Доклады Академии наук СССР
1971. Том 200, № 4

УДК 519.1

МАТЕМАТИКА

А. Ю. ЛЕВИН

АЛГОРИТМ КРАТЧАЙШЕГО СОЕДИНЕНИЯ ГРУППЫ ВЕРШИН ГРАФА

(Представлено академиком Л. В. Канторовичем 9 II 1971)

1. Рассматривается дискретная задача о кратчайшем соединении, которую можно сформулировать следующим образом. Дан конечный неориентированный связный граф G с вершинами $1, 2, \ldots, m$ и заданными (положительными) длинами ребер. Среди вершин графа G выделена некоторая группа из n вершин ($2 \leqslant n \leqslant m$), скажем, $1, 2, \ldots, n$. Требуется найти связный подграф графа G, среди вершин которого содержатся $1, 2, \ldots, n$ и который имеет минимальную суммарную длину ребер. Очевидно, искомый подграф не содержит циклов и поэтому имеет структуру дерева. Итак, кратко говоря, требуется «связать» вершины $1, 2, \ldots, n$ минимальным деревом (точнее, поддеревом графа G).

Крайние случаи $n = 2$ и $n = m$ хорошо известны. При $n = 2$ получаем задачу о кратчайшем пути, при $n = m$ — задачу о минимальном дереве без «дополнительных» вершин. Для обеих этих задач известны простые и весьма эффективные алгоритмы ([1-5]). Общий случай значительно сложнее. Ниже предлагается алгоритм решения, практически приемлемый при небольших n. С ростом m, n требуемый объем памяти ведет себя как $O(2^n m + m^2)$, а трудоемкость алгоритма — как $O(3^n m + 2^n m^2)$.

В дальнейшем длина ребра (i, j) обозначается через a_{ij} ($= a_{ij} > 0$). Через Γi обозначается совокупность вершин графа G, смежных с вершиной i; если ψ — некоторое множество вершин, то $\Gamma(\psi)$ — совокупность вершин, смежных хотя бы с одной вершиной из ψ. Через $\rho(i_1, i_2, \ldots, i_k)$ обозначается длина (т. е. суммарная длина ребер) минимального дерева, связывающего вершины $i_1, i_2, \ldots, i_k$, а через $D(i_1, i_2, \ldots, i_k)$ — само минимальное дерево (точнее, какое-либо из минимальных деревьев, которых, разумеется, может быть несколько).

Если называть каждую вершину дерева, имеющую степень (относительно этого дерева) $s \geqslant 3$, узлом кратности $s - 2$, то сумма кратностей всех узлов дерева равна числу висячих вершин минус два. Применим это соображение, легко доказываемое по индукции, к искомому минимальному дереву $D(1, 2, \ldots, n)$. Так как висячими вершинами в $D(1, 2, \ldots, n)$ могут быть только $1, 2, \ldots, n$, то число узлов в $D(1, 2, \ldots, n)$ заведомо не превосходит $n - 2$. (Аналогичный факт справедлив и для непрерывных задач о кратчайшем соединении.)

Это дает простой рецепт решения задачи при $n = 3$. Именно:

$$\rho(1, 2, 3) = \min_{1 \leqslant i \leqslant m} [\rho(i, 1) + \rho(i, 2) + \rho(i, 3)].$$

Если i_0 — минимизирующая вершина, то $D(1, 2, 3)$ получается объединением ребер кратчайших путей $D(i_0, 1)$, $D(i_0, 2)$, $D(i_0, 3)$ (при $i_0 \leqslant 3$ один из этих путей, естественно, аннулируется).

Та же идея перебора всевозможных претендентов на роль узлов в $D(1, 2, \ldots, n)$ реализуема и при $n > 3$, но вряд ли практически целесообразна: трудоемкость соответствующего алгоритма при $n = 4$ ведет себя как cm^3, а при любом фиксированном $n > 4$ — как cm^{n-2}.

3. Предлагаемый алгоритм основан на вычислении и запоминании всевозможных величин вида

$$\rho(i_1, i_2, \ldots, i_k, i), \tag{1}$$

$$1 \leqslant k \leqslant n-1, \quad 1 \leqslant i_1 < i_2 < \ldots < i_k \leqslant n-1, \quad 1 \leqslant i \leqslant m.$$

Каждая величина вида (1) именуется далее координатой, или, подробнее, k-координатой вершины i; таким образом, каждая вершина имеет $\binom{n-1}{k}$ k-координат. На первом этапе алгоритма находятся (и запоминаются) 1-координаты всех вершин, на втором — 2-координаты,..., $(n-1)$-м — $(n-1)$-координата всех вершин i. Значение ее при $i=n$ даст $\rho(1, 2, \ldots, n)$; найденные координаты позволят также восстановить $D(1, 2, \ldots, n)$.

