' p_m'(z')\right)^{-1} = g_q(z'), \tag{5.7}$$

$$\mathbf{P}(A_r''(z'')) \to s_r'' p_r''(z'') \left(s_1'' p_1''(z'') + \cdots + s_m'' p_m''(z'')\right)^{-1} = g_r(z'') \tag{5.8}$$

(при $n \to \infty$). События $A_q'(z'), A_r''(z'')$ зависимы (хотя области H', H'', как и выборки S', S'', не пересекаются); причиной зависимости является лишь связь (5.3) между величинами n_k' и n_k'', $k = 1, \cdots, m$. Поэтому $A_q'(z'), A_r''(z'')$ условно независимы при данном $\vec{n}' = (n_1', \cdots, n_m')$:

$$\mathbf{P}(A_q'(z') A_r''(z'') | \vec{n}') = \mathbf{P}(A_q'(z') | \vec{n}') \mathbf{P}(A_r''(z'') | \vec{n}'). \tag{5.9}$$

В силу (5.6) для почти любой последовательности $\vec{n}_1^{'}, \vec{n}_2^{'}, \cdots$ применима теорема 1, так что с учетом (5.9)

$$\mathbf{P}\big(A_q^{'}(z')|\vec{n}_n^{'}\big) \overset{\text{п.н.}}{\to} g_q(z'), \quad \mathbf{P}\big(A_r^{''}(z'')|\vec{n}_n^{'}\big) \overset{\text{п.н.}}{\to} g_r(z'') \quad (n \to \infty),$$
$$\mathbf{P}\big(A_q^{'}(z')A_r^{''}(z'')|\vec{n}_n^{'}\big) \overset{\text{п.н.}}{\to} g_q(z')g_r(z'') \quad (n \to \infty).$$

Из п.-н.-сходимости к константе равномерно ограниченных случайных величин следует сходимость к той же константе их средних:

$$\mathbf{P}\big(A_q^{'}(z')A_r^{''}(z'')\big) = \mathbf{EP}\big(A_q^{'}(z')A_r^{''}(z'')|\vec{n}_n^{'}\big) \to g_q(z')g_r(z'') \tag{5.10}$$

при $n \to \infty$. *Далее, пусть* $A_k(z)$ *- событие* $f(z) \in S_k \quad (k = 1, \cdots, m)$, где $f(z)$ есть ближайший к z элемент выборки S. В случае $z \in \mathbf{R}_+^N$

$$f(z) = f_n(z) \overset{\text{п.н.}}{\to} z \quad (n \to \infty). \tag{5.11}$$

Если, скажем, $p_k(z) > 0$, *то в любой окрестности точки* z *почти наверное найдутся элементы из* S_k, так как они независимы и имеют одну и ту же ненулевую вероятность попадания в окрестность. Отсюда следует (5.11). Так как $z'(z'')$ - внутренняя точка $H'(H'')$, то почти наверное $A_q^{'}(z') = A_q(z'), A_r^{''}(z'') = A_r(z'')$ для всех достаточно больших n.

Итак, предельные вероятности при $n \to \infty$ для событий $A_q(z')$, $A_r(z'')$, $A_q(z')A_r(z'')$ совпадают с найденными предельными вероятностями для $A_q^{'}(z')$, $A_r^{''}(z')$, $A_q^{'}(z')A_r^{''}(z')$ соответственно (п.-н.-сходимость индикаторов влечет сходимость их математических ожиданий). С учетом (5.7), (5.8), (5.10) имеем поэтому при $n \to \infty$

$$\mathbf{P}\big(A_q(z')A_r(z'')\big) - \mathbf{P}\big(A_q(z')\big)\mathbf{P}\big(A_r(z'')\big) \to 0. \tag{5.12}$$

Пусть $x_i \in S_q, x_j \in S_r$. Величина $c_{qr} = cov(I_i, I_j)$, равная

$$\mathbf{P}\big(f(x_i) \in S_q, f(x_j) \in S_r\big) - \mathbf{P}\big(f(x_i) \in S_q\big)\mathbf{P}\big(f(x_j) \in S_r\big), \tag{5.13}$$

представляется интегралом I_n по $\mathbf{R}^N \times \mathbf{R}^N$ от функции

$$\Big(\mathbf{P}\big(A_q(z')A_r(z'')\big) - \mathbf{P}\big(A_q(z')\big)\mathbf{P}\big(A_r(z'')\big)\Big)p_q(z')p_r(z''). \tag{5.14}$$

В (5.14) по сравнению с (5.13) n_q, n_r, если $q \neq r$, уменьшаются на 1 в связи с исключением "текущих" значений $x_i = z', x_j = z''$ (при $q = r \quad n_q$ уменьшается на 2). Это не влияет на предел (5.13) при $n \to \infty$. Итак, достаточно убедиться, что $I_n \to 0$ при $n \to \infty$.

Подинтегральная функция $\varphi(z', z'') = \varphi_n(z', z'')$ имеет суммируемую в $\mathbf{R}^N \times \mathbf{R}^N$ мажоранту $p_q(z')p_r(z'')$. Остается показать, что $\varphi_n(z', z'') \to 0$ почти всюду в $\mathbf{R}^N \times \mathbf{R}^N$ при $n \to \infty$. Для пар (z', z'') таких, что $p_q(z')p_r(z'') = 0$, это тривиально. "Диагональ" $z' = z''$ имеет нулевую меру; то же относится к множеству пар (z', z'') таких, что z' или z'' не является лебеговой точкой для одной из плотностей $p_1, \cdots, p_m$. Остальные пары $(z', z'') \in \mathbf{R}^N \times \mathbf{R}^N$ удовлетворяют условиям (5.1) и для них нужное соотношение получено в (5.12).

Лемма 3 доказана. Это завершает и обоснование теоремы 3.

6. Доказательство основного результата

6.1. Начнем с утверждения а), относящегося к случаю однородности. Согласно результатам §2 и §4 (теорема 2) в этом случае

$$\mathbf{E}\gamma_n = -\sqrt{n}\frac{m-1}{n-1}(< 0), \quad \mathbf{D}\gamma_n \leq \frac{2(n-m)(m-1)}{n-3} \leq 4(m-1).$$

Утверждение а) следует из неравенства Чебышева; в качестве $C(\epsilon, m)$ можно выбрать величины $2\sqrt{(m-1)\epsilon^{-1}}$ (они могут быть уменьшены).

В случае неоднородности, согласно (3.21) и (5.1), имеем

$$\mathbf{E}\Big(\frac{\gamma_n}{\sqrt{n}}\Big) \to \alpha, \quad \mathbf{D}\Big(\frac{\gamma_n}{\sqrt{n}}\Big) \to 0 \quad (n \to \infty),$$

где α - некоторая положительная постоянная.Итак, $\gamma_n/\sqrt{n}$ п.н. сходится к $\alpha(> 0)$. Отсюда сразу следует утверждение б).

Результат, названный выше основным, полностью доказан.

7. О количественных уточнениях

7.1. С качественной точки зрения основной результат устанавливает некоторое - теоретически достаточно важное - свойство статистики γ; доказательство носит конструктивный характер, т.к. подходящие значения $C(\epsilon, m)$ явно указываются. Однако существует и количественная сторона дела, особенно важная в приложениях. С этой точки зрения полученная простая формула для возможных пороговых значений (а именно, $2\sqrt{(m-1)\epsilon^{-1}}$) является не лучшим вариантом. Она, как отмечалось, дает завышенные пороговые значения, т.е. вероятности ошибок первого рода фактически оказываются меньше - и обычно существенно меньше - чем ϵ. Это, естественно, ведет к увеличению ошибок второго рода (при тех или иных альтернативах), т.е. к понижению чувствительности критерия при отклонениях от однородности.

Некоторые напрашивающиеся способы уточнения, будучи небесполезными в отдельных ситуациях, не позволяют, однако, снять данную трудность. Так, возвращение к более точным (и более громоздким) оценкам для $D\gamma$ не дает значительного эффекта. Попытка использовать вместо неравенства Чебышева асимптотическую нормальность γ_n (еще нуждающуюся в строгом обосновании) бесполезна для небольших выборок, да и для больших менее эффективна, чем обычно - ведь для $D\gamma_n$ известна лишь оценка сверху. Что касается употребительных на практике правил типа "3σ", то их дефектом является, как известно, отсутствие четкого представления о размере критерия.

7.2. Радикального решения вопроса можно добиться на другом пути. Он не приводит к какой-либо простой и "оптимальной" во всех случаях формуле для пороговых значений, поскольку носит алгоритмический характер. Речь идет о перестановочном методе, восходящем еще к Фишеру (подробнее, см., напр., [3,5]). Перестановочный подход применялся, главным образом, в скалярном случае, но здесь многомерность не создает принципиальных трудностей. Для удобства читателей напомним этот подход в общих чертах.

Обозначим через $\{S\}, \{S_i\}$ множества элементов выборок $S, S_i (i = 1, \ldots, m)$. Пусть, далее, S^0 - наблюдаемая реализация выборки S; соответствующее S^0 значение критической статистики обозначим через γ^0. Для нахождения в данной ситуации вероятности ошибки первого рода (при отклонении H_0) будем рассуждать следующим образом. Пусть гипотеза H_0 выполняется. Генерирование выборки S можно представить как два последовательных этапа: на первом этапе генерируется множество $\{S\}$, на втором равновероятно выбирается одно из разбиений $\{S\}$ на m множеств

$$\{S\} = \{S_1\} \cup \cdots \cup \{S_m\} \quad (|S_i| = n_i, \quad i = 1, \ldots, m). \tag{7.1}$$

Равновероятность является следствием H_0: все выборочные элементы равноправны, поэтому каждое из разбиений (7.1) имеет одну и ту же вероятность $n_1! \cdots n_m! (n!)^{-1}$. Каждому разбиению и соответствующим "виртуальным" подвыборкам $S_1, \cdots, S_m$ отвечает свое значение γ.

Представим себе, что в нашем случае первый этап уже выполнен, а второй еще нет, т.е. $\{S^0\}$ фиксировано, а γ пока является случайной величиной, распределение которой (фактически условное) однозначно определено и не зависит от p_0. Основную роль далее играет условная вероятность

$$\epsilon = P(\gamma \geq \gamma^0 | \{S^0\}).$$

Если отклонять гипотезу H_0 в случае $\gamma \geq \gamma^0$, то ϵ становится, таким образом, условным размером критерия (условие $\{S\} = \{S^0\}$ имеет нулевую вероятность, что здесь несущественно).

При реализациях малых выборок значение ϵ зачастую удается вычислить точно, используя комбинаторные соображения или полный перебор. Однако высокая точность здесь обычно не требуется, так что естественным общим способом приближенного вычисления ϵ является метод Монте-Карло.

Величина ϵ является адекватной мерой степени достоверности H_0; если H_0 отклоняется при $\epsilon \leq \epsilon_1$ (где ϵ_1 - заданный пороговый размер), то и безусловный размер критерия при любой p_0 также не превосходит ϵ_1. (Ввиду произвольности ϵ_1 значение ϵ, очевидно, более информативно, чем сам факт отклонения или неотклонения H_0).

