1. Емеличев В. А., Ковалев М. М., Кравцов М. К. Многогранники, графы, оптимизация. М., 1981.
2. Емеличев В.А., Кравцов М.К. Дискрет. мат. 1991. Т. 3. Вып. 2. С. 3.
3. Balas E., Saltzman M.J. Discrete Appl. Math. 1989. Vol. 23. № 3. P. 201.
4. Кравцов М.К., Кравцов В.М., Лукшин Е.В. Проблемы теоретической кибернетики: Тез. докл. XII Междунар. конф. М., 1999. Ч. 1. С. 119.
5. Кравцов М.К., Лукшин Е.В. Изв. вузов. Математика. 1999. № 12. С. 65.

Поступила в редакцию 17.06.99.

УДК 519.1

Н.А. НАУМОВИЧ

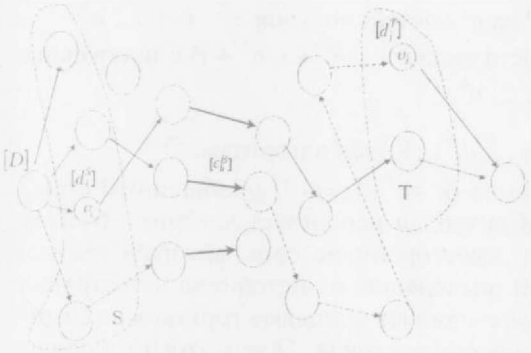
ГЕНЕРАТОР ЗАДАЧ О МИНИМАЛЬНОМ РАЗРЕЗЕ

An algorithm for generation of mincut-maxflow problem instances with with apriori known optimal solution is suggested.

Экземпляры задач о минимальном разрезе используются в качестве тестовых при практическом анализе многих комбинаторных алгоритмов оптимизации, в частности задачи о минимизации субмодулярной функции [1]. Алгоритм, описанный в данной статье, предназначен для генерации сетей с предписанным количеством вершин и дуг, а также с априори известным решением задачи о минимальном разрезе – множеством дуг, составляющим минимальный разрез.

Алгоритм MinCutNetGen генерации задач о максимальном потоке с известным минимальным разрезом

Алгоритм MinCutNetGen генерирует сеть, состоящую из двух частей: *левосторонней* и *правосторонней сети*, а также соединяющего их *моста* (рисунок).



Структура сети, сгенерированной алгоритмом MinCutNetGen. Жирными стрелками выделен «мост». S – множество источников, T – множество стоков. Для некоторых вершин и дуг указаны их пропускные способности

Параметрами алгоритма являются: n^L – количество вершин левосторонней сети; m^L – количество дуг левосторонней сети; n^S – количество источников; n^R – количество вершин правосторонней сети; m^R – количество дуг правосторонней сети; n^T – количество стоков; n^B – количество дуг в мосте; D – целочисленное суммарное производство в источниках; C – априорная целочисленная суммарная пропускная способность дуг моста ($C \le D$). Апостериорная суммарная пропускная способность дуг моста не превосходит C и определяется в процессе генерации сети $[u^{\min}, u^{\max}]$ – диапазон изменения верхней пропускной способности дуг (обеих сетей), причем сгенерированная алгоритмом верхняя пропускная способность некоторых дуг может превосходить $[u^{\max}]$, если это требуется для обеспечения вывоза всего объема производства D из источников. Сгенерированная сеть имеет $n=n^L+n^R+2$ вершины и $m=m^L+n^S+n^B+m^R+n^T$ дуг. На значения параметров накладываются следующие ограничения:

связность: $m^L \ge n^L, m^R \ge n^R$;

допустимость: $D \geq n^S$. Алгоритм использует следующие промежуточные величины: d_i^S – производство в источнике i, полученное равномерным распределением суммарного производства D между n^S источниками, $i=1,\ldots, n^S$, $D = \sum_i^{n^S} d_i^S$; d_i^B – потребление в i-й вершине левой части моста, $i=1,\ldots, n^B$, $D = \sum_i^{n^B} d_i^B$; c_i^B, $i=1,\ldots, n^B$ – величины, полученные равномерным распределением C на n^B частей; d^T – потребление в i-м стоке, полученное в процессе работы алгоритма, $i=1,\ldots, n^T$.

Алгоритм MinCutNetGen генерации сети с известным минимальным разрезом для задачи о максимальном потоке

Вход: $n^L, m^L, n^S, m^S, n^R, m^R, n^T, n^B, D, C, u^{\min}, u^{\max}$.

Выход: v^C – величина минимального разреза; минимальный разрез, которым является, в частности, мост между левосторонней и правосторонней сетями.

Шаг 1. Получить d_i^S, $i=1,\ldots, n^S$, равномерно распределив D между n^S источниками.

Шаг 2. [Сгенерировать левостороннюю сеть.] Вызвать NetGen($0, n^L, m^L, n^S, m^S, n^B, d^B, u^{\min}, u^{\max}$).

Шаг 3. [Сгенерировать правостороннюю сеть.] Вызвать NetGen($n^L, n^R, m^R, n^B, d^B, n^T, d^T, u^{\min}, u^{\max}$).

Шаг 4. Сгенерировать n^S дуг вида (s,i), где s – *обобщенный источник* с пропускными способностями d_i^S, $i=1,\ldots, n^S$.

Шаг 5. Сгенерировать n^T дуг вида (i,t), где t – *обобщенный сток* с пропускными способностями d_i^T – потребление в i-м стоке, полученное в процессе работы алгоритма, $i=1,\ldots, n^T$.

