

## ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ С ПРИМЕНЕНИЕМ НЕСТАНДАРТНОГО НАБОРА БЛОКОВ

Е. Ф. ЖИГАЛОВА, Т. Г. МАКСИМЕНКО

(Представлена научно-техническим семинаром кафедры автоматизированных систем управления и лаборатории управления)

В настоящее время при решении задачи компоновки конструктивных элементов (модулей) вычислительного устройства в блоки наибольшее распространение получил способ «разрезания» схемы на блоки по критерию максимума внутриблочных соединений. Совокупность модулей, вошедших в один блок, в дальнейшем будем называть структурами, а набор блоков, полученный таким способом, нестандартным (ННБ). Очевидно, что модули структур связаны между собой функционально, но в общем случае структуры могут состоять из несвязных подструктур. Разрезание на блоки по названному критерию имеет свою положительную сторону в том, что получаемые блоки имеют достаточно высокий коэффициент заполнения  $K_3$ .

$$K_3 = \frac{n}{N}, \quad (1)$$

где

$N$  — максимально возможное количество модулей в блоке,

$n$  — фактическое количество модулей в блоке.

В то же время этот способ имеет недостаток, заключающийся в том, что совершенно не учитывается топология структур и каждая схема представляется своим набором структур. В результате такого подхода проектируемое устройство в некотором смысле становится «уникальным». Если при этом учесть тот факт, что повторное производство узлов вычислительных устройств обходится в несколько раз дешевле изготовления нового узла, то будет понятно желание, если не все, то часть блоков из набора, полученного от предыдущих разрезов, использовать при конструировании новых устройств.

Статистический анализ принципиальных схем вычислительных устройств показывает, что в устройствах одного и того же класса содержится тождественные структуры, что является хорошей предпосылкой возможности использования структур из ННБ во вновь спроектированных устройствах.

Здравый смысл подсказывает, что вероятность использования всей структуры блока меньше вероятности использования ее части. Назовем коэффициентом использования блока величину  $K_n$ , равную

$$K_n = \frac{m}{n}, \quad (2)$$

$n$  — определена выше.

Применение блоков с низким коэффициентом  $K_n$  приводит к увеличению числа блоков в структуре, что может существенно снизить экономический эффект от применения данного набора блоков, а также привести к неоправданному увеличению габаритов устройства. На рис. 1 показана зависимость числа блоков  $P$  в устройстве от величины коэффициента  $K$ . Крутизна этой характеристики для различных классов вычислительных устройств различна, поэтому нижняя, рациональная граница коэффициента  $K_n^*$  должна определяться в каждом конкретном случае. Таким образом, применяемость нестандартных блоков лежит в диапазоне  $K_n^* \leq K_n \leq 1$ .

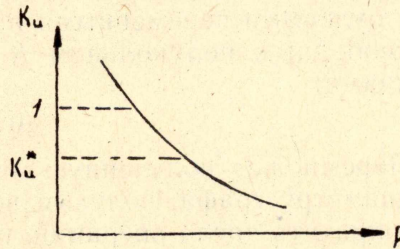


Рис. 1

Для дальнейшего изложения данного вопроса будем пользоваться аппаратом теории графов, который позволяет принципиальную схему ВУ, а также структуры блоков интерпретировать ориентированными графами. В терминах этой теории условия, которым должна отвечать каждая используемая подструктура, запишутся:

$$K_n^* \leq K_n \leq 1, \quad (1)$$

$$\forall u \in U \exists x_i, y_j \in X \{P(x_i, u, y_j)\}, \quad (2)$$

$$\forall L' \exists x_i, x_j \in X \{L' \cap L^{-1}(x_i, u) \cap L^+(x_j, v) \cap L^+(x_i, u) \cap L^-(x_j, v)\}, \quad (3)$$

где  $L = (x, u, p)$  — граф структуры блока;

$L'$  — часть графа  $L$  или граф подструктуры блока.

Условия (2), (3) соответственно означают, что используемая подструктура вместе со своими вершинами должна содержать и инцидентные им дуги; подструктура должна содержать вершины, соответствующие контактам входа и выхода разъема. Целесообразно данную задачу решать в два этапа. На первом, подготовительном, априорно формировать подструктуры, удовлетворяющие условиям (2), (3) и (1). На втором — решать задачу покрытия полученным набором структур (подструктур) схемы вычислительного устройства.

