ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Январь—июнь 1999. Серия 2. Том 6, № 1, 3–11

УДК 519.87+519.854.33

ЗАДАЧА РАЗМЕЩЕНИЯ НА СЕТИ С ОГРАНИЧЕННЫМИ ПРОПУСКНЫМИ СПОСОБНОСТЯМИ КОММУНИКАЦИЙ*)

И. П. Вознюк

Рассмотрена задача о наилучшем размещении пунктов производства в вершинах сети с ограниченными пропускными способностями коммуникаций. Для решения задачи на древовидной сети предложен алгоритм динамического программирования с временем работы $T = O(n^3b^2)$ и требуемой памятью $\Pi = O(n^2b)$, где n — число вершин сети, b — максимальный объем спроса. Если сеть является цепью, то алгоритм решения задачи имеет оценки $T = O(n^3)$, $\Pi = O(n)$. В общем случае реализован метод ветвей и границ. Приведены результаты численного эксперимента.

Большое внимание исследователей (см., например, [1, 3–5, 8]) уделяется рассмотрению следующей задачи. Задана сеть коммуникаций, связывающая пункты потребления некоторого продукта и возможные места его производства. Требуется разместить пункты производства таким образом, чтобы суммарные затраты на открытие предприятий и транспортные расходы по доставке заданных объемов продукта от производителей к потребителям были минимальны. В данной статье эта задача изучается при ограничениях на пропускные возможности коммуникаций сети. Объемы производства продукта предполагаются «неограниченными». Ситуация с ограничениями на объемы производства легко сводится к рассматриваемому случаю.

1. Постановка задачи

Рассмотрим сеть коммуникаций, представленную графом $G = (N, E)$. Вершины из множества $N = \{1, \dots, n\}$ обозначают пункты потребления некоторого продукта и возможные места его производства. Ребра из E соответствуют коммуникациям сети, по которым происходит доставка

*) Работа выполнена при финансовой поддержке Федеральной целевой программы «Интеграция».

продукта от производителей к потребителям. Известны объем спроса b_i и затраты g_i^0 на размещение производства в пункте i, $i \in N$. Каждая коммуникация $e \in E$ характеризуется стоимостью c_e транспортировки по ней единицы продукта и максимальным количеством a_e продукта, поставляемого по этой коммуникации.

Требуется указать те пункты производства из N, при открытии которых удовлетворяется спрос на производимый продукт и суммарные затраты на открытие предприятий и доставку продукта минимальны.

Введем вспомогательные обозначения и переменные величины:

P_{ij} — множество всех простых путей, ведущих в графе G из вершины i в вершину j;

Q_e — множество таких троек (i, j, p), что путь p проходит через ребро e и $p \in P_{ij}$;

$g_{ij}^p = \sum\limits_{e \in p} c_e$ — стоимость транспортировки единицы продукта по пути $p \in P_{ij}$;

x_i — переменная выбора, равная единице, если в пункте $i \in N$ открывается предприятие, и равная нулю в противном случае;

x_{ij}^p — количество продукта, поставляемое из пункта производства i в пункт потребления j по пути $p \in P_{ij}$.

В принятых обозначениях задача формулируется следующим образом: найти минимум суммарных затрат

$$\sum_{i \in N} g_i^0 x_i + \sum_{i,j \in N} \sum_{p \in P_{ij}} g_{ij}^p x_{ij}^p \tag{1}$$

по всем переменным x_i, x_{ij}^p при условиях

$$\sum_{i \in N} \sum_{p \in P_{ij}} x_{ij}^p = b_j, \quad j \in N, \tag{2}$$

$$\sum_{(i,j,p) \in Q_e} x_{ij}^p \leqslant a_e, \quad e \in E, \tag{3}$$

$$0 \leqslant x_{ij}^p \leqslant b_j x_i, \quad p \in P_{ij}, \quad i, j \in N, \tag{4}$$

$$x_i \in \{0, 1\}, \quad i \in N. \tag{5}$$

Равенство (2) обеспечивает удовлетворение спроса в пункте $j \in N$. Соотношение (3) учитывает ограничение на пропускную способность ребра $e \in E$. Неравенства (4) не позволяют поставлять продукт из пункта i, если в нем не размещено производство. Постановку (1)–(5) будем кратко называть *задачей размещения*.