Если требуется получить критерий точно (или почти точно) заданного размера, то следует преодолеть еще некоторые технические трудности. Дело в том, что перестановочное распределение дискретно (так же как и выборочное распределение, возникающее при применении метода Монте-Карло). Поэтому, если критическая область имеет вид $\gamma \geq c$, то размер критерия может принимать лишь конечное число значений (расположенных обычно достаточно "густо"). Эта трудность, естественно, преодолевается с помощью рандомизации.

Преимущества перестановочного метода очевидны - это его точность и общность. В частности, не требуется никаких ограничений на объем выборок; отметим также, что данный подход не связан с конкретным видом критической статистики. Алгоритмический характер не может считаться принципиальным недостатком, тем более, что трудоемкость перестановочного метода вполне приемлема.

8. О возможных обобщениях

8.1. В заключение остановимся на ряде обобщений, представляющих определенный интерес.

Естественно считать, что близость элементов определяется стандартной евклидовой нормой в $\mathbf{R}^N$. Однако фактически все остается справедливым и для других норм - скажем, для максимума модулей компонент и т.п.

Линейность пространства также несущественна; векторные операции встречались лишь в выражении $\| x-y \|$, т.е. для обозначения расстояния. Правда, на произвольные метрические пространства основной результат не распространяется (легко указать контрпримеры для несепарабельного случая).

Основной результат сохраняет силу для вырожденных распределений, сосредоточенных на некоторых N_1-мерных многообразиях в $\mathbf{R}^N$ с N_1-мерными плотностями ($1 \le N_1 < N$). (Интересны, в частности, распределения на сфере). Априорной информации об этих многообразиях, как и о самом факте вырожденности, не требуется.

8.2. Что касается распределений общего типа, то уже для формальной применимости γ-критерия нужна единственность (почти наверное) ближайшего элемента. Если таковая отсутствует (скажем, в дискретном случае) то возможна рандомизация, т.е. равновероятный выбор одного из претендентов. Поскольку для этой цели применимы малые непрерывные, независимые и одинаково распределенные возмущения - что возвращает нас к непрерывному случаю - значение $\mathbf{E}\gamma_n$ и оценка (4.1) для $\mathbf{D}\gamma_n$ (при H_0) сохраняются.

Отсюда нетрудно перейти к более удобной - и уже не рандомизированной - модификации γ-критерия. Именно, пусть для любых i, j элемент x_{ij} имеет r_{ij} ближайших среди $n-1$ остальных элементов S (при совпадениях подсчет ведется с учетом "кратности"), из которых r'_{ij} принадлежат выборке S_i. Определим статистику γ'_n той же формулой (1.3), где теперь

$$\theta_i = \frac{r'_{i1}}{r_{i1}} + \ldots + \frac{r'_{in_i}}{r_{in_i}} \quad (i = 1, \ldots, m).$$

Это - обобщение γ_n на случай неединственности ближайших (для непрерывных распределений γ_n и γ'_n совпадают почти наверное). Рассмотрим события, определяющие для всех элементов выборки S совокупности ближайших к ним элементов S. Данные $(2^{n-1}-1)^n$ событий индуцируют разбиение пространства выборок, причем γ'_n есть условное математическое ожидание γ_n относительно этого разбиения. Следовательно (см., напр., [10]), $\mathbf{E}\gamma'_n = \mathbf{E}\gamma_n, \mathbf{D}\gamma'_n \le \mathbf{D}\gamma_n$ (так что теорема 2 остается в силе).

8.3. Сфера применимости γ-критерия (или γ'-критерия) радикально расширяется, если отказаться от жесткого требования полной состоятельности Из утверждений а), б) (см.§1) важно теперь лишь первое, основанное на теореме 2, доказанной без существенных ограничений. Теперь критерий может применяться в произвольном метрическом пространстве (с оговорками по поводу измеримости). Элементами выборок могут быть, например, фрагменты выборочных случайных полей; близость определяется по метрике в адекватном функциональном пространстве (для выборочных элементов или функций от них). Пусть, например, сравниваются выборочные траектории стационарных эргодических гауссовских процессов на некотором интервале времени; гипотеза H_0 состоит в том, что эти процессы идентичны. Здесь естественно применять метрику, учитывающую разность выборочных средних и расстояние (в какой-либо из функциональных норм) между выборочными автоковариациями.

8.4. Отпадает необходимость и в аксиомах метрики. От интерпретации $f(x)$ как ближайшего к x элемента можно перейти к произвольному отображению $f: S \to S$ без неподвижных точек (ради краткости считаем, что все элементы S различны). Формально требуется лишь, чтобы f определялось значениями элементов S и не зависело от их индексации. В то же время с точки зрения чувствительности к альтернативам целесообразно определять $f(x)$ как "наиболее сходный" с x среди остальных элементов выборки S в некотором естественном смысле (связанном с конкретикой задачи).

8.5. Для приложений существенно, что формализация f не обязательна и f может определяться экспертной оценкой. Сравним два варианта экспертизы. При первом эксперту предъявляются выборки $S_1, \ldots, S_m$ и задается вопрос о наличии или отсутствии однородности. Каков бы ни был ответ, надежность его основывается лишь на вере в качественность экспертизы. Второй вариант таков. Эксперту предъявляется выборка S в целом (без указания $S_1, \ldots, S_m$) и предлагается для каждого элемента выборки S указать "наиболее сходный" с ним среди $n-1$ остальных; затем применяется γ-критерий. Здесь при отклонении H_0, т.е. при $\gamma > C(m,\epsilon)$, вывод делается с уровнем риска, не превосходящим ϵ - притом без априорных сведений о качестве экспертизы. (Сам факт больших значений γ свидетельствует - в вероятностном смысле - как о нарушении H_0, так и о качественности экспертной оценки).

8.6. Независимость элементов выборки S также не является необходимым требованием. Именно, будем трактовать теперь гипотезу однородности H_0 как перестановочность совместного распределения всех элементов S. Как известно, это менее ограничительное требование содержательно отвечает понятию однородности; в то же время оно достаточно для справедливости утверждения а) §1. Сама форма γ-критерия показывает, что он ориентирован на альтернативы, допускающие перестановочность не в целом, а лишь внутри выборок $S_1, \ldots, S_m$.

Сказанное выше показывает, что формально γ-критерий применим почти без ограничений - не требуется конечномерности и линейности пространства, независимости выборочных элементов, предположений, связанных с метрикой (да, собственно, и числового характера данных). В части, касающейся ошибок первого рода, при этом ничего не меняется. Правда, при столь "неограниченной" общности уже не гарантируется состоятельность против всех альтернатив.

8.7. Рассмотренный γ-критерий связан с тем соображением, что при неоднородности близкие элементы каждой выборки S_i как бы "тяготеют" друг к другу. Эта идея, конечно, не нова; упомянем, например, классический критерий серий [11] для скалярных выборок, где применяется сходное соображение. Можно предложить другие многомерные статистики родственного характера. Так, естественным выглядит "расширение кругозора", когда учитывается не только ближайший к $x \in S$ элемент $x^1 = f(x)$ среди остальных элементов S, но и последующие - в порядке возрастания расстояний до x - элементы $x^2, x^3, \ldots$. При этом вклад в статистику элемента $x \in S_i$ может определяться числом r таким, что $x^1, \ldots, x^r \in S_i$, $x^{r+1} \notin S_i$. В другом варианте r определяется как минимальное k, при котором $x^k \in S_i$. Пока неизвестно, распространяется ли на соответствующие статистики основной результат; с этой точки зрения применение статистики γ представляется в настоящее время наиболее оправданным.

8.8. В заключение отметим, что проводились достаточно обширные численные эксперименты, которые свидетельствуют о практической эффективности γ-критерия. Так, приемлемый уровень достоверности при выявлении сильно выраженной неоднородности весьма часто обеспечивался уже при малых выборках $S_1, \ldots, S_m$, содержащих не более 5 - 10 элементов (что вполне естественно). Более подробные данные здесь не приводятся, поскольку полный произвол в выборе плотностей неизбежно придает любым экспериментам весьма частный характер.

Литература

[1] Кендалл М.Дж., Стьюарт А. Многомерный статистический анализ и временные ряды. М.:Наука, 1976. 736 с.

[2] Большев Л.Н., Смирнов Н.В. Таблицы математической статистики. М.: Наука, 1965. 464 с.

[3] Леман Э. Проверка статистических гипотез. М.:Наука, 1979. 408 с.

[4] Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Основы моделирования и первичной обработки данных. М.:Финансы и статистика, 1983. 472 с.

[5] Кокс Д.,Хинкли Д. Теоретическая статистика. М.:Мир, 1978. 560 с.

[6] Гаек Я.,Шидак З. Теория ранговых критериев. М.:Наука,1971. 376 с.

[7] Боровков А.А. Математическая статистика. Дополнительные главы. М.:Наука, 1984. 144 с.

[8] Левин А.Ю. О состоятельном многомерном непараметрическом критерии однородности //Усп. матем. наук. 1993. Т.48, N 6. С.155-156.

[9] Шилов Г.Е., Гуревич Б.Л. Интеграл, мера и производная. М.:Наука, 1964. 212 с.

[10] Ширяев А.Н. Вероятность. М.: Наука, 1980. 576 с.

[11] Wald A., Wolfowitz J. On a test whether two samples are from the same population //Ann. Math. Statist. 1940. V. 11. P. 147-162.

Модел. и анализ информ. систем. Т.9, №2 (2002) 45–49

УДК 519.673

Сети W-нейронов в задаче планирования оптимальных путей для точечных роботов

Майоров В.В., Шабаршина Г.В., Анисимова И.М.
Ярославский государственный университет
150 000, Ярославль, Советская, 14

В статье рассматривается задача планирования оптимальных путей для двух точечных роботов на правильной треугольной решетке. Часть вершин решетки выполняет роль препятствий на плоскости. При движении роботы должны огибать препятствия и не иметь позиций для столкновения. Основная идея нахождения безопасных кратчайших путей состоит в замене движения двух роботов на перемещение одного обобщенного. Для решения задачи предлагается использовать нейронную сеть, элементами которой являются W - нейроны.

1. Введение

Развитие теории нейронных сетей идет как в области создания новых нелинейных элементов, которые, взаимодействуя между собой, порождают системы со сложным поведением, так и в разработке новых архитектур нейронных сетей [1]–[3]. Наряду с этим идет активный поиск областей приложения нейронных сетей в системах управления, робототехнике, системах обработки изображений.

Рассматриваемая здесь задача планирования пути обслуживает широкий круг прикладных проблем. Например, задача генерации пути автономных передвижных средств, задачи навигации и т.д.

В данной работе решение задачи о планировании оптимальных путей для двух роботов основано на применении эффективной идеи замены их движения на перемещение одного обобщенного робота. Идея предложена одним из авторов статьи (В. В. М.), реализация ее, отличная от данной, представлена в [4] на основе нейронной сети волнового распространения.

2. Постановка задачи

Рассмотрим на плоскости правильную треугольную решетку, длину ребра которой можно считать равной единице. Обозначим вершины через $V_{i,j}$ $(1 \le i \le N,\ 1 \le j \le M)$. Назовем окрестностью вершины $V_{i,j}$ множество $U_{i,j}$, состоящее из шести смежных с ней вершин. Обозначим через $e_{i,j}^{k,l}$ ребро, соединяющее точку $V_{i,j}$ с точкой $V_{k,l}$, где $V_{k,l} \in U_{i,j}$.