### Алгоритм разбиения структур блоков на подструктуры

Разбиение на подструктуры можно производить путем последовательного исключения элементов из структуры с одновременной проверкой на выполнимость условий (2) и (3). Количество операций существенно сокращается, если структуру, представленную ориентированным графом, разбивать на подмножества конечных последовательностей элементов  $x_0, x_1, x_2 \dots x_{k-1}, x_k$ , для которых истинно высказывание:  $P(x_0, u, x_1) P(x_1, u_2, x_2) \dots P(x_{k-1}, u_k, x_k)$ , где  $x_0$  и  $x_k$  — вершины, соответствующие контактам входа и выхода разъема блока. Тогда задача разбиения на подструктуры сводится к отысканию всех маршрутов в графе с фиксированным началом и концом. Так как нас интересуют не вообще все маршруты, а лишь начинающиеся и заканчивающиеся в вершинах, соответствующих контактам разъема, и идущие в направлении ори-

ентации графа, то это позволяет упростить алгоритм, представленный в [1], следующим образом. «Усовершенствованная» матрица смежности  $R_u$  строится на основе матрицы смежности  $R$  данного графа  $L = (x, u, p)$  с булевыми переменными и матрицы инцидентий  $H$  этого же графа над свободным полукольцом  $K$  с наложенными определяющими соотношениями:

$$\xi\theta \neq \theta\xi; \eta\xi = 0; \xi\theta = \xi\tau = 1.$$

Перемножая полученную матрицу  $R_u$  саму на себя  $l$  раз, где  $l_{\max}$  равно диаметру графа, получим все маршруты длины  $l$ . Здесь также можно сократить число операций, если считать нулевыми строки, соответствующие вершинам  $x_i \in x_1$ , для которых истинно  $\Gamma x^{-1} \neq \emptyset$ , иначе говоря, это вершины, не соответствующие входным контактам разъема.

Проиллюстрируем процесс разбиения на примере графа, не содержащего циклы, изображенного на рис. 2, а.

«Усовершенствованная» матрица смежности  $R_u$  данного графа имеет вид:

$$R_u = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \left\| \begin{array}{cccccccc} 0 & u_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_2 & 0 & u_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & u_4 & 0 & 0 \\ 0 & 0 & 0 & u_6 & 0 & 0 & 0 & u_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right\| \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \end{array}$$

Пусть вершины  $[x_1, x_7]$  соответствуют входным контактам разъема

блока, а вершины  $[x_6, x_8]$  — выходным. Тогда в соответствии с изложенным запишем матрицу  $R_u^*$ :

$$R_u^* = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \left\| \begin{array}{cccccccc} 0 & u_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_7 & 0 & 0 & 0 & 0 & 0 \end{array} \right\| \begin{array}{l} 1 \\ 7 \end{array} \end{array}$$

В результате перемножения матрицы  $R_u^*$  на  $R_u$  получаем матрицу  $R_{u1}^*$ . Матрица  $R_{u2}^* = R_{u1}^* \times R_u$  и т. д.  $R_{ul}^* = R_{u_{l-1}}^* \times R_u$ . Процесс перемножения закончится тогда, когда матрица  $R_{ul}^*$  станет нулевой. Так, в данном примере имеем

$$R_{u2}^* = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \left\| \begin{array}{cccccccc} 0 & 0 & 0 & u_1 & u_2 & u_3 & 0 & 0 & 0 & u_1 & u_5 & u_8 \\ & & & + u_1 & u_5 & u_6 & u_7 & u_3 & u_4 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & \end{array} \right\| \begin{array}{l} 1 \\ 7 \end{array} \end{array}$$

$$R_{u3}^* = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \left\| \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & u_1 & u_2 & u_3 & u_4 & 0 & 0 \\ & & & & & + u_1 & u_5 & u_6 & u_4 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & \end{array} \right\| \begin{array}{l} 1 \\ 7 \end{array} \end{array}$$

