ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Ноябрь—декабрь 2012. Том 19, № 6. С. 9–22

УДК 519.7

ЭФФЕКТИВНЫЙ АЛГОРИТМ РЕШЕНИЯ ДВУХЭТАПНОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ НА ДРЕВОВИДНОЙ СЕТИ *)

Э. Х. Гимади, А. А. Курочкин

Аннотация. Рассматривается двухэтапная задача размещения производства на древовидной сети при условии, что затраты на транспортировку единицы продукции из пункта в пункт равны сумме длин рёбер в цепи, соединяющей эти пункты. Предложен алгоритм для точного решения данной задачи с трудоёмкостью $O(nm^3)$, где n — число пунктов спроса конечного продукта, m — ограничение сверху на число возможных пунктов размещения производства каждого этапа.

Ключевые слова: двухэтапная задача размещения производства, полиномиальный алгоритм, древовидная сеть.

Введение

Класс многоэтапных задач размещения характеризуется наличием нескольких этапов производства, на которых осуществляется обработка сырья, прежде чем готовый продукт поступает конечному потребителю. Классическим примером двухэтапной задачи размещения производства является добыча и переработка нефти, когда сырая нефть из скважины поставляется сначала на нефтеперерабатывающий завод, а уже после готовый продукт поступает на автозаправочные станции, которые в данном случае выступают в роли конечных потребителей.

Как известно, в общем случае задача размещения NP-трудна даже в классической одноэтапной постановке [1, 5]. Поэтому для многоэтапной задачи размещения производства представляется целесообразным проведение исследований в следующих двух направлениях.

1. *Поиск специальных подклассов задачи, для решения которых возможно построение точных алгоритмов полиномиальной сложности.*

*)Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 12–01–00093а, 10–07–00195а и 12–01–33028мол_а_вед), целевой программы Президиума РАН (проект № 227) и междисциплинарного интеграционного проекта СО РАН (№ 7Б).

В [2, 4] для метрической k-этапной задачи размещения предприятий на цепи построен алгоритм с трудоёмкостью $O\left(n^3 \sum\limits_{r=1}^{k} m_r\right)$, где n — число пунктов спроса конечного продукта, m_r — число возможных мест открытия предприятий на r-м этапе. Алгоритм полиномиален при константном числе этапов. Для метрической двухэтапной задачи на цепи в [2, 4] представлен алгоритм с временно́й сложностью $O(n\, m_1\, m_2)$ или $O(nm^2)$, где m — ограничение сверху на число возможных пунктов размещения производства каждого этапа. В [4] анонсирована идея построения точного алгоритма решения двухэтапной задачи размещения на древовидной сети за время $O(nm^3)$ без представления соответствующего обоснования.

2. *Построение приближённых алгоритмов решения с гарантированными оценками точности.* Например, в [3] для метрической k-этапной задачи размещения предприятий построен комбинаторный алгоритм с оценкой точности 3.27, что значительно улучшает предыдущий рекорд, равный 6 [7]. Для случаев $k = 2$ и $k = 3$ построены алгоритмы с оценками 2.4211 и 2.8446 соответственно. Позже для случая $k = 2$ в [6] с использованием техники жадных алгоритмов построен полиномиальный алгоритм с оценкой точности 1.77.

Ниже рассматривается двухэтапная задача размещения на сети в виде дерева при естественном условии, что затраты по транспортировке единицы продукта из пункта в пункт равны сумме длин рёбер в цепи, соединяющей эти пункты. Результатом настоящей статьи является строгое обоснование точного алгоритма A решения этой задачи с временно́й сложностью $O(nm^3)$.

1. Общая постановка и определения

Пусть $N = \{1, \ldots, n\}$ — множество пунктов спроса конечного продукта, а $M_r \subset N$ — множество возможных пунктов размещения предприятий r-го этапа, $r = 1, 2$. Известны затраты $g_i^r \geqslant 0$ на размещение (открытие) предприятия r-го этапа в пункте $i \in M_r$. Для каждого пункта потребления $j \in N$ заданы объём спроса $b_j \geqslant 0$ и транспортные затраты $c_{ij} \geqslant 0$, связанные с доставкой единицы продукции из пункта i в j. Для удовлетворения спроса пункта $j \in N$ необходимый объём продукта b_j должен пройти по следующей цепочке: открытое предприятие 2-го уровня $\to$ открытое предприятие 1-го уровня $\to$ пункт j.

