

УДК 681.322.06:681.3.323

АЛГОРИТМЫ МЕЖМАШИННЫХ ОБМЕНОВ И РЕКОНФИГУРАЦИИ  
ГРАФОВ СВЯЗЕЙ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С  
ПРОГРАММИРУЕМОЙ СТРУКТУРОЙ

Э.А. Монахова

Технико-экономические ограничения приводят к невозможности построения полных графов связей вычислительных систем (ВС) с программируемой структурой [1,2]. Поэтому установление пути передачи данных между двумя элементарными машинами (ЭМ) ВС происходит через транзитные ЭМ, и, следовательно, возникает экстремальная задача поиска кратчайших путей на графе сети связи. Для некоторых классов регулярных графов можно рассчитывать пути непосредственно по описанию графа. Примером таких графов являются хордовые кольца [3], остовы  $n$ -мерных булевых кубов [4]. Процедура определения кратчайших путей по описанию графа имеет смысл, если она достаточно проста и не требует задания графа в табличной или списковой форме. И здесь в первую очередь представляют интерес графы с параметрическим описанием [5-7].

В настоящей работе показана возможность эффективного решения задачи поиска кратчайших путей по описанию графа для класса диофантовых оптимальных структур [7,8] с аналитически вычисляемым параметрическим описанием [5], рассмотренных в качестве графов связей вычислительных систем, и предложены:

1) децентрализованные, адаптивные, отказоустойчивые путевые процедуры как альтернатива табличному методу организации путевых процедур;

2) алгоритмы трансляционных и трансляционно-циклических межмашинных обменов;

3) алгоритмы реконфигурации ВС, сохраняющие при изменении числа ЭМ принадлежность графов связей ВС к рассматриваемому классу.

## §1. Решение экстремальной задачи поиска пути на графе сети связи

Рассмотрим в качестве модели сети связи ВС с программируемой структурой, состоящей из  $N$  ЭМ, графы, принадлежащие классу  $D^*$  оптимальных двумерных диофантовых структур с аналитически вычисляемым описанием [5]:  $D_2^*$ -граф  $\in D^*$  есть  $D_2$ -граф с числом вершин  $N$  и параметрическим описанием вида  $\{N; 1, 1+1\}$ , где  $1 = \lceil (\sqrt{2N-1} - 1)/2 \rceil$  ( $\lceil [x] \rceil$  - ближайшее целое  $x$ ). Число  $N$  называется порядком  $D_2^*$ -графа, числа  $1$  и  $1+1$  - его образующими [7].

Оптимальные  $D_2^*$ -графы, рассматриваемые как элементы класса  $D^*$ , обладают следующими свойствами [5]:

- а) существуют для любого натурального  $N \geq 5$ ;
- б) имеют аналитически вычисляемое параметрическое описание и диаметр;
- в) имеют описание вида  $\{N; 1, 1+1\}$  для данного  $1$  при изменении  $N$  от  $21^2-1$  до  $21^2+41+3$ .

Заметим, что хордовые кольца [3], являющиеся подграфами  $D_2$ -графов, свойствами "а"- "в" не обладают.

Будем сопоставлять вершины  $D_2^*$ -графа элементарным машинам ВС, а ребра - каналам, соединяющим соседние ЭМ. Абсолютная адресация ЭМ [7] соответствует нумерации вершин  $D_2^*$ -графа числами  $0, \dots, N-1$ .

Для двух вершин  $D_2^*$ -графа  $i, j \in \overline{0, N-1}$  относительный адрес [7] вершины  $j$  относительно  $i$  есть вектор  $A_{i,j} = (x^{i,j}, y^{i,j})$ , где  $|x^{i,j}| (|y^{i,j}|)$  равен числу дуг с отметкой  $1$  ( $1+1$ ), лежащих на кратчайшем пути из  $i$  в  $j$ , а  $\text{sgn} x^{i,j}$  ( $\text{sgn} y^{i,j}$ ) задает направление движения в кратчайшем пути по (+) или против (-) образующей  $1$  ( $1+1$ ). Относительный адрес  $A_{i,j}$  задает все множество кратчайших путей из вершины с номером  $i$  в вершину с номером  $j$ . Выходные полюса каждой ЭМ перенумерованы числами  $1, 2, 3, 4$ , как показано на рис. 1, а, и соответствуют образующим  $1+1, 1, -(1+1), -1$ . Внутри ЭМ имеется таблица, называемая таблицей образующих, которая ставит в соответствие номеру выходного направления имя образующей и ее знак.

Для геометрического представления  $D_2^*$ -графа достаточно каждому единичному квадрату с центром в точке  $(x, y)$ ,  $|x|, |y| \leq \lceil (\sqrt{2N-1} - 1)/2 \rceil$ , с целочисленными координатами  $x$  и  $y$  сопоставить вершину  $D_2^*$ -графа с номером  $(x1 + y(1+1)) \bmod N$ . При этом вершины связаны ребром, если для их номеров  $i$  и  $j$  выполняется  $(i-j) \bmod N \in \{1, 1+1, -1, -(1+1)\}$ . Легко видеть, что при таком геометрическом представлении все вершины  $D_2^*$ -графа располагаются

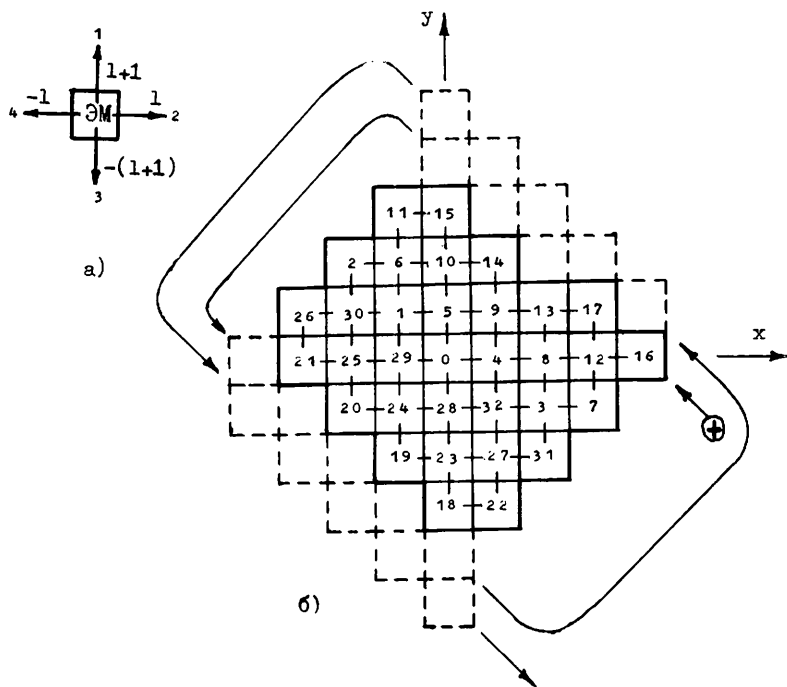


Рис. I

внутри ромба с диагональю  $2d^*$ . Число  $d^*$  является диаметром  $D_2^*$ -графа и вычисляется [5] по формуле  $d^* = [(\sqrt{2N-1} - 1)/2]$ .

