В Ы Ч И С Л И Т Е Л Ь Н Ы Е   С И С Т Е М Ы

Сборник трудов

1963 г. Института математики СО АН СССР Выпуск 6

О МАШИННОМ МЕТОДЕ ПРОЕКТИРОВАНИЯ СОЕДИНЕНИЙ

В.Л. Харченко

При создании вычислительных систем высокой производительности [I] возникает задача оптимального размещения большого числа физически однородных элементов и тесно связанная с ней задача построения соединений между ними.

В настоящей статье предлагается вариант решения задачи о построении соединений между элементами с помощью электронной вычислительной машины (ЭВМ).

Будем считать, что на плоскости C задана конечная совокупность элементов M, разделенная на k наборов элементов M_i таких, что: $M_i \cap M_j \neq 0$ при $i \neq j$ и $\bigcup_{i=1} M_i = M$.

Плоскость C с заданной на ней совокупностью элементов M разбита на одинаковые площадки c_i таким образом, что в пределах каждой из них содержится не более одного элемента $m_i \in M$.

Две площадки c_i и c_j называются соседними, если их границы имеют общую часть, содержащую более одной точки.

Для каждой площадки $c_i \in C$ определено подмножество всех соседних площадок $N(c_i)$, состоящее из n площадок плоскости C. Число $n \geq 1$ зависит от способа разбиения плоскости C на элементарные площадки. Например, $n = 4$, если плоскость C разбита на одинаковые квадратные площадки, как показано на рис. I.

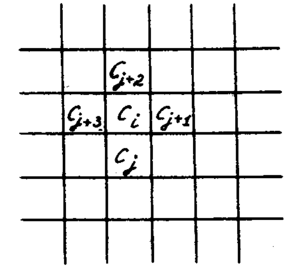

Рис. I

В любом случае, если $c_j \in N(c_i)$, то $c_i \in N(c_j)$.

Любые два элемента m_i и m_j можно соединить друг с другом, если на плоскости C найдется цепь площадок

$$p(c_0, c_m) = \{c_0, c_1, c_2, \ldots, c_n\},$$

где c_0 – площадка, заключающая в себе m_i;

c_m – площадка, заключающая m_j, $c_{i+1} \in N(c_i)$, $i = 0,1,\ldots,m-1$;

$\pi(c_0, c_m)$ – множество всех возможных цепей, имеющих в качестве исходного и конечного звеньев, соответственно, площадки c_0 и c_m.

Каждому соединению, состоящему из цепи площадок c_i, ставится в соответствие целое неотрицательное число, которое задается конкретным признаком соединения (например, длина, количество пересечений и т.д.). Таким образом, на $\pi(c_i, c_j)$ определена некоторая неотрицательная целочисленная функция f, которая называется функцией веса соединения и которая всегда удовлетворяет следующим условиям:

F1. $f[p(c_i, c_i)] = 0$.

F2. Если $p(c_0, c_m) = \{c_0, c_1, c_2, \ldots, c_i, \ldots, c_{m-1}, c_m\}$ и

$$p(c_m, c_0) = \{c_m, c_{m-1}, \ldots, c_i, \ldots, c_1, c_0\}, \quad c_i \in N(c_{i+1}), \quad (i = 0,1,2,\ldots,m-1),$$

то $f[p(c_0, c_m)] = f[p(c_m, c_0)]$.

F3. Если $p(c_0, c_m) = \{c_0, c_1, c_2, \ldots, c_{i-1}, c_i, \ldots, c_m\}$ и

$$p(c_0, c_i) = \{c_0, c_1, c_2, \ldots, c_{i-1}, c_i\},$$

(где $p(c_0, c_i)$ – частичная цепь от $p(c_0, c_m)$ $(i = 0,1,2,\ldots,m-1)$), то $f[\rho(c_0, c_i)] \leq f[\rho(c_0, c_m)]$.