Предполагается, что каждые пункт спроса конечной продукции и пункт производства любого этапа получают продукцию только от одного производителя, при этом предприятие 1-го этапа получает продукцию

от предприятия 2-го этапа. Всюду далее будем говорить, что предприятие $i \in M_r,\ r = 1, 2$, *участвует в обслуживании* пункта спроса $j \in N$, если i задействовано в цепочке предприятий, через которые j получает конечный продукт. Отметим также, что в одном пункте могут находиться одновременно предприятия 1-го и 2-го этапов.

Требуется выбрать подмножества пунктов размещения предприятий каждого этапа $I^r \subseteq M_r,\ r = 1, 2$, и осуществить назначение выбранных предприятий на пункты спроса так, чтобы минимизировать суммарные затраты на открытие всех выбранных предприятий и на транспортировку продукта от открытых предприятий к надлежащим пунктам спроса.

Дадим общую математическую постановку двухэтапной задачи размещения. Необходимо минимизировать

$$\sum_{i \in M_1} g_i^1 x_i + \sum_{k \in M_2} g_k^2 y_k + \sum_{j \in N} b_j \sum_{k \in M_2} \sum_{i \in M_1} (c_{ki} + c_{ij}) x_{kij} \longrightarrow \min$$

по переменным x_i, y_k, x_{kij} при заданных ограничениях:

$$\sum_{k \in M_2} \sum_{i \in M_1} x_{kij} = 1, \quad j \in N,$$

$$\sum_{k \in M_2} x_{kij} \leqslant x_i, \quad j \in N,\ i \in M_1,$$

$$\sum_{i \in M_1} x_{kij} \leqslant y_k, \quad j \in N,\ k \in M_2,$$

$$x_i, y_k, x_{kij} \in \{0, 1\},$$

где x_i, y_k, $i \in M_1$, $k \in M_2$, — переменные выбора предприятий 1-го и 2-го этапа соответственно (если $x_i = 1$, то в соответствующем решении предприятие 1-го этапа $i \in M_1$ открыто), x_{kij}, $k \in M_2$, $i \in M_1$, $j \in N$, — транспортные переменные, задающие предприятия, участвующие в обслуживании пункта спроса j (если $x_{kij} = 1$, то в соответствующем решении пункт спроса j получает продукт от предприятия 1-го этапа $i \in M_1$, которое в свою очередь получает продукт от предприятия 2-го этапа $k \in M_2$).

Дадим формулировку задачи в терминах переменных назначения. Для этого понадобится ввести несколько дополнительных определений:

$\pi^r = \left(\pi_1^r, \ldots, \pi_n^r\right)$ — вектор назначения предприятий r-го этапа, где π_j^r — номер пункта из M_r, в котором размещено (открыто) предприятие r-го этапа, обслуживающее пункт спроса j, $r = 1, 2$, $1 \leqslant j \leqslant n$;

$\pi = (\pi^1, \pi^2)$ — пара векторов назначения, которую также будем называть решением задачи;

$I^r(\pi) = \bigcup\limits_{j \in N} \{\pi_j^r\}$ — множество предприятий r-го этапа, задействованных (открытых) в решении $\pi, r = 1, 2$.

Двухэтапная задача размещения в терминах переменных назначения записывается в более компактном виде:

$$\sum_{i \in I^1(\pi)} g_i^1 + \sum_{k \in I^2(\pi)} g_k^2 + \sum_{j \in N} b_j (c_{\pi_j^2 \pi_j^1} + c_{\pi_j^1 j}) \longrightarrow \min_{\pi}.$$

В работе рассматриваем двухэтапную задачу размещения на древовидной сети. Более конкретно, рассматриваем задачи, в которых матрице транспортных затрат (c_{ij}) поставлена в соответствие ациклическая сеть $G = (N, E)$, где $N = \{1, \ldots, n\}$ и $E = \{e_k \mid 1 \leqslant k < n\}$ — множества вершин и рёбер соответственно. Вершины в сети соответствуют пунктам спроса. Стоимость транспортировки единицы продукта c_{ij} из пункта i в пункт j определяем как сумму длин рёбер в цепи, соединяющей эти пункты. Отметим также, что $c_{ij} = c_{ji}$ и $c_{ik} \leqslant c_{ij} + c_{jk}$ для любых $i, j, k \in N$.