На рис. I, б изображен  $D_2^*$ -граф  $\{33; 4, 5\}$ . Для простоты изображения показаны только ребра между вершинами  $x_1 + y(1+1)$  и  $x'1 + y'(1+1)$ , для которых выполняется либо  $|x-x'|=1$ ,  $y = y'$ , либо  $x = x'$ ,  $|y-y'|=1$ . Числа, стоящие в квадратах, есть номера вершин. Относительный адрес вершины с номером  $i \in \{1, N-1\}$  относительно вершины с номером 0 совпадает при таком геометрическом представлении с прямоугольными координатами  $(x, y)$  вершины  $i$ . При возрастании  $N$  от  $21^2 - 1$  до  $21^2 + 41 + 3$  (в области оптимальности описания  $\{N; 1, 1+1\}$ ) ромб последовательно заполняется новыми вершинами (квадраты, обозначенные пунктиром, в порядке, указанном стрелкой).

Число  $i^*(N)$  - это номер той вершины  $D_2^*$ -графа, начиная с которого все вершины расположатся в нижней части ромба левее линий  $x = -y$  для  $x \leq 0$  и  $x-1 = -y$  для  $x > 0$ . В [5] даны следующие правила определения  $i^*$ :

$$\begin{aligned} i^* &= N-1^2+1 && \text{при} && 21^2-1 \leq N \leq 21^2+1, \\ i^* &= 1^2+1+1 && \text{при} && 21^2+1 \leq N \leq 21^2+21+1, \\ i^* &= N-1^2-1 && \text{при} && 21^2+21+1 \leq N \leq 21^2+31+2, \\ i^* &= 1^2+21+2 && \text{при} && 21^2+31+2 \leq N \leq 21^2+41+3. \end{aligned}$$

Для  $D_2^*$ -графа, изображенного на рис. 1, б,  $i^* = 18$ .

Покажем, что для  $D_2^*$ -графа между относительным адресом  $A_{ij}$  и абсолютными адресами  $i$  и  $j$  существует простая и явная связь, не связанная с решением сравнений  $j-i \equiv 1x^{1j} + (1+1)y^{1j} \pmod{N}$ .

По свойству  $D_n$ -графов [7]  $(x^{0k}, y^{0k}) = (-x^{0, N-k}, -y^{0, N-k})$  для любого  $k \in \overline{0, N-1}$ .

Для  $k \geq i^*$  из выражения  $i^*$  через  $N$  следует  $N-k < i^*$  и, таким образом, определение относительного адреса  $A_{jk}$  для всех  $k \in \overline{i^*, N-1}$  сводится к определению относительного адреса вершины с номером  $(N-k)$ , лежащей в верхней части ромба (правее линий  $x = -y$  для  $x \leq 0$  и  $x-1 = -y$  для  $x > 0$ ), и заменой знаков полученных координат на противоположные.

Для вершин с номерами  $0 \leq k < i^*$  кратчайшие пути из вершины с номером 0 аналитически определяет следующая

**ТЕОРЕМА I.** В  $D_2^*$ -графе порядка  $N$  пусть  $0 \leq k < i^*$  - абсолютный адрес вершины и  $\Delta_k = k \bmod 1$ . Тогда относительный адрес  $A_{0k} = (x^{0k}, y^{0k})$  определяется следующим образом:

$$x^{0k} = \left[ \frac{k}{1} \right] - \Delta_k, \quad y^{0k} = \Delta_k \quad \text{при} \quad \Delta_k - 1 \leq \left[ \frac{k}{1} \right] - \Delta_k < 1+1; \quad (1)$$

$$x^{0k} = \left[ \frac{k}{1} \right] - \Delta_k + (1+1), \quad y^{0k} = \Delta_k - 1 \quad \text{при} \quad \left[ \frac{k}{1} \right] - \Delta_k < \Delta_k - 1; \quad (1')$$

$$x^{0k} = \left[ \frac{k}{1} \right] - \Delta_k - (1+1), \quad y^{0k} = \Delta_k + 1 \quad \text{при} \quad \left[ \frac{k}{1} \right] - \Delta_k \geq 1+1. \quad (1'')$$

ДОКАЗАТЕЛЬСТВО. Пусть  $0 \leq k < i^*(N)$  - номер вершины  $D_n^*$ -графа, max  $i^*(N) = 1^2 + 21 + 2$ . Требуется для любого натурального  $0 \leq k \leq 1^2 + 21 + 1$  определить  $x^{0k} = x$ ,  $y^{0k} = y$ , где  $x, y$  - координаты вершины с номером  $k$ , таковы, что

$$k = x1 + y(1+1). \quad (2)$$

При доказательстве будем использовать тот факт, что для любых натуральных  $k$  и  $1$  число  $k$  единственным образом представимо через целую часть и остаток от деления на  $1$ :

$$k = \left[ \frac{k}{1} \right] 1 + \Delta_k, \quad 0 \leq \Delta_k < 1. \quad (3)$$

Случай (I). Рассмотрим область ромба, ограниченную следующими прямыми:

$$\left. \begin{array}{l} 0 \leq y < 1 \\ -y \leq x \\ y-1 \leq x < 1+1 \end{array} \right\}.$$

Отсюда  $x+y \geq 0$  и  $0 \leq y < 1$ . Таким образом, если представить записанный в виде (2) номер  $k$  вершины из этой области в виде  $k = (x+y)1 + y$ , то в силу единственности выражения (3) получим:

$$\left. \begin{array}{l} \left[ \frac{k}{1} \right] = x+y \\ \Delta_k = y \end{array} \right\},$$

т.е.  $x = \left[ \frac{k}{1} \right] - \Delta_k$ ,  $y = \Delta_k$ . И эти формулы имеют место при выполнении условия  $\Delta_k - 1 \leq \left[ \frac{k}{1} \right] - \Delta_k = x < 1+1$ .

Случай (I'). Рассмотрим область ромба, ограниченную следующими прямыми:

$$\left. \begin{array}{l} y < 0 \\ -y+1 \leq x \leq y+1 \end{array} \right\}.$$

Отсюда  $x+y-1 \geq 0$  и  $0 < 1+y < 1$ . Таким образом, если представить номер  $k$  вершины из этой области, записанный в виде (2), в следующем виде:  $k = (x+y-1)1 + (1+y)$ , то в силу единственности выражения (3) для  $k$  получим:

$$\left. \begin{array}{l} \left[ \frac{k}{1} \right] = x+y-1 \\ \Delta_k = 1+y \end{array} \right\},$$

т.е.  $x = \left[ \frac{k}{1} \right] - \Delta_k + (1+1)$ ,  $y = \Delta_k - 1$ .

Эти формулы имеют место при выполнении условия  $\left[\frac{k}{1}\right] \leq 2y+1-1 = 2\Delta_k - 1 - 1$ , т.е. при  $\left[\frac{k}{1}\right] < 2\Delta_k - 1$ .

Случай (I''). Для вершин с номерами  $k=1^2+1, 1^2+21, 1^2+21+1$  координаты равны соответственно  $(0,1), (1,1), (0, 1+1)$ . Вычисленные по формулам (I'') координаты совпадают с данными. Это имеет место при выполнении условия  $x \geq 0$ , т.е.  $\left[\frac{k}{1}\right] - \Delta_k \geq 1+1$ . Теорема доказана.

Заметим, что в определении  $A_{0k}$  по формулам (I)-(I'') меньше операций, чем в аналогичных вычислениях для хордовых колец [3].

Основной проблемой при организации межмашинных обменов в ВС является определение кратчайших путей между вершинами  $D_2^*$ -графа по известным относительным адресам их по отношению к третьей вершине (без потери общности будем считать этой третьей вершиной нулевую).

