ИЗВЕСТИЯ
ТОМСКОГО ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА имени С. М. КИРОВА

Том 211 1970

# ОБ ОДНОМ АЛГОРИТМЕ ПРЕОБРАЗОВАНИЯ КОМПОНЕНТ СВЯЗНОСТИ

В. К. ПОГРЕБНОЙ

(Представлена научно-техническим семинаром НИИ АЭМ при ТПИ)

Необходимость в разработке алгоритма преобразования компонент связности возникает на стадии трассировки монтажных соединений при решении задач автоматизации конструкторского этапа проектирования вычислительных устройств.

В схемах вычислительных устройств часто встречаются ситуации, когда некоторое множество контактов (элементов компоненты) должно быть непосредственно связано. В случае, когда число контактов в таком множестве превышает два, возникает задача минимизации суммарной длины проводов, связывающих контакты данного множества.

В терминах теории графов рассматриваемая задача заключается в построении минимального связывающего дерева. Точные методы решения данной задачи разработаны для случаев, когда число ребер $\alpha$ инцидентных одной вершине неограничивается [1], [2]. Использование данных методов с наложением ограничений на число $\alpha$ вызывает значительное отклонение суммарной длины ребер полученного дерева от соответствующей суммы в минимальном дереве. Если при $\alpha=3$ такое отклоненение в практических примерах монтажа является в какой-то мере допустимым, то при $\alpha = 2$ оно существенно возрастает, что свидетельствует о необходимости разработки специального метода решения данной задачи.

Задача построения минимального дерева с $\alpha = 2$ возникает не только в связи с техническими ограничениями на число проводов, присоединяемых к одному контакту. Чаще она возникает при наличии в схемах последовательных цепей либо в случае, когда такие цепи формируются искусственно [3]. Как правило, в таких цепях заведомо известны контакты начала и конца. Если к тому же расстояние между такими контактами невелико (например, соседние контакты разъема), то возникает задача отыскания фактически замкнутой цепи с минимальной суммой длин ребер.

Алгоритм [4], учитывающий ограничение на число $\alpha$, является достаточно эффективным при построении цепей, в которых заведомо не фиксируется начало и конец. Однако при наличии данного ограничения его эффективность заметно снижается. Кроме того, эффективность данного алгоритма существенно зависит от конкретного расположения вершин в графе.

Рассматриваемую задачу можно также сформулировать как задачу дискретного программирования. Однако практика свидетельствует о значительных трудностях, возникающих при решении данных задач в такой постановке.

Предлагаемый метод решения данной задачи позволяет получить близкую к оптимальной как замкнутую, так и разомкнутую цепь с заданным началом и концом.

Будем рассматривать граф с $n$ вершинами и матрицей $L = \|l_{ij}\|$ расстояний между ними. Задача состоит в обходе всех $n$ вершин замкнутой цепью минимальной длины (разомкнутую цепь считаем частным случаем замкнутой).

Исходной цепью при построении замкнутой цепи будем называть цепь вида $m — m$, $m \in n$. В случае построения разомкнутой цепи с заданными вершинами начала $m$ и конца $\kappa$, исходная цепь имеет вид $m — \kappa$. На рис. 1 схематично показана исходная цепь $m — m$. Точками условно показаны места, которые займут в конечной цепи оставшиеся $n - 1$ вершин.

Попытаемся расширить данную цепь, т. е. включить в нее одну из оставшихся вершин. Встает вопрос, какую из вершин включить в данную цепь в первую очередь. При включении некоторой вершины $\lambda$ получается цепь $m — \lambda — m$. Вершина $\lambda$ в такой цепи имеет возможность «занять» одно из $n - 1$ мест, т. е. ей предоставляется максимальная возможность в процессе включения следующих вершин передвигаться по всем местам цепи $m — m$.

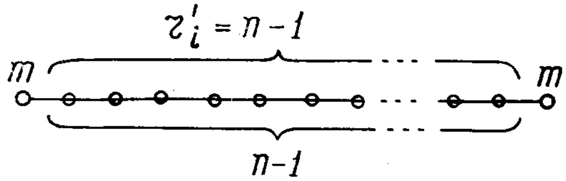


Рис. 1.

Возможность передвижения $i$-ой вершины на $f$-ом шаге включения будем выражать понятием степени свободы $r_i^f$. Очевидно, что на первом шаге $r_i^1 = n - 1$. На последующих шагах степень свободы уменьшается и определяется выражением.

$$r_i^f = n - f.$$

Сравним между собой две вершины $i$ и $j$, причем $l_{im} > l_{jm}$. По отношению к вершине $j$ есть большее основание предполагать, что в конечной цепи она займет место, более близкое к вершине $m$, чем вершина $i$. Вершина $j$ «притянута» к вершине $m$ и поэтому обладает меньшей «подвижностью», т. е. меньшей степенью свободы. Обладая большей степенью свободы, вершина $i$ в меньшей мере будет «мешать» включению в цепь остальных вершин, поэтому при выборе вершины для включения в цепь она предпочтительнее, чем вершина $j$.

В общем случае, на каждом $f$-ом шаге, включаемая вершина $\lambda$, определяется из условия

$$\max \sum_{i \in Q_f} l_{\lambda i}\,, \qquad (2)$$

где $Q_f$ — множество вершин, составляющих исходную цепь для $f$-го шага.