Очевидно, что матрица  $R_{u_4}^*$  будет нулевой. Наличие слагаемых в элементах матрицы  $R_{ui}^*$  говорит о наличии параллельных маршрутов. Итак, мы получили следующие маршруты, начинающиеся и заканчивающиеся соответственно в вершинах:

$$1 u_1 \quad 2 u_2 \quad 3 u_3 \quad 4 u_4 \quad 6, \quad (1)$$

$$1 u_1 \quad 2 u_5 \quad 5 u_8 \quad 8, \quad (2)$$

$$1 u_1 \quad 2 u_5 \quad 5 u_9 \quad 4 u_4 \quad 6, \quad (3)$$

$$7 u_7 \quad 3 u_3 \quad 4 u_4 \quad 6. \quad (4)$$

Легко убедиться, что эти маршруты удовлетворяют условиям (2) и (3). Анализ полученных маршрутов показывает, что маршруты (1) и (3) начинаются и заканчиваются в одних и тех же вершинах  $[x_1, x_7]$ . Это означает, что они не могут использоваться каждый в отдельности и являются составными частями одного «сложного» маршрута, проходящего по вершинам  $[1, 2, 3, 5, 4]$ . Отсюда следует, что подструктуры, соответствующие маршрутам, начинающимся и заканчивающимся в одних и тех же вершинах, должны объединиться в одну. В соответствии с этим структура, изображенная данным графом, разбивается на три подструктуры, включающие следующие элементы:

$$1, 2, 3, 4, 5, 6, \quad (1)$$

$$1, 2, 5, 8, \quad (2)$$

$$7, 3, 4, 6. \quad (3)$$

На практике известно, что схема вычислительного устройства, представленная в реальном базисе, содержит «специфические» соединения, связывающие некоторые совокупности элементов в неделимую группу. Примером таких связей могут служить модули, реализующие на входе конъюнкцию входных сигналов. Очевидно, что через вершину графа, соответствующую этому модулю, пройдет столько маршрутов, сколько переменных он объединяет. Однако подструктуры, соответствующие этим маршрутам, не могут работать автономно, так как это привело бы к существенному изменению алгоритма функционирования устройства. Можно сразу оговориться, что в некоторых случаях с целью расширения функциональных возможностей модулей можно разбивать также структуры по входам при условии, что электрическая схема соединений не будет нарушаться. Так, РТМ элементов ИС серии 155 допускает не использовать до двух входов, оставляя их свободными. В этом случае элемент типа ЛБ 551 может выполнять функцию элементов ЛБ 553 и ЛБ 554, а элемент ЛБ 554 — функцию элемента ЛБ 553. Поэтому при разбиении достаточно произвести перекодировку типов модулей. В остальных случаях для разбиения структур в сам алгоритм разбиения необходимо внести некоторые изменения. Так, зафиксировав вершины, соответствующие модулям, являющимся носителями специфических соединений, произвести разбиение в соответствии с описанным алгоритмом. Затем произвести «склеивание» маршрутов, содержащих фиксирование вершины. Рассмотрев алгоритм разбиения структур, графы которых не содержат циклы, перейдем к рассмотрению более сложного случая, когда граф содержит циклы. В качестве примера возьмем граф, изображенный на рис. 2, б.

Пусть вершины  $[x_1, x_7]$  соответствуют входным контактам разъема блока, а вершина  $[x_5]$  — выходному контакту. Вершины  $[x_2, x_5, x_4, x_6]$  входят в цикл  $Q$ . В соответствии с описанным алгоритмом разобьем дан-

ный граф на маршруты, приравнивая к нулю те слагаемые элементов матрицы  $R_{ul}^*$ , в которых какой-либо сомножитель встречается более одного раза, исключая тем самым повторное прохождение по одному и тому же маршруту. Тогда получим следующие маршруты, начинающиеся в вершинах  $[x_1, x_7]$ :

$$1 u_1 2 u_3 3 u_4 4 u_5 5, \quad (1)$$

$$1 u_1 2 u_3 3 u_4 4 u_5 6 u_2 2, \quad (2)$$

$$7 u_7 6 u_2 2 u_3 3 u_4 4 u_5 6, \quad (3)$$

$$7 u_7 6 u_2 2 u_3 3 u_4 4 u_6 6, \quad (4)$$

где маршруты (2) и (4) не удовлетворяют условию (3) и маршруты (1) и (3) не удовлетворяют условию (2). Полученное множество маршрутов будет удовлетворять условиям (2) и (3), если каждый маршрут дополнить соответственно дугами  $(u_6, u_2)$ ,  $(u_5)$ ,  $(u_6)$ ,  $(u_5)$ , в результате чего получаем, что маршруты (1) и (2), (3) и (4) попарно совпадают, т. е. данный граф разбивается лишь на два маршрута, удовлетворяющих условиям (2), (3):

$$\left\{ \begin{array}{l} 1 u_1 2 u_3 3 u_4 4 u_5 5 u_6 6, \\ 7 u_7 6 u_2 2 u_3 3 u_4 4 u_6 6 u_5 5, \end{array} \right. \quad (1a)$$

$$\left\{ \begin{array}{l} 1 u_1 2 u_3 3 u_4 4 u_5 5 u_6 6, \\ 7 u_7 6 u_2 2 u_3 3 u_4 4 u_6 6 u_5 5, \end{array} \right. \quad (2a)$$