При рассмотрении двух точек  $i$  и  $j$  с координатами соответственно  $A_{0i} = (x^{0i}, y^{0i})$  и  $A_{0j} = (x^{0j}, y^{0j})$  в прямоугольной системе координат на бесконечной плоскости относительный адрес  $A_{ij} = (x^{ij}, y^{ij})$  вычисляется по формуле

$$A_{ij} = A_{0j} - A_{0i} \quad (4)$$

и всегда задает кратчайший путь между ними.

Вследствие замкнутости  $D_2^*$ -графа, кратчайшие пути между его вершинами вычисляются по формуле (4) не всегда. Обозначим через  $\alpha$  величину, равную  $|x^{0j} - x^{0i}| + |y^{0j} - y^{0i}|$ . Ясно, что если  $\alpha > d^*$ , то формула (4) неприменима для определения  $A_{ij}$ , так как между  $i$  и  $j$  существует путь длины, меньшей или равной  $d^*$ . При  $\alpha = d^*$  в одних случаях формула (4) применима, в других нет. Например, в  $D_2^*$ -графе  $\{I8; 3, 4\}$  пусть  $A_{0i} = (0, 1)$  и  $A_{0j} = (-2, 0)$ . Тогда  $\alpha = |-2-0| + |0-1| = 3 = d^*$ , но  $A_{ij} = (0, 2) \neq A_{0j} - A_{0i}$ .

Справедлива следующая

**ТЕОРЕМА 2.** В  $D_2^*$ -графе порядка  $N$  пусть  $A_{0i} = (x^{0i}, y^{0i})$  и  $A_{0j} = (x^{0j}, y^{0j})$  — относительные адреса вершин  $i$  и  $j$ . Если

$$|x^{0j} - x^{0i}| + |y^{0j} - y^{0i}| < d^*, \quad (5)$$

то  $A_{ij} = A_{0j} - A_{0i}$ .

**ДОКАЗАТЕЛЬСТВО.** По свойству  $D_n$ -графов [7], если  $j-1 \equiv j' - i' \pmod{N}$ , то  $A_{ij} = A_{i'j'}$ , поэтому

$$A_{ij} = \begin{cases} A_{0, j-1} & \text{при } j > i, \\ A_{0, N+j-1} & \text{при } j < i, \end{cases}$$

т.е.  $A_{ij} = A_{0k}$  и для вершины  $k$   $|x^{0k}| + |y^{0k}| < d^*$ . А всем вершинам  $k$ , для которых выполняется последнее неравенство,  $A_{0k}$  задает кратчайший путь из 0 в  $k$ . Тогда и  $A_{ij}$ , вычисленный по формуле (4), задает кратчайший путь между  $i$  и  $j$ , т.е. является относительным адресом. Теорема доказана.

**ТЕОРЕМА 3.** В  $D_2^*$ -графе порядка  $N$  пусть  $A_{0i} = (x^{0i}, y^{0i})$  и  $A_{0j} = (x^{0j}, y^{0j})$  — относительные адреса вершин  $i$  и  $j$ . Если  $|x^{0j} - x^{0i}| + |y^{0j} - y^{0i}| > d^*$ , то  $A_{ij} = A_{0k}$ , где

$$k = \begin{cases} \beta & \text{при } \beta > 0, \\ N + \beta & \text{при } \beta < 0 \end{cases} \quad (6)$$

$$\text{и } \beta = (x^{0j} - x^{0i})1 + (y^{0j} - y^{0i})(1+1).$$

**ДОКАЗАТЕЛЬСТВО.** В теореме утверждается, что кратчайший путь из вершины  $i$  в вершину  $j$  при  $\alpha > d^*$  не вычисляется по формуле (4), а совпадает с кратчайшим путем из вершины 0 в вершину  $k$ , где  $k$  определяется по формуле (6). Заметим, что  $\beta = x^{0j}1 + y^{0j}(1+1) - [x^{0i}1 + y^{0i}(1+1)] = j - i$  и так же, как в доказательстве теоремы 2,  $A_{ij} = A_{0k}$ , где  $k$  вычисляется по формуле (6). Теорема доказана.

Полученные теоремы используются при организации алгоритма поиска кратчайшего пути (§2). При этом в случае  $\alpha = d^*$  не распознается, когда  $A_{ij}$  можно вычислять по формуле (4), а когда нет (вследствие сложности процедуры распознавания), и  $A_{ij}$  определяется так же, как и в случае  $\alpha > d^*$ .

Результаты теорем 1-3 полностью переносятся на любой выделенный связный подграф  $D_2^*$ -графа при задании на нем относительной адресации вершин, принимаая за нулевую любую фиксированную вершину подграфа.

## §2. Алгоритмы парных обменов

В ВС с программируемой структурой межмашинные обмены организуются между машинами виртуальной подсистемы [1]. Универсальные алгоритмы [2] прокладывает трассы передачи данных только через ЭМ, включенные в подсистему. При этом пути ищутся не кратчайшие, а

близкие к ним. В данной работе такое ограничение не накладывается, а именно используется все множество кратчайших путей, что позволяет определять альтернативные пути в случае отказов ЭМ и линий связи и учитывать распределение нагрузки ЭМ системы при поиске кратчайших путей.

В противоположность табличному методу задания путевых функций, используемых при определении кратчайших путей, в ВС, построенных на основе  $D_2^*$ -графов, возможен аналитический расчет кратчайших путей, проводимый в динамике вычислительного процесса.

Рассмотрим путевую процедуру (алгоритм определения трассы передачи данных между ЭМ), эффективно реализующую парные взаимодействия машин произвольной виртуальной подсистемы (дифференцированный обмен) и основанную на операторе поиска кратчайшего пути.

Передача данных из передающей ЭМ в принимающую начинается с формирования в передающей ЭМ коммутационного сообщения. Структура сообщения включает служебную часть и текст сообщения. В служебную часть входят признаки используемого типа обмена, необходимые адреса и т.д.

Итак, пусть необходимо передать коммутационное сообщение из передающей ЭМ в принимающую кратчайшим путем при возможных отказах ЭМ и линий связи. Относительный адрес первоначально определяется в передающей ЭМ и определяет все множество кратчайших путей между передающей и принимающей ЭМ. Относительный адрес входит в служебную часть коммутационного сообщения и модифицируется в каждой транзитной ЭМ по пути в принимающую так, чтобы текущий относительный адрес задавал все множество кратчайших путей из этой транзитной ЭМ в принимающую.

При реальном выполнении путевой процедуры выбор транзитной ЭМ может происходить согласно относительному адресу и текущим состояниям соседних ЭМ. А именно: при поступлении в ЭМ коммутационного сообщения путевая процедура выбирает направление, принадлежащее множеству кратчайших путей до принимающей ЭМ. Очевидно, выбор должен исключать отказавшее направление или отказавшую ЭМ.

Если при этом существует более одной альтернативы для выбора, то в качестве транзитной ЭМ выбирается предпочтительно ЭМ с меньшей нагрузкой - с меньшим числом поступивших сообщений. Модификация относительного адреса при этом заключается в изменении на единицу одной из его координат.



Если все направления, принадлежащие множеству кратчайших путей до принимающей ЭМ, отказались, то выбор производится среди оставшихся направлений по степени нагрузки. В этом случае, если состояние, определяющее транзитную ЭМ от принимающей ЭМ, больше диаметра системы, путевая процедура вычисляет относительный адрес заново.