Чередующуюся последовательность вершин и ребер

$$V_{i,j}, e_{i,j}^{k,l}, V_{k,l}, \ldots, e_{p,q}^{r,s}, V_{r,s}$$

будем называть путем (траекторией), соединяющим две фиксированные вершины $V_{i,j}$ и $V_{r,s}$. Длина пути — сумма длин входящих в него ребер.

Отметим на решетке две вершины, в которые поместим точечных роботов. Роботы функционируют в дискретном времени. За один такт каждый из них должен по ребру переместиться в одну из смежных вершин. Будем считать, что часть вершин решетки выполняет роль препятствий на плоскости, т.е. траектория движения робота не может содержать эти точки. Кроме того, в любой момент времени роботы не могут одновременно находиться в одной и той же вершине.

Задача состоит в отыскании для каждого робота кратчайшего пути, соединяющего заданные для него начальную и конечную точки. При этом пути должны огибать препятствия и быть безопасными, то есть не иметь возможных позиций столкновения.

Ниже в качестве вспомогательной рассматривается задача нахождения кратчайшего пути для одного робота. Для ее решения используется нейронная сеть, элементами которой являются W-нейроны.

Для решения основной задачи введем на плоскости систему координат, выбрав в качестве начала одну из вершин решетки, направления осей по образующим решетки. Тогда положение роботов на плоскости будем характеризовать целочисленными координатами $(a_1(t), b_1(t))$, $(a_2(t), b_2(t))$ соответственно. Здесь t — текущий момент времени.

В четырехмерном пространстве рассмотрим множество Ω, точек вида (a_1, b_1, a_2, b_2), где (a_1, b_1) и (a_2, b_2) — координаты вершин решетки. Тогда движению роботов на плоскости соответствует перемещение точки $(a_1(t), b_1(t), a_2(t), b_2(t))$ в пространстве Ω. Здесь первые две координаты описывают передвижение первого робота, а третья и четвертая - второго робота. Далее к множеству Ω применим алгоритм вспомогательной задачи.

3. Описание W - нейрона

W - нейроны были введены в работах [5], [6]. для решения задачи хранения и воспроизведения последовательности бинарных векторов.

В любой дискретный момент времени t W - нейрон находится в одном из трех состояний. Для описания состояния W - нейрона введем величину $s(t)$. Значение $s(t) = 0$ соответствует состоянию ожидания (покоя), $s(t) = 1$ - состоянию возбуждения, при $s(t) = -1$ нейрон находится в состоянии рефрактерности - невосприимчивости к внешнему воздействию. Состояние возбуждения длится один такт. Пусть в момент времени t W-нейрон возбужден, т.е. $s(t) = 1$. Тогда в следующие $r_0 \geq 1$ тактов W -нейрон находится в рефрактерном состоянии: $s(t+i) = -1$ $(i = 1, \ldots, r_0)$. В момент времени $t + r_0 + 1$ он обязательно переходит в состояние ожидания, т.е. $s(t + r_0 + 1) = 0$.

В каждый момент времени t W - нейрон формирует выходной сигнал $x(t) = \delta_0(s(t) - 1)$, где $\delta_0(v)$ - импульсная функция: $\delta_0(0) = 1$ и $\delta_0(v) = 0$ при $v \neq 0$. Тем самым, выходной сигнал отличен от нуля и равен единице только в состоянии возбуждения.

W - нейрон имеет синаптические входы двух типов: суммирующие и безусловно тормозящие. Если в момент времени t W - нейрон находится в состоянии ожидания и на один из безусловно тормозящих синапсов поступает сигнал, то в следующий момент времени элемент оказывается в состоянии рефрактерности: $s(t+1) = -1$ и остается в нем r_0 тактов: $s(t+i) = -1$ для $i = 1, \ldots, r_0$. Для описания влияния суммирующих синаптических входов определим для W - нейрона величину $u(t)$, которую назовем мембранным потенциалом. Пусть $x_i(t)$ - сигнал (равен либо нулю, либо единице), поступающий в момент времени t на i - ый синапс. Тогда положим

$$u(t) = q\delta_0(s(t-1))u(t-1) + \left[\sum_{i=1}^{N} w_i x_i(t)\right] \delta_0(s(t)),$$

где параметр $0 \leq q < 1$, N-число синапсов. Числа w_i в формуле мембранного потенциала назовем синаптическими весами. Введем число u_0, которое назовем пороговым значением мембранного потенциала. Если в момент времени t значение $u(t) \geq u_0$, то в следующий момент времени $s(t+1) = 1$, т.е. W - нейрон переходит в возбужденное состояние и формирует выходной сигнал $x(t+1) = 1$. Удобно записать: $s(t+1) = \theta(u(t) - u_0)$, где $\theta(v)$ - функция Хевисайта, т.е. $\theta(v) = 0$ при $v < 0$ и $\theta(v) = 1$ при $v \geq 0$.

Отметим, что интересные приложения сетей из W - нейронов связаны с особенностями суммирования нейроном входных сигналов не только по пространству, но и по времени. Однако в данной работе вполне достаточно использовать упрощенную модель W - нейрона, характеризуя его как элемент, способный находиться в одном из трех состояний и переходящий в возбужденное состояние, если по соседству оказалось достаточное количество возбужденных нейронов.

4. Нахождение кратчайшего пути точечного робота на правильной решетке

В узлах правильной решетки размещены W -нейроны, каждый из которых может оказывать действие на шесть смежных с ним элементов. Будем считать, что нейроны, соответствующие препятствиям, имеют безусловно тормозящие синаптические входы, остальные — суммирующие. Обозначим через A и B W - нейроны сети, соответствующие начальному и конечному положению робота.

Обозначим через $w_{i,j}^{k,l}$ - синаптический коэффициент воздействия (k,l) -го нейрона, расположенного в вершине $V_{k,l}$ на (i,j)-ый нейрон, расположенный в вершине $V_{i,j}$, причем $V_{i,j} \in U_{k,l}$. Будем считать связи между нейронами сети симметричными, т.е. $w_{i,j}^{k,l} = w_{k,l}^{i,j}$. Первоначально положим все синаптические веса равными единице.

Снабдим каждый W - нейрон внешним входом. Если в момент времени t нейрон находится в состоянии ожидания $(s(t) = 0)$, то внешний сигнал $x_{вн}(t) = -1$ (абсолютно тормозящий) переводит его в следующий момент времени в состояние рефрактерности, которое продлится r_0 тактов. В свою очередь, сигнал

$x_{вн}(t) = 1$ (возбуждающий) переводит W - нейрон в следующий момент времени на один такт в возбужденное состояние (генерируется единичный выходной сигнал). Внешний нулевой сигнал не оказывает действия на нейрон.

Пусть в нулевой момент времени все W - нейроны находятся в состоянии покоя. Подадим внешний сигнал $x_{вн} = 1$ на нейрон A, соответствующий начальному положению робота. В момент времени $t = 1$ W - нейрон сгенерирует импульс. Будем считать, что пороговые значения мембранных потенциалов у каждого элемента сети одинаковы и равны $u_0 = 1$. Это означает, что в момент времени $t = 2$ W - нейроны окрестности нейрона A перейдут на один такт в возбужденное состояние или на r_0 тактов в рефрактерное, если они соответствовали препятствиям. Сам нейрон A переходит в рефрактерное состояние.

Процесс генерации импульсов нейронами сети сопровождается модификацией имеющихся синаптических связей. Если нейрон получает сигнал от другого нейрона и, вследствие этого, становится активным, то вес связи между этими нейронами согласно правилу Хебба [7], следует увеличить. С учетом симметрии положим такие коэффициенты равными двум: $w_{i,j}^{k,l} = w_{k,l}^{i,j} = 2$. Если нейроны генерируют импульсы одновременно, то связь между ними, по крайней мере, не усиливается. Такие коэффициенты оставляем без изменений. Не меняются также веса тех связей, активизация которых не привела к генерации импульса.

Процесс последовательной генерации импульсов нейронами сети заканчивается, как только волна возбуждения достигает точки B. Назовем его первым тактом прохождения волны возбуждения по сети. Заметим, что здесь определяется длина минимального пути AB. По окончании первого такта все нейроны приводятся в состояние покоя. На втором такте ненулевой сигнал подается на внешний вход нейрона B. Пороговые значения мембранных потенциалов полагаем равными $u_0 = 2$. Это означает, что на втором такте в возбужденное состояние будут переходить только те нейроны, синаптические входы которых имеют достаточный вес. Изменение синаптических коэффициентов происходит аналогично изменениям на первом такте. Второй такт заканчивается, как только волна возбуждения, порожденная нейроном, находящимся в точке B, достигает нейрона в точке A. За вторым тактом следует третий: ненулевой сигнал подается на внешний вход нейрона A, и т.д. Каждый такт прохождения волны возбуждения по сети будем сопровождать изменением синаптических весов:

1) Если для двух соседних нейронов $s_{k,l}(t)s_{i,j}(t+1) = 1$, где $s_{k,l}(t)$ и $s_{i,j}(t+1)$ состояния их в моменты времени t и $t+1$ соответственно, то синаптические веса $w_{i,j}^{k,l}$ и $w_{k,l}^{i,j}$ увеличиваем на единицу. Если $s_{k,l}(t)s_{i,j}(t+1) = 0$, то веса не изменяются.

2) Если для двух соседних нейронов (i, j) и (k, l) $s_{k,l}(t)s_{i,j}(t) = 1$, то синаптические веса связей $w_{i,j}^{k,l}$ и $w_{k,l}^{i,j}$ оставляем без изменений.

3) Если в окрестности $V_{i,j}$ (i, j)-го нейрона, генерирующего импульс в данный момент времени t, нет элемента с индексом (k, l), такого что $s_{k,l}(t+1)s_{i,j}(t) = 1$, то синаптические коэффициенты $w_{i,j}^{m,n}$ и $w_{m,n}^{i,j}$ для любой пары индексов (m, n) полагаем равными единице.

4) Процесс распространения импульсов по сети останавливается, если волна возбуждения достигла нейрона, соответствующего точке A или B.

Обозначим через A_k - множество W-нейронов, генерирующих импульсы на κ - ом шаге. Заметим, что, начиная с некоторого номера k, $A_{k+1} \subset A_k$. Действительно, если пороговое значение мембранного потенциала достаточно велико, то синаптические веса связей между нейронами должны принимать большие значения. В противном случае согласно третьему пункту алгоритма, если генерация импульса данным W-нейроном не влечет за собой генерацию импульсов элементами его окрестности, то коэффициенты связей полагаются равными единице. На следующем такте такой нейрон генерировать ненулевой выходной сигнал не может.

Если $A_k = A_{k+1}$, то множество вершин, соответствующих этим нейронам, входит в оптимальные маршруты. Ребра, их соединяющие, имеют веса $k + l$. Предположим, что здесь содержится путь, длина которого больше длины минимального пути AB. Вернемся к нейронной сети. Волна импульсов, генерируемых нейронами, находящимися в вершинах этого маршрута, достигает нейрона B. Но к этому моменту он уже перешел в рефрактерное состояние. Это означает, что на k-ом такте синаптические веса нейрона, находящегося в окрестности нейрона B, полагаются равными единице. Таким образом, входить в множество A_{k+1} он не может. Приведенный алгоритм сходится за конечное число шагов.