каждый из которых включает цикл  $Q$ . Отсюда следует, что если маршрут проходит по дугам, входящим в цикл, то этот маршрут должен включить в себя и сам цикл. Можно предложить следующий способ разбиения структур, графы которых содержат цикл.

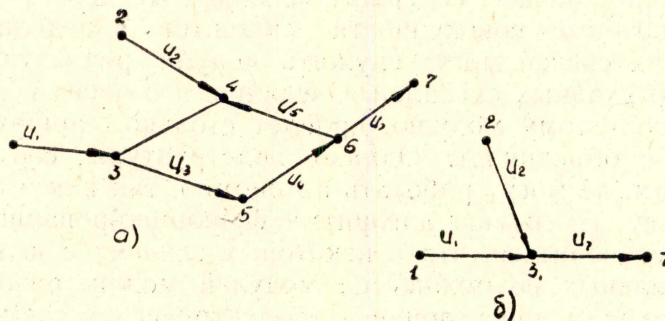


Рис. 3.

Вначале определить подмножество вершин и дуг, входящих в цикл  $Q$ , затем «стянуть» цикл  $Q$ , отождествив при этом все его вершины. В результате граф, содержащий цикл  $Q$ , превратится в граф без цикла. Проиллюстрируем эту процедуру на примере графа, изображенного на рис. 3, а, где вершины  $[x_3, x_4, x_5, x_6]$  и дуги  $[u_3, u_4, u_5, u_6]$  составляют цикл  $Q$ . На рис. 3, б, представлен тот же граф после стягивания цикла. В соответствии с алгоритмом граф, изображенный на рис. 3, б, разбивается на маршруты:

$$\left\{ \begin{array}{l} 2 u_2 3' u_7 7, \\ 1 u_1 3' u_7 7, \end{array} \right. \quad (1)$$

$$\left\{ \begin{array}{l} 2 u_2 3' u_7 7, \\ 1 u_1 3' u_7 7, \end{array} \right. \quad (2)$$

удовлетворяющие условиям (2), (3). Теперь достаточно каждому маршруту вместо вершины  $3'$  приписать сам цикл  $Q$ , тогда маршруты (1) и (2) будут иметь вид:

$$\begin{cases} 2 u_2 (3 u_3 5 u_4 6 u_5 4 u_6 3) u_7 7, & (1) \\ 1 u_1 (3 u_3 5 u_4 6 u_5 4 u_6 3) u_7 7. & (2) \end{cases}$$

Таким образом, задачи разбиения на подструктуры делятся на следующие основные этапы:

- 1) выделение контуров в графе;
- 2) разбиение на элементарные маршруты;
- 3) построение подструктур, удовлетворяющих условиям (2) и (3).

В [4] описан алгоритм выделения контуров в ориентированном графе. Для реализации этого алгоритма в ЭВМ класса М 220 принята форма представления исходной информации в виде списка функциональных связей между элементами в блоке. Далее эта исходная информация программным путем представляется двумерным массивом  $A [1:N, 1:4]$  вида  $A (M_i \Psi \mu_{ij} M_j)$ , где

$M_i, M_j$  — номера информационно связанных элементов,

$\mu_{ij}$  — функция связи,

$\Psi$  — фиктивный столбец для отметки контактов,

$N$  — число строк в массиве.

