ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль—декабрь 2005. Серия 2. Том 12, № 2, 85–96

УДК 519.87+519.854

НЕКОТОРЫЕ СЛУЧАИ ПОЛИНОМИАЛЬНОЙ РАЗРЕШИМОСТИ ЗАДАЧИ О ВЫБОРЕ ЦЕН НА ПРОДУКЦИЮ

Ю. В. Шамардин

Продолжается исследование задачи об оптимальном выборе производителем цен на продукцию при условии обязательного удовлетворения потребительского спроса. Рассматривается ситуация, когда каждый пункт производства доступен ровно двум потребителям. Показано, что в этом частном случае задача остается NP-трудной. Найдены три случая ее полиномиальной разрешимости.

1. Постановка задачи

В работе [2] было начато изучение следующей задачи. Пусть известен список из n пунктов производства (предприятий) некоторого продукта и задан перечень m потребителей этого продукта. Каждое предприятие имеет «неограниченный» объём производства, т. е. способно удовлетворить спрос любого числа потребителей. Производитель, владеющий всеми предприятиями, вправе устанавливать в каждом пункте производства любую цену на выпускаемый продукт.

Относительно каждого потребителя i, $1 \leqslant i \leqslant m$, известны объём спроса b_i, множество предприятий, доступных данному потребителю, и ценовой порог d_i, характеризующий покупательную способность потребителя. Потребитель выбирает из всех доступных предприятий то, на котором цена продукта наименьшая, но делает покупку только в случае, если эта цена не превышает d_i. Предполагается, что производитель обязан устанавливать такие цены, чтобы весь потребительский спрос был удовлетворён.

В этих условиях требуется найти цены продукта на каждом предприятии, при которых суммарный доход производителя максимален.

В [2], в частности, исследовалась ситуация, когда каждому потребителю доступны только два предприятия. В данной статье рассматривается противоположный случай, когда каждое предприятие доступно

ровно двум потребителям. Показано, что в этом случае задача остается NP-трудной. Выделены три случая полиномиальной разрешимости задачи с помощью динамического программирования. Перейдём к формальной постановке.

Введём обозначения:

$G = (V, E)$ — граф с множеством вершин $V = \{1, \dots, m\}$, соответствующих потребителям, и множеством рёбер E, $|E| = n$, соответствующих предприятиям (смежность вершин $i, j \in V$ означает, что пункт производства $\{i, j\} \in E$ доступен потребителям i и j);

x_{ij} — цена единицы продукта на предприятии $\{i, j\} \in E$;

$x = (x_{ij})$ — вектор цен, выбираемых производителем;

$N_i = \{j \in V \mid \{i, j\} \in E\}$ — окрестность вершины $i \in V$ (предполагаем, что каждому потребителю $i \in V$ доступно хотя бы одно предприятие, т. е. граф G не имеет изолированных вершин и $N_i \neq \varnothing$);

$\varphi_i(x) = \min\{x_{ij} \mid j \in N_i\}$ — минимальная цена продукта среди предприятий, доступных потребителю $i \in V$.

Требуется найти максимум суммарного дохода производителя, т. е. максимальное значение функции

$$\Phi(x) = \sum_{i \in V} b_i \varphi_i(x) \tag{1}$$

при ограничениях обязательного удовлетворения спроса

$$\varphi_i(x) \leqslant d_i, \quad i \in V, \tag{2}$$

и возможных значений цен

$$x_{ij} \in \{d_i, d_j\}, \quad \{i, j\} \in E. \tag{3}$$

Здесь вместо условия неотрицательности переменных x_{ij} выписано включение (3), что справедливо в силу леммы 1 из [2].

2. Севе́дение общего случая к двудольному графу

Покажем, что задачу (1)–(3) можно свести к эквивалентной задаче, граф которой является двудольным.

Положим $c_i = \min\{d_j \mid j \in N_i\}$ при $i \in V$. Множество вершин V графа G разобьём на три подмножества:

$$V_1 = \{i \in V \mid d_i < c_i\}, \quad V_2 = \bigcup_{i \in V_1} N_i, \quad V_3 = V \setminus (V_1 \cup V_2). \tag{4}$$

Вершины из V_1 соответствуют потребителям с меньшими чем у соседей ценовыми порогами. Множество V_2 состоит из вершин, смежных с вершинами из V_1. Все остальные вершины включаются в V_3. Нетрудно видеть, что вершины из V_1 не смежны между собой. Для любых двух смежных вершин $i \in V_1$ и $j \in V_2 \cup V_3$ верно неравенство $d_i < d_j$.