В описанном ниже алгоритме  $A_{0i}$  - относительный адрес ЭМ, исполняющей путевую процедуру;  $A_{0j}$  - относительный адрес принимающей ЭМ;  $A_{ij}$  - относительный адрес принимающей ЭМ относительно ЭМ, исполняющей путевую процедуру;  $A$  - множество номеров четырех выходных направлений ЭМ;  $B$  - множество номеров выходных направлений, принадлежащих кратчайшему пути;  $C$  - множество номеров отказавших выходных направлений; " $B \setminus C > 0$ " означает, что на входной полюс с номером 0 (т.е. из самой ЭМ, реализующей путевую процедуру) поступило коммутационное сообщение с признаком "дифференцированный обмен"; " $B \setminus C < 0$ " означает, что на входной полюс с номером  $h$  ( $h = 1, 4$ ) поступило коммутационное сообщение с признаком "дифференцированный обмен".

Алгоритм поиска кратчайшего пути при наличии отказов

1. Если  $B \setminus C > 0$ , то на 2. Если  $B \setminus C < 0$ , то, если "отказа нет", то, если  $A_{ij} = 0$ , то идти на 6, иначе на 5, иначе на 2.

2. Если условие (5) выполнено, то вычислить  $A_{ij}$  по формуле (4), идти на 5, иначе на 3.

3. Вычислить  $k$  по формуле (6). Если  $k \geq i^*$ , то  $k := N - k$ ,  $\text{sgn}(k) := \langle - \rangle$ , идти на 4, иначе  $\text{sgn}(k) := \langle + \rangle$ , идти на 4.

4. Вычислить  $A_{ok}$  по (I)-(I"),  $A_{ij} := \text{sgn}(k)A_{ok}$ , идти на 5.

5. Если  $v \in C$ , то из  $A \setminus C$  выбрать любое направление, в сообщении сформировать "отказ" и послать сообщение по выбранному направлению, идти на 6; иначе из  $B \setminus C$  выбрать любое направление, в сообщении сформировать "отказа нет", модифицировать  $A_{ij}$  (если это  $x^{ij}$ , то  $x^{ij} := x^{ij} - \text{sgn}x^{ij} \cdot 1$ , если  $y^{ij}$ , то  $y^{ij} := y^{ij} - \text{sgn}y^{ij} \cdot 1$ ) и послать сообщение по соответствующему выходному направлению, идти на 6.

6. Конец.

Модифицированный алгоритм, учитывающий при выборе направления степень нагрузки соседней ЭМ, получается из вышеописанного добавлением в п.5 критерия загруженности ЭМ (т.е. выбирать не любое направление, а направление к наименее загруженной ЭМ).

При наличии отказавших ЭМ и линий связи отк а з о у с т о й ч и в ы й алгоритм поиска кратчайшего пути должен находить кратчайший путь до принимающей ЭМ, если он существует, или за конечное время определить, что нужного пути не существует.

Вышеизложенный алгоритм еще не гарантирует отказоустойчивой работы, но при следующих дополнительных действиях, внесенных в него, отказоустойчивость достигается.

Каждая транзитная ЭМ (включая и принимающую) запоминает номер входного (с которого пришло сообщение) направления. Номер направления (с адресом передающей ЭМ и номером сообщения в ней (чтобы различать разные сообщения, отправленные одной ЭМ)) хранится в ЭМ до тех пор, пока по нему не пройдет в обратном направлении снова то же сообщение.

Рассмотрим работу алгоритма в случае, когда из  $(i+1)$ -й ЭМ сообщение вернулось в  $i$ -ю ЭМ по тому направлению, номер которого запоминали для данного сообщения. При этом возможно два случая:

- 1) принимающая ЭМ не достигнута (признак = "нет");
- 2) принимающая ЭМ достигнута и возвращаемся по найденному кратчайшему пути для его закрепления и передачи по нему сообщения (признак = "есть").

Рассмотрим случай I. Если сообщение передается обратно по тому направлению, по которому передавалось в  $(i+1)$ -ю ЭМ, то это означает, что в  $(i+1)$ -й ЭМ все другие выходные направления уже просмотрены и оказались либо отказавшими, либо тупиковыми. При этом все заблокированные (те каналы связи, по которым сообщение прошло как в прямом направлении, так и в обратном, блокируются для данного сообщения) в  $(i+1)$ -й ЭМ выходные направления разблокируются. При попадании сообщения в  $i$ -ю ЭМ направление, ведущее в  $(i+1)$ -ю ЭМ, блокируем с тем, чтобы при выборе направления из  $i$ -й ЭМ снова не пойти по уже пройденному раз направлению. Таким образом, отсекаются тупиковые ветви.

Просматривание всех возможных путей достигается тем, что перебираем все оставшиеся направления по критериям: принадлежности к кратчайшему пути из  $i$ -й в принимающую и степени нагруженности соседней ЭМ.

Если число оставшихся после блокировки направлений не больше единицы, то при  $i$ , совпадающем с номером  $r$  передающей ЭМ, это означает, что принимающая ЭМ недостижима из передающей, а при  $i \neq r$  принимающая ЭМ недостижима из  $i$ -й ЭМ и возвращаемся в  $(i-1)$ -ю

ЭМ, разблокировав предварительно в  $i$ -й все заблокированные направления и заблокировав путь из  $(i-1)$ -й ЭМ в  $i$ -ю ЭМ.

Если имеет место случай 2, то закрепляем каналы, по которым попадает сообщение в каждую транзитную ЭМ, и передаем сообщение. При передаче сообщения все заблокированные на время поиска пути направления снова разблокировываются.

Итак, приведенные выше дополнения, внесенные в работу алгоритма, гарантируют корректность алгоритма поиска кратчайшего пути при наличии отказов, и таким образом, предложенная путевая процедура, реализующая парные взаимодействия машин подсистемы, является отказоустойчивой.

Свойства предложенного алгоритма поиска кратчайших путей:

1) полное отсутствие таблиц путевых функций (экономия памяти по сравнению с [2] порядка  $O(\log_2 N)$  бит);

2) память, требуемая для поля адреса в сообщении, в  $4\log_2 N / \log_2 \sqrt{N}$  раз меньше требуемой в  $D(z, m)$  адресации [2];

3) возможность определения альтернативных путей в случае отказов ЭМ и линий связи;

4) возможность адаптации к распределению нагрузки машин системы.

### §3. Трансляционные и трансляционно-циклические обмены

Трансляционные и трансляционно-циклические обмены [9] составляют большую часть межмашинных обменов в ВС с программируемой структурой. Существующие универсальные алгоритмы [2, 10] предложены для произвольных графов связей ВС и могут быть применены для  $D_2^*$ -графов, но при этом никак не учитываются свойства последних, благодаря которым возможно получение более эффективных алгоритмов.

В настоящей работе рассматривается организация трансляционных и трансляционно-циклических обменов на подсистемах, граф связи которых совпадает с графом связи всей системы, при предположении абсолютной надежности всех ЭМ и линий связи между ними.