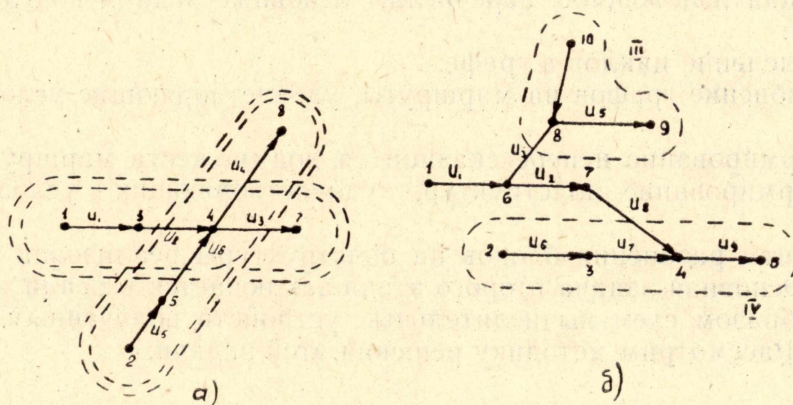


Рис. 4

В результате разбиения структур блоков получаем некоторое множество маршрутов с фиксированными началами и концами. При этом в зависимости от топологии графа это множество может состоять как из непересекающихся подмножеств, так и из пересекающихся. Пересекающимися будем называть такие подмножества маршрутов, которые включают в себя хотя бы одну общую дугу либо вершину. На рис. 4, а, б изображены графы различной топологии. В соответствии с данным алгоритмом разбиения получаем, что граф (4, а) разбивается на маршруты: I — (1  $u_1$  3  $u_2$  4  $u_4$  8); II — (1  $u_1$  3  $u_2$  4  $u_3$  7) III — (2  $u_5$  5  $u_6$  4  $u_4$  8); IV — (2  $u_5$  5  $u_6$  4  $u_3$  7), где вершины  $[x_1, x_2]$  соответствуют контактам входа, а вершины  $[x_7, x_8]$  — выхода. Все маршруты графа (4, а) проходят через вершину  $x_4$ , т. е. все маршруты пересекаются. Граф, изображенный на рис. 4, б, разбивается на маршруты: I — (1  $u_1$  6  $u_2$  7  $u_8$  4  $u_9$  5), II — (1  $u_1$  6  $u_9$  8  $u_4$  9), III — (2  $u_6$  3  $u_7$  4  $u_9$  5), IV — (10,  $u_4$  8  $u_5$  9). Здесь маршруты III и IV не содержат ни одной общей вершины и являются непересекающимися. Рассмотрим один способ проверки на пересечение маршрутов, легко реализующийся на ЭВМ. Если все множество полученных маршрутов разбить на такие подмножества маршрутов, которые начинаются (заканчиваются) в какой-либо одной вершине, то становится очевидным следующее. Если граф  $L = (x, u, p)$  раз-

бывается на пересекающиеся маршруты, начинающиеся соответственно в вершинах  $x_i$  и  $x_j$ , в которых, по крайней мере, два маршрута  $\mu_i \subset T_i$  и  $\mu_j \subset T_j$  заканчиваются в одной и той же вершине  $x_k$ , т. е.  $x_k \in \mu_i$  и  $x_k \in \mu_j$ . Например, граф, изображенный на рис. 4, а, содержит два подмножества  $T_1$ , объединяющее маршруты I и II, и  $T_2$ , объединяющее маршруты III и IV. Здесь имеем, что вершина  $x_7 \in T_1$  и  $x_7 \in T_2$ , то есть подмножества  $T_1$  и  $T_2$  пересекаются. Граф, изображенный на рис. 4, б, содержит подмножество  $T_1$  объединяющее маршруты I и II, и подмножества  $T_3$  и  $T_2$ , состоящие соответственно из маршрутов III и IV. Здесь подмножества  $T_2$  и  $T_3$  не содержат ни одной общей вершины, то есть не являются пересекающимися. Информация о пересечениях маршрутов в дальнейшем будет использоваться при решении задачи покрытия схем нестандартным набором блоков, которая будет рассмотрена ниже. Полученные в результате разбиения подструктуры являются минимальными подструктурами, удовлетворяющими условиям (2) и (3), для которых в общем случае условие (1) слева не выполняется. Для получения подструктур, удовлетворяющих всем трем условиям, достаточно произвести объединение минимальных подструктур таким образом, чтобы выполнялось условие  $K_{ii}^* \leq K_{ii}$ .

Обобщая изложенное, перечислим основные задачи подготовительного этапа:

1. Выделение циклов в графе.
2. Разбиение графов на маршруты, удовлетворяющие условиям (2) и (3).
3. Формирование непересекающихся подмножеств маршрутов.
4. Формирование подструктур, удовлетворяющих условиям (1), (2), (3).