На первом этапе с помощью $(n-1)$-кратного применения алгоритма «развертывания» [2, 4] из вершин 1, 2,..., $n-1$ вычисляются 1-координаты, т. е. кратчайшие расстояния от каждой из вершин графа G до каждой из вершин 1, 2,..., $n-1$.

Рассмотрим k-й этап алгоритма $(2 \leqslant k \leqslant n-1)$. Он состоит из $\binom{n-1}{k}$ аналогичных подэтапов, на каждом из которых вычисляются величины (1), $i = 1, 2, \ldots, m$, для того или иного набора $\{i_l\}$. Достаточно описать один из этих подэтапов, т. е. вычисление величины (1), обозначаемой далее для краткости через ρ_i, при фиксированных $i_1, i_2, \ldots, i_k$.

Величины $\rho_{i_1} = \ldots = \rho_{i_k} = \rho(i_1, i_2, \ldots, i_k)$ известны из $(k-1)$-го этапа. Вычисление остальных ρ_i осуществляется в две стадии. На первой стадии для всех $i \neq i_1, \ldots, i_k$ вычисляется так называемая квазикоордината (или, подробнее, k-квазикоордината) μ_i. Для этого рассматриваются всевозможные разбиения множества вершин $\Omega = \{i_1, i_2, \ldots, i_k\}$ на два непустых непересекающихся подмножества

$$\Omega = \Omega_u^1 \cup \Omega_u^2 \quad (i_1 \in \Omega_u^1), \quad u = 1, 2, \ldots, 2^{k-1}-1. \tag{2}$$

Квазикоордината вершины i определяется формулой

$$\mu_i = \min_u [\rho(\Omega_u^1, i) + \rho(\Omega_u^2, i)], \tag{3}$$

где $\rho(\Omega_u^j, i)$ — длина минимального дерева, связывающего вершину i и все вершины множества Ω_u^j $(j = 1, 2)$. Каждая величина вида $\rho(\Omega_u^j, i)$ известна из предыдущих этапов, так как является q-координатой вершины i с некоторым $q \leqslant k-1$. Найдя все квазикоординаты, упорядочиваем их в порядке возрастания μ_i, после чего переходим ко второй стадии — вычислению координат ρ_i.

Вторая стадия, в свою очередь, делится на ряд однотипных шагов, смысл которых легко уяснить, если ввести для искомых координат ρ_i другую нумерацию — в порядке возрастания:

$$\rho^1 < \rho^2 < \ldots < \rho^s \quad (1 \leqslant s \leqslant m-k+1).$$

При этом одна величина ρ^j может, конечно, быть координатой для нескольких вершин (в частности, $\rho^1 = \rho_{i_1} = \ldots = \rho_{i_k}$). На j-м шаге определяется величина ρ^j, и все вершины i, для которых $\rho_i = \rho^j$, снабжаются этой координатой. Таким образом, через s шагов все вершины будут снабжены координатами.

Далее Π^l, Φ^l — множества вершин, снабженных координатами на l-м шаге и на первых l шагах соответственно $(\Phi^l = \Pi^1 \cup \ldots \cup \Pi^l)$.

Пометкой на (r-м шаге) вершины $i \notin \Phi^{r-1}$ называется величина

$$\lambda_i^r = \min_{j \in \Phi^{r-1} \cap \Gamma_i} (\rho_j + a_{ij}) \quad (= \infty \text{ при } \Phi^{r-1} \cap \Gamma_i = \phi). \tag{4}$$

Первый шаг состоит в том, что наименьшей координатой $\rho^1 = \rho(i_1, \ldots, i_k)$ (известной из предыдущего этапа) снабжаются, помимо $i_1, \ldots, i_k$, также все вершины, квазикоординаты которых равны ρ^1. Затем эти вершины исключаются из упорядоченного квазикоординатного массива.

Опишем общий, r-й шаг ($r \geqslant 2$). Он начинается с вычисления пометок вершин $i \notin \Phi^{r-1}$, которое при $r = 2$ проводится по формуле (4), а при $r > 2$ — по более рациональной формуле

$$\lambda_i^r = \min\{\lambda_i^{r-1}, \min_{j \in \Pi^{r-1} \cap \Gamma_i} (\rho_j + \alpha_{ij})\} \quad (= \lambda_i^{r-1} \text{ при } \Pi^{r-1} \cap \Gamma_i = \phi). \tag{5}$$

Ясно, что в пересчете нуждаются лишь вершины $i \notin \Gamma(\Pi^{r-1}) \setminus \Phi^{r-1}$.

Координата ρ^r находится как минимум из оставшихся квазикоординат и текущих пометок:

$$\rho^r = \min\{\min_{i \notin \Phi^{r-1}} \mu_i, \min_{i \notin \Phi^{r-1}} \lambda_i^r\} \tag{6}$$

(ввиду упорядоченности квазикоординат $\min \mu_i$ определяется без вычислений). Далее координатой ρ^r снабжаются все вершины, минимизирующие правую часть (6), т. е. для которых выполнено хотя бы одно из соотношений

$$\mu_i = \rho^r, \quad \lambda_i^r = \rho^r \quad (i \notin \Phi^{r-1}). \tag{7}$$

Эти вершины исключаются из упорядоченного квазикоординатного массива и присоединяются к Φ^{r-1}, образуя вместе с ним множество Φ^r. Затем переходим к следующему шагу.