Соединение $\rho^1(c_i, c_j)$ из $\pi(c_i, c_j)$ имеет минимальный вес от f_1, если $f_1[\rho'(c_i,c_j)] \leq f_1[\rho(c_i, c_j)]$ для всех

$$\rho(c_i, c_j) \in \pi(c_i, c_j).$$

Соединение $\rho^{1,2}(c_i, c_j)$ имеет минимальный вес от f_1, f_2, если $\rho^{1,2}(c_i, c_j) \in \pi'(c_i, c_j)$, где $\pi'(c_i, c_j)$ – множество всех соединений, имеющих минимальный вес от f_1, т.е. $\pi'(c_i,c_j) = \{\rho'(c_i,c_j) \mid f_1[\rho'(c_i,c_j)] \leq f_1[\rho(c_i, c_j)]$ для всех

$$\rho(c_i,c_j) \in \pi(c_i,c_j)\} \text{ и } f_2[\rho^{1,2}(c_i,c_j)] \leq f_2[\rho(c_i,c_j)].$$

Аналогично можно определить соединение $\rho^{1,2,3,\dots,z}(c_i, c_j)$, которое и имеет минимальный вес от $F = \{f_1, f_2, f_3, \dots, f_z\}$.

Для каждого набора элементов нужно построить минимальное число соединений, связывающих все элементы в некоторый конечный связный граф без циклов [3], вершинами которого являются все элементы данного набора M_i, а ребрами – соединения между элементами. Суммарный вес F построенных соединений для каждого набора M_i должен быть минимальным.

Пусть, например, необходимо построить кратчайшее соединение между площадками c^* и c^{**} (см. рис.2. Площадки, заключающие в себе символ „×", являются запрещенными для использования их в качестве звеньев цепи, соединяющей c^* и c^{**}). В этом случае вес соединения определяется значением одной функции f_1.

			×	×		c^{**}	
	×			×			
	×			×			
		×					
		×			×	×	
c^*		×					

Рис. 2

Из множества цепей $\pi(c^*, c^{**})$, соединяющих площадки c^* и c^{**}, выбирается цепь $\rho^1(c^*, c^{**})$, содержащая минимальное число звеньев, где первым звеном является c^*, а последним c^{**}.

Если площадка c_i свободна (т.е. не содержит никакого элемента m_i из совокупности M и не является звеном ранее выбранной цепи), то

$$f_1[\rho(c^*, c_i)] = \left\{ min \left\{ f_1[\rho(c^*, c_j)] \,\middle|\, c_j \in N(c_i) \right\} \right\} + 1$$

для всех c_j, для которых значение $f_1[\rho(c^*, c_j)]$ было определено.

Свободным площадкам, соседним с c^*, ставится в соответствие значение функции веса $f_1[\rho(c^*, c_i)] = 1$. На следующем шаге свободным площадкам, соседним с теми, для которых было определено значение $f = 1$, ставится в соответствие значение веса $f_1[\rho(c^*, c_i^1)] = 2$ до тех пор, пока не будет достигнута площадка c_i, содержащая элемент c^{**}.

Затем выбирается такая цепь $\rho(c^*, c^{**})$ (рис. 3), для которой значение $f_1[\rho(c^*, c_i)]$ было бы минимальным из всех имеющихся, а значения $f_1[\rho(c^*, c_i)]$ для последовательности частичных цепей выбранной цепи составляли бы ряд членов монотонно убывающей арифметической прогрессии.

5	6	7	×	×	12	c^{**}	14
4	×	8	9	×	11	12	13
3	×	9	8	×	10	11	12
2	3	×	7	8	9	10	11
1	2	×	6	7	×	×	10
c^*	1	×	5	6	7	8	9
1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9

Рис. 3

В работе [6] показано, что в процессе решения этой задачи на ЭВМ в целях экономии памяти для величины $f_1[\rho(c^*; c_i)]$ не обязательно использовать весь натуральный ряд чисел 0,I,2,3,4,5,...,n и т.д., а достаточно ограничиться последовательностью чисел I,2,3,I,2,3,I,... и т.д.

В этом случае выбирается цепь $\rho(c^*, c^{**})$, такая, чтобы значения $f_1[\rho(c^*, c_i)]$ для последовательности всех соответствующих частичных цепей составляли убывающую последовательность чисел по модулю 3, начиная с $f_1[\rho(c^*, c^{**})]$ и кончая $f_1[\rho(c^*, c^*)]$.