Для каждого $j \in N$ обозначим через N_j множество потомков вершины j (максимальное поддерево исходного дерева с корневой вершиной j), $I_j^r(\pi) = \bigcup \{\pi_k^r \mid k \in N_j\}$ множество предприятий r-го этапа, обслуживающих клиентов множества N_j в назначении π.

Индуктивно введём специальное обозначение $\mu_j(\pi)$ для самого близкого к вершине $j \in N$ предприятия второго этапа, задействованного в назначении π (отметим, что это предприятие обязательно является открытым): для корня дерева $j = 1$ положим $\mu_1(\pi) = \arg\min\limits_{k \in I^2(\pi)} c_{1k}$, для вершины j, $1 < j \leqslant n$, положим

$$\mu_j(\pi) = \begin{cases} \mu_i(\pi), & \text{если } c_{j\mu_i(\pi)} = \min\limits_{k \in I^2(\pi)} c_{jk}, \text{ где } i \text{ — отец } j, \\ \arg\min\limits_{k \in I^2(\pi)} c_{jk} & \text{в противном случае.} \end{cases}$$

Иногда для удобства будем опускать π в обозначении $\mu_j(\pi)$ и писать просто μ_j, если понятно, о каком именно назначении идёт речь.

Замечание 1. Если вершина j — сын вершины i и $\mu_j \neq \mu_i$ в назначении π, то $\mu_j \in N_i$.

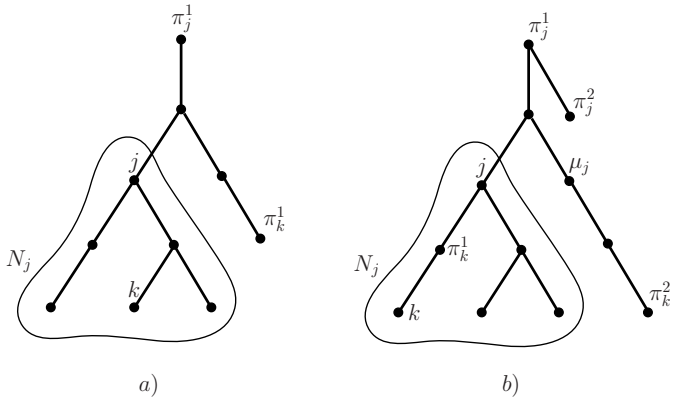
Доказательство. Допустим напротив, что $\mu_j \notin N_i$. В силу определения μ_j и условия $\mu_j \neq \mu_i$ имеем $c_{j\mu_j} < c_{j\mu_i}$. С другой стороны,

поскольку $j \in N_i$ и $\mu_j \notin N_i$, получаем

$$c_{j\mu_j} = c_{ji} + c_{i\mu_j} \geqslant c_{ji} + c_{i\mu_i} \geqslant c_{j\mu_i};$$

противоречие. Следовательно, исходное утверждение верно.

Введём дополнительные определения. Пусть дано некоторое решение двухэтапной задачи размещения на сети π. Для каждой вершины j *плохими вершинами первого этапа* будем называть вершины $k \in N_j$, для которых $\pi^1_k \notin N_j \cup \{\pi^1_j\}$. *Плохими вершинами второго этапа для* j будем называть вершины $k \in N_j$, для которых $\pi^2_k \notin N_j \cup \{\pi^2_j\} \cup \mu_j$. Наглядная иллюстрация данных определений приведена на рис. 1. Вершины, для которых выполнены обратные включения, будем называть *хорошими вершинами первого этапа для* j и *хорошими вершинами второго этапа для* j соответственно. Будем называть вершину k *плохой для* j, если она плохая вершина первого или второго этапа для j. Будем называть вершину k *плохой*, если она плохая для какой-либо вершины дерева. Вершину k будем называть *хорошей вершиной для* j, если она хорошая вершина первого и второго этапа для j. Вершину k будем называть *хорошей*, если она хорошая для всех вершин дерева. Обозначим количество плохих вершин первого и второго этапа для вершины j через $\nu^1_j(\pi)$ и $\nu^2_j(\pi)$ соответственно. Положим $\nu_j(\pi) = \nu^1_j(\pi) + \nu^2_j(\pi)$. Величину $\nu(\pi) = \sum\limits_{j \in N} \nu_j(\pi)$ будем называть *индексом* решения π.



a) *b)*

Рис. 1. a) k — плохая вершина 1-го этапа для j; b) k — хорошая вершина 1-го этапа и плохая вершина 2-го этапа для j

2. Основные утверждения

Лемма 1. *Если в оптимальном решении π вершина k хорошая для j, а j хорошая для i, то k хорошая для i.*

Доказательство. Докажем два свойства, из которых следует утверждение леммы.

