ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
1975 год Выпуск 63

УДК 681.142.4

АЛГОРИТМЫ РАСПОЗНАВАНИЯ ПОДСИСТЕМ ЗАДАННЫХ СТРУКТУР В ОДНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ

Н.Н.Миренков, С.Б.Фиерман

Одна из задач диспетчера однородной вычислительной системы (ОВС) [1] – организация подсистем заданной конфигурации из имеющегося набора элементарных машин (ЭМ) [2]. Сложность такой организации в реальных системах связана с функциональной неоднородностью (например, занятием некоторых подсистем приоритетными задачами, которые нельзя прерывать, или выходом машин из строя), которая приводит к необходимости объединения в единое целое "разбросанных" ЭМ.

Подсистема ОВС рассматривается как подмножество ЭМ, обладающее возможностью реализации, по крайней мере, одного режима функционирования из некоторого заданного набора. Режимы функционирования определяются классами решаемых задач и способами использования ресурсов подсистемы. Для подсистем выделяются [2,3] следующие основные режимы работы: диспетчера, автономной работы ЭМ, параллельной обработки, профилактики.

Для разных режимов функционирования необходимы, вообще говоря, подсистемы разной структуры. Однако можно найти достаточно универсальные структуры, на которых реализуются фактически все основные режимы. Структура считается универсальной, если она может выполнить любую схему обменов между ЭМ из полного набора схем [4]. Поскольку схемы обменов в разных структурах реализуются с различной скоростью, то важно для каждого режима выявить наиболее часто используемые схемы обменов и, опираясь на это, наиболее эффективные структуры.

Установлено, что для n-мерных ОВС, в которых каждая ЭМ связана с $2 \times n$ соседними, наиболее эффективны структуры подсистем*), представленные на рис.1. Анализ возможности реализации таких структур на заданном подмножестве ЭМ – одна из важных функций Главного диспетчера ОВС, управляющего разбиением системы на подсистемы [5].

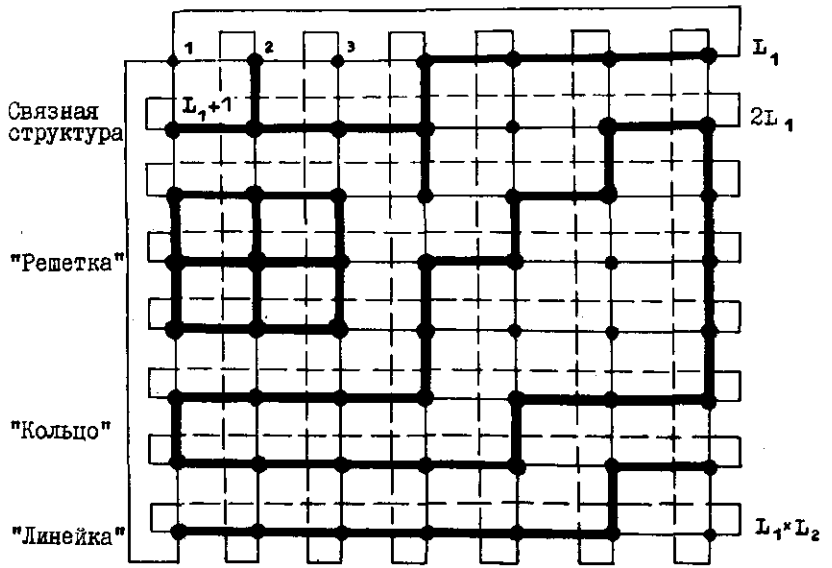


Рис.1. Основные структуры подсистем

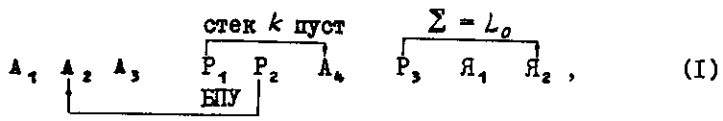
1. Связная структура.

ОПРЕДЕЛЕНИЕ 1. Подмножество элементарных машин $M = \bigcup_{k=1}^{L_0} M_{i_k}$ позволяет реализовать подсистему "связной" структуры, если для каждой пары ЭМ (M_{i_1}, M_{i_2}) существует связывающий их путь, проходящий через $M_i \in M$.

ОПРЕДЕЛЕНИЕ 2. Две ЭМ M_{i_1} и M_{i_2} называются соседними, если модуль разности $i_1 - i_2$ равен одной из следующих величин 1, L_1, $L_1 - 1$, $L_1(L_2 - 1)$, где L_1, L_2 – число ЭМ в ОВС соответственно по осям X, Y.