Первая задача состоит в следующем: требуется организовать передачу коммутационного сообщения из любой ЭМ системы во все остальные (трансляционный обмен) и при этом избежать передач избыточных копий сообщения. Последнее требование вызвано тем, чтобы минимизировать нагрузку на сети. Возможное решение этой задачи заключается в следующем:

1) построить на системе  $(D_2^*$ -графе) покрывающее дерево [11] с корнем в передающей ЭМ;

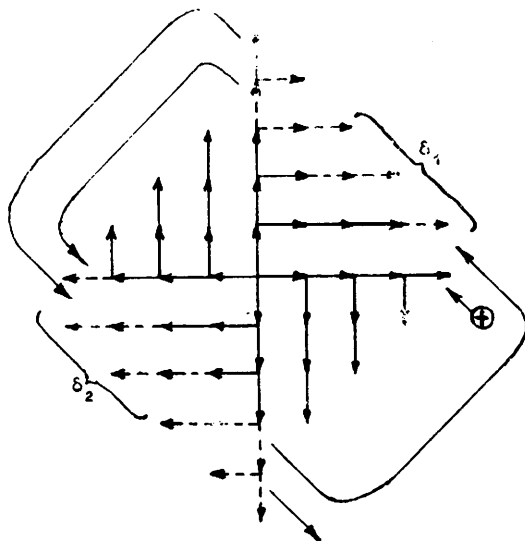


Рис. 2

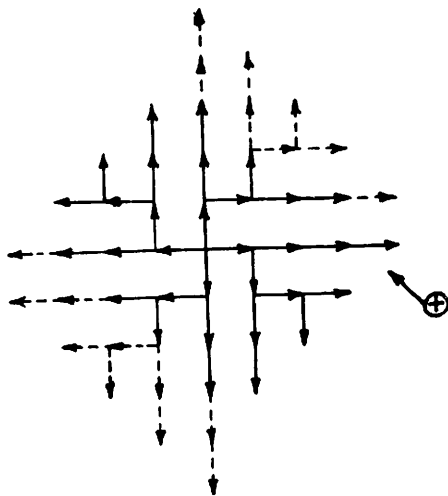


Рис. 3

2) передавать копии коммутационного сообщения только по направленным ребрам этого дерева.

Для  $D_2^*$ -графа  $\{33; 4, 5\}$ , изображенного на рис. 1, б, на рис. 2 и 3 представлены два из возможных покрывающих деревьев (сплошными линиями). Стрелками указан порядок появления новых ребер покрывающего дерева (изображены пунктиром) при увеличении  $N$  в области оптимальности описания  $\{N; 4, 5\}$ .

Алгоритм организации трансляционного обмена рассмотрим на примере дерева, изображенного на рис. 2, как более простого.

Предварительные пояснения.

" $VX_0 > 0$ " означает, что на входной полюс с номером 0 (т.е. из самой ЭМ, реализующей путьную процедуру) поступило коммутационное сообщение, которое необходимо передать всем остальным ЭМ системы.

" $VX_i > 0$ " ( $i \in \{1, 2, 3, 4\}$ ) означает, что на входной полюс с номером

1 из соседней ЭМ поступило коммутационное сообщение, которое не - обходимо принять данной ЭМ (послать в Вых $\emptyset$ ) и передать копии его в те ЭМ (послать в Вых $i$ ,  $i \in \{1, 2, 3, 4\}$ ), которые определяются путевой процедурой.

(Вых $i \rightarrow$  Вых $j$ ) означает, что первое сообщение с входного полюса  $i$  пересылается в выходной полюс  $j$ .

Для определения номеров выходных полюсов и при необходимости пересылки копий сообщения алгоритм трансляционного обмена формирует в служебной части коммутационного сообщения параметры  $\{R, U, V\}$ , где:  $R \in \{0, d^*\}$  - счетчик расстояния ЭМ, исполняющей путевую процедуру, от передающей ЭМ;  $U \in \{0, 1\}$  определяет число направлений (одно для  $U = 0$  и два для  $U = 1$ ), по которым ЭМ, исполняющая путевую процедуру, передает копии сообщения;  $V \in \{0, d^*+1\}$  определяет при  $R = d^*-1$ : нужна дальнейшая передача копии сообщения ( $V > 0$ ) или не нужна ( $V = 0$ ).

При изменении  $N$  в области оптимальности будем различать два случая:

$$21^2-1 \leq N \leq 21^2+21+1, \quad (7)$$

$$21^2+21+1 \leq N \leq 21^2+41+3 \quad (7')$$

Соответственно вычисляем  $\delta = N - (21^2 - 1)$  для случая (7) и  $\delta = N - (21^2 + 21 + 1)$  для случая (7'). По значению  $\delta$  вычисляем в обоих случаях  $\delta_1$  и  $\delta_2$ : если  $\delta \leq 1+1$ , то  $\delta_1 := \delta$ ,  $\delta_2 := 0$ , иначе  $\delta_1 := 1+1$ ,  $\delta_2 := \delta - \delta_1$ . Число  $\delta_1$  ( $\delta_2$ ) означает количество ребер покрывающего дерева, имеющих для данного  $N$  на последнем ярусе в I (III) квадранте (см. рис. 6, а). Путевая процедура в передающей ЭМ формирует в сообщении из  $\delta_1$  и  $\delta_2$  параметр  $V$ .

Таким образом ЭМ, исполняющей путевую процедуру, при принятии сообщения, прошедшего  $R = d^*-1$  шагов от передающей ЭМ, копия сообщения передается, если она необходима. И если в ЭМ пришло сообщение, прошедшее  $R = d^*$  шагов от передающей, то его необходимо принять, так как оно первый раз попадает в данную ЭМ. Действия передающей ЭМ при трансляционном обмене и ЭМ, исполняющей трансляционный обмен, отражают соответственно блок-схемы, представленные на рис. 4 и 5.

Вторая задача состоит в следующем: организовать передачу коммутационных сообщений из каждой ЭМ системы во все остальные

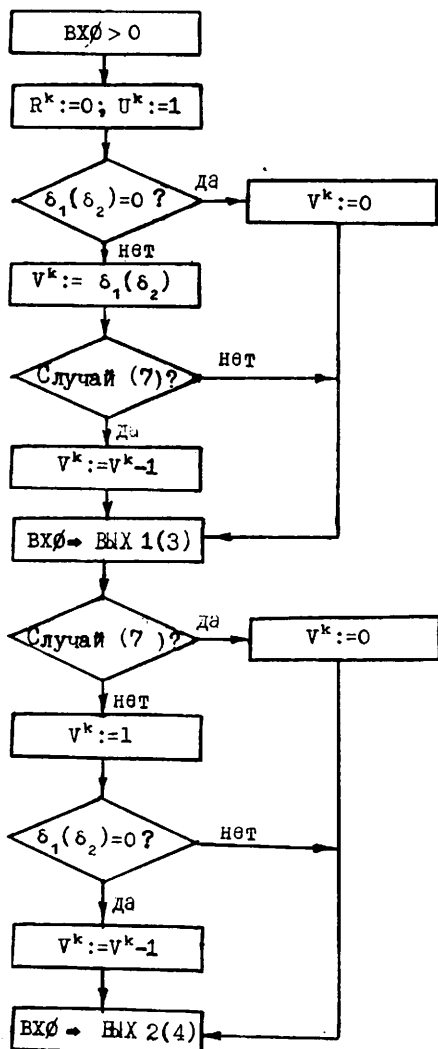


Рис. 4

ботки путевой процедурой сообщений, поступающих на входные полюса.

(трансляционно-циклический обмен) и при этом избежать передач избыточных копий сообщений.

Ниже предлагается один из возможных алгоритмов трансляционно-циклического обмена, основанный на вышеизложенном алгоритме трансляционного обмена.