(i) Если k — хорошая вершина 1-го этапа для j, j — хорошая вершина 1-го этапа для i, то k — хорошая вершина 1-го этапа для i. Другими словами, имеем $\pi_k^1 \in N_j \cup \{\pi_j^1\}$, $\pi_j^1 \in N_i \cup \{\pi_i^1\}$. Следовательно, либо $\pi_k^1 \in N_j \subset N_i$, либо $\pi_k^1 = \pi_j^1 \in N_i \cup \{\pi_i^1\}$. Таким образом, $\pi_k^1 \in N_i \cup \{\pi_i^1\}$.

(ii) Если k — хорошая вершина 2-го этапа для j, j — хорошая вершина 2-го этапа для i, то k — хорошая вершина 2-го этапа для i. Другими словами, имеем $\pi_k^2 \in N_j \cup \{\pi_j^2\} \cup \mu_j$, $\pi_j^2 \in N_i \cup \{\pi_i^2\} \cup \mu_i$. Рассмотрим все случаи:

(ii.1) $\pi_k^2 \in N_j \subset N_i$ и k — хорошая вершина 2-го этапа для i;

(ii.2) $\pi_k^2 = \pi_j^2 \in N_i \cup \{\pi_i^2\} \cup \mu_i$ и k — хорошая 2-го этапа для i;

(ii.3) $\pi_k^2 = \mu_j$; в силу замечания 1 либо $\mu_j = \mu_i$, либо $\mu_j \in N_i$.

В любом случае получаем, что k — хорошая вершина 2-го этапа для i. Лемма 1 доказана.

Докажем фундаментальное свойство, которым обладают оптимальные решения задачи. Это свойство будет использовано для построения алгоритма поиска оптимального решения.

Теорема 1. *Существует оптимальное решение двухэтапной задачи размещения на сети, в котором для любой вершины $t \in N$ выполнены свойства* ($I1$):

$$I_t^1(\pi) \subset N_t \cup \{\pi_t^1\}, \quad I_t^2(\pi) \subset N_t \cup \{\pi_t^2\} \cup \mu_t(\pi).$$

Доказательство. Используя данные выше определения, теорему можно переформулировать следующим образом: существует оптимальное решение двухэтапной задачи размещения на сети с нулевым индексом, т. е. существует оптимальное решение π, для которого $\nu(\pi) = 0$.

Допустим, что теорема неверна, т. е. $\nu(\pi) > 0$ для любого оптимального решения π (в любом оптимальном решении найдётся вершина, для которой не выполняются заявленные в теореме включения). Из всех оптимальных решений выберем решение σ с минимальным индексом: $\nu(\sigma) = \min_\pi \nu(\pi) > 0$. Из всех плохих вершин, соответствующих решению σ, выберем ту, которая имеет минимальное расстояние до корня дерева. Обозначим её через k, а отца вершины k — через j. Вершина j, очевидно, является хорошей для всех остальных вершин, поскольку находится на меньшем расстоянии до корня дерева, чем k. Заметим, что k — плохая вершина для j. Действительно, если предположить противное

(k — хорошая вершина для j), то в силу замечания 1 вершина k хорошая для всех вершин, что противоречит нашему предположению.

Вершина k может быть плохой вершиной 1-го этапа или плохой вершиной 2-го этапа для j. Покажем, что в обоих случаях получим противоречие.