*) Для наглядности алгоритмы рассматриваются на двумерных E_n-графах [6].

А л г о р и т м[*], распознающий связные структуры, может быть представлен в виде следующей операторной схемы:



(I)

где A_1 – $k := 0$; выбирает произвольную ЭМ $M_{i_0} \in M$ и заносит её номер в стек с номером k;

A_2 – $k := k + 1$;

A_3 – в стек k заносит номера машин, которые 1) не содержатся ни в одном из предыдущих стеков; 2) соседние с машинами из стека $k-1$;

P_1 – передает управление (ПУ) на A_4, если стек k пуст;

P_2 – безусловно передает управление (БПУ) на A_2 ;

A_4 – вычисляет общее количество Σ ЭМ, зарегистрированных во всех стеках;

P_3 – ПУ на Я$_2$, если $\Sigma = L_0$ (L_0 – число ЭМ в подмножестве М);

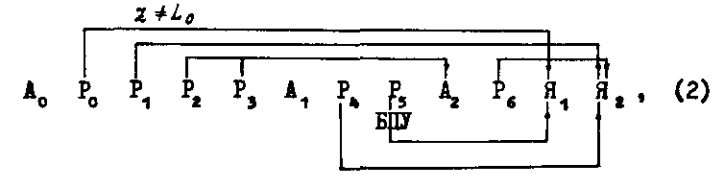
Я$_1$ – оператор окончания, сигнализирующий о невозможности реализовать связную подсистему;

Я$_2$ – оператор успешного окончания.

2. С т р у к т у р а "р е ш е т к а".

ОПРЕДЕЛЕНИЕ 3. Подмножество ЭМ $M = \bigcup_{k=1}^{L_0} M_{i_k}$ позволяет реализовать подсистему структуры "решетка" размером $n_1 \times n_2 = L_0$, если оно умещается на прямоугольнике таких же размеров[**].

А л г о р и т м, распознающий эту структуру, может быть представлен в виде следующей операторной схемы:



(2)

[*] Алгоритмы решения подобной задачи см. также в [7,8].

[**] Расстояние между соседними ЭМ считается равным единице.

где A_0 – вычисляет $z = n_1 \times n_2$;

P_0 – ПУ на Я$_1$, если $z \neq L_0$;

P_1 – ПУ на Я$_2$, если $(n_1 \times n_2 = L_1 \times L_2) \wedge (n_1 \times n_2 = L_1 + L_2)$;

P_2 – ПУ на A_2, если $(n_1 = L_1) \vee (n_1 = L_2)$, при этом в стек заносится n_1 ;

P_3 – ПУ на A_2 , если $(n_2 = L_1) \vee (n_2 = L_2)$, при этом в стек заносится n_2 ;

A_1 – вычисляет $\alpha = 2(n_1 + n_2) - 8$, $\beta = L_0 - 2(n_1 + n_2) + 4$;

P_4 – ПУ на Я$_2$, если выполняются следующие соотношения: $L(2) = 4$, $L(3) = \alpha$, $L(4) = \beta$, $L(j)$ – число ЭМ в подмножестве М, которые имеют по j соседей из M ;

P_5 – БПУ на Я$_1$;

A_2 – вычисляет $\alpha = 2X$, где X – содержимое стека и $\beta = L_0 - \alpha$;

P_6 – ПУ на Я$_2$, если $(L(3) = \alpha) \wedge (L(4) = \beta)$;

Я$_1$ – оператор окончания, сигнализирующий о невозможности реализовать "решетку";

Я$_2$ – оператор успешного окончания.

2. С т р у к т у р а "л и н е й к а".

ОПРЕДЕЛЕНИЕ 4. Подмножество ЭМ $M = \bigcup_{k=1}^{L_0} M_{i_k}$ позволяет реализовать подсистему структуры "линейка", если через его элементы можно провести гамильтонову цепь [9].

ОПРЕДЕЛЕНИЕ 5. Пусть $\bar{M}_i \subset M$ – подмножество ЭМ, соседних с элементом M_i; G_{M_i} – число ЭМ в $\bar{M}_i$. Шаг из M_i в $M_\alpha \in \bar{M}_i$ называется н а и б о л е е в е р о я т н ы м, если $G_{M_\alpha} = \min_{\gamma \in \bar{M}_i} G_\gamma$.