Трансляционно-циклический обмен рассматривается как суперпозиция трансляционных обменов, выполненных для каждой ЭМ системы. Выход на трансляционно-циклический обмен происходит в каждой ЭМ независимо от других по мере возникновения коммутационного сообщения. В результате каждая ЭМ передаст всем остальным ЭМ свои данные без избыточных передач (таково свойство предложенного трансляционного обмена) и получит данные от всех ЭМ, но порядок получения этих данных не будет определен. Включение в сообщение адреса передающей ЭМ лишь частично решает проблему, поскольку приходится упорядочивать сообщения в принимающих ЭМ.

Введем упорядоченность в сам процесс обра-

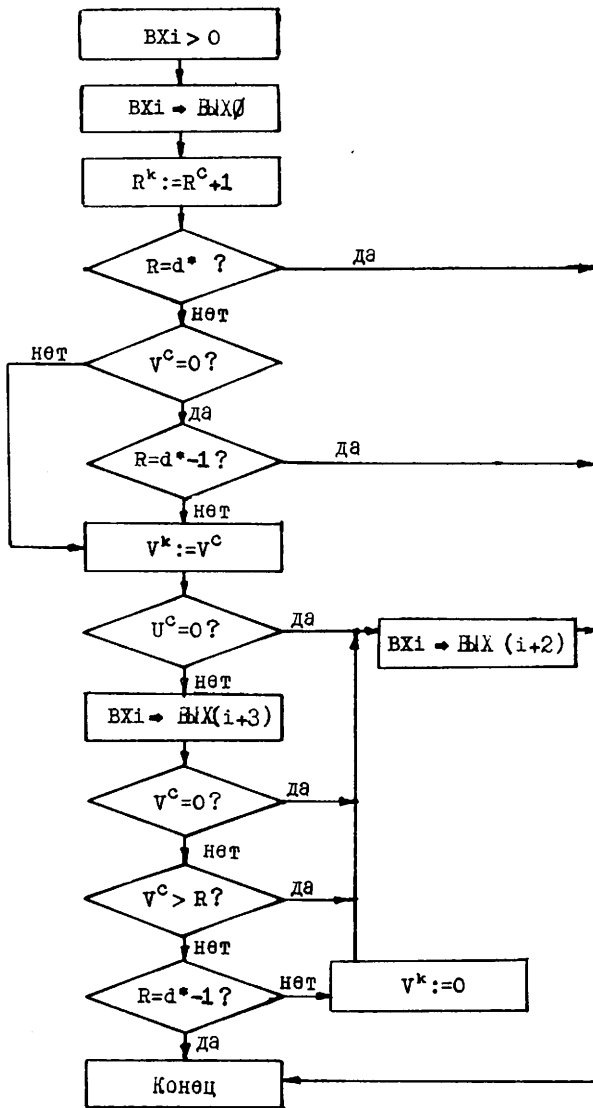


Рис.5. Действия ЭМ, исполняющей трансляционный обмен.

Куда передавать, или номера выходных направлений для посту -  
пвшего сообщения, определяет алгоритм трансляционного обмена.

Последовательность приема с входных направлений и обработки  
поступивших сообщений для алгоритма трансляционного обмена зада-  
ется следующими правилами:

- 1) первым обработать сообщение, поступившее на  $VX\emptyset$  ;
- 2)  $i := 1$  ;
- 3) обработать  $F_R(i)$  сообщений, поступивших на  $VXi$  ;
- 4)  $i := i+1$  ; если  $i > 2n$  , то  $i := 1$  , перейти на 3.

Функция  $F_R(i)$  задает число сообщений, прошедших расстояние  
 $R$  от передающей ЭМ до ЭМ, исполняющей путевую процедуру, и посту-  
пивших на  $VXi$  .

$$\text{Для } R \in \overline{1, d^* - 1} \text{ для всех } i \in \overline{1, 4} \quad F_R(i) = R .$$

$$\text{Для } R = d^* \text{ и } \delta_2(\delta_1) = 1+1$$

$$F_R(2) (F_R(4)) = \delta_2(\delta_1) - 1 ,$$

$$F_R(1) (F_R(3)) = \begin{cases} 1 & \text{в случае (7),} \\ 1 & \text{в случае (7') .} \end{cases}$$

$$\text{Для } R = d^* \text{ и } \delta_2(\delta_1) < 1+1$$

$$F_R(2) (F_R(4)) = \delta_2(\delta_1) ,$$

$$F_R(1) (F_R(3)) = \begin{cases} 1-1 & \text{в случае (7),} \\ 0 & \text{в случае (7') .} \end{cases}$$

Значения  $F_R(i)$  становятся понятны, если рассмотреть дерево  
стока, или дерево путей, по которым в произвольную ЭМ системы  
(она соответствует центральному квадрату ромба) стекаются (пере-  
даются) сообщения из остальных ЭМ системы. Это дерево изображено  
на рис. 6, а, и получается как суперпозиция путей, проходящих со-  
общением в центральную ЭМ ромба из каждой ЭМ ромба по соответст-  
вующему ей дереву передач, изображенному на рис.2. Пунктирные ли-  
нии соответствуют ребрам дерева стока, появляющимся при изменении  
числа  $N$  в области оптимальности, а стрелки задают последователь-  
ность появления этих ребер. При этом все ЭМ системы относительно  
принимающей оказываются разбиты на 4 области, соответствующие ка-  
налам, по которым сообщения из них поступают в принимающую ЭМ. По-



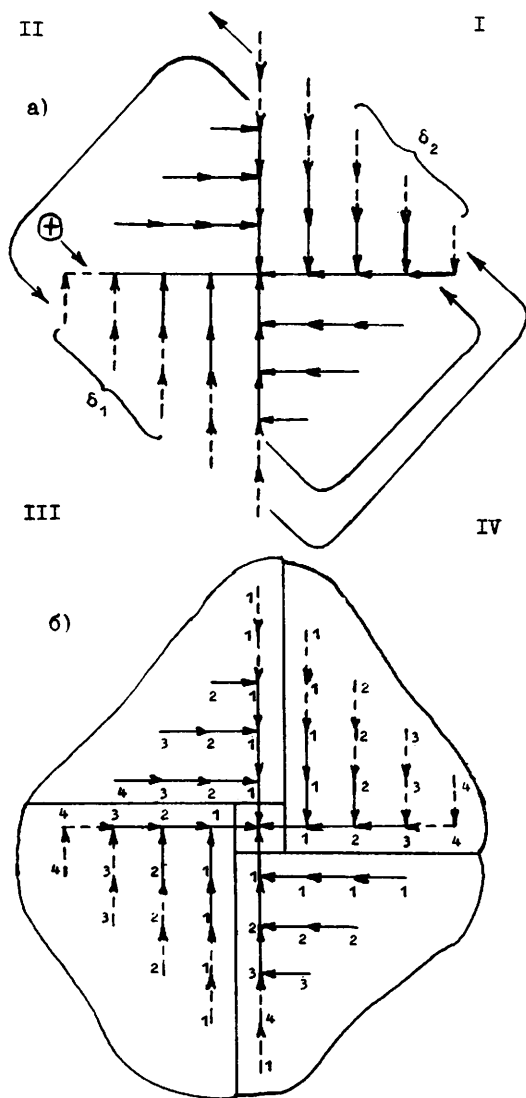


Рис. 6

рядок поступления сообщений от ЭМ, находящихся на одинаковом расстоянии, указан цифрами на рис. 6, б. При  $R \in \overline{1, d^* - 1}$  на каждый канал принимающей ЭМ поступает  $R$  сообщений от ЭМ, принадлежащих соответствующей области. При  $R = d^*$  число сообщений зависит от значения  $N$  и легко вычисляется по  $\delta_1$  и  $\delta_2$  (рис. 6, а), где  $\delta_1$  и  $\delta_2$  имеют тот же смысл и значение, что и  $\delta_1$  и  $\delta_2$  на рис. 2.