Числа a_e, b_i считаем целыми. Полагаем $a = \max\limits_{e\in E} a_e$, $b = \max\limits_{i\in N} b_i$. Предполагаем, что $a \leqslant nb$, поскольку поток продукта по коммуникации сети заведомо не превосходит nb.

Отметим используемое ниже и легко доказываемое утверждение.

Лемма 1. *Существует такое оптимальное решение задачи* (1)–(5), *что*

1) *каждый пункт производства не является транзитной вершиной для потоков продукта из других предприятий*;

2) *все потоки продукта, идущие через данное ребро, имеют одно и то же направление.*

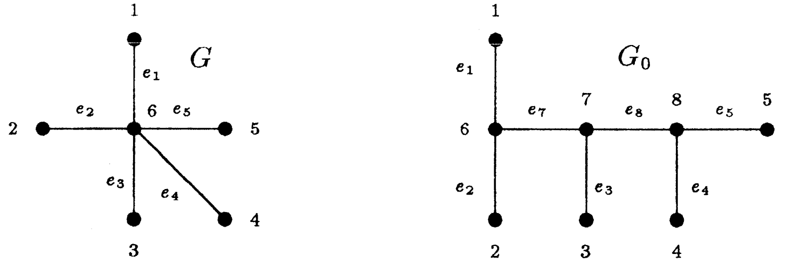
2. Задача размещения на древовидной сети

Пусть граф $G = (N, E)$ является деревом. Дерево называется *двоичным*, если степени всех его вершин не превосходят 3.

Лемма 2. *Задача размещения на дереве сводится к задаче размещения на двоичном дереве, содержащем не более* $2n$ *вершин.*

Доказательство. Исходное дерево $G = (N, E)$ заменим двоичным деревом $G_0 = (N_0, E_0)$, вводя для каждой вершины степени более 3 дополнительные вершины и ребра. Эта операция продемонстрирована для вершины 6 степени 5 (см. рисунок). Числовые данные исходной задачи дополним характеристиками новых элементов сети. Используя в качестве примера этот рисунок, получаем

$$g_7^0 = g_8^0 = \infty, \quad b_7 = b_8 = 0, \quad c_{e_7} = c_{e_8} = 0, \quad a_{e_7} = a_{e_8} = nb.$$



Пример преобразования дерева в двоичное дерево

Нетрудно убедиться, что задача размещения на дереве G_0 равносильна исходной.

Покажем, что $|N_0| \leqslant 2n$. Это очевидно, если $n = 1$. Пусть $n \geqslant 2$. Обозначим через d_i степень вершины $i \in N$ в дереве G. Положим

$M = \{i \in N \mid d_i > 3\}$. При переходе от дерева G к дереву G_0 для каждой вершины $i \in M$ вводится $d_i - 3$ новых вершин. Поэтому

$$|N_0| = n + \sum_{i \in M}(d_i - 3) = n + \sum_{i \in M} d_i - 3|M|.$$

Так как $d_i \geqslant 1$ при всех $i \in N$ и

$$\sum_{i \in M} d_i = 2(n-1) - \sum_{i \in N \setminus M} d_i,$$

получаем

$$|N_0| = n + 2(n-1) - \sum_{i \in N \setminus M} d_i - 3|M| \leqslant n + 2(n-1) - |N \setminus M| - |M| \leqslant 2n.$$

Следовательно, $|N_0| \leqslant 2n$. Лемма 2 доказана.

Теорема 1. *Задача размещения на двоичном дереве решается методом динамического программирования за время $T = O(n^3 b^2)$ при объеме памяти $\Pi = O(n^2 b)$.*

Доказательство. Двоичное дерево $G = (N, E)$ представим как корневое и ориентированное. В качестве корня возьмем произвольную вершину $\alpha \in N$. Все ребра ориентируем в направлении от корня.