ОПРЕДЕЛЕНИЕ 6. ЭМ M_i считается элементом сгущения, если $G_{M_i} = 4$. Подмножество, состоящее из связной структуры элементов сгущения и соседей последних, называется подмножеством с г у щ е н и я .

ОПРЕДЕЛЕНИЕ 7. Подмножество M называется п о в ы ш е н н о й с в я з н о с т и, если среди всех дуг, связывающих эти элементы, не существует таких, из которых удаление одной или двух (инцидентных одной вершине) сделало бы подмножество несвязным.

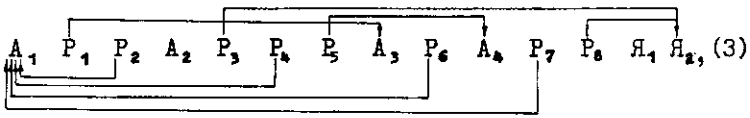
ОПРЕДЕЛЕНИЕ 8. Дуга (M_{i_1}, M_{i_2}) называется б л и ж а й ш е й п а р а л л е л ь н о й дуге (M_{i_3}, M_{i_4}), если машины $M_{i_1}, M_{i_2}, M_{i_3}, M_{i_4}$ являются вершинами квадрата, сторона которого равна единице.

Распознавание подсистем линейной структуры выполняется в два этапа (предполагается, что рассматриваемое подмножество позволяет реализовать связную структуру):

– на первом выделяются подмножества повышенной связности,

– на втором прокладываются гамильтоновы цепи в выделенных подмножествах.

Первый этап является подготовительным и реализуется с помощью алгоритма (I) путем проверки связности каждого подмножества, получающегося из заданного исключением дуги, не имеющей ближайшей параллельной и элемента, имеющего четырех соседей:



где A_1 – выбирает очередной элемент рассматриваемого множества;

P_1 – ПУ на A_3, если выбранный элемент имеет 4 соседних;

P_2 – ПУ на A_1, если для каждой дуги, связывающей выбранный элемент с соседними, существует ближайшая параллельная дуга;

A_2 – с помощью алгоритма (I) проверяет для каждой дуги, не имеющей ближайшей параллельной, связность подмножества, из которого она исключена (для элемента с двумя соседними достаточно проверить одну дугу, для элемента с тремя соседними проверяются все дуги);

P_3 – ПУ на $Я_2$, если выявились три дуги, исключение каждой из которых делает (под)множество несвязным;

P_4, P_6 – ПУ на A_1, если (под)множество оказалось связным;

A_3 – проверяет с помощью алгоритма (I) связность (под)множества, из которого исключен выбранный элемент;

P_5 – ПУ безусловно на A_4;

A_4 – разбивает (под)множество, содержащее выбранный элемент, на два подмножества: в первое – включаются ЭМ, находящиеся в стеках после работы алгоритма (I), во второе – остальные ЭМ разбиваемого (под)множества. Если выбранный элемент имеет четыре соседних, то он помечается и включается в оба подмножества, иначе – он помечается вместе с соседним элементом, определяемым дугой, выбрасывание которой делает (под)множество несвязным, и включаются они в разные подмножества;

P_7 – ПУ на A_1, если не все элементы исходного множества проверены;

P_8 – ПУ на $Я_2$, если существуют выделенные подмножества, у которых число помеченных элементов больше двух;

$Я_1$ – оператор успешного окончания работы алгоритма;

$Я_2$ – оператор окончания, сигнализирующий о невозможности организовать в исходном множестве линейную структуру.

В результате работы алгоритма (3) исходное множество разбивается на подмножества повышенной связности, в каждом из которых существуют помеченные элементы (см. рис. 2).