Как восстановить путь, зная синаптические веса связей? Ненулевой внешний сигнал посылаем на нейрон A. Если вес $w_{i,j}^{k,l} = k + 1$, то $V_{i,j}, e_{i,j}^{k,l}, V_{k,l}$ — элемент оптимального пути. В общем случае на правильной треугольной решетке путь не единственный. Как правило, мы имеем серию эквивалентных путей, то есть путей, одинаковых по длине.

5. Нахождение кратчайших путей для двух точечных роботов на правильной решетке

Напомним постановку задачи. На правильной треугольной решетке выбраны две вершины, в которые помещены точечные роботы. Каждый из них за один такт по ребру перемещается в одну из смежных вершин. Часть вершин решетки выполняет роль препятствий на плоскости, т.е. траектория движения робота не может содержать эти точки. Кроме того, в любой момент времени роботы не могут одновременно находиться в одной и той же вершине.

Задача состоит в отыскании для каждого робота кратчайшего пути, соединяющего заданные для него начальную и конечную точки. При этом пути должны огибать препятствия и быть безопасными, то есть не иметь возможных позиций столкновения.

Положение роботов на плоскости можно описать целочисленными координатами $(a_1(t), b_1(t))$, $(a_2(t), b_2(t))$ соответственно. Здесь t — текущий момент времени.

Рассмотрим множество Ω, принадлежащее четырехмерному пространству, точек вида (a_1, b_1, a_2, b_2), где (a_1, b_1) и (a_2, b_2) — координаты вершин решетки. Тогда движению роботов на плоскости соответствует перемещение точки $(a_1(t), b_1(t), a_2(t), b_2(t))$ в пространстве Ω. Здесь первые две координаты описывают передвижение первого робота, а третья и четвертая - второго робота.

Обозначим через $P \subset \Omega$ - область препятствий. Она представляет собой ту часть множества Ω, в которой невозможно передвижение соответствующей точки $(a_1(t), b_1(t), a_2(t), b_2(t))$.

Приведем формальное описание множества P. Пусть (c, d) координаты произвольной вершины, принадлежащей множеству вершин — препятствий на плоскости. Тогда точка (a_1, b_1, a_2, b_2) принадлежит множеству P, если $a_1 = c$, $b_1 = d$ или $a_2 = c$, $b_2 = d$. По смыслу задачи это условие означает,что траектории движения роботов в R^2 не должны пересекать препятствия.

К множеству P мы должны отнести также те точки из множества Ω, координаты которых удовлетворяют условиям: $a_1 = b_1$, $a_2 = b_2$. Траектории движения роботов в R^2 не должны пересекаться друг с другом.

После того как описаны множество Ω и множество P, в точках Ω размещаем W-нейроны. При этом нейроны, помещенные в точки множества P, имеют абсолютно тормозящие синапсы. Далее к нейронной сети применяется алгоритм решения вспомогательной задачи о нахождении оптимального пути движения для одного робота. Из найденного пути - набора смежных вершин в $\Omega \backslash \Psi$, который начинается в начальной точке (a_1, b_1, a_2, b_2), выделяются оптимальные маршруты для каждого робота.

6. Заключение

Очевидно, что исследование сетей W - нейронов является перспективной тематикой. При конструировании формальных нейронов далеко не всегда привлекаются биологические данные. Однако они оказываются весьма полезными. Рассмотренные в настоящей работе W - нейроны, как и нервные клетки, могут находиться в состоянии возбуждения, рефрактерности и состоянии покоя. С рефрактерностью связана возможность применения сети W - нейронов для решения поставленной задачи.

Заметим, что основная идея решения задачи — замена движения двух роботов на перемещение одного обобщенного, т.е. переход в многомерное пространство, позволяет распространить планирование путей на случай k роботов.

Литература

[1] Лоскутов А.Ю., Михайлов А.С. Введение в синергетику. М.: Наука. 1990. 270 с.

[2] Ижикевич Е.М., Малинецкий Г.Г. О возможной роли хаоса в нейросистемах // Докл. АН России. 1992. Т. 326. №4. С. 626 -632.

[3] Chua L. O., Yang L. Cellular Neural Networks: Theory //IEEE Trans/ Circ/ And Syst. 1988. V. 35, №10.

[4] Лебедев Д.В. Планирование оптимальных путей плоских роботов на основе использования волновой нейросетевой архитектуры // Моделирование и анализ информационных систем. 2000. Т. 7. 1. С.30-38.

[5] Майоров В.В., Шабаршина Г.В. Сообщение о сетях W - нейронов // Моделирование и анализ информационных систем. 1997. В. 4. С. 37 - 50.

[6] Майоров В.В., Шабаршина Г.В. Сети W - нейронов в задаче ассоциативной памяти // ЖВМиМФ.2001. Т.41. №8. С. 1298-1296.

[7] Hebb D.O. The organization of Behavior. N.Y.: Wiley. 1949. 335 p.

Модел. и анализ информ. систем. Т.9, №1 (2002) 50–58

УДК 519.872

Приоритетное множество для системы массового обслуживания $M|G|1$

Черменский П.П.
Ярославский государственный университет
150 000, Ярославль, Советская, 14

1. Введение

В работе рассматривается система массового обслуживания, состоящая из одного прибора и бесконечной очереди, на вход которой поступают требования одного типа, разрешается прерывание обслуживания. Для этой системы найдено параметрическое описание приоритетного множества – множества всех значений средних длин очередей в системе. Задача решалась путём сведения исходной системы к системе с ветвящимися потоками вторичных требований. Решение основано на результатах работ [1]-[4].

2. Постановка задачи и результат

Опишем исходную систему. Система $M|G|1$ состоит из одного прибора и бесконечной очереди. На вход системы поступает пуассоновский поток требований. Интенсивность потока заявок равна λ. Длительность обслуживания требований с вероятностью α_1 равна a_1, с вероятностью α_2 равна $a_1 + a_2$, с вероятностью $\alpha_3 = 1 - (\alpha_1 + \alpha_2)$ равна $a_1 + a_2 + a_3$ ($\alpha_i \geqslant 0$, $\alpha_1 + \alpha_2 + \alpha_3 = 1$). Будем считать, что выполняется условие существования стационарного режима: $\lambda(a_1 + (1-\alpha_1)a_2 + (1-\alpha_1-\alpha_2)a_3) < 1$. Допускается прерывание обслуживания требований с последующим дообслуживанием. Правило, выбирающее порядок обслуживания требований, назовем дисциплиной обслуживания.

Через $\widetilde{L}$ обозначим среднюю длину очереди требований при некоторой дисциплине обслуживания. Определим приоритетное множество:

$$M = \{\widetilde{L}, \text{ по всем дисциплинам обслуживания.}\} \tag{1}$$

В работе показано, что приоритетное множество M *является выпуклой оболочкой точек* $\{\widetilde{L}\}$ в пространстве, которая принадлежит объединению следующих множеств:

Множество $M(\sigma_i)$ определяется как множество точек $\{\widetilde{L}\}$, принадлежащих множествам, заданным уравнениями:

$$\begin{aligned}\widetilde{L} = {} & \frac{\rho_1^2}{2(1-\rho_1)}\Big\{1 + \frac{p_1p_2(\rho_2+\rho_3)}{(1-\rho)(1-(\rho_1+\rho_2))} + \frac{p_1(\rho_1+\rho_2)}{1-(\rho_1+\rho_2)}\Big\} + \frac{p_1(1-p_1)\lambda^2}{2(1-\rho_1)}(a_2-y_1)^2 + \\ & + \frac{p_1^2p_2(1-p_2)\lambda^2}{2(1-\rho_1)(1-(\rho_1+\rho_2))}(a_3-z_1)^2 + \frac{p_1(1-p_1)p_2\lambda^2}{2(1-\rho_1)}(a_3-z_2)^2 + \frac{p_1(1-p_1)\lambda\rho_1}{1-\rho_1}y_1 + \\ & + \frac{p_1p_2\lambda\rho_1}{1-\rho_1}(z_2-z_1) + \frac{p_1p_2\lambda(\rho_1+\rho_2)}{1-(\rho_1+\rho_2)}z_1 + \Big\{p_1\rho_1(\rho_1+\rho_2) + \frac{\rho_2^2}{2(1-\rho_1)}\Big\}\frac{1-\rho+p_2(\rho_2+\rho_3)}{(1-\rho)(1-(\rho_2+\rho_3))} \\ & + \Big\{p_2\rho_1(\rho_2+\rho_3) + p_2\rho_2\rho + \frac{\rho_3^2}{2(1-(\rho_1+\rho_2))}\Big\}\frac{1}{1-\rho}\end{aligned} \tag{2}$$

$$\begin{aligned}\widetilde{L} = {} & \frac{p_1(1-p_1)\lambda^2}{2(1-\rho_1)}(a_2-y_2)^2 + \frac{p_1(1-p_1)p_2\lambda^2}{2(1-\rho_1)}(a_3-z_1)^2 + \frac{p_1\lambda\rho_1}{1-\rho_1}(a_2-y_2) + \frac{p_1p_2\lambda\rho_1}{1-\rho_1}z_1 + \\ & + \Big\{p_1\rho_1\rho + \frac{\rho_2^2}{2(1-\rho_1)} + \frac{p_1^2p_2\lambda^2a_3^2}{2(1-\rho_1)} + \frac{\rho_2\rho_3}{1-\rho_1}\Big\}\frac{1}{1-\rho} + \frac{\rho_1^2}{2(1-\rho_1)}\Big(1 + \frac{p_1\rho}{1-\rho}\Big)\end{aligned} \tag{3}$$

$$
\begin{aligned}
\widetilde{L} = {} & \frac{p_1(1-p_1)\lambda^2\rho}{2(1-\rho_1)(1-\rho)}(a_2-y_1)^2 + \frac{\lambda\rho_3(2+\rho_1(p_1-1))}{2(1-\rho_1)(1-\rho)}(a_2-y_1) + \frac{p_1\lambda\rho_1}{1-\rho_1}y_1 + \\
& + \frac{\rho_1^2+\rho_3^2}{2(1-\rho_1)}\Big(1+\frac{p_1\rho}{1-\rho}\Big) + \Big(p_1\rho_1\rho + \frac{p_1\lambda^2 a_2^2}{2(1-\rho_1)}\Big)\frac{1}{1-\rho}
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
\widetilde{L} = {} & \frac{p_1p_2\lambda(\rho_1+\rho_2)}{1-(\rho_1+\rho_2)}z_1 + \Big\{\frac{\rho_3^2}{2(1-(\rho_1+\rho_2))} + p_2p_2\lambda\rho(a_1+a_2)\Big\}\frac{1}{1-\rho} + \\
& + \frac{\rho_1^2+2\rho_1\rho_2+p_1\lambda^2a_2^2}{2(1-(\rho_1+\rho_2))}\Big(\frac{p_1p_2\rho}{1-\rho}+1\Big)
\end{aligned} \tag{5}
$$

$$
\widetilde{L} = \Big\{\rho_1^2/2 + p_1\lambda^2a_2^2/2 + p_1p_2\lambda^2a_3^2 + \rho_1\rho_2 + \lambda a_2\rho_3 + \rho_1\rho_3\Big\}\frac{1}{1-\rho}, \tag{6}
$$

где константы $\rho_1, \rho_2, \rho_3, \rho$ равны: $\rho_1 = \lambda a_1$, $\rho_2 = \lambda p_1 a_2$, $\rho_3 = \lambda p_1 p_2 a_3$, $\rho = \rho_1 + \rho_2 + \rho_3$. Параметры p_1, p_2: $\alpha_1 = (1-p_1)$, $\alpha_2 = p_1(1-p_2)$, $\alpha_3 = p_1p_2$. А произвольные параметры y_1, z_1, z_2 удовлетворяют ограничениям: $0 \leqslant y_1 \leqslant a_2$, $0 \leqslant z_1 < z_2 \leqslant a_3$.