Рассмотрим поддерево G_i с корнем в вершине i. Через l обозначим вершину, смежную с вершиной i по дуге $e = (l, i)$, и через y — суммарный поток продукта по коммуникации e. По лемме 1 достаточно рассматривать потоки продукта, имеющие одно и то же направление. Условимся считать $y > 0$, если поток продукта идет по дуге e «внутрь» дерева G_i, и $y < 0$, если он идет в обратном направлении «вне» дерева G_i.

Через $S_i(y)$ обозначим оптимум в задаче размещения на поддереве G_i при условии, что поток продукта по дуге e равен y, и через $S_i(y, x_i)$ — аналогичный оптимум при фиксированном значении переменной x_i. В эти величины включим и затраты на транспортировку $|y|$ единиц продукта по дуге e.

Перейдем к составлению рекуррентных уравнений динамического программирования для расчета величины $S_\alpha(0)$, равной оптимуму исходной задачи. Параметр y пробегает в уравнениях целочисленный отрезок $[-a_e, a_e]$. Полагаем $S_i(y) = S_i(y, x_i) = \infty$, если $y \notin [-a_e, a_e]$.

Очевидно, что $S_i(y) = \min\{S_i(y, 0),\ S_i(y, 1)\}$. В зависимости от характера вершины i возможны три ситуации.

1. Вершина i концевая. Тогда

$$S_i(y, 0) = \begin{cases} c_e y, & \text{если } y \geqslant b_i, \\ \infty, & \text{если } y < b_i, \end{cases} \qquad S_i(y, 1) = c_e|y| + g_i^0.$$

2. Из вершины i выходит одна дуга $e' = (i, j)$. В этом случае

$$S_i(y,0) = c_e|y| + S_j(y - b_i), \quad S_i(y,1) = c_e|y| + g_i^0 + \min_{z\in[0,a_{e'}]} S_j(z).$$

3. Из вершины i выходят две дуги $e' = (i, j)$ и $e'' = (i, k)$. Тогда

$$S_i(y,0) = c_e|y| + \min_{z\in[-a_{e'},a_{e'}]} \{S_j(z) + S_k(y - b_i - z)\},$$

$$S_i(y,1) = c_e|y| + g_i^0 + \min_{z\in[0,a_{e'}]} S_j(z) + \min_{z\in[0,a_{e''}]} S_k(z).$$

Алгоритм динамического программирования состоит из «прямого хода», на котором по выписанным уравнениям рассчитываются величины $S_i(y)$, и «обратного хода», на котором восстанавливается решение исходной задачи. Прямой ход алгоритма требует $O(na^2)$ арифметических операций при объеме памяти $O(na)$ для хранения промежуточных результатов. Обратный ход осуществляется за $O(n)$ операций. Расчеты удобнее вести, предварительно перенумеровав вершины таким образом, что если дуга (i, j) принадлежит дереву G, то $i > j$. Подобная нумерация строится за $O(n)$ операций [6].

В целом, учитывая условие $a \leqslant nb$, получаем требуемые оценки времени работы алгоритма $T = O(n^3b^2)$ и памяти $\Pi = O(n^2b)$. Теорема 1 доказана.

В силу леммы 2 и теоремы 1 задача размещения на произвольном дереве G решается с теми же оценками времени $T = O(n^3b^2)$ и памяти $\Pi = O(n^2b)$. Более экономичный алгоритм можно предложить в частном случае, когда исходное дерево G является цепью.