Алгоритм трансляционно-циклического обмена заканчивает работу после обработки  $R_B(i)$  сообщений, поступивших на  $BX_i$  при  $i = 4$  и  $R = d^*$ .

В отличие от универсальных алгоритмов [2] в предложенных алгоритмах трансляционных и трансляционно-циклических обменов не требуется задания адресов передающих ЭМ и табличного задания трансляционных функций. Роль последних выполняют

#### 64. Алгоритмы реконфигурации вычислительных систем

Исследованные в [5] свойства оптимальных  $D_2^*$ -графов позволяют простым способом производить перенумерацию ЭМ и перекоммутацию их связей в ВС при наращивании числа ЭМ и сохранении оптимальности структуры ее связей. В ВС, построенных на основе  $D_2^*$ -графов, обеспечивается преемственность программного обеспечения при изменении числа ЭМ. Все предлагаемые в §I-3 алгоритмы межмашинных обменов могут быть реализованы при любом  $N$ . Таким образом, число ЭМ в системе оказывается единственным параметром, на который должно настраиваться программное обеспечение.

В [5] приводятся оценки необходимо изменяемого (программно и аппаратно) числа связей при реконфигурации системы, вызванной изменением числа ЭМ, с условием сохранения оптимальности структуры связей. Под аппаратным изменением понимается пересоединение связи, а под программным изменением — замена имени образующей в таблице образующих, соответствующей данной связи.

Ниже рассмотрены конкретные алгоритмы перестройки макроструктуры при подключении к ВС, состоящей из  $N$  ЭМ,  $(N+1)$ -й ЭМ. Алгоритмы сформулированы в терминах теории графов.

Будем считать  $N \in W_1$ , если  $2l^2 - 1 \leq N \leq 2l^2 + 4l + 2$ , и  $N \in W_2$ , если  $2l^2 + 4l + 1 \leq N \leq 2l^2 + 4l + 3$ , где  $l$  — любое натуральное число. Первый интервал задает область значений  $N$ , при которых возможна перестройка оптимального  $D_2^*$ -графа  $\{N; 1, 1+1\}$  в оптимальный  $D_2^*$ -граф  $\{N+1; 1, 1+1\}$ . Указанную перестройку осуществляет алгоритм I. Второй интервал задает область значений  $N$ , при которых возможна перестройка оптимального  $D_2^*$ -графа  $\{N; 1, 1+1\}$  в оптимальный  $D_2^*$ -граф  $\{N+1; 1+1, 1+2\}$ , т.е. переход к оптимальному описанию того же вида, но с увеличенными на единицу образующими. Данную перестройку осуществляет алгоритм II.

Пусть  $0 \leq i \leq N$  — номер, присваиваемый новой вершине.

##### Алгоритм I

1. Для всех  $j \in \overline{1, N-1}$  делать:  $j := j+1$ . (Перенумерацию вершин можно избежать при  $i=N$ .)

2. Для всех  $j \in [i-1, i-1]$  и  $j \in [i+1, i+1]$  делать: направлениям связи по образующим соответственно  $1, -1$  присвоить программно значения образующих  $1+1, -(1+1)$ .

3. Для всех  $j \in [i+1, i+1]$  делать: линию связи  $(j, j-(1+2))$  пересоединить в линию  $(j, j-1)$ . Для вершины  $j = i+(1+1)$  линию связи  $(j, j-(1+2))$  пересоединить в линию  $(j, j-(1+1))$ .

4. Для вершины  $i$  построить две новые линии связи:  $(i, i-1)$  и  $(i, i-(1+1))$ .

ТЕОРЕМА 4. Для  $N \in W_1$  алгоритм I задает преобразование  $D_2^*$ -графа  $\{N; 1, 1+1\}$  в  $D_2^*$ -граф  $\{N+1; 1, 1+1\}$ . При этом 1 связей изменяются программно,  $1+1$  связей пересоединяются и две новые связи присоединяются.

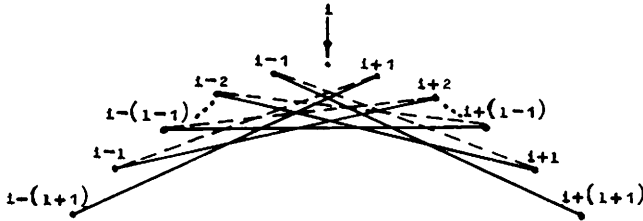


Рис. 7

ДОКАЗАТЕЛЬСТВО. На рис.7 изображен фрагмент  $D_2^*$ -графа  $\{N; 1, 1+1\}$  после выполнения первого шага алгоритма. Пунктирной линией изображены линии связи, участвующие во 2 шаге алгоритма. Сплошной линией обозначены линии связи, которые подлежат пересоединению по шагу 3. Теорема будет доказана, если после выполнения шагов 2-4 все вершины с номерами от  $i-(1+1)$  до  $i+(1+1)$  по-прежнему будут связаны линиями связи с образующими  $\pm 1, \pm(1+1)$ , так как остальные вершины  $D_2^*$ -графа в перестройке не участвуют и остаются неизменными после работы алгоритма. Линии связи с образующими  $1, -1$ , соединяющие до выполнения шага I вершины  $i-r$  и  $i+1-r$ ,  $r \in [1, 1]$ , после выполнения его связывают уже вершины  $i-r$  и  $i+(1+1)-r$ ,  $r \in [1, 1]$ , что соответствует связям по образующим  $1+1, -(1+1)$ . Это изменение 1 образующих программно реализуется шагом 2. После выполнения шага 3 вершины с номерами  $j \in [i+1, i+1]$  связаны образующими  $1, -1$  с вершинами  $j \in [i, i-(1-1)]$ , а вершина  $i$  с вершиной  $i+(1+1)$  образующей  $1+1$ . Заметим, что у вершин из  $[i+1, i+1]$  и  $[i-1, i-(1-1)]$  уже есть полный набор образующих  $\pm 1, \pm(1+1)$ , а у вершины  $i$  есть образующие  $1$  и  $(1+1)$ . Пересоединяется всего  $1+1$  линия связи. Теперь у вершин  $i-(1+1)$  и  $i-1$  отсутствуют соответственно образующие  $1+1$  и  $1$ , а у вершины  $i$  отсутствуют образующие  $-(1+1)$  и  $-1$ . После выполнения шага 4 все образующие оказываются на месте, т.е. построен  $D_2^*$ -граф  $\{N+1; 1, 1+1\}$ . Теорема доказана.

Таким образом, в  $BC$ , построенных на основе  $D_2^*$ -графов, при наращивании  $\mathcal{M}$  возможно ценой небольшой перестройки переходить снова к оптимальной структуре с тем же параметрическим описанием. Например, для  $BC$  с числом  $\mathcal{M}$  порядка 100 при подключении новой  $\mathcal{M}$  7 связей необходимо пересоединить, 6 переименовать и 2 построить заново.