3. Схема нахождения приоритетного множества

Опишем основные этапы нахождения приоритетного множества M.

1) Выберем достаточно малый параметр θ и рассмотрим те дисциплины обслуживания, которые допускают прерывание обслуживания только в моменты времени, кратные θ. Для таких дисциплин обслуживания рассматриваемая система сводится к системе с ветвящимися потоками вторичных требований (СВПВТ), изученной в работах [2] и [1]. Обозначим через L_i $(1 \leqslant i \leqslant k+m+s=n)$ средние длины очередей в этой системе.

2) В работе [4] показано, что приоритетное множество для СВПВТ является многогранником в R^n, обозначим его $P(\theta)$.

3) Пусть $M_P(\theta)$ – проекция $P(\theta)$ в пространстве R. При этой проекции точка с координатами $(L_1, \ldots, L_n)$ переходит в точку $L = \sum\limits_{i=1}^{n} L_i$. Проекции вершин многогранника $P(\theta)$, в которых достигается минимум линейной функции

$$
F = \sum_{i=1}^{n} L_i \tag{7}
$$

и будут вершинами многогранника $M_P(\theta)$. Таким образом, для нахождения многогранника $M_P(\theta)$ нужно найти все те вершины многогранника $P(\theta)$, в которых достигается минимум линейной функции (7) при всевозможных значениях C.

4) Приоритетное множество M находится как $\lim\limits_{\theta\to 0} M_P(\theta)$.

4. Сведение к СВПВТ

Опишем подробнее первый этап решения. СВПВТ строится следующим образом. Выберем $0 < \theta \ll 1$, так чтобы $a_1 = k\theta$, $a_2 = m\theta$, $a_3 = s\theta$, для простоты будем считать k, m, s натуральными. Будем рассматривать те дисциплины обслуживания, которые допускают прерывания обслуживания только в моменты времени, кратные θ. В результате получаем систему, имеющую $k+m+s=n$ новых типов требований, которая

1) система состоит из одного прибора и $k+m+s$ типов требований;

2) на вход системы поступает пуассоновский поток заявок типа с интенсивностью λ;

3) каждое из $(m+k+s)$ типов требований имеет постоянное время обслуживания θ;

4) в каждый момент времени обслуживается только одно требование, прерывание обслуживания не допускается;

5) требования одного типа обслуживаются в порядке поступления;

6) после завершения обслуживания требований i-го типа, с вероятностью q_{ij} появляется требование j-го типа, которое ставится в очередь на обслуживание, где:

$$q_{ij} = \begin{cases} 1, & j = i+1,\ i \neq k, i \neq k+m, i \neq k+m+s. \\ p_1, & i = k, j = k+1. \\ p_2, & i = k+m, j = k+m+1. \\ 0, & \text{в остальных случаях.} \end{cases} \tag{8}$$

Таким образом, вышеописанная СВПВТ совпадает с системой, изученной в [1].

5. Дисциплины обслуживания

Для нахождения минимума функции (7) нам необходимо описать те дисциплины обслуживания, при которых он достигается. Эта задача решалась в работах [3] и [4]. Приведем необходимые нам результаты этих работ.

Каждая дисциплина обслуживания, при которой достигается минимум произвольной линейной функции от средних длин очередей L_i в СВПВТ, задается подстановкой $\sigma = (\sigma_1, \ldots, \sigma_n)$ множества индексов $\{1, \ldots, n\}$ следующим образом: требования, имеющие тип σ_i, будут ставиться на обслуживание раньше требований типа σ_j, если $i < j$. Опишем вспомогательные величины, которые необходимы для вычисления этих подстановок.

Введём величины γ_i – математическое ожидание суммарной длительности обслуживания (без учета ожидания) требований типа i и всех требований типа $j > i$. Ясно, что:

$$\gamma_i = \begin{cases} (k-i+1+p_1m+p_1p_2s)\theta, & 1 \leqslant i \leqslant k. \\ (k+m-i+1+p_2s)\theta, & k+1 \leqslant i \leqslant k+m. \\ (k+m+s-i+1)\theta, & k+m+1 \leqslant i \leqslant k+m+s. \end{cases} \tag{9}$$

Следуя работе [3], приоритетными индексами требований i-го типа будем называть величины R_i, которые определяются следующим образом:

$$R_i = \begin{cases} \max\left\{\dfrac{1}{\gamma_i}, \dfrac{1-p_1}{(k-i+1)\theta}, \dfrac{1-p_1p_2}{(k-i+1+p_1m)\theta}\right\}, & 1 \leqslant i \leqslant k. \\ \max\left\{\dfrac{1}{\gamma_i}, \dfrac{1-p_2}{(k+m-i+1)\theta}\right\}, & k+1 \leqslant i \leqslant k+m. \\ \dfrac{1}{\gamma_i}, & k+m+1 \leqslant i \leqslant k+m+s. \end{cases}$$

В работе [3] доказано, что дисциплина обслуживания, при которой достигается минимум функции (7), заключается в следующем: если $R_i > R_j$, то требование типа i должно ставиться на обслуживание раньше требования типа j; если же два требования имеют одинаковый приоритетный индекс, то порядок их обслуживания может быть выбран произвольно.

Из этих результатов следует, что подстановки σ для СВПВТ, определяющие дисциплины обслуживания, при которых достигается минимум функции (7), будут следующими:

$$\sigma = (*, 1, w_2, w_2-1, \ldots, w_1+1, v_1, v_1-1, \ldots, k+1, w_1, w_1-1, \ldots, m+k+1) \tag{10a}$$

$$\sigma = (*, 1, v_2, v_2-1, \ldots, v_1+1, w_1, w_1-1, \ldots, m+k+1, v_1, v_1-1, \ldots, k+1) \tag{10b}$$

$$\sigma = (*, 1, v_1, v_1-1, \ldots, k+1) \tag{10c}$$

$$\sigma = (*, 1, w_1, w_1-1, \ldots, m+k+1) \tag{10d}$$

$$\sigma = (*, 1) \tag{10e}$$

Произвольные параметры v_1, v_2, w_1, w_2 являются натуральными и изменяются в следующих пределах: $k+1 < v_1 < v_2 \leqslant k+m$, $k+m+1 < w_1 < w_2 \leqslant k+m+s$. При этом '$*$' – означает любую подстановку оставшихся индексов из множества $\{1, \ldots, k+m+s\}$.

Из описания СВПВТ очевидно следует, что для любых требований (за исключением требований 1, $k+1$ и $k+m+1$ типов), порядок обслуживания между ними можно менять произвольным образом, если выполняется следующее условие: приоритетные индексы R_i всех этих требований должны удовлетворять одному и тому же из неравенств:

$$R_1 \geqslant R_i \geqslant R_{k+1} \geqslant R_{k+m+1}; \qquad R_1 \geqslant R_i \geqslant R_{k+m+1} \geqslant R_{k+1};$$

Данное свойство было использовано при вычислении подстановок (10a)–(10e).

6. Сведения, необходимые для вычисления вершин $M_P(\theta)$

Для нахождения вершин многогранника $M_P(\theta)$ необходимо вычислить величины L_i (средние длины очередей требований в СВПВТ) для заданных дисциплин обслуживания, задаваемых подстановками, при которых реализуется минимум функции (7). Все необходимые нам подстановки описаны в уравнении (10a)–*(10e). Приведем необходимые нам результаты работ* [1] *и* [4].

Введём величины x_{ij}, являющиеся решениями следующей системы линейных уравнений:

$$\begin{cases} \sum\limits_{i=1}^{n} a_{ij}x_{ir} + \sum\limits_{i=1}^{n} a_{ir}x_{ij} = G_{jr}, & 1 \leqslant j \leqslant r \leqslant n. \\ x_{\sigma_i \sigma_j} = 0, & \text{если } i > j. \end{cases} \tag{11}$$

где константы a_{ij} и G_{ij} определяются как:

$$\begin{array}{ll} a_{ii} = 1,\ i \neq 1; & a_{ii+1} = -1,\ i \notin \{k, k+m, k+m+s\}; \\ a_{kk+1} = -p_1, & a_{k+mk+m+1} = -p_2, \\ a_{11} = 1 - \lambda\theta; & a_{i1} = -\lambda\theta,\ i \neq 1; \\ a_{ij} = 0, \text{ в остальных случаях.} & \end{array} \tag{12}$$

$$\begin{array}{l} G_{ij} = G_{ji}, \quad \forall i,j; \\ G_{ij} = 0, \quad i,j \neq 1; \\ G_{11} = 2\lambda^2\theta^2(\lambda k + p_1\lambda m + p_1p_2\lambda s)/2 = 2\lambda\theta W_1; \end{array} \qquad G_{1j} = \begin{cases} \lambda^2\theta, & 1 \leqslant j \leqslant k. \\ p_1\lambda^2\theta, & k+1 \leqslant j \leqslant k+m. \\ p_1p_2\lambda^2\theta, & k+m+1 \leqslant j \leqslant k+m+s. \end{cases} \tag{13}$$

Тогда, как показано в работе [1], средние длины очередей для СВПВТ находятся следующим образом.

$$L_j = \begin{cases} \sum\limits_{i=1}^{n} x_{ij}\theta + \theta W_1, & j = 1. \\ \sum\limits_{i=1}^{n} x_{ij}\theta, & j \neq 1. \end{cases} \tag{14}$$

Для решения уравнений (11) и вычисления L_j будет полезным следующее утверждение:

Утверждение 1. *Пусть приоритетный индекс R_j для системы СВПВТ удовлетворяет неравенству: $R_j \geqslant R_1$. Тогда $L_j = 0$.*

Доказательство. Обозначим через t_0 момент поступления первого требования в систему (это будет требование 1-го типа). Через t_α, $\alpha \in N$ обозначим моменты завершения обслуживания требований. Доказательство проведем индукцией по α. Через $L_j(t)$ обозначим длину очереди требований типа j в момент времени t. Из описания системы ясно, что во время обслуживания некоторого требования в системе могут появляться требования только 1-го типа. Следовательно, $L_j(t) = L_j(t_{\alpha-1})$ для $t_{\alpha-1} < t < t_\alpha$ и $j \neq 1$.

Если в момент времени t_1 в системе появляется требование j типа, то оно сразу ставится на обслуживание (в соответствии с определением дисциплины обслуживания, так как из него следует, что в системе одновременно не может быть два или более требований, приоритет обслуживания которых выше, чем у требований 1-го типа). Очевидно, что $L_j(t_1) = 0$. Пусть в некоторый момент времени $t_{\alpha-1}$ выполняется

$L_j(t_{\alpha-1}) = 0$. Рассмотрим систему в момент t_α. Если в этот момент в системе появилось требование типа j, то оно сразу ставится на обслуживание и длина соответствующей очереди не меняется. Итак, доказано, что $L_j(t_\alpha) = 0$ в произвольный момент времени t_α, а следовательно, и в произвольный момент времени t выполняется $L_j(t) = 0$. □

Следствие 1. *Если приоритетный индекс R_j для системы СВПВТ удовлетворяет неравенству: $R_j \geqslant R_1$, тогда для любого i: $x_{ij} = 0$.*

Доказательство. Из утверждения (1) следует, что $L_j = 0$. Величины $x_{ij} = 0$, $\forall i$ будут удовлетворять уравнениям (11), (14), и поэтому являются их решениями. □

7. Вычисление средних длин очередей для СВПВТ

7.1.