В множестве рёбер E графа G выделим подмножество

$$E_{12} = \Big\{ \{i, j\} \in E \mid i \in V_1,\ j \in V_2 \Big\}.$$

Лемма 1. *Пусть* $x^* = (x^*_{ij})$ *— оптимальное решение задачи* (1)–(3). *Тогда вектор* $x = (x_{ij})$ *с компонентами*

$$x_{ij} = x^*_{ij}, \quad \{i, j\} \in E_{12},$$
$$x_{ij} = \max\{d_i,\ d_j\}, \quad \{i, j\} \in E \setminus E_{12}, \tag{5}$$

также является оптимальным решением задачи (1)–(3).

Доказательство. Покажем, что вектор x является допустимым решением задачи (1)–(3). Ограничения (3), очевидно, выполняются. Пусть i — произвольная вершина из V_1. Поскольку x^* — допустимое решение, найдётся вершина $j \in V_2$ такая, что $x^*_{ij} = d_i$. Следовательно, $x_{ij} = d_i$. Поэтому $\varphi_i(x) = d_i$, т. е. ограничения (2) выполняются при любом $i \in V_1$. Пусть $i \in V_2 \cup V_3$. По построению разбиения (4) найдётся вершина $j \in V$ такая, что $d_i \geqslant d_j$. Поэтому $\varphi_i(x) \leqslant x_{ij} \leqslant d_i$ и вектор x удовлетворяет условиям (2) и (3).

Поскольку $x_{ij} \geqslant x^*_{ij}$ при каждом $\{i, j\} \in E$, $\Phi(x) \geqslant \Phi(x^*)$. Так как x^* — оптимальное решение, то $\Phi(x) = \Phi(x^*)$ и вектор x оптимален. Лемма 1 доказана.

В силу леммы 1 при решении задачи (1)–(3) можно ограничиться векторами x, удовлетворяющими (2), (3) и (5). Условия (2) достаточно рассматривать как равенства $\varphi_i(x) = d_i,\ i \in V_1$. Выражение (1) принимает вид

$$\Phi(x) = \sum_{i \in V_1 \cup V_3} b_i d_i + \sum_{i \in V_2} b_i \varphi_i(x),$$

где вторая сумма является переменной величиной. Удаляя из задачи (1)–(3) константы и несущественные ограничения, приходим к следующей эквивалентной формулировке.

Задан двудольный граф $G = (V_1, V_2, E)$ с долевыми множествами вершин V_1 и V_2, где $|V_1| + |V_2| = m$, и множеством рёбер E, $|E| = n$. Ценовые пороги удовлетворяют неравенствам

$$d_i < d_j, \quad i \in V_1,\ j \in V_2,\ \{i, j\} \in E.$$

Требуется найти максимум функции

$$\Phi(x) = \sum_{i \in V_2} b_i \varphi_i(x) \tag{6}$$

при условиях

$$\varphi_i(x) = d_i, \quad i \in V_1, \tag{7}$$

$$x_{ij} \in \{d_i,\ d_j\}, \quad \{i, j\} \in E. \tag{8}$$

Далее задача (6)–(8) рассматривается как основная. Она выглядит проще задачи (1)–(3), но имеет ту же вычислительную сложность, что и задача о выборе цен в общей постановке [2].

Лемма 2. *Задача* (6)–(8) *является* NP-*трудной.*

Доказательство. Покажем, что к задаче (6)–(8) сводится следующая NP-трудная задача о покрытии множества подмножествами [1].

Пусть заданы множество $K = \{1, \dots, k\}$ и семейство его непустых подмножеств $K_1, \dots, K_s$, объединение которых равно K. Множество $P = \{K_j \mid j \in L\}$, где $L \subseteq \{1, \dots, s\}$, называется *покрытием* множества K, если объединение множеств K_j, $j \in L$, равно K. Требуется найти покрытие P^* минимальной мощности.