Шаг 6. Равномерно разделить C на n^B частей, получая c_i^B, $i=1,\ldots, n^B$.

Шаг 7. Сгенерировать n^B дуг моста вида $(n^L - n^B + i, n^L + i)$ с пропускными способностями $\min(c_i^B, d_i^B)$, $i=1,\ldots, n^B$.

Шаг 8. Вычислить $v^C = \sum_{i=1}^{n^B} \min(c_i^B, d_i^B)$. Конец алгоритма. ⊗

Обобщенный источник имеет номер $(n^L+n^R+1)=n-1$, обобщенный сток – номер $(n^L+n^R+2)=n$. Алгоритм MinCutNetGen использует алгоритм NetGen, генерирующий правостороннюю и левостороннюю сети, который вначале генерирует непересекающиеся цепи, выходящие из источника и состоящие из случайного количества дуг, затем соединяет концевые вершины сгенерированных цепей со случайным количеством стоков. После этого генерируются дополнительные дуги до достижения необходимого количества m дуг. Пропускные способности дуг генерируются таким образом, чтобы весь продукт, произведенный в источниках, мог быть привезен в стоки. Пусть Distribute (D, n, d) – процедура, которая, используя датчик псевдослучайных равномерно распределенных чисел, делит целое число D на n частей $d_i > 0$, $i=1,\ldots,n$ таким образом, что $D = \sum_{i=1}^{n} d_i$.

Приведем описание алгоритма NetGen генерации потоковой сети.

Вход: n – количество вершин, m – количество дуг; n^0 – сдвиг в нумерации вершин (первая вершина сети имеет порядковый номер n^0+1); n^S – количество источников; d_i^S, $i=1,\ldots, n^S$ – производство в i-м источнике; n^T – количество стоков; $[u^{\min}, u^{\max}]$ – диапазон изменения пропускных способностей дуг.

Выход: потоковая сеть, вершины которой перенумерованы так, что первая вершина имеет номер n^0+1, d_i^T – потребление в i-м стоке, $i=1,\ldots, n^T$.

Шаг 1. [Сгенерировать цепи.] Вначале сеть состоит только из вершин и не имеет дуг. Пусть S – множество источников, T – множество стоков. Если $n=n^S+n^T$, то промежуточных вершин нет, и цепи состоят только из вершин-источников. Иначе – определить, сколько дуг должно быть в каждой из цепей: **Distribute** $(n-n^S-n^T,\ n^S,\ N^C)$, $N^C=(\eta_1^C,\ldots,\eta_{n^S}^C)$. Сгенерировать n^S вершинно непересекающихся цепей, причем i-я цепь состоит из η_i^C дуг, и пропускная способность каждой j-й дуги i-й цепи определяется как $c_{ij}=\max(d_i^S,u_j)$, где u_j – случайное число из диапазона $[u^{\min}, u^{\max}]$. Для каждой i-й цепи вычислить d_i^{CE} – количество продукта, которое может быть привезено в концевую вершину цепи, $i=1,\ldots, n^S$.

Шаг 2. Для $i\in 1,\ldots, n^S$ определить T_i – множество стоков, с которыми должна быть соединена i-я цепь. Распределить поток i-й цепи между $|T_i|$ стоками: **Distribute** $(d_i^{CE},|T_i|,D^{CT})$, где $D^{CT}=(d_1^{CT},\ldots,d_{|T_i|}^{CT})$ – результирующий вектор, и соединить каждую концевую вершину i-й цепи с $|T_i|$ стоками $j\in T_i$ дугами пропускной способности d_j^{CT}.

Шаг 3. Пусть $\overline{m}$ – количество уже сгенерированных дуг. Если $m>\overline{m}$, то сгенерировать $m-\overline{m}$ различных дуг с верхней пропускной способностью из диапазона $[u^{\min},u^{\max}]$. Перенумеровать вершины полученной сети начиная с n^0. ⊗

Теорема. В построенной потоковой сети:

1) дуги моста B являются одним из минимальных разрезов;

2) величина минимального разреза равна $v^C=\sum_{i=1}^{n^B}\min(c_i^B,d_i^B)$.

Доказательство теоремы следует из описания алгоритма. ⊗

1. Свами М., Тхуласираман К. Графы, сети и алгоритмы. М., 1984.

Поступила в редакцию 25.01 2000.

ВНИМАНИЮ ЧИТАТЕЛЕЙ

В статье В.А. Емеличева, Ю.В. Никулина «О радиусе сильной устойчивости векторной траекторной задачи», опубликованной в № 1 журнала за 2000 г., на с. 49 по техническим причинам допущена опечатка.

Перед формулировкой теоремы следует заменить введеные обозначения

$$\varphi^n(A)=\min_{t\in P(A)}\ \max_{t\in\overline{P}(A)}\ \min_{i\in N_n}\ \Gamma_i(t,t,A),$$

$$\psi^n(A)=\min_{t\in\overline{P}(A)}\ \max_{t\in P(A)}\ \min_{i\in N_n}\ \Gamma_i(t,t,A).$$

на

$$\varphi^n(A)=\max_{t\in P(A)}\ \min_{t\in\overline{P}(A)}\ \max_{i\in N_n}\ \Gamma_i(t,t,A),$$

$$\psi^n(A)=\min_{t\in\overline{P}(A)}\ \max_{t\in P(A)}\ \min_{i\in N_n}\ \Gamma_i(t,t,A).$$