В связи с вышесказанным условие F3 примет вид:

F3' Если $\rho(c_0, c_i)=\{c_0, c_1, c_2, \dots, c_{i-2}, c_{i-1}, c_i\}$ и

$\rho(c_0, c_{i-1})=\{c_0, c_1, c_2, \dots, c_{i-2}, c_{i-1}\}$,

то

$$\underset{\text{по } mod\{3\}}{f_1[\rho(c_0, c_{i-1})]} < \underset{\text{по } mod\{3\}}{f_1[\rho(c_0, c_i)]}$$

Если нужно построить соединение между c^* и c^{**}, имеющее минимальное число пересечений с находящимися на плоскости соединениями, то функция веса f_2 задается следующими условиями:

I. Если площадка c_i свободна, то $f_2[\rho(c^*, c_i)] =$

$$= min\left\{f_2[\rho(c^*, c_j) \mid c_j \in N(c_i)]\right\} \quad \text{для всех } c_j,$$

для которых значение $f_2[\rho(c^*, c_j)]$ было определено.

2. Если площадка c_i является звеном цепи ранее построенного соединения, то $f_2[\rho(c^*, c_i)] =$

$$= \left\{min\left\{f_2[\rho(c^*, d_1)], f_2[\rho(c^*, d_2)]\right\}\right\}+1,$$

если хотя бы одно из значений $f_2[\rho(c^*, d_1)]$ или $f_2[\rho(c^*, d_2)]$ было определено. Здесь d_1 и d_2 – площадки, соседние с c_i. Они находятся по обе стороны цепи, звеном которой является c_i.

Пример: на рис.4 показано ранее построенное соединение $\overline{AB}$. Нужно построить соединение между c^* и c^{**}, имеющее минимальное число пересечений с $\overline{AB}$.

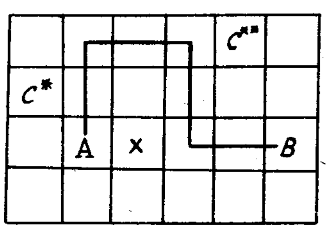

Рис. 4

В соответствии с условиями I и 2 определяются все значения $f_2[\rho(c^*, c_i)]$ (см. рис.5). Затем выбирается цепь площадок, соединяющая c^{**} и c^* так, чтобы значения $f_2[\rho(c^*, c_i)]$, соответствующие всем частичным цепям этой цепи, составляли монотонную невозрастающую последовательность чисел.

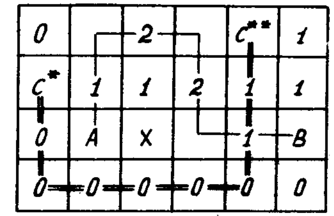

Рис. 5

На рис. 5 показаны все значения $f_2[\rho(c^*, c_i)]$ и искомое соединение между c^* и c^{**}.

Если набор M_i содержит q элементов, где $q > 2$, то решается задача построения минимального дерева из теории графов [3], но с некоторой модификацией.

Например, нужно соединить между собой 4 элемента (рис.6). Сначала строится соединение с минимальным весом между двумя элементами (рис. 6а). Затем выбирается третий элемент m_2, и строится соединение с минимальным весом от m_2 до $\overline{m_3\ m_4}$ (рис. 6б). И, наконец, строится соединение с минимальным весом от m_1 до $\overline{m_3\ m_4\ m_2}$ (рис. 6в).

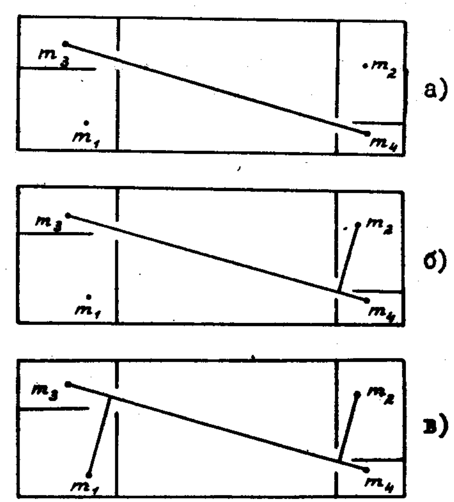

Рис. 6

Может оказаться, что веса соединений для некоторых пар элементов, принадлежащих к одному набору M_i, равны друг другу. В таком случае может существовать не один вариант минимального дерева. Эта проблема не рассматривается в данной статье, поэтому для простоты будем считать, что веса соединений между любыми парами элементов в одном наборе M_i различны и, следовательно, существует единственный вариант минимального дерева.

Геометрическая конфигурация отдельного соединения во многих случаях определяется очередностью его построения.