Рассмотрим двудольный граф $G = (V_1, V_2, E)$ с долями $V_1 = K$ и $V_2 = \{1, \dots, s\}$. Множество рёбер E состоит из пар $\{i, j\}$ таких, что $i \in V_1$, $j \in V_2$ и $i \in K_j$ (таким образом, окрестность N_j вершины $j \in V_2$ равна K_j). Объём спроса b_j полагается равным 1 при любом $j \in V_2$. Ценовой порог d_i равен 1, если $i \in V_1$, и равен 2, если $i \in V_2$.

Пусть $x^* = (x^*_{ij})$ — оптимальное решение задачи (6)–(8) при вышеописанных исходных данных. Положим $L^* = \{j \in V_2 \mid \varphi_j(x^*) = 1\}$. С учётом принятых обозначений получаем

$$\Phi(x^*) = 2(s - |L^*|) + |L^*| = 2s - |L^*|;$$

при этом $P^* = \{K_j \mid j \in L^*\}$ является покрытием множества K в силу ограничений (7).

Пусть $P = \{K_j \mid j \in L\}$ — произвольное покрытие множества K. Рассмотрим вектор $x = (x_{ij})$, где $i \in N_j$ и $j \in V_2$, с компонентами $x_{ij} = 1$, если $j \in L$, и $x_{ij} = 2$, если $j \in V_2 \setminus L$. Ограничения (7) выполняются, поскольку P — покрытие множества $K = V_1$. Вычисляя разность между $\Phi(x^*)$ и $\Phi(x) = 2s - |L|$, получаем

$$\Phi(x^*) - \Phi(x) = |L| - |L^*| \geqslant 0,$$

так как $\Phi(x^*) \geqslant \Phi(x)$. В силу произвольности P покрытие P^* имеет минимальную мощность. Искомое сведение получено. Лемма 2 доказана.

Пусть вектор $x = (x_{ij})$ удовлетворяет условию (8). Будем говорить, что вершина $j \in V_2$ *покрывает* смежную с ней вершину $i \in V_1$, если $x_{ij} = d_i$, и *не покрывает* вершину i, если $x_{ij} = d_j$.

В следующих двух леммах выясняются используемые ниже свойства оптимальных решений.

Лемма 3. *В задаче* (6)–(8) *найдётся такое оптимальное решение* $x^* = (x^*_{ij})$, *что для каждой вершины* $j \in V_2$ *имеется такое число* c_j, *что* $x^*_{ij} = d_i$ *при* $i \in N_j$ *и* $d_i \geqslant c_j$ *и* $x^*_{ij} = d_j$ *при* $i \in N_j$ *и* $d_i < c_j$. *Иными словами, вершина* j *покрывает все смежные с ней вершины с ценовыми порогами не менее* c_j *и не покрывает все остальные смежные с ней вершины.*

Доказательство. Пусть $x' = (x'_{ij})$ — некоторое оптимальное решение задачи (6)–(8), и пусть j — произвольная вершина из V_2. Если вершина j не покрывает ни одну смежную с ней вершину из N_j, то в качестве c_j выбираем любое число, удовлетворяющее неравенству

$$c_j > \max\{d_i \mid i \in N_j\}.$$

В противном случае полагаем $c_j = \min\{d_i \mid i \in N_j,\ x'_{ij} = d_i\}$.

Рассмотрим вектор $x^* = (x^*_{ij})$, где $i \in N_j$ и $j \in V_2$, с компонентами $x^*_{ij} = d_i$, если $d_i \geqslant c_j$, и $x^*_{ij} = d_j$, если $d_i < c_j$. Вектор x^* является искомым, поскольку ограничения (7), (8) выполняются и $\Phi(x^*) = \Phi(x')$. Лемма 3 доказана.

Лемма 4. *Имеется такое оптимальное решение задачи* (6)–(8), *что каждая вершина из* V_1 *покрывается ровно одной вершиной из* V_2.

Доказательство. Пусть $x' = (x'_{ij})$ — некоторое оптимальное решение задачи (6)–(8). Каждой вершине $i \in V_1$ поставим в соответствие такую вершину $j_i \in V_2$, которая покрывает i. Рассмотрим вектор $x^* = (x^*_{ij})$, где $i \in N_j$ и $j \in V_2$, с компонентами $x^*_{ij} = d_i$, если $j = j_i$, и $x^*_{ij} = d_j$, если $j \neq j_i$. Этот вектор является искомым, поскольку ограничения (7), (8) выполняются и $\Phi(x^*) = \Phi(x')$. Лемма 4 доказана.