Теорема 2. *Задача размещения на цепи решается методом динамического программирования за время $T = O(n^3)$ при объеме памяти $\Pi = O(n)$.*

Доказательство. Вершины цепи занумеруем числами $1, \ldots, n$ в порядке ее обхода. К цепи добавим вершины 0, $n+1$ и ребра $e' = \{0, 1\}$, $e'' = \{n, n+1\}$. Введем числовые характеристики новых элементов сети:

$$g_0^0 = g_{n+1}^0 = 0, \quad b_0 = b_{n+1} = 0, \quad a_{e'} = a_{e''} = 0, \quad c_{e'} = c_{e''} = \infty.$$

Пусть $0 \leqslant i < j \leqslant n+1$. Через $h(i, j)$ обозначим минимум транспортных затрат, необходимых для покрытия спроса в пунктах $i, i+1, \ldots, j$ при условии, что продукт поставляется из пунктов i и j. Полагаем $h(i, j) = \infty$, если удовлетворение спроса невозможно из-за ограничений на пропускные способности ребер цепи.

Введем обозначение $f(i, j) = g_i^0 + h(i, j)$ и рассмотрим задачу: найти минимум суммы

$$\sum_{k=1}^{s+1} f(i_{k-1}, i_k) \tag{6}$$

по переменным $i_1, \ldots, i_s$ и s при ограничениях

$$0 = i_0 < i_1 < \ldots < i_s < i_{s+1} = n+1, \quad s = 1, \ldots, n. \tag{7}$$

Нетрудно видеть, что любое оптимальное решение $\{i_1, \ldots, i_s\}$ задачи (6), (7) соответствует допустимому решению исходной задачи, в котором предприятия открываются в пунктах $i_1, \ldots, i_s$. С другой стороны, если в оптимальном решении исходной задачи предприятия размещаются в пунктах $i_1, \ldots, i_s$, то с учетом леммы 1 минимум суммарных затрат равен сумме (6). Следовательно, исходная задача размещения на цепи сводится к задаче (6), (7).

Задача (6), (7) решается следующей схемой динамического программирования [3]:

$$S(0) = 0, \quad S(j) = \min_{0 \leqslant i < j} \{S(i) + f(i,j)\}, \; j = 1, \ldots, n+1.$$

Оптимум в задаче (6), (7) равен $S(n+1)$.

Величина $f(i,j)$ вычисляется за $O(j-i)$ арифметических операций. Поэтому расчет величины $S(j)$ требует $O(j^2)$ операций. В целом прямой ход динамического программирования реализуется за $O(n^3)$ операций при объеме памяти $O(n)$ для хранения промежуточных результатов. Восстановление оптимального решения задачи (6), (7) осуществляется обратным ходом за $O(n)$ операций. Следовательно, задача размещения на цепи решается алгоритмом с временной сложностью $T = O(n^3)$ и памятью $\Pi = O(n)$. Теорема 2 доказана.

3. Задача размещения на сети общего вида

Пусть сеть коммуникаций $G = (N, E)$ имеет произвольную структуру. Если $a_e = nb$ при всех $e \in E$, то задача (1)–(5) равносильна NP-трудной задаче размещения предприятий на сети без ограничений на пропускные способности коммуникаций, для решения которой использовался метод ветвей и границ [3, 8]. Этим же методом будем решать рассматриваемую задачу размещения. Основными элементами метода являются две описываемые ниже процедуры. Это вычисление нижней оценки оптимума функционала (1) на подмножестве вариантов размещения и «зондирование» подмножества, т. е. построение приближенного решения. Суть метода удобнее излагать, перейдя от постановки (1)–(5) к равносильной формулировке, в которой перемещение продукта описывается потоками по отдельным коммуникациям сети G.

Каждое ребро $e = \{i,j\} \in E$ заменим парой дуг (i,j) и (j,i). Множество всех дуг обозначим через $\vec{E}$. Через y_{ij} обозначим количество продукта, перемещаемое по дуге $(i,j) \in \vec{E}$, через a_{ij} и c_{ij} — пропускную

способность дуги (i,j) и стоимость транспортировки по ней единицы продукта, полагая $a_{ij} = a_e$, $c_{ij} = c_e$, где $e = \{i,j\} \in E$.