#### Алгоритм П

1. Выполнить шаг I алгоритма I.
2. Выполнить шаг 2 алгоритма I.
3. Для всех  $j \in [i-(1+1), i-1]$  и  $j \in [i+1, i+(1+1)]$  делать: направления связи по образующим соответственно  $1+1, -(1+1)$  присвоить программно значения образующих  $1+2, -(1+2)$ .
4. Для всех  $j \in [i+1, i-(1+2)]$  линию связи  $(j, j+1)$  пересоединить в линию  $(j, j+1+2)$ . Для вершины  $j = i-(1+1)$  линию связи  $(j, j+1)$  пересоединить в линию  $(j, j+1+1)$ .
5. Для вершины  $i$  построить две новые линии связи:  $(i, i+1+1)$  и  $(i, i+1+2)$ .

ТЕОРЕМА 5. Для  $N \in W_2$  алгоритм П задает преобразование  $D_2^*$ -графа  $\{N; 1, 1+1\}$  в  $D_2^*$ -граф  $\{N+1; 1+1, 1+2\}$ . При этом  $2i+1$  связей изменяются программно,  $N-1$  связей пересоединяются и две новые связи присоединяются.

ДОКАЗАТЕЛЬСТВО. Алгоритм основан на том, чтобы при включении в  $D_2^*$ -граф новой вершины с номером  $i$  и соответствующем перенумеровании вершин сохранить, где можно, образующую  $1+1$ , а образующую  $1$  заменить на  $1+2$ . Теорема будет доказана, если после выполнения шагов I-5 алгоритма П каждая из  $N+1$  вершин будет иметь образующие  $\pm(1+1), \pm(1+2)$ .

После выполнения шага 4 у всех вершин  $j \in [i+1, i-(1+2)]$  появляется линия связи с образующей  $1+2$  (вместо  $1$ ). У всех  $j \in [i+1+(1+2), i]$  появляется линия связи с образующей  $-(1+2)$ . Заметим, что у вершины с номером  $i$  появляется образующая  $-(1+2)$ . Таким образом, у всех вершин с номерами  $j \in [i+1+3, i-(1+2)]$  есть образующие  $1+2$  и  $-(1+2)$ . Образующие  $1+1, -(1+1)$  у этих вершин не меняются при работе алгоритма.

Остается рассмотреть вершины с номерами из интервала  $[i-(1+1), i+(1+2)]$ .

У вершины  $j=i-(l+1)$  есть все четыре образующие: есть  $-(l+2)$ , образующая  $l+1$  программно изменилась в  $l+2$  после шага 3, образующая  $l$  пересоединилась в  $l+1$ , образующую  $-(l+1)$  алгоритм не меняет. При этом при пересоединении линии связи  $l$  в  $l+1$  у вершины с номером  $i$  появляется образующая  $-(l+1)$ .

После  $l-3$  шагов у всех вершин с номерами  $j \in [i-1, i+1]$  образующая  $l$  заменяется программно на  $l+1$ , а образующая  $l+1$  - на  $l+2$  и после шага 4 появляется образующая  $-(l+2)$ . Образующая  $-(l+1)$  не менялась, т.е. у вершин из интервала  $[i-1, i+1]$  есть все образующие.

Соответственно у вершин с номерами  $j \in [i+1, i+1]$  после  $l-3$  шагов образующая  $l$  заменяется программно на  $-(l+1)$ , образующая  $-(l+1)$  - на  $-(l+2)$ , и вместо  $l$  появляется образующая  $l+2$ . Образующая  $l+1$  не менялась, т.е. у вершин из интервала  $[i+1, i+1]$  есть все образующие после  $l-4$  шагов.

Осталось построить у вершины  $j=i+1+1$  образующую  $-(l+1)$ , у вершины  $j=i+1+2$  образующую  $-(l+2)$ , у вершины  $i$  две образующие  $l+1$ ,  $l+2$ , что и выполняется на шаге 5 алгоритма. Теорема доказана.

### Заключение

В качестве графов связей ВС с программируемой структурой рассмотрен класс оптимальных диофантовых структур размерности два. Свойство регулярности данных структур позволяет решать экстремальную задачу поиска пути на графе связей аналитически в динамике вычислительного процесса. Полученный в статье алгоритм поиска кратчайших путей в отличие от существующих универсальных алгоритмов не требует памяти под таблицы путевых функций, позволяет определять альтернативные пути в случае отказов ЭМ и линий связи, дает возможность учитывать распределение нагрузки машин системы при поиске кратчайших путей. Использование относительной адресации ЭМ при расчете кратчайших путей дает также уменьшение памяти, требуемой для поля адреса в сообщении, по сравнению с  $D(z, m)$ -адресацией, используемой в универсальных алгоритмах.

Свойство регулярности рассматриваемых структур используется в предложенной в статье организации трансляционных и трансляционно-циклических обменов. В полученных алгоритмах обменов не используются адреса передающих ЭМ и таблицы трансляционных функций.

В ВС, построенных на основе рассмотренного класса оптимальных диофантовых структур, возможно сохранение принадлежности к данному классу структур при изменении числа ЭМ. В настоящей работе предложены алгоритмы реконфигурации, позволяющие простым способом производить перестройку системы при наращивании числа ЭМ и сохранении оптимальности структуры ее связей.

### Л и т е р а т у р а

1. КОРНЕЕВ В.В., ХОРОШЕВСКИЙ В.Г. Вычислительные системы с программируемой структурой. - Электронное моделирование, 1979, №1, с. 42-52.

2. КОРНЕЕВ В.В., МОНАХОВ О.Г. Организация межмашинных взаимодействий в вычислительных системах с программируемой структурой. - Электронное моделирование, 1980, № 5, с. 16-22.

3. ARDEN B.W., LEE H. Analysis of Chordal Ring Network. - IEEE Trans. on Computers, 1981, v.C-30, N 4, p.291-295.

4. SULLIVAN H., BASHKOW T.R. A large scale, homogeneous, fully distributed parallel machine. - Computer Architecture News, 1977, v.5, N 7, p.105-124.

5. МОНАХОВА Э.А. Об аналитическом задании оптимальных двумерных диофантовых структур однородных вычислительных систем. - В кн.: Однородные вычислительные системы (Вычислительные системы, вып. 90). Новосибирск, 1981, с. 81-92.

6. МОНАХОВ О.Г. Параметрическое описание структур однородных вычислительных систем. - В кн.: Вопросы теории и построения вычислительных систем (Вычислительные системы, вып. 80). Новосибирск, 1979, с. 3-17.

7. ВОРОБЬЕВ В.А. Простейшие структуры однородных вычислительных систем. - В кн.: Вычислительные системы. Вып. 60. Вопросы теории и построения вычислительных систем. Новосибирск, 1974, с. 35-49.

8. КОРНЕЕВ В.В. О макроструктуре однородных вычислительных систем. - Там же, с. 17-34.

9. КОСАРЕВ Ю.Г. О схемах обмена между ветвями параллельных алгоритмов. - В кн.: Вычислительные системы. Вып. 51, Новосибирск, 1972, с. 70-75.

10. КОРНЕЕВ В.В. Организация параллельных вычислений в вычислительных системах с программируемой структурой. - Новосибирск, 1981. - 48 с. (Препринт/ИМ СО АН СССР: ОВС-13).

11. ХАРАРИ Ф. Теория графов. - М.: Мир, 1973. - 300 с.

Поступила в ред.-изд.отд.  
16 августа 1982 года