3. Некоторые полиномиально разрешимые случаи задачи (6)–(8)

Теорема 1. *Если граф* $G = (V_1, V_2, E)$ *является деревом, то задача* (6)–(8) *решается методом динамического программирования с полиномиальной сложностью.*

Доказательство. Возьмём произвольную вершину $q \in V_1$. Дерево G будем рассматривать как корневое с корнем в вершине q. Рёбра графа G заменим дугами, ориентируя их в направлении «от корня».

Пусть i — не корневая вершина и вершина p — её предок, т. е. $(p, i) \in E$. Через $f_i(d)$, $d \in \{d_i, d_p\}$, обозначим оптимум задачи (6)–(8), рассматриваемой на поддереве с корнем в вершине i, при условии, что значение переменной x_{ip} равно d. Через f_q обозначим оптимум исходной задачи (6)–(8).

Перейдём к вычислению величин $f_i(d)$ и f_q при прямом ходе динамического программирования. Для висячих вершин $i \in V$ дерева G естественно принять следующие значения:

$$f_i(d_i) = 0, \quad f_i(d_p) = -\infty, \ i \in V_1,$$
$$f_i(d) = b_i d, \quad d \in \{d_i, d_p\}, \ i \in V_2.$$

Пусть вершина i не является висячей или корневой вершиной в дереве G и пусть p — её предок. Через $M_i = N_i \setminus p$ обозначим множество потомков вершины i. Рассмотрим отдельно два случая: 1) $i \in V_1$; 2) $i \in V_2$.

1) Пусть $i \in V_1$. Рассмотрим величину $f_i(d_i)$. Поскольку вершина i покрывается вершиной $p \in V_2$, в силу леммы 4 можно считать, что вершины из M_i не покрывают i. Поэтому

$$f_i(d_i) = \sum_{j \in M_i} f_j(d_j), \quad i \in V_1. \tag{9}$$

Рассмотрим величину $f_i(d_p)$. Вершина p не покрывает вершину i. По лемме 4 достаточно перебрать ситуации, когда ровно одна вершина из M_i покрывает вершину i. Следовательно,

$$f_i(d_p) = \max_{j \in M_i} \Big\{ f_j(d_i) + \sum_{k \in M_i \setminus j} f_k(d_k) \Big\}, \quad i \in V_1. \tag{10}$$

2) Пусть $i \in V_2$. Рассмотрим величину $f_i(d)$, $d \in \{d_i, d_p\}$, и соответствующую задачу. Если в оптимальном решении вершина i не покрывает ни одной вершины из M_i, то оптимум в этом случае (обозначим его через $g_i(d)$) равен $g_i(d) = b_i d + \sum_{j \in M_i} f_j(d_i)$. Пусть вершина i покрывает некоторые вершины из M_i. Для краткости записи положим

$$M_i(t) = \{j \in M_i \mid d_j < t\}$$

и обозначим через $h_i(d)$ соответствующий оптимум. С учётом леммы 3 получаем

$$h_i(d) = \max_{j \in M_i} \Big\{ b_i \min\{d,\ d_j\} + \sum_{k \in M_i(d_j)} f_k(d_i) + \sum_{k \in M_i \setminus M_i(d_j)} f_k(d_k) \Big\}.$$

Поэтому

$$f_i(d) = \max\{g_i(d),\ h_i(d)\}, \quad d \in \{d_i, d_p\},\ i \in V_2. \tag{11}$$

Прямой ход динамического программирования заканчивается вычислением величины f_q — оптимума исходной задачи (6)–(8). Поскольку $q \in V_1$, аналогично формуле (10) получаем

$$f_q = \max_{j \in N_q} \Big\{ f_j(d_q) + \sum_{k \in N_q \setminus j} f_k(d_k) \Big\}. \tag{12}$$

Сложность вычислений по рекуррентным соотношениям (9)–(12) не превосходит $O(m^3)$ арифметических операций; при этом требуются $O(m)$ ячеек памяти, где m — число вершин дерева G. Теорема 1 доказана.

Вернёмся к общей задаче (6)–(8) и рассмотрим ещё два случая её полиномиальной разрешимости, представленных теоремами 2 и 3.