4. Не останавливаясь подробно на обосновании приведенного алгоритма вычисления координат, отметим, что в его основе по существу лежит следующая очевидная альтернатива:

а) если i не является висячей вершиной в $D(\Omega, i)$, то $D(\Omega, i)$ есть объединение двух минимальных деревьев вида $D(\Omega^1, i)$, $D(\Omega^2, i)$, где Ω^1, Ω^2 таковы, что

$$\Omega^1 \cup \Omega^2 = \Omega, \quad \Omega^1 \cap \Omega^2 = \phi, \quad \Omega^1 \neq \phi, \quad \Omega^2 \neq \phi;$$

б) если i является висячей вершиной в $D(\Omega, i)$, то для вершины j, с которой i соединена в $D(\Omega, i)$ ребром, имеет место равенство $\rho_j = \rho_i - \alpha_{ij}$.

Варианты а), б) соответствуют первому и второму из соотношений (7) (Оба соотношения (7) могут выполняться и одновременно, так как $D(\Omega, i)$ может быть не единственным.)

5. После вычисления всех координат строится дерево $D(1, 2, \ldots, n)$. Удобнее рассмотреть несколько более общий вопрос о построении произвольного дерева вида (как и ранее, $\Omega = \{i_1, \ldots, i_k\}$)

$$D(\Omega, i) \quad (1 \leqslant i_1 < \ldots < i_k \leqslant n-1, \ 1 \leqslant i \leqslant m, \ i \neq i_1, \ldots, i_k).$$

Проверяем, существует ли вершина $j \in \Gamma i$ такая, что $\rho_j = \rho_i - \alpha_{ij}$. Если существует, то (i, j) — одно из ребер $D(\Omega, i)$, и задача сводится к отысканию $D(\Omega, j)$. В противном случае восстанавливаем минимизирующее в (3) разбиение $\Omega = \Omega^1 \cup \Omega^2$ (величины $\rho(\Omega^1_{u}, i)$, $\rho(\Omega^2_{u}, i)$, будучи координатами вершины i, известны), после чего дело сводится к отысканию $D(\Omega^1, i)$, $D(\Omega^2, i)$.

Итак, указанная процедура в одном случае сводит задачу к отысканию минимального дерева с меньшим (на единицу) числом ребер, а в другом — к отысканию двух минимальных деревьев с меньшим числом соединяемых вершин. К этим деревьям вновь применяется та же процедура и т. д. Для полного построения $D(\Omega, i)$ понадобится не более $k + l - 2$ повторений процедуры, где l — число вершин в $D(\Omega, i)$.

6. Требуемый объем памяти (см. п. 1) обусловлен тем, что запоминаются $2^{n-1} - 1$ координат каждой из m вершин; кроме того, $O(m^2)$ ячеек нужно для запоминания графа и длин дуг. Трудоемкость складывается, в основном, из первой и второй стадий вычисления координат. Вычисление

любой из $\binom{n}{k}$ k-квазикоординат отдельной вершины по формуле (3) требует $2^{n-1}-2$ сравнений, так что общий вклад первой стадии есть $O(3^n m)$. Вклад же второй стадии равен $O(2^n m^2)$, ибо трудоемкость второй стадии отдельного подэтапа, как можно проверить, равна $O(m^2)$. Последнее относится и к любому подэтапу первого этапа, т. е. к вычислению расстояний до какой-либо вершины j $(1 \leqslant j \leqslant n-1)$. Отметим, что Дж. Данциг ([4]) также дает оценку $O(m^2)$ для трудоемкости решения задачи о кратчайшем пути, не учитывая, однако, упорядочивания соседей каждой вершины по длинам дуг. Эта работа, необходимая при предложенном в ([4]) методе, сама по себе требует, вообще говоря, $cm^2 \ln m$ операций. Если же организовать схему «развертывания» аналогично описанной выше второй стадии отдельного подэтапа (с тем упрощением, что здесь не участвуют квазикоординаты), то за $O(m^2)$ операций получим полное решение задачи о кратчайшем пути. Для графов общего вида не существует алгоритмов с меньшей (в смысле порядка) трудоемкостью.

Поступило
2 II 1971

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

[1] O. Boruvka, Práce Moravské Prirodovedecké Spolecnosti, 3, 1926. [2] G. Minty, Oper. Res., 5 (1957). [3] Р. К. Прим, Кибернетич. сборн., в. 2, ИЛ, 1961. [4] Дж. Данциг, Линейное программирование, его обобщения и применения, М., 1966. [5] С. И. Зуховицкий, Л. И. Авдеева, Линейное и выпуклое программирование, М., 1967.