Например, нужно выполнить построение минимальных соединений для набора элементов M_i. Если оно выполняется в первую очередь, то на плоскости C еще нет ни одного построенного соединения для других наборов $M_j \subset M$, поэтому ничто не препятствует построению абсолютно минимальных соединений для набора M_i.

Если же на плоскости C уже появилось хотя бы одно соединение для любого другого набора M_j, то построение минимальных соединений для набора M_i нужно выполнять с учетом уже имеющихся на плоскости соединений. При этом существует большая вероятность, что построенные соединения не будут совпадать с абсолютно минимальными для этого набора M_i.

Следовательно, максимальное число возможных вариантов всей совокупности соединений может достигать $\kappa!$, где κ – число различных наборов M_i в совокупности M. Но так как не все $\kappa!$ вариантов являются приемлемыми решениями, поэтому возникает проблема их перебора.

Возможности существующих ЭВМ позволяют выполнять перебор всех вариантов для этой задачи, если их число не превышает 10! Практически перебор вариантов можно существенно сократить.

Для этого выбираются любые два набора элементов M_i и M_j. На плоскости независимо друг от друга строятся кратчайшие соединения для всех возможных пар элементов набора M_i. Обозначим множество этих соединений через $\pi^*(M_i)$. Аналогичные построения выполняются для набора M_j, в результате образуется $\pi^*(M_j)$.

Если найдется хотя бы одна цепь $\rho_i(M_i) \in \pi^*(M_i)$, пересекающаяся, по крайней мере, с одной цепью $\rho_j(M_j) \in \pi^*(M_j)$, то, значит, наборы M_i и M_j, соответствующие этим соединениям элементов, зависимые. Такая процедура выполняется для всех возможных пар M_i и M_j.

В результате можно разделить все наборы элементов на независимые группы. Внутри каждой группы будут находиться зависимые наборы элементов. Для каждой независимой группы автономно решается задача построения соединений и выполняется полный перебор различных вариантов соединений. По условиям задачи число переборов для решения всей задачи в таком случае равно $\kappa_1!+\kappa_2!+\ldots+\kappa_i!\ldots+\kappa_p!$, где p – число независимых групп и $\sum_{i=1}^{p}\kappa_i=\kappa$.

Например, если нужно выполнить построение соединений для 20 наборов элементов и удается разделить все наборы элементов хотя бы на две независимые группы по 10 наборов в каждой, то такая задача уже может быть решена с помощью ЭВМ.

В результате проведенных исследований установлено, что в каждом случае наборы элементов могут быть разделены на независимые группы и число наборов в одной группе обычно не превышает 6, а в большинстве случаев бывает и меньше 6.

В соответствии с предложенным алгоритмом была составлена программа для проектирования с помощью ЭВМ монтажных схем печатной технологии для случая, когда печатный монтаж ведется на одной стороне платы и ее размеры не превышают 300x300 мм^2 [5]. Время решения задачи на машине не более 30 минут.

Мы рассмотрели построение соединений на одной плоскости. Но иногда использование только одной плоскости может оказаться недостаточным. В некоторых случаях для построения всех соединений может потребоваться столько плоскостей, сколько имеется непересекающихся наборов M_i в совокупности M. Поэтому возникает необходимость в решении задачи об определении минимального числа плоскостей для выполнения всех соединений.

Не менее важна задача о выборе оптимального взаимного расположения элементов. Комплексно решая вышеуказанные задачи на ЭВМ, можно добиться полной автоматизации процесса оптимального размещения элементов и построения соединений между ними.

Литература

1. Э.В. Евреинов, Ю.Г.Косарев. О возможности построения вычислительных систем высокой производительности. Новосибирск, 1962. Из-во СО АН СССР
2. C.J. Lee. An algorithm for Path Connections and Its Applications. IRE Transactions on Electronic Computers, v. EC - 10, N 3, 1961.
3. К. Берж. Теория графов и ее применения. ИЛ, Москва, 1962.
4. Gill S. The Use of Computers in Designing Computers. Research Applied in Industry, v. 15, N4, April 1962.
5. В.А.Тюренков. Программа расположения печатных проводников на плате. Отчёт. Институт математики СО АН СССР, 1963.
6. В.А.Тюренков. Алгоритм нахождения кратчайшего пути (Данный сборник).