Пусть $M_1, \ldots, M_s$ — набор некоторых множеств. Будем говорить, что этот набор имеет *древовидную структуру*, если выполняется условие

$$M_k \cap M_l \in \{\varnothing,\ M_k,\ M_l\}, \quad 1 \leqslant k, l \leqslant s,$$

т. е. любые два множества либо не пересекаются, либо в одном из них содержится другое.

Теорема 2. *Если набор* $\{N_j \mid j \in V_2\}$ *окрестностей вершин графа* $G = (V_1, V_2, E)$ *имеет древовидную структуру, то задача* (6)–(8) *решается методом динамического программирования с полиномиальной сложностью.*

Доказательство. Через $M_1, \ldots, M_s$ обозначим все различные окрестности среди $\{N_j \mid j \in V_2\}$ и через V_{2k} — множество вершин из V_2, имеющих одну и ту же окрестность M_k, т. е.

$$V_{2k} = \{j \in V_2 \mid N_j = M_k\}, \quad k = 1, \ldots, s.$$

Ниже потребуется следующая

Лемма 5. *Задача* (6)–(8) *имеет такое оптимальное решение, что в*

каждом из множеств V_{2k}, $1 \leqslant k \leqslant s$, не более одной вершины участвует в покрытии и эта вершина покрывает все смежные с ней вершины, ценовые пороги которых не менее некоторой величины c_k.

Доказательство. Пусть $x' = (x'_{ij})$ — некоторое оптимальное решение задачи (6)–(8). Через K обозначим список индексов $k = 1, \dots, s$ таких, что хотя бы одна вершина из V_{2k} участвует в покрытии. Если $k \in K$, то полагаем

$$c_k = \min\{d_i \mid x'_{ij} = d_i,\ i \in M_k,\ j \in V_{2k}\} \tag{13}$$

и обозначим через j_k вершину из V_{2k}, на которой достигается минимум (13).

Рассмотрим следующий вектор $x^* = (x^*_{ij})$. Если $i \in M_k$, $k \in K$, $j = j_k$ и $d_i \geqslant c_k$, то $x^*_{ij} = d_i$. Во всех остальных случаях $x^*_{ij} = \max\{d_i,\ d_j\}$. Вектор x^* искомый, поскольку он удовлетворяет (7), (8) и $\Phi(x^*) = \Phi(x')$. Лемма 5 доказана.

Продолжим описание конструкций, необходимых для построения схемы динамического программирования.

Среди множеств $M_1, \dots, M_s$ выберем максимальное по мощности. Пусть это будет M_s. Достаточно рассмотреть случай, когда $M_s = V_1$. В противном случае задача (6)–(8) распадается на независимые задачи меньшей размерности. Вершины V_1 графа G занумеруем числами $1, \dots, m_1$ по возрастанию величин d_i, т. е. $d_1 \leqslant \ldots \leqslant d_{m_1}$.

Отношение включения между окрестностями $M_1, \dots, M_s$ опишем с помощью ориентированного дерева $T = (S, D)$ с множеством вершин $S = \{1, \dots, s\}$ и множеством дуг D. Дерево T рассматривается как корневое дерево с корнем s. Дуга (k, l) лежит в D, если окрестность M_l содержится в M_k и является максимальной по включению среди окрестностей, содержащихся в M_k.

В соответствии с методом динамического программирования и с учётом леммы 5 введём семейство задач, аналогичных задаче (6)–(8). Через $f_k(t)$, где $k = 1, \dots, s$ и $t = 0, 1, \dots, m_1$, обозначим оптимум «усеченной» задачи (6)–(8), рассматриваемой на подграфе $G' = (V'_1, V'_2, E')$ графа $G = (V_1, V_2, E)$, где $V'_1 = M_k$, а множество V'_2 состоит из тех вершин множества V_2, окрестности которых содержатся в множестве M_k или совпадают с ним. В ограничениях (7) индекс i пробегает множество $L_{kt} = M_k \cap [1, t]$. В частности, при $t = 0$ множество L_{k0} пусто и ограничений (7) нет.

Перейдём к составлению рекуррентных соотношений для вычисления величины $f_s(m_1)$ — оптимума исходной задачи (6)–(8), используя введённое выше дерево T.

Пусть k — висячая вершина в дереве T. По определению величины $f_k(t)$ и с учётом леммы 5 получаем следующие выражения.