Алгоритм разбиения блоков на подструктуры реализован на языке АЛЬФА. Основная задача второго этапа заключается в реализации наилучшим образом схем вычислительных устройств полученным набором структур. Рассмотрим методику решения этой задачи.

### Покрывание схем вычислительных устройств нестандартным набором блоков (ННБ)

Прежде чем дать формальное описание постановки данной задачи покрытия, рассмотрим некоторые ее физические особенности и сформулируем основные требования к покрытию. При покрытии схем заданным набором нестандартных блоков в каждой принятой к реализации структуре

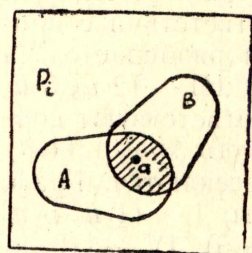


Рис. 5

должны быть использованы все модули, а также функциональные связи между ними, иначе говоря, коэффициент использования  $K_{ii}$  каждой структуры должен быть равен единице. Это требование вытекает из самого способа формирования структур, описанного выше. Очевиден также и тот факт, что каждый элемент схемы должен быть отнесен лишь к одной какой-либо структуре. Практически при покрытии схемы набором структур некоторые элементы, а также совокупности элементов, могут быть отнесены к нескольким структурам.

Это является следствием того, что, с одной стороны, в реальной схеме некоторые совокупности элементов, топологии функциональных связей которых тождественны каким-либо структурам из заданного набора, пересекаются, как это показано на рис. 5, где  $P$  — схема вычислитель-

ного устройства,  $A$  и  $B$  представляют собой области, соответствующие некоторым совокупностям функционально связанных элементов, принадлежащих схеме  $P_i$ , точка  $a$  отображает элемент, одновременно входящий в обе совокупности. С другой стороны, в схеме могут содержаться совокупности элементов, интерпретирующий граф которых тождествен графу, полученному отождествлением изоморфных частей двух, трех и т. д. графов, интерпретирующих структуры из заданного набора, как это показано на рис. 6. На рис. 6,  $a$  и 6,  $b$  изображены гра-

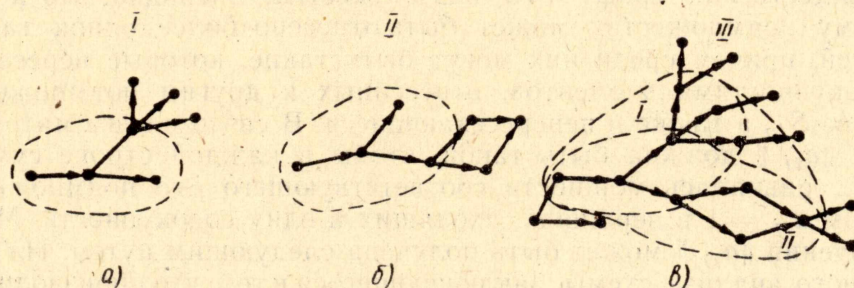


Рис. 6

фы двух структур I и II, содержащие изоморфные части, отмеченные пунктиром, а на рис. 6,  $b$  представлена часть графа, интерпретирующей принципиальную схему, где пунктиром отмечены подграфы, изоморфные графам I (рис. 6,  $a$ ) и II (рис. 6,  $b$ ). Очевидно, что данная схема не может быть реализована обеими структурами так, чтобы не нарушались сформулированные выше требования. Отсюда следует, что к реализации может быть принята лишь одна структура из приведенного набора, наилучшим образом реализующая схему. Распространяя сделанный вывод на множество структур из нестандартного набора блоков, приходим к тому, что задача покрытия принципиальной схемы заданным набором структур является комбинаторной, заключающаяся в оптимальном выборе набора структур.