Рассмотрим перестановку 10а.

$$x_{j1} = \begin{cases} \lambda L_1 + \lambda^2(j-1)\theta, & 1 \leqslant j \leqslant k. \\ p_1\lambda^2(j-v_1)\theta, & v_1 < j \leqslant k+m. \\ p_1p_2\lambda^2(j-w_2)\theta, & w_2 < j \leqslant n. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_1 = \Big\{\lambda^2 k(k-1)\theta^2 + p_1\lambda^2(k+m-v_1+1)(k+m-v_1)\theta^2 + \\ + p_1p_2\lambda^2(n-w_2+1)(n-w_2)\theta^2 + \theta W_1\Big\}\frac{1}{2(1-\lambda k\theta)} \tag{15}$$

Рассмотрим $w_1 < r \leqslant w_2$:

$$x_{jr} = \begin{cases} \lambda L_r + p_1p_2\lambda^2\theta, & 1 \leqslant j \leqslant k. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_r = \frac{p_1p_2\lambda^2 k\theta^2}{1-\lambda k\theta}, \quad w_1 < r \leqslant w_2. \tag{16}$$

Рассмотрим $k+1 < r \leqslant v_1$:

$$x_{jr} = \begin{cases} \lambda L_r + p_1\lambda^2\theta, & 1 \leqslant j \leqslant k. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_r = \frac{p_1\lambda^2 k\theta^2}{1-\lambda k\theta}, \quad k+1 < r \leqslant v_1. \tag{17}$$

$$x_{jk+1} = \begin{cases} \lambda L_{k+1} + p_1\lambda L_1 + p_1\lambda^2 k\theta, & 1 \leqslant j \leqslant k. \\ p_1 x_{kk+1} + \dfrac{p_1^2\lambda^2\theta(j-(k+1))}{1-\lambda k\theta}, & k+1 \leqslant j \leqslant v_1. \\ p_1 x_{kk+1} + \dfrac{p_1^2\lambda^2\theta(v_1-(k+1))}{1-\lambda k\theta}, & v_1 < j \leqslant k+m. \\ \dfrac{p_1^2p_2\lambda^2\theta(j-w_1)}{1-\lambda k\theta}, & w_1 < j \leqslant w_2. \\ \dfrac{p_1^2p_2\lambda^2\theta(w_2-w_1)}{1-\lambda k\theta}, & w_2 < j \leqslant n. \\ 0, & k+m+1 \leqslant j \leqslant w_1. \end{cases}$$

$$L_{k+1} = \Big\{(p_1\lambda L_1 + p_1\lambda^2 k\theta)(k\theta + p_1 m\theta) + \frac{p_1^2\lambda^2\theta^2(v_1-(k+1))}{1-\lambda k\theta}\Big[\frac{v_1-k}{2} + k + m - v_1\Big] + \\ + \frac{p_1^2 p_2\lambda^2\theta(w_2-w_1)}{1-\lambda k\theta}\Big[\frac{w_2-w_1+1}{2} + n - w_2\Big]\Big\}\frac{1}{1-(\lambda k\theta + p_1\lambda m\theta)} \tag{18}$$

Рассмотрим $k+m+1 < r \leqslant w_1$:

$$x_{jr} = \begin{cases} \lambda L_r + p_1 p_2\lambda^2\theta, & 1 \leqslant j \leqslant k. \\ p_1 x_{kr}, & k+1 \leqslant k+m. \\ 0, & k+m+1 \leqslant j \leqslant n. \end{cases}$$

$$L_r = \frac{p_1 p_2\lambda^2\theta^2(k+p_1 m)}{1-(\lambda k\theta + p_1\lambda m\theta)}, \quad k+m+1 < r \leqslant w_1. \tag{19}$$

$$x_{jk+m+1} = \begin{cases} \lambda L_{k+m+1} + p_1 p_2\lambda^2(k+m+1-v_1)\theta, & 1 \leqslant j \leqslant k. \\ p_1 x_{kk+m+1} + p_2 x_{k+mk+1}, & k+1 \leqslant j \leqslant k+m. \\ p_2 x_{k+mk+m+1} + \dfrac{p_1^2 p_2^2\lambda^2\theta(j-(k+m+1))}{1-(\lambda k\theta + p_1\lambda m\theta)}, & k+m+1 \leqslant j \leqslant w_1. \\ p_2 x_{k+mk+m+1} + \dfrac{p_1^2 p_2^2\lambda^2\theta(w_1-(k+m+1))}{1-(\lambda k\theta + p_1\lambda m\theta)}, & w_1 < j \leqslant n. \end{cases}$$

$$L_{k+m+1} = \Big\{p_1 p_2\lambda^2\theta(k\theta + p_1 m\theta + p_1 p_2 s\theta)(k+m+1-v_1) + (p_2 m\theta + p_2^2 s\theta)(p_1^2\lambda L_1 + p_1\lambda L_{k+1}) + \\ + (p_2 m\theta + p_2^2 s\theta)\Big[p_1^2\lambda^2 k\theta + \frac{p_1^2\lambda^2\theta(v_1-(k+1))}{1-\lambda k\theta}\Big] + \\ + \frac{p_1^2 p_2^2\lambda^2\theta^2(w_1-(k+m+1))}{1-(\lambda k\theta + p_1\lambda m\theta)}\Big[\frac{w_1-(k+m)}{2} + n - w_1\Big]\Big\}\frac{1}{1-(\lambda k\theta + p_1\lambda m\theta + p_1 p_2\lambda s\theta)} \tag{20}$$

7.2.

Рассмотрим перестановку 10b.

$$x_{j1} = \begin{cases} \lambda L_1 + \lambda^2(j-1)\theta, & 1 \leqslant j \leqslant k. \\ p_1\lambda^2(j-v_2)\theta, & v_2 < j \leqslant k+m. \\ p_1 p_2\lambda^2(j-w_1)\theta, & w_1 < j \leqslant n. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_1 = \Big\{\lambda^2 k(k-1)\theta^2 + p_1\lambda^2(k+m-v_2+1)(k+m-v_2)\theta^2 + \\ + p_1 p_2\lambda^2(n-w_1+1)(n-w_1)\theta^2 + \theta W_1\Big\}\frac{1}{2(1-\lambda k\theta)} \tag{21}$$

Рассмотрим $v_1 < r \leqslant v_2$:

$$x_{jr} = \begin{cases} \lambda L_r + p_1\lambda^2\theta, & 1 \leqslant j \leqslant k. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_r = \frac{p_1\lambda^2 k\theta^2}{1-\lambda k\theta}, \quad v_1 < r \leqslant v_2. \tag{22}$$

Рассмотрим $k+m+1 < r \leqslant w_1$:

$$x_{jr} = \begin{cases} \lambda L_r + p_1 p_2\lambda^2\theta, & 1 \leqslant j \leqslant k. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_r = \frac{p_1 p_2 \lambda^2 k \theta^2}{1-\lambda k\theta}, \quad k+m+1 < r \leqslant w_1. \tag{23}$$

$$x_{jk+m+1} = \begin{cases} \lambda L_{k+m+1} + p_1 p_2 \lambda^2 (k+m+1-v_2)\theta, & 1 \leqslant j \leqslant k. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_{k+m+1} = \frac{p_1 p_2 \lambda^2 k (k+m+1-v_2)\theta^2}{1-\lambda k\theta} \tag{24}$$

Рассмотрим $k+1 < r \leqslant v_1$:

$$x_{jr} = \begin{cases} \lambda L_r + p_1 \lambda^2 \theta, & 1 \leqslant j \leqslant k. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_r = \frac{p_1 \lambda^2 k \theta^2}{1-\lambda k\theta}, \quad k+1 < r \leqslant v_1. \tag{25}$$

$$x_{jk+1} = \begin{cases} \lambda L_{k+1} + p_1 \lambda L_1 + p_1 \lambda^2 k\theta, & 1 \leqslant j \leqslant k. \\ p_1 x_{kk+1} + \dfrac{p_1^2 \lambda^2 \theta}{1-\lambda k\theta}(j-(k+1)), & k+1 \leqslant j \leqslant v_2. \\ p_1 x_{kk+1} + \dfrac{p_1^2 \lambda^2 \theta}{1-\lambda k\theta}(v_2-(k+1)), & v_2 < j \leqslant k+m. \\ p_1 x_{kk+m+1} + p_2 x_{k+mk+1} + \dfrac{p_1^2 p_2 \lambda^2 \theta}{1-\lambda k\theta}(j-(k+m+1)), & k+m+1 \leqslant j \leqslant w_1. \\ p_1 x_{kk+m+1} + p_2 x_{k+mk+1} + \dfrac{p_1^2 p_2 \lambda^2 \theta}{1-\lambda k\theta}(w_1-(k+m+1)), & w_1 < j \leqslant n. \end{cases}$$

$$\begin{aligned} L_{k+1} = \Big\{ &(k\theta + p_1 m\theta + p_1 p_2 s\theta)(p_1 \lambda L_1 + p_1 \lambda^2 k\theta) + \frac{p_1^2 \lambda^2 \theta^2 (v_2-(k+1))}{1-\lambda k\theta}\left[\frac{v_2-k}{2} + k + m - v_2\right] + \\ &+ \frac{p_1^2 p_2 \lambda^2 s\theta^2 (v_2-(k+1))}{1-\lambda k\theta} + \frac{p_1^2 p_2 \lambda^2 \theta^2 (w_1-(k+m+1))}{1-\lambda k\theta}\left[\frac{w_1-(k+m)}{2} + n - w_1\right] + \\ &+ p_1 s\theta(\lambda L_{k+m+1} + p_1 p_2 \lambda^2 (k+m+1-v_1))\theta \Big\} \frac{1}{1-(\lambda k\theta + p_1 \lambda m\theta + p_1 p_2 \lambda s\theta)} \end{aligned} \tag{26}$$

7.3.