Если $L_{kt} = \varnothing$, то

$$f_k(t) = \sum_{j \in V_{2k}} b_j d_j. \tag{14}$$

Если $L_{kt} \neq \varnothing$, то

$$f_k(t) = \max_{q \in V_{2k}} \Big\{ b_q \min_{i \in L_{kt}} d_i + \sum_{j \in V_{2k} \setminus q} b_j d_j \Big\}. \tag{15}$$

Пусть вершина k в дереве T не является висячей. Если $L_{kt} = \varnothing$, то по определению величины $f_k(t)$ получаем

$$f_k(t) = \sum_{j \in V_{2k}} b_j d_j + \sum_{l \in P_k} f_l(t). \tag{16}$$

Здесь и ниже $P_k = \{l \in S \mid (k,l) \in D\}$ — множество потомков вершины k дерева T.

Пусть $L_{kt} \neq \varnothing$. Через M'_k обозначим множество тех вершин из M_k, которые не лежат ни в одной из окрестностей, содержащихся в M_k, и положим

$$L'_{kt} = M'_k \cap [1, t].$$

Возможны два случая: 1) $L'_{kt} = \varnothing$ и 2) $L'_{kt} \neq \varnothing$.

1) Пусть $L'_{kt} = \varnothing$. Через $g_k(t)$ и $h_k(t)$ обозначим оптимумы рассматриваемой задачи соответственно при условиях, когда вершины из V_{2k} участвуют и не участвуют в покрытии. Обычными рассуждениями динамического программирования и с учетом леммы 5 получаем

$$g_k(t) = \max_{q \in V_{2k}} \max_{r \in L_{kt}} \Big\{ b_q d_r + \sum_{j \in V_{2k} \setminus q} b_j d_j + \sum_{l \in P_k} f_l(r-1) \Big\}, \tag{17}$$

$$h_k(t) = \sum_{j \in V_{2k}} b_j d_j + \sum_{l \in P_k} f_l(t). \tag{18}$$

В этом случае

$$f_k(t) = \max\{g_k(t),\ h_k(t)\}. \tag{19}$$

2) Пусть $L'_{kt} \neq \varnothing$. Через l_{kt} обозначим минимальный элемент из L'_{kt}. Поскольку вершины из L'_{kt} покрываются вершинами из V_{2k}, аналогично (17) получаем

$$f_k(t) = \max_{q \in V_{2k}} \max_{r \in L_{kt} \cap [1, l_{kt}]} \Big\{ b_q d_r + \sum_{j \in V_{2k} \setminus q} b_j d_j + \sum_{l \in P_k} f_l(r-1) \Big\}. \tag{20}$$

Вычисления по схеме (14)–(20) заканчиваются нахождением величины $f_s(m_1)$ — оптимума исходной задачи (6)–(8). Вычислительная сложность схемы равна $O(m^5)$ арифметических операций при объёме памяти $O(m^2)$, где m — число вершин графа G. Теорема 2 доказана.

Теорема 3. *Если набор $\{N_i \mid i \in V_1\}$ окрестностей вершин графа $G = (V_1, V_2, E)$ имеет древовидную структуру, то задача (6)–(8) решается методом динамического программирования с полиномиальной сложностью.*

Доказательство. Опишем ряд преобразований задачи (6)–(8), приводящих её к виду, удобному для применения динамического программирования.

1) Через $M_1, \dots, M_s$ обозначим все различные окрестности среди $\{N_i \mid i \in V_1\}$ и положим $V_{1k} = \{i \in V_1 \mid N_i = M_k\}, \quad k = 1, \dots, s$. По лемме 3 в оптимальном решении задачи (6)–(8) все вершины множества V_{1k} покрываются одной вершиной из M_k, которая покрывает вершину $i^* \in V_{1k}$ с минимальным ценовым порогом $d_{i^*} = \min\{d_i \mid i \in V_{1k}\}$. Удаление вершин из $V_{1k} \setminus i^*$ не влияет на оптимум задачи. Поэтому далее будем предполагать, что все окрестности $\{N_i \mid i \in V_1\}$ различны и каждой окрестности N_i соответствует ровно одна вершина i.