Рассмотрим формальную постановку данной задачи покрытия. Пусть имеется конечный граф  $L = (x, u, p)$ , относящийся к классу графов Берга, и некоторое конечное семейство графов  $S_j = (y, v)$ . Известно также, что  $x \gg y_j$ ;  $j = 1, 2, \dots, N$ . Требуется найти такое покрытие графа  $L$ , т. е. набор подмножеств  $S_j$ , изоморфных соответствующим подграфам графа  $L$ , при котором любая вершина  $x_i \in X$  может принадлежать только одному из подмножеств  $S_j$ , а также этот набор должен быть таким, чтобы в него вошло как можно больше вершин графа  $L$ . Введем матрицу пересечений  $\|a_{ij}\|$ , в столбцах которой расположены вершины графа  $L$ , а в строках — номера графов из  $S_j$ , где  $a_{ij} = 1$ , если  $i$ -я вершина принадлежит  $j$ -му подмножеству. Тогда данная задача покрытия в терминах дискретного программирования является взвешенной задачей покрытия, максимизирующей линейную форму:

$$z(x) = \sum_{j=1}^m b_j x_j \quad (1)$$

при условиях

$$\sum_{j=1}^m a_{ij} x_j \leq 1, \quad i = \overline{1, n}, \quad (2)$$

$$x_j = \begin{cases} 1, & \text{если } i\text{-я структура входит в покрытие,} \\ 0, & \text{в противном случае,} \end{cases}$$



где

$b_j$  — мощность  $j$ -й структуры,

$$a_{ij} = \begin{cases} 1, & \text{если } i\text{-я вершина отнесена к } j\text{-му подмножеству,} \\ 0, & \text{в противном случае.} \end{cases}$$

Специфика данной задачи такова, что к  $j$ -му подмножеству могут быть отнесены лишь те вершины, топология функциональных связей которых тождественна графу  $j$ -го подмножества. Очевидно, что к некоторому  $j$ -му подмножеству может быть отнесено более одной таких совокупностей, причем среди них могут быть такие, которые пересекаются с совокупностями элементов, отнесенных к другим подмножествам семейства  $S_j$ , а также и непересекающиеся. В связи с этим матрица пересечений  $\|a_{ij}\|$  должна быть такой, чтобы в каждой строке сумма значений  $a_{ij}$  равнялась мощности соответствующего  $j$ -го подмножества и при этом  $a_{ij} = 1$  в вершинах, входящих в одну совокупность. Матрица ограничений  $\|a_{ij}\|$  может быть получена следующим путем. На базе структурного анализа схемы, заключающегося в том, что производится независимое покрытие схемы каждой структурой из семейства  $S_j$  с фиксацией покрытых элементов, строится матрица  $\|r_{ij}\|$ , столбцы которой соответствуют номерам элементов схемы, а строки — номерам структур  $S_j$ , где

$$r_{ij} = \begin{cases} 1, & \text{если } i\text{-я вершина покрылась } j\text{-й структурой;} \\ 0, & \text{в противном случае.} \end{cases}$$

Сумма элементов  $i$ -й строки может быть подсчитана по выражению:  $\sum_j x_{ij} b_j$ , где  $b_j$  — мощность  $j$ -й структуры,  $x_{ij}$  — количество  $i$ -х совокупностей элементов схемы, покрытых  $j$ -й структурой. Причем после покрытия схемы  $j$ -й структурой первая восстанавливается полностью. Алгоритм структурного анализа разработан в [3].

Полученная матрица  $\|r_{ij}\|$  может быть легко преобразована в матрицу  $\|a_{ij}\|$  разбиением каждой строки на число строк, равное количеству совокупностей, покрытых соответствующей  $j$ -й структурой. При этом должны выполняться следующие два условия:

1. Каждая вновь полученная строка должна содержать только одну совокупность  $x_{ij}$ . 2. Поэлементная сумма этих строк должна быть равна исходной строке. Размерность матрицы  $\|a_{ij}\|$  может быть сокращена за счет вычеркивания нулевых столбцов. Используя результаты [5], данную задачу можно решать, модифицируя незначительно симплекс-метод.

#### ЛИТЕРАТУРА

1. А. А. Корбут, Ю. Ю. Финкельштейн. Дискретное программирование. М., «Наука», 1969.
2. А. А. Зыков. Теория конечных графов. Новосибирск, «Наука», 1969.
3. В. К. Погребной. Разработка и исследование алгоритмов автоматизации технического этапа проектирования вычислительных устройств. Диссертация на соискание ученой степени кандидата технических наук. Томск, 1970.
4. Л. Я. Лейфман, Л. Г. Петрова. Некоторые алгоритмы для анализа ориентированных графов. Сб.: «Выч. системы». СО АН СССР, вып. II, 1964.
5. В. А. Трубин. О решении задачи выбора. Сб. «Математические методы в экономике», вып. 3., Киев, 1968.