Рассмотрим перестановку 10c.

$$x_{j1} = \begin{cases} \lambda L_1 + \lambda^2 (j-1)\theta, & 1 \leqslant j \leqslant k. \\ p_1 \lambda^2 (j-v_1)\theta, & v_1 < j \leqslant k+m. \\ p_2 x_{k+m1} + p_1 p_2 \lambda^2 (j-(k+m))\theta, & k+m+1 \leqslant j \leqslant n. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$\begin{aligned} L_1 = \Big\{ &\lambda^2 k(k-1)\theta^2 + p_1 \lambda^2 (k+m-v_1+1)(k+m-v_1)\theta^2 + \\ &+ 2p_1 p_2 \lambda^2 s(k+m-v_1)\theta^2 + p_1 p_2 \lambda^2 s(s+1)\theta^2 + \theta W_1 \Big\} \frac{1}{2(1-\lambda k\theta)} \end{aligned} \tag{27}$$

Рассмотрим $k+1 < r \leqslant v_1$:

$$x_{jr} = \begin{cases} \lambda L_r + p_1 \lambda^2 \theta, & 1 \leqslant j \leqslant k. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_r = \frac{p_1\lambda^2 k\theta^2}{1-\lambda k\theta}, \quad k+1 < r \leqslant v_1. \tag{28}$$

$$x_{jk+1} = \begin{cases} p_1\lambda L_1 + \lambda L_{k+1} + p_1\lambda^2 k\theta, & 1 \leqslant j \leqslant k. \\ p_1 x_{kk+1} + \dfrac{p_1\lambda^2\theta}{1-\lambda k\theta}(j-(k+1)), & k+1 \leqslant j \leqslant v_1. \\ p_1 x_{kk+1} + \dfrac{p_1\lambda^2\theta}{1-\lambda k\theta}(v_1-(k+1)), & v_1 < j \leqslant k+m. \\ p_1p_2 x_{kk+1} + +\dfrac{p_1p_2\lambda^2\theta}{1-\lambda k\theta}(v_1-(k+1)), & k+m+1 \leqslant j \leqslant n. \end{cases}$$

$$L_{k+1} = \Big\{(p_1\lambda L_1 + p_1\lambda^2 k\theta)(m\theta + p_1 k\theta + p_1p_2 s\theta) + \frac{p_1\lambda^2\theta^2(v_1-(k+1))}{1-\lambda k\theta}\Big[m+k-v_1+\frac{v_1-k}{2}\Big]+ \\ + \frac{p_1p_2\lambda^2 s\theta^2(v_1-(k+1))}{1-\lambda k\theta}\Big\}\frac{1}{1-(\lambda m\theta + p_1\lambda k\theta + p_1p_2\lambda s\theta)} \tag{29}$$

7.4.

Рассмотрим перестановку 10d.

$$x_{j1} = \begin{cases} \lambda L_1 + \lambda^2(j-1)\theta, & 1 \leqslant j \leqslant k. \\ p_1 x_{k1} + p_1\lambda^2(j-k)\theta, & k+1 \leqslant j \leqslant k+m. \\ p_1p_2\lambda^2(j-w_1)\theta, & w_1 < j \leqslant n. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_1 = \Big\{\lambda^2 k(k-1)\theta^2/2 + p_1\lambda^2(k-1)m\theta^2 + \\ + p_1\lambda^2(m+1)m\theta^2/2 + p_1p_2\lambda^2(n-w_1)(n-w_1+1)\theta^2/2 + \theta W_1\Big\}\frac{1}{1-(\lambda k\theta + p_1\lambda m\theta)} \tag{30}$$

Рассмотрим $m+k+1 < r \leqslant w_1$:

$$x_{jr} = \begin{cases} \lambda L_r + p_1p_2\lambda^2\theta, & 1 \leqslant j \leqslant k. \\ p_1\lambda L_r + p_1^2p_2\lambda^2\theta, & k+1 \leqslant j \leqslant k+m. \\ 0, & \text{в остальных случаях.} \end{cases}$$

$$L_r = \frac{p_1p_2\lambda^2(k+p_1m)\theta^2}{1-(\lambda k\theta + p_1\lambda m\theta)}, \quad k+m+1 < r \leqslant w_1. \tag{31}$$

$$x_{jk+m+1} = \begin{cases} \lambda L_{k+m+1} + p_1p_2\lambda L_1 + p_1p_2\lambda^2(m+k)\theta, & 1 \leqslant j \leqslant k. \\ p_1 x_{kk+m+1}, & k+1 \leqslant j \leqslant k+m. \\ p_2 x_{k+mk+m+1} + \dfrac{p_1^2p_2^2\lambda^2\theta(j-(k+m+1))}{1-(\lambda k\theta + p_1\lambda m\theta)}, & k+m+1 \leqslant j \leqslant w_1. \\ x_{w_1k+m+1}, & w_1 < j \leqslant n. \end{cases}$$

$$L_{m+k+1} = \Big\{\frac{p_1^2p_2^2\lambda^2\theta(w_1-(k+m+1))}{1-(\lambda k\theta + p_1\lambda m\theta)}\Big(\frac{w_1-(k+m)}{2}+n-w_1\Big)+ \\ + (p_1p_2L_1 + p_1p_2\lambda(m+k)\theta)(\lambda k\theta + p_1\lambda m\theta + p_1p_2\lambda s\theta)\Big\}\frac{1}{1-(\lambda k\theta + p_1\lambda m\theta + p_1p_2\lambda s\theta)} \tag{32}$$

7.5.

Рассмотрим перестановку 10е.

$$x_{j1}=\begin{cases}\lambda L_1+\lambda^2(j-1)\theta, & 1\leqslant j\leqslant k.\\ p_1x_{k1}+p_1\lambda^2(j-k)\theta, & k+1\leqslant j\leqslant k+m.\\ p_2x_{k+m1}+p_1p_2\lambda^2(j-(k+m))\theta, & w_1<j\leqslant n.\\ 0, & \text{в остальных случаях.}\end{cases}$$

$$L_1=\Big\{\lambda^2k(k-1)\theta^2/2+p_1\lambda^2m(m+1)\theta^2/2+p_1p_2\lambda^2s(s+1)\theta^2/2+ \\ +p_1\lambda^2(k-1)m\theta^2+p_1p_2\lambda^2ms\theta^2+p_1p_2\lambda^2(k-1)s\theta^2+\theta W_1\Big\}\frac{1}{1-(\lambda k\theta+p_1\lambda m\theta+p_1p_2\lambda s\theta)} \qquad (33)$$

8. Построение приоритетного множества M

Чтобы найти приоритетное множество M, необходимо вычислить $\widetilde{L}$. Для этого надо найти пределы $\lim\limits_{\theta\to0}\sum\limits_1^n L_i$ при всевозможных дисциплинах обслуживания задаваемых подстановками (10а)-(10е) – они и будут искомой величиной $\widetilde{L}$. В результате получаются формулы (2), задающие соответствующие точки $\{\widetilde{L}\}$ в одномерном пространстве, выпуклой оболочкой которых является приоритетное множество M. При этом произвольные параметры v_1, v_2, w_1, w_2 и y_1, y_2, z_1, z_2 будут связаны следующим образом:

$$y_1=\lim_{\theta\to0}(v_1-(k+1)); \qquad y_2=\lim_{\theta\to0}(v_1-(k+1));$$
$$z_1=\lim_{\theta\to0}(w_1-(k+m+1)); \qquad z_2=\lim_{\theta\to0}(w_2-(k+m+1));$$

Литература

[1] *Китаев М.Ю., Рыков В.В.* Система обслуживания с ветвящимися потоками вторичных требований // АиТ. 1980. 9. С. 52-61.

[2] *Климов Г. П.* Системы обслуживания с разделением времени. I // ТВП. 1974. Вып. 3. Т. XIX. С. 558-576.

[3] *Климов Г. П.* Системы обслуживания с разделением времени. II // ТВП. 1978. Вып. 2. Т. XXIII. С. 331-339.

[4] *Тимофеев Е.А.* Оптимизация средних длин очередей в системе обслуживания с ветвящимися потоками вторичных требований // АиТ. 1995. С. 60-67.

Модел. и анализ информ. систем. Т.9, №2 (2002) 59–62

УДК 519.2

Статистически оцениваемые инварианты динамических систем

Тимофеев Е.А.
Ярославский государственный университет
150 000, Ярославль, Советская, 14
E-mail: tim@uniyar.ac.ru

В работе описаны новые инварианты мер и динамических систем, названные *статэнтропией*. Показано, что при достаточно общих ограничениях предлагаемая статистическая оценка этих инвариантов является состоятельной. Для марковских сдвигов показано, что предложенная оценка является состоятельной оценкой энтропии и HP-спектра размерностей.

1. Символические динамические системы

Напомним построение символических динамических систем (см., напр., [3, 5]), для которых и будут вычисляться инварианты.

Пусть (X, T, P) — динамическая система с дискретным временем, порожденная непрерывным отображением $T : X \to X$, где X – компактное метрическое пространство с метрикой d, P – инвариантная мера T, которая является борелевской и вероятностной (т.е. $P(X) = 1$).

Пусть α – конечная случайная величина на вероятностном пространстве (X, P), которая принимает значения из множества $S = \{0, 1, 2, \ldots, s-1\}$ (разбиение пространства X). Построим символическую динамическую систему (Ω, σ, μ) с преобразованием - сдвигом σ и инвариантной мерой μ. Для этого рассмотрим пространство $S^{\mathbb{N}}$, состоящее из всех s-ичных последовательностей $\omega = \{\omega_n\}_{n=0}^{\infty}$, $\omega_n \in S$ с метрикой

$$\rho(x, y) = \sum_{n=0}^{\infty} \theta^{-n} |x_n - y_n|, \tag{1}$$

где $\theta > 1$ – некоторый параметр. Положим

$$\Omega = \{\omega \ : \ \omega = (\alpha(x), \alpha(T(x)), \ldots, \alpha(T^n(x)), \ldots), \ \forall x \in X\}. \tag{2}$$

Ясно, что $\sigma(\Omega) \subset \Omega$. Мера P индуцирует инвариантную меру μ на Ω стандартным образом.

2. Статэнтропия меры

Пусть Ω – компактное метрическое пространство с метрикой ρ и пусть μ – борелевски регулярная вероятностная мера на Ω.

Пусть задана строго монотонная положительная измеряющая функция $\gamma(t)$.

Через $r = \nu(t, x)$ будем обозначать радиус шара $B(x, r)$ с центром в точке x, мера которого равна t.

На вероятностном пространстве (Ω, μ) введем функционал $N_\gamma(u)$, положив

$$N_\gamma(u) = \gamma^{-1} \left(\int_\Omega \gamma(|u(x)|) d\mu(x) \right). \tag{3}$$

Нижней и верхней статэнтропиями меры μ относительно метрики ρ будем называть величины

$$\underline{\eta}(\gamma, \mu, \rho) = \underline{\lim}_{t \to 0+} \frac{\log t}{\log N_\gamma(\nu(t, \cdot))}, \quad \overline{\eta}(\gamma, \mu, \rho) = \overline{\lim}_{t \to 0+} \frac{\log t}{\log N_\gamma(\nu(t, \cdot))}. \tag{4}$$

Если $\underline{\eta}(\gamma, \mu, \rho) = \overline{\eta}(\gamma, \mu, \rho) = \eta(\gamma, \mu, \rho)$, то величину $\eta(\gamma, \mu, \rho)$ будем называть *статэнтропией меры μ*.

3. Статистическая оценка

Статистическая оценка статэнтропии меры является не сложным обобщением оценки из [2].

Пусть дано компактное метрическое пространство Ω с метрикой ρ и пусть μ – борелевски регулярная вероятностная мера на Ω.

Пусть $\xi_1, \xi_2, \dots, \xi_n$ – независимые одинаково распределенные на Ω по мере μ случайные величины.

Статистическая оценка $\eta_n(\gamma, \rho)$ статэнтропии строится следующими простыми вычислениями:

- находим вспомогательную случайную величину

$$r_n(\gamma) = \frac{1}{n}\sum_{j=1}^{n}\gamma\left(\min_{i:i\neq j}{}^{(k)}\rho(\xi_i, \xi_j)\right), \tag{5}$$

  где $\min^{(k)}\{x_1, \dots, x_N\} = x_k$, если $x_1 \leqslant x_2 \leqslant \dots \leqslant x_N$;

- полагаем оценкой статэнтропии случайную величину

$$\eta_n(\gamma, \rho) = -\frac{\log n}{\log \gamma^{-1}(r_n(\gamma))}. \tag{6}$$

Целое число k, $1 \leqslant k \ll n$ нужно для того, чтобы обеспечить существование средних значений каждого слагаемого в сумме (5) (для убывающих функций $\gamma(t)$). Если k достаточно большое, то далее будет показано, что оценка не зависит от k.