К рассматриваемой сети добавим две новые вершины s, t и дуги (s,i), (i,t), $i \in N$. Расширенную таким образом сеть обозначим через $\vec{G}$. Через y_{si} и y_{it} обозначим потоки по добавленным дугам. Эти величины соответствуют количествам продукта, вывозимого из предприятия, открытого в пункте i, и поставляемого в пункт i со всех предприятий.

Пусть $x = (x_1, \dots, x_n)$ — некоторый вариант размещения предприятий. Положим

$$N_i = \{j \in N \mid \{i,j\} \in E\},\ i \in N, \quad B = \sum_{i \in N} b_i.$$

Величина $F(x)$, включающая затраты на открытие предприятий и минимум транспортных расходов в варианте размещения x, задается в виде

$$F(x) = \sum_{i \in N} g_i^0 x_i + \min_{(y_{si}, y_{ij}, y_{it})} \sum_{(i,j) \in \vec{E}} c_{ij} y_{ij}. \tag{8}$$

Минимум берется при ограничениях

$$y_{si} + \sum_{j \in N_i} y_{ji} = \sum_{j \in N_i} y_{ij} + y_{it}, \quad i \in N, \tag{9}$$

$$0 \leqslant y_{ij} \leqslant a_{ij},\ (i,j) \in \vec{E}, \quad 0 \leqslant y_{it} \leqslant b_i,\ i \in N, \tag{10}$$

$$\sum_{i \in N} y_{si} = \sum_{i \in N} y_{it} = B, \tag{11}$$

$$0 \leqslant y_{si} \leqslant nbx_i, \quad i \in N. \tag{12}$$

Поиск минимума в (8) представляет собой известную задачу о потоке минимальной стоимости. Методы ее решения изложены, например, в [2, 7, 9]. Если ограничения (9)–(12) несовместны (максимальная мощность потока из источника s в сток t меньше B), то полагаем $F(x) = \infty$.

Обозначим через X множество всех векторов $x = (x_1, \dots, x_n)$ с компонентами 0 и 1. Задача размещения предприятий заключается в поиске минимума величины $F(x)$ по всем вариантам $x \in X$.

Пусть k — целое число из отрезка $[1, n]$ и $z = (z_1, \dots, z_k)$ вектор с компонентами 0 и 1. Рассмотрим подмножество вариантов размещений

$$X(z) = \{x \in X \mid x_i = z_i,\ i = 1, \dots, k\}.$$

Нижняя оценка. Для каждого $i \in N$ вычислим величину Y_i максимального объема продукта, который может поставляться из пункта производства i во все пункты потребления с учетом пропускных способностей коммуникаций. Это можно сделать, решив при всех $i \in N$

задачу о максимальном потоке [2, 7] в сети $\vec{G}$ из источника s в сток t при ограничениях (9), (10) и условиях

$$0 \leqslant y_{si} \leqslant nb, \quad y_{sj} = 0,\ j \neq i.$$

Поскольку при любом потоке продукта в сети $\vec{G}$ выполняются неравенства $y_{si} \leqslant Y_i$, $i \in N$, в качестве нижней оценки минимума функции $F(x)$ на множестве $X(z)$ можно принять величину

$$H(z) = \sum_{i=1}^{k} g_i^0 z_i + \min_{(y_{si}, y_{ij}, y_{it})} \left\{ \sum_{i=k+1}^{n} \frac{g_i^0}{Y_i} y_{si} + \sum_{(i,j)\in\vec{E}} c_{ij} y_{ij} \right\}.$$

Минимум берется при ограничениях (9)–(11) и условиях

$$0 \leqslant y_{si} \leqslant Y_i z_i,\ i = 1, \ldots, k, \quad 0 \leqslant y_{si} \leqslant Y_i,\ i = k+1, \ldots, n. \tag{13}$$

Эта оценочная задача представляет собой задачу о потоке минимальной стоимости. Полагаем $H(z) = \infty$, если ограничения (9)–(11), (13) несовместны.