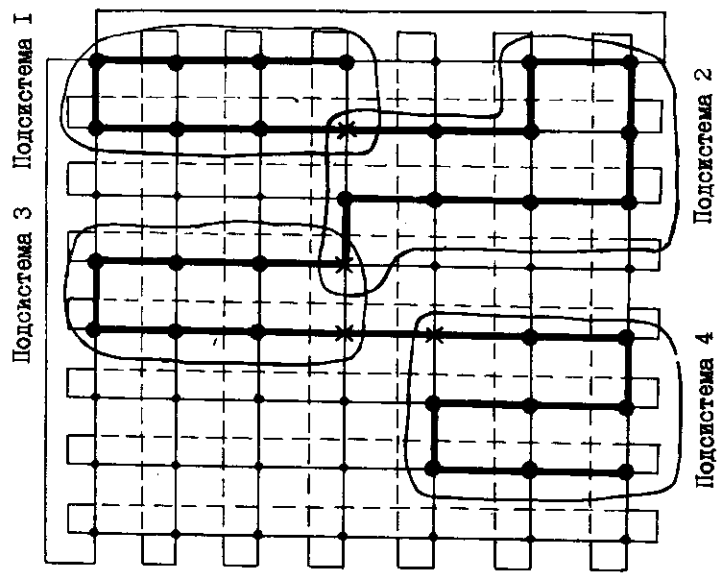


Рис. 2. Подмножества повышенной связности;
* – помеченные элементы

Второй этап по распознаванию линейных структур – прокладывание гамильтоновых цепей в этих подмножествах. Причем различаются случаи, когда помечены начальный и конечный элементы цепи (подсистемы 2 и 3, см. рис. 2) и когда задан только начальный элемент (подсистемы I,4). Если исходное множество само повышенной связности, то за начальный элемент берется ЭМ с ми-

нимальным числом соседних. Алгоритм прокладывания цепи представляет собой преимущественное движение наиболее вероятными шагами. Через несколько шагов проверяется связность оставшейся части рассматриваемого подмножества. Если связность имеет место, то проложенный путь считается приемлемым, иначе делается один или несколько шагов назад и прокладывание возобновляется по новому направлению. При одношаговом отступлении алгоритм является точным, при многошаговом – эвристическим. Последнее используется в подмножествах сгущения и позволяет значительно сократить время реализации при большом числе машин в подсистеме.

Операторная схема алгоритма имеет следующий вид:

(4)

где A_0 – $z := 0$;

A_1 – $\alpha := 1$, если среди путей, исходящих из выделенной точки, существуют непомеченные направления; иначе $\alpha := 0$ (помеченным считаются: направление в сторону конечного элемента цепи, когда не все остальные элементы пройдены, и направление, которое раньше привело к неудачной попытке, см. операторы A_5 и A_6);

P_0 – ПУ на P_6, если $\alpha = 0$;

A_2 – из выделенной точки делает наиболее вероятный шаг, исключая помеченные направления; $z := z + 1$; новая точка становится выделенной;

P_1 – ПУ на A_1, если $z \le k$ (k – целая константа, связанная с конкретной реализацией);

A_3 – $\beta := 1$, и последняя точка пути назначается выделенной, если элементы подмножества, не охваченные пройденными шагами, позволяют реализовать связную структуру; иначе $\beta := 0$;

P_2 – ПУ на A_0, если $\beta = 1$;

A_4 – $\gamma := 1$, если k последних элементов в пройденном пути принадлежат одному подмножеству сгущения; иначе $\gamma := 0$;

P_3 – ПУ на A_6, если $\gamma = 1$;

A_5 – делает шаг назад по неудавшемуся пути (неудавшееся направление помечается);

P_4, P_5 – безусловно ПУ на A_3;

A_6 – делает m шагов назад по неудавшемуся пути, m – константа, связанная с конкретной реализацией (последняя точка оставшегося пути назначается выделенной, неудавшееся направление помечается);

P_6 – ПУ на $Я_1$, если выделенной точкой является начальная точка цепи;

P_7 – ПУ на $Я_2$, если длина пройденного пути равна длине искомой цепи;

P_8 – безусловно ПУ на A_4;

$Я_1$ – оператор конца, сигнализирующий о невозможности реализовать линейную структуру;

$Я_2$ – оператор успешного окончания.

4. Структура "кольцо".

ОПРЕДЕЛЕНИЕ 9. Подмножество ЭМ $M = \bigcup_{k=1}^{L_0} M_{i_k}$ позволяет реализовать подсистему структуры "кольцо", если через его элементы можно провести гамильтонов цикл [9].

Таблица

Номер эксперимента	K	m	L	L_0	t[сек]
I	3	1	25	17	5
	10	1	25	17	3
	20	1	25	17	3
2	1	1	25	20	4
	3	1	25	20	2
	10	1	25	20	1
3	3	1	36	21	4
	5	1	36	21	4
	7	1	36	21	4
	12	1	36	21	3
4	3	1	64	34	100
	3	2	64	34	45
	3	3	64	34	40
	3	5	64	34	35
	3	7	64	34	38
5	3	1	64	24	32
	3	2	64	24	9
	3	3	64	24	10
	3	4	64	24	12

Алгоритм распознавания структуры "кольцо" может быть также представлен схемой (4), так как гамильтонов цикл есть гамильтонова цепь, у которой начальный и конечный элементы являются соседними.