4. Ограничения на меру μ и метрику ρ

В этом разделе будут описаны все ограничения на меру μ, метрику ρ и измеряющую функцию $\gamma(t)$, выполнение которых необходимо для доказательства свойств статэнтропии.

Напомним, что *нижней и верхней локальными размерностями меры μ в точке x* называются величины

$$\underline{d}_\mu(x) = \underline{\lim}_{u\to 0}\frac{\log\mu(B(x,u))}{\log u}, \quad \overline{d}_\mu(x) = \overline{\lim}_{u\to 0}\frac{\log\mu(B(x,u))}{\log u}.$$

Если $\underline{d}_\mu(x) = \overline{d}_\mu(x) = d_\mu(x)$, то величина $d_\mu(x) = d(x)$ называется *локальной размерностью меры μ в точке x*.

Будем предполагать, что для меры μ и метрики ρ справедливы неравенства

$$\exists \overline{d} \geqslant \underline{d} > 0 \ : \ \underline{d} \leqslant \underline{d}_\mu(x) \leqslant \overline{d}(x) \leqslant \overline{d}, \quad \forall x \in \Omega. \tag{7}$$

Второе ограничение состоит в том, что $\underline{\eta}(\gamma, \mu, \rho) = \overline{\eta}(\gamma, \mu, \rho) = \eta(\gamma, \mu, \rho)$.

Третье ограничение используется для нахождения статэнтропии в пространстве последовательностей $\Omega \subset S^{\mathbb{N}}$. Введем функцию $\tau(t, x)$, положив для каждой точки $x \in \Omega \subset S^{\mathbb{N}}$

$$\tau(t,x) = \sup\{r \ : \ C_n(x) \subset B(x,r), \ \mu(C_n(x)) \leqslant t, \mu(C_{n-1}(x)) > t\}, \tag{8}$$

где $C_n(x)$ – цилиндр, первые n координат которого совпадают с координатами точки x.

Будем предполагать, что выполнено условие конечности индекса Морана [4], т.е. для некоторой константы L

$$\nu(t/L, x) \leqslant \tau(t,x) \leqslant \nu(t,x), \quad \forall x \in \Omega, t > 0 \tag{9}$$

На рассматриваемые измеряющие функции $\gamma(t)$ наложим три следующих ограничения.

1. Функция $\gamma(t)$ удовлетворяет Δ_2-условию (см., напр., [1]) В этом случае существует такая константа $q > 0$, что

$$xt \leqslant \max\{\gamma^{-1}(2^q x^q \gamma(t)), \gamma^{-1}(2^{-q}x^{-q}\gamma(t))\}, \quad \forall t > 0, x > 1. \tag{10}$$

2. Функционал $N_\gamma(u)$ удовлетворяет Δ_2-условию

$$\exists C \ : \ N_\gamma(2u)) \leqslant CN_\gamma(u) \quad \forall u(x). \tag{11}$$

3. Функция $\gamma(t)$ должна возрастать (убывать) не слишком медленно при $t \to 0$.

$$\exists \kappa > 0, \varepsilon_0 > 0 \ : \ \forall 0 < \varepsilon < \varepsilon_0 \quad 1 - \frac{\gamma(t^{1\pm\varepsilon})}{\gamma(t)} = \mathcal{O}(t^{\kappa\varepsilon}). \quad \forall 0 < t < t_0, \tag{12}$$

где знак $\pm$ выбирается в зависимости от возрастания (+) или убывания (−) функции γ.

5. Свойства статэнтропии

Для статэнтропии $\eta(\gamma,\mu,\rho)$, можно доказать справедливость следующих свойств.

1. Пусть Ω_1, Ω_2 – компактные метрические пространства с метриками ρ_1 и ρ_2, пусть μ_1 – борелевски регулярная вероятностная мера на Ω_1, пусть $F:\Omega_1\to\Omega_2$ – гомеоморфизм, пусть выполнено условие (11) и пусть существует статэнтропия $\eta(\gamma,\mu_1,\rho_1)$ для тройки (Ω_1,ρ_1,μ_1), тогда статэнтропия существует для тройки $(\Omega_2,\rho_2,\mu_2=\mu_1\circ F^{-1})$ и равна той же самой функции $\eta(\gamma,\mu_1,\rho_1)$.

2. Пусть μ, $\tilde\mu$ – борелевски регулярные вероятностные меры на Ω такие, что для некоторой константы $C>1$

$$C^{-1}\mu(S)\leqslant\tilde\mu(S)\leqslant C\mu(S)\quad\forall S\subset\Omega,$$

пусть для функции $\gamma^{-1}(t)$ выполнено Δ_2-условие и пусть существует статэнтропия $\eta(\gamma,\mu,\rho)$ для тройки (Ω,ρ,μ), тогда статэнтропия существует для тройки $(\Omega,\rho,\tilde\mu)$ и равна той же самой функции $\eta(\gamma,\mu,\rho)$.

3. *Пусть выполнено условие* (7), *тогда*

$$\underline{d}\leqslant\underline{\eta}(\gamma,\mu,\rho)\leqslant\overline{\eta}(\gamma,\mu,\rho)\leqslant\overline{d}.$$

4. Пусть выполнено условие (7) и пусть для μ-почти всех точек $x\in\Omega$ существует $d(x)$, тогда существует статэнтропия $\eta(-\log,\mu,\rho)$ и

$$\eta(-\log,\mu,\rho)=\left(\int_\Omega d(x)^{-1}d\mu(x)\right)^{-1}.$$

Теорема 1. *Пусть существует $\eta(\gamma,\mu,\rho)$ и пусть функция $\gamma(t)$ удовлетворяет Δ_2-условию и условию (12), тогда при $k\underline{d}>q$, где q определено в (10), существует*

$$\lim_{n\to\infty}\frac{\log\gamma^{-1}(\mathsf{E}r_n(\gamma))}{\log n}=-\frac{1}{\eta(\gamma,\mu,\rho)}.$$

Это утверждение, в частности, показывает, что оценка $\eta_n(\gamma,\rho)$ не зависит от параметра k (если он достаточно большой).

Теорема 2. *Пусть существует $\eta=\eta(\gamma,\mu,\rho)$ и выполнено условие (7) и пусть для заданного q из (10) параметр k удовлетворяет ограничению*

$$(k-2)\underline{d}>2q. \tag{13}$$

Пусть функция $\gamma(t)$ удовлетворяет Δ_2-условию, условию (12) и для некоторого $d>0$

$$\frac{\gamma(t^{1/\eta(\gamma^2,\mu,\rho)})}{\gamma(t^{1/\eta(\gamma,\mu,\rho)})}=\mathcal{O}(t^{(d-1)/2}), \tag{14}$$

тогда для $c<\min\{\underline{d}^2,1/\overline{d}^2,d\}$

$$\mathsf{D}\left[\frac{r_n(\gamma)}{\gamma(n^{-1/\eta})}\right]=\mathcal{O}(n^{-c}). \tag{15}$$

Здесь и далее γ^2 обозначает квадрат функции γ.

Теорема 3. *Пусть сдвиг σ является эргодическим преобразованием с конечной энтропией $h(\sigma)$ в символической динамической системе (Ω,σ,μ) и пусть выполнены условия (9), (7), тогда*

$$\eta(-\log,\mu,\rho)=\frac{h(\sigma)}{\log\theta}.$$

6. Сдвиги Маркова

Пусть задана марковская цепь с множеством состояний $M = \{0, 1, \ldots, m-1\}$, матрицей переходных вероятностей $||a_{ij}||_0^{m-1}$ и стационарным распределением $\{p_i : i \in M\}$ на M, где

$$p_j = \sum_{i=0}^{m-1} p_i a_{ij}, \quad j \in M.$$

Для простоты будем считать, что $a_{ij} > 0$.

Возьмем метрическое пространство $X = M^{\mathbb{N}}$ с метрикой (1) с параметром $\theta = 1/2$. Мера P определяется как марковская мера по начальному распределению p_i и матрице переходных вероятностей a_{ij}. Преобразование T будет сдвигом, который и называется (односторонним) сдвигом Маркова.

Энтропия $h(T)$ сдвига Маркова хорошо известна (см., напр., [3]).

$$h(T) = -\sum_{i=0}^{m-1} \sum_{j=0}^{m-1} p_i a_{ij} \log a_{ij}.$$

Выберем случайную величину α так, чтобы на цилиндрах $C_0, C_1, \ldots, C_{m-1}$ она принимала значения $0, 1, \ldots, m-1$ (образующее разбиение). По случайной величине α построим пространство $\Omega = M^{\mathbb{N}}$ и выберем метрику (1) с произвольным параметром $\theta > 1$. Возьмем меру $\mu = P$.

Нетрудно проверить, что условия (9), (7) выполнены.

Применяя утверждение 3, получим, что для измеряющей функции $\gamma(t) = -\log t$ статэнтропия совпадает с энтропией с точностью до множителя $1/\log\theta$. Неравенство (14) выполнено с $d = 1/2$. Поэтому статистическая оценка (5)-(6) (при достаточно большом k) является состоятельной оценкой энтропии.

Найдем статэнтропию для функций $\gamma(t) = t^q$. Обозначим через $\Phi(\beta)$ спектральный радиус матрицы $||b_{ij}^{\beta}||_0^{m-1}$, где

$$b_{ij} = \frac{a_{ij} p_i}{p_j}, \quad i, j \in M.$$

Теорема 4. *Пусть $\gamma(t) = t^q$, $\Omega = M^{\mathbb{N}}$ с метрикой, заданной в (1), и марковской мерой μ, тогда $\eta(\gamma, \mu, \rho) = \frac{q}{1-\beta(q)}$, где $\beta(q)$ является корнем уравнения $\Phi(\beta) = \theta^q$.*

Сравнивая статэнтропию меры μ с *HP-спектром размерностей* [5, с.182], получим, что статэнтропия меры μ с измеряющей функцией $\gamma(t) = t^q$ совпадает с HP-спектром размерностей, построенным по параметру $\beta(q)$.

Нетрудно проверить, что неравенство (14) выполнено с $d = 2q(\beta) - q(2\beta) > 0$. Поэтому, статистическая оценка (5)-(6) для $\gamma(t) = t^q$ (и достаточно большого k) является состоятельной.

Литература

[1] Красносельский М.А., Рутицкий Я.Б. Выпуклые функции и пространства Орлича. М.: ГИФМЛ. 1958.

[2] Майоров В.В., Тимофеев Е.А. Статистическая оценка обобщенных размерностей //Мат. заметки. 2002. Т.71. №5. С. 697 - 712.

[3] Синай Я.Г. Введение в эргодическую теорию. Ереван : Изд-во ЕГУ. 1973.

[4] Темпельман А.А. Размерность случайных фракталов в метрических пространствах //Теория вероят. и ее примен. 1999. Т.44, №3.

[5] Pesin Ya. Dimension Theory in Dynamical Systems: Contemporary Views and Applications. Chicago : The University of Chicago Press, 1997.