Зондирование подмножества. Приближенное решение задачи размещения на множестве $X(z)$ будем строить, используя аналогично [5] процедуру покоординатного спуска. Обозначим через d_i вектор размерности n, в котором координата i равна 1, а остальные координаты равны 0. Пусть y_{si}^* — компоненты оптимального решения оценочной задачи. Алгоритм состоит из следующих шагов.

Шаг 1. Строится вектор $x \in X(z)$ с компонентами $x_i = 1$, если $y_{si}^* > 0$, и $x_i = 0$, если $y_{si}^* = 0$, где $i = k+1, \ldots, n$.

Шаг 2. Находится величина $h = \max\{F(x) - F(x - d_i) \mid i = k+1, \ldots, n,$ $x_i = 1\}$. Пусть максимум достигается на номере i_0.

Шаг 3. Если $h \geqslant 0$, то полагается $x = x - d_{i_0}$ и повторяется шаг 2. В противном случае вычисления прекращаются.

Для проверки работоспособности предложенных конструкций была разработана программная реализация метода ветвей и границ, использующая одновременную схему ветвления [3]. Расчеты проводились на ПЭВМ IBM PC 486DX с тактовой частотой 100 Мгц. Рассматривались задачи размещения на сетях с числом вершин $n = 50$. Вершины i и j соединялись ребром $e = \{i, j\}$ с вероятностью 0,5. Величины b_i, a_e, c_e выбирались случайным образом равномерно из целочисленного отрезка $[1, 10]$. Начальные затраты g_i выбирались тем же способом из отрезка $[10^3, 10^4]$.

В серии из 10 задач время работы программы изменялось от 3 до 20 мин и составило в среднем 9 мин. Число вершин в дереве ветвлений колебалось от 49 до 353 и в среднем равно 182.

В заключение выражаю искреннюю признательность моему научному руководителю Э. Х. Гимади за предложенную тему исследований и всестороннюю поддержку и Ю. В. Шамардину за помощь при написании статьи.

ЛИТЕРАТУРА

1. **Агеев А. А.** Полиномиальный алгоритм решения задачи размещения на последовательно-параллельной сети // Управляемые системы: Сб. науч. тр. Новосибирск: Ин-т математики СО АН СССР, 1990. Вып. 30. С. 3–16.
2. **Адельсон-Вельский Г. М., Диниц Е. А., Карзанов А. В.** Потоковые алгоритмы. М.: Наука, 1975.
3. **Береснев В. Л., Гимади Э. Х., Дементьев В. Т.** Экстремальные задачи стандартизации. Новосибирск: Наука, 1978.
4. **Гимади Э. Х.** Эффективный алгоритм решения задачи размещения с областями обслуживания, связными относительно ациклической сети // Управляемые системы: Сб. науч. тр. Новосибирск: Ин-т математики СО АН СССР, 1983. Вып. 23. С. 12–23.
5. **Гимади Э. Х., Глебов Н. И., Дементьев В. Т.** Об одном методе построения нижней оценки и приближенного решения с апостериорной оценкой точности для задачи стандартизации // Управляемые системы: Сб. науч. тр. Новосибирск: Ин-т математики СО АН СССР, 1974. Вып. 13. С. 26–31.
6. **Пападимитриу Х., Стайглиц К.** Комбинаторная оптимизация. М.: Мир, 1985.
7. **Форд Л. Р., Фалкерсон Ф. Р.** Потоки в сетях. М.: Мир, 1963.
8. **Cornuejols G., Nemhauser G. L., Wolsey L. A.** The uncapacitated facility location problem // Mirchandani P. B., Francis R. L. (eds.) Discrete Location Theory: Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley and Sons Inc., 1990. P. 119–171.
9. **Orlin J. B.** A faster strongly polynomial minimum cost flow algorithm // Oper. Res. 1993. V. 41, N 2. P. 338–350.

Адрес автора:

Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск,
Россия

Статья поступила
14 апреля 1998 г.,
переработанный вариант —
13 февраля 1999 г.