Примечание 1. Алгоритмы (3) и (4) могут эффективно использоваться при решении и более общих задач; например, задано ℓ ЭМ, требуется распознать в них подсистему структуры "линейка" с числом машин $k \le \ell$.

Примечание 2. При невозможности реализовать на имеющихся машинах нужную структуру диспетчер прерывает работу

одной или нескольких соседних подсистем и производит новое распределение машин по подсистемам.

5.Численные эксперименты по проверке описанных алгоритмов были реализованы на машине "Минск-22". Моделировалась двумерная система с числом ЭМ: 25,36,64. Основное внимание уделялось алгоритму (4), типичные результаты реализации которого приведены в таблице (k, m – константы, используемые соответственно в операторах P_1 и A_6 алгоритма (4); L – общее число ЭМ в системе; L_0 – число машин в предполагаемой подсистеме, t –время распознавания линейной структуры).

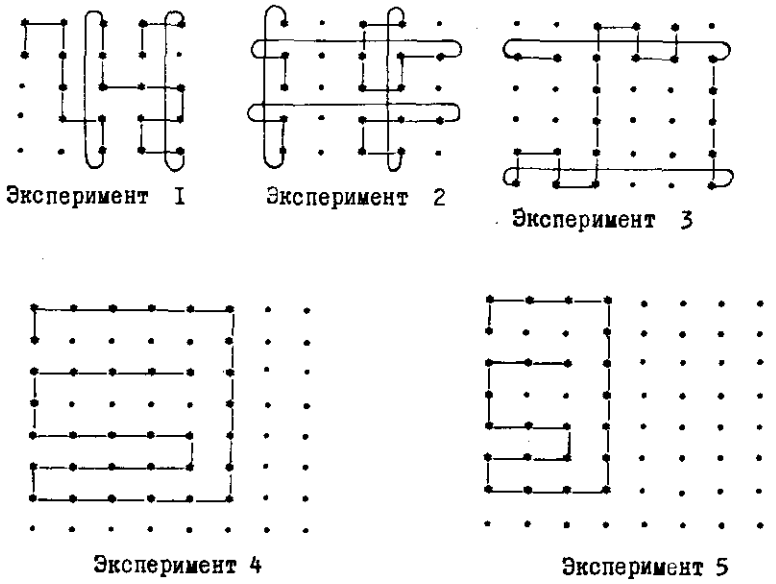


Рис. 3.

В экспериментах выявлялись гамильтоновы цепи, представленные схемами на рис. 3 (ЭМ подсистемы помечены значком "*").

Авторы благодарят Н.Л.Резник, оказавшую большую помощь в проведении численных экспериментов.

Рассмотренные алгоритмы являются эффективными, не требуют больших затрат машинного времени и могут применяться в

операционных системах для многомерных многомашинных однородных комплексов. Эти алгоритмы использованы в управляющей программе системы МИНИМАКС.

Литература

I. ЕВРЕИНОВ Э.В., КОСАРЕВ Ю.Г. Однородные универсальные вычислительные системы высокой производительности.Новосибирск. "Наука", I966.

2. МИРЕНКОВ Н.Н. Диспетчирование в однородных вычислительных системах. –В кн.: Материалы 3-й Всесоюзной конференции по однородным вычислительным системам и средам. Таганрог,I972, с. 5I–54.

3. МИРЕНКОВ Н.Н. МИНИМАКС – вычислительная система коллективного пользования. –В кн.: Вычислительные системы. Вып.60. Новосибирск, I974, с. II5–I28.

4. КОСАРЕВ Ю.Г. О схемах обмена между ветвями параллельных алгоритмов. –В кн.: Вычислительные системы. Вып. 5I. Новосибирск, I972, с. 70–75.

5. КЕРБЕЛЬ В.Г., КОРНЕЕВ В.Д., МИРЕНКОВ Н.Н.,ЩЕРБАКОВ Е.В. Управляющая программа системы МИНИМАКС. –В кн.: Вычислительные системы. Вып. 60. Новосибирск, с. I29–I42.

6. ВОРОБЬЕВ В.А. Простейшие структуры однородных вычислительных систем. –В кн.: Вычислительные системы. Вып. 60. Новосибирск, I974, с. 35–49.

7. УШАКОВ И.А. Об одном алгоритме проверки связности графа. –"Техническая кибернетика", I967, № 4, с.85–86.

8. ЗАКРЕВСКИЙ А.Д. Алгоритмы синтеза дискретных автоматов, М., "Наука", I97I.

9. ОРЕ О. Теория графов. М., "Наука", I968.

Поступила в ред.–изд.отд.
9 июня I973 года