В рассматриваемом случае, после включения вершины $\lambda$, исходной цепью для включения следующей, например, $\kappa$-ой вершины является цепь $m — \lambda — m$. В этом случае $Q_f = \{m, \lambda\}$. При включении $\kappa$-ой вершины безразлично, в какое звено цепи она будет включена. Однако на последующих шагах возникает вопрос, в какое звено цепи необходимо включать очередную вершину $\lambda$. Такое звено $l_{ij}$ на каждом шаге определяется из условия

$$min \quad [l_{\lambda i} + l_{\lambda j} - l_{ij}]. \qquad (3)$$

Ниже приводится логическая схема алгоритма преобразования компонент связности с числом $\alpha = 2$. При этом используются обозначения, принятые в [5].

$$U = A_0 A_1 P_1 \overset{1}{\uparrow} A_2 A_3 \overset{4}{\downarrow} A_4 \overset{3}{\downarrow} A_5 A_6 P_2 \overset{2}{\uparrow} A_7 w \overset{3}{\uparrow} \overset{1}{\downarrow} A_8 w \overset{4}{\uparrow} \overset{2}{\downarrow} P_3 \overset{5}{\uparrow} A_{10} \overset{6}{\downarrow}$$
$$\overset{6}{\downarrow} \overset{7}{\downarrow} A_{11} A_{12} A_{13} P_4 \overset{5}{\uparrow} A_{14} A_{15} w \overset{8}{\uparrow} \overset{5}{\downarrow} A_9 w \overset{7}{\uparrow} \overset{8}{\downarrow} A_{\kappa},$$

где $A_0$ — ввести в ЭЦВМ исходные данные о координатах вершин компоненты

$$\{x_1 y_1,\ x_2\ y_2,\ \ldots, x_n y_n\},\ \{m,\ \kappa\}.$$

$A_1$ — на множестве вершин $n$ построить полный граф и представить его в матричной форме. Элемент матрицы $l_{ij}$ определяется по формуле

$$l_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\ .$$

$P_1$ — проверить, определяется замкнутая цепь или цепь между вершинами $m$, $\kappa$. Если $P_1 = 1$, то перейти к оператору $A_2$, при $P_1 = 0$ перейти к $A$.

$A_2$ — выбрать из матрицы максимальнй элемент $l_{ij}$.

$A_3$ — номера вершин $i$ и $j$ занести в $РП_1$.

$A_4$ — просуммировать построчно элементы матрицы, соответствующие номерам $РП_1$ при условии, что сумма элементов равна нулю, если одно из слагаемых равно нулю. Суммы занести в $РП_2$.

$A_5$ — из полученных в $РП_2$ сумм выбрать максимальную.

$A_6$ — номер вершины, соответствующий данной сумме занести в $РП_1$.

$P_2$ — проверить условие $Q_f < n$.
Если $P_2 = 1$, то перейти к $A_7$, при $P_2 = 0$ перейти к $P_3$.

$A_7$ — просуммировать построчно элементы столбца матрицы с номером, занесенным в $РП_1$ по $A_6$ с соответствующими суммами в $РП_2$. Результаты занести в $РП_2$, предварительно исключив из него предыдущие суммы. Передать управление на $A_5$.

$A_8$ — номера вершин $m$ и $\kappa$ занести в $РП_1$ и передать управление на $A_4$.

$A_9$ — записать в $РП_3$ исходную цепь $m—\kappa$ и передать управление на $A$.

$P_3$ — проверить, определяется замкнутая цепь или цепь между вершинами $m$ и $\kappa$. Если $P_3 = 1$, то перейти к $A_{10}$, при $P_3 = 0$ перейти к $A_9$.

$A_{10}$ — запись в $РП_3$ исходную цепь $m—m$.

$A_{11}$ — для каждого звена исходной цепи определяется величина $l = l_{\lambda i} + l_{\lambda j} - l_{ij}$, где $\lambda$ очередной номер из $РП_1$, подлежащий включению в исходную цепь.

$A_{12}$ — из всех значений $l$ выбрать $l_{\min}$.

$A_{13}$ — номер $\lambda$, соответствующий $l_{\min}$ и $l_{ij}$ включить в исходную цепь между вершинами $i$ и $j$.

$P_4$ — проверить, все ли номера вершин содержаться в $РП_3$. Если $P_4 = 1$, то перейти к $A_{14}$, при $P_4 = 0$ перейти к $A_{11}$.

$A_{14}$ — определить сумму длин звеньев цепи в $РП_3$.

$A_{15}$ — печать последовательности вершин цепи $РП_3$ и суммы длин ее звеньев.

$A_к$ — конец алгоритма.

По данному алгоритму составлена программа на машину М-20. Анализ результатов решения ряда практических задач показал преимущество предложенного алгоритма по сравнению с известными.

## ЛИТЕРАТУРА


1. К. Б е р ж. Теория графов и ее применение. Изд-во иностранной литературы, 1962.
2. Р. К. П р и м. Кратчайшие связывающие сети и некоторые обобщения. «Кибернетический сборник», № 2. Изд-во иностранной литературы, 1961.
3. Автоматизация некоторых этапов проектирования вычислительных устройств Отчет по научно-исследовательской теме. Томск, ТПИ, 1969.
4. М. Е. Ш т е й н, В. С. Г а й д а е н к о В. С. О задачах трассировки монтажных соединений. В кн.: Применение вычислительных машин для проектирования цифровых устройств. «Советское радио», М., 1968.
5. Синтез дискретных автоматов и управляющих устройств. «Наука», М., 1968.