2) Рассмотрим две вершины i и j из V_1 такие, что $N_i \supset N_j$ и $d_i \geqslant d_j$. По лемме 3 в оптимальном решении вершина i покрывается вершиной из N_j, которая покрывает вершину j. Удаление вершины i не влияет на оптимум задачи. Поэтому далее предполагаем, что если $N_i \supset N_j$, то $d_i < d_j$.

3) Пусть окрестность N_{m_1} имеет наибольшую мощность среди $\{N_i \mid i \in V_1\}$ (здесь $m_1 = |V_1|$). Если $N_{m_1} \subset V_2$, то с учетом условия доказываемой теоремы задача (6)–(8) распадается на независимые задачи меньшей размерности. Поэтому далее предполагаем, что $N_{m_1} = V_2$.

4) Рассмотрим произвольную окрестность N_i, где $i \in V_1$, которая не является минимальной по включению среди окрестностей $N_1, \dots, N_{m_1}$. Через L_i обозначим множество тех вершин из N_i, которые не принадлежат объединению окрестностей, содержащихся в N_i. Если $L_i \neq \varnothing$, то к множеству V_1 добавим новую вершину (обозначим ее, например, через r) с окрестностью $N_r = L_i$ и ценовым порогом $d_r = \min\{d_j \mid j \in L_i\}$. Очевидно, окрестность N_r является минимальной по включению и, таким образом, каждая вершина из V_2 находится в некоторой минимальной по включению окрестности вершины из V_1.

Последнее преобразование не влияет на оптимум задачи, не нарушает свойств, полученных в результате предыдущих преобразований, и

позволяет более компактно изложить расчётную схему динамического программирования.

Для описания этой схемы введём ориентированное дерево $T = (V_1, D)$ с множеством дуг D и множеством вершин V_1, которые ассоциируются с вершинами первой доли графа G и их окрестностями. Дерево T описывает включение окрестностей $N_1, \ldots, N_{m_1}$ и рассматривается как корневое с корнем m_1. Дуга (i, j) лежит в D, если $N_i \supset N_j$ и окрестность N_j является максимальной по включению среди окрестностей, содержащихся в N_i.

Через T_i обозначим поддерево с корнем $i \in V_1$ и через f_i — оптимум задачи (6)–(8), рассматриваемой относительно дерева T_i. Перейдём к вычислению величин f_i, $i \in V_1$, среди которых f_{m_1} — оптимум исходной задачи.

Пусть вершина $i \in V_1$ дерева T является висячей. По определению получаем
$$f_i = \max_{j \in N_i} \Big\{ b_j d_i + \sum_{k \in N_i \setminus j} b_k d_k \Big\}. \tag{21}$$

Пусть вершина $i \in V_1$ дерева T не является висячей. Через W_i обозначим множество висячих вершин дерева T_i, через P_{ij} — множество вершин дерева T_i, принадлежащих пути из корневой вершины i в висячую вершину $j \in W_i$, и через Q_{ij} — множество вершин дерева T_i, смежных с вершинами пути P_{ij}, но не принадлежащих ему. Отметим, что если из дерева T_i удалить вершины P_{ij}, то вершины из Q_{ij} окажутся корневыми в оставшихся поддеревьях.

В силу леммы 3 и пункта 2) справедливо следующее утверждение. Если в оптимальном решении рассматриваемой подзадачи вершина i покрывается некоторой вершиной $k \in N_j$, где $j \in W_i$, то все вершины из пути P_{ij}, содержащиеся в первой доле двудольного графа подзадачи, покрываются вершиной k. Пользуясь этим фактом и древесной структурой задачи, получаем
$$f_i = \max_{j \in W_i} \max_{k \in N_j} \Big\{ b_k d_i + \sum_{l \in N_j \setminus k} b_l d_l + \sum_{q \in Q_{ij}} f_q \Big\}. \tag{22}$$

Вычисления по схеме (21), (22) требуют $O(m^4)$ арифметических операций и $O(m)$ ячеек памяти, где m — число вершин графа G. Теорема 3 доказана.

ЛИТЕРАТУРА

1. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

2. **Дементьев В. Т., Шамардин Ю. В.** Задача о выборе цен на продукцию при условии обязательного удовлетворения спроса // Дискрет. анализ и исслед. операций. Сер. 2. 2002. Т. 9, № 2. С. 31–40.

Адрес автора:

Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск,
Россия.
E-mail: orlab@math.nsc.ru

Статья поступила
18 сентября 2005 г.