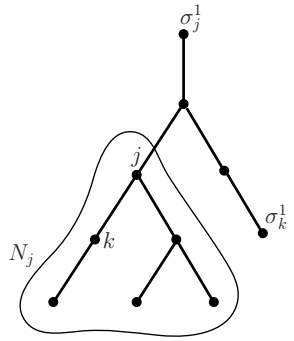


Рис. 2. Иллюстрация случая 1

Случай 1. Пусть k — плохая вершина 1-го этапа для j, т. е. $k \in N_j$, $\sigma_k^1 \notin N_j$ и $\sigma_k^1 \neq \sigma_j^1$. В силу оптимальности решения σ имеем

$$c_{\sigma_j^2\sigma_j^1} + c_{\sigma_j^1 j} \leqslant c_{\sigma_k^2\sigma_k^1} + c_{\sigma_k^1 j}, \quad c_{\sigma_k^2\sigma_k^1} + c_{\sigma_k^1 k} \leqslant c_{\sigma_j^2\sigma_j^1} + c_{\sigma_j^1 k}. \tag{1}$$

Действительно, если нарушается, например, первое неравенство, то

$$c_{\sigma_j^2\sigma_j^1} + c_{\sigma_j^1 j} > c_{\sigma_k^2\sigma_k^1} + c_{\sigma_k^1 j}.$$

Тогда в решении σ можно заменить цепочку предприятий, участвующих в обслуживании клиента j (σ_j^2 и σ_j^1 заменить на σ_k^2 и σ_k^1), что приведёт к уменьшению значения целевой функции исходной задачи за счёт снижения транспортных расходов на доставку продукта клиенту j. Другими словами, решение σ не будет оптимальным.

Поскольку $\sigma_k^1 \notin N_j$, а $j, k \in N_j$, последнее неравенство представимо в виде

$$c_{\sigma_k^2\sigma_k^1} + c_{\sigma_k^1 j} + c_{jk} \leqslant c_{\sigma_j^2\sigma_j^1} + c_{\sigma_j^1 k}. \tag{2}$$

Складывая неравенства (1) и (2), получаем

$$c_{\sigma_j^1 j} + c_{jk} \leqslant c_{\sigma_j^1 k}. \tag{3}$$

В то же время имеет место и обратное неравенство $c_{\sigma_j^1 j} + c_{jk} \geqslant c_{\sigma_j^1 k}$, так как граф представляет собой дерево. Следовательно, во всех неравенствах (1)–(3) должен стоять знак равенства. В частности, имеет место

$$c_{\sigma_j^2\sigma_j^1} + c_{\sigma_j^1 k} = c_{\sigma_k^2\sigma_k^1} + c_{\sigma_k^1 k}. \tag{4}$$

Рассмотрим назначение π, заданное следующим образом: $\pi_i^r = \sigma_i^r$ для $i \neq k,\ r = 1, 2$, и $\pi_k^r = \sigma_j^r,\ r = 1, 2$. Другими словами, назначения π и σ совпадают на всех вершинах, кроме k. Поэтому все вершины за исключением k, которые были хорошими (плохими) в решении σ, также остаются хорошими (плохими) в решении π. В силу равенства (4) стоимость решения, соответствующего назначению π, совпадает со стоимостью оптимального решения σ, т. е. решение π также оптимально. При этом в решении π вершина k хорошая для j, так как $\pi_k^r = \sigma_j^r = \pi_j^r,\ r = 1, 2$. Более того, так как j является отцом k, в силу замечания 1 вершина k хорошая и для всех остальных вершин дерева в решении π. Таким образом, новых плохих вершин в решении π не появилось (поскольку π совпадает с σ на всех вершинах кроме k, а k хорошая для всех вершин в решении π). В то же время число вершин, плохих для j, уменьшилось на одну и $\nu_j(\pi) < \nu_j(\sigma)$. Из последних двух предложений следует, что $\nu(\pi) < \nu(\sigma)$. Таким образом, решение π оптимально и имеет индекс меньше $\nu(\sigma)$, что приводит к противоречию с исходным предположением.

Случай 2. Пусть k — плохая вершина 2-го этапа для j, т. е. $k \in N_j$, $\sigma_k^2 \notin N_j$, $\sigma_k^2 \neq \sigma_j^2$ и $\sigma_k^2 \neq \mu_j$. Если к тому же k — плохая вершина 1-го этапа, то по доказанному в предыдущем пункте получим противоречие. Поэтому можно считать, что k — хорошая вершина 1-го этапа для j и выполнено одно из включений $\sigma_k^1 \in N_j$ или $\sigma_k^1 = \sigma_j^1 \notin N_j$. Рассмотрим оба варианта.

Подслучай 2.1: $\sigma_k^1 \in N_j$ (рис. 3(а)). В силу неравенства $c_{\mu_j j} \leqslant c_{\sigma_k^2 j}$, верного по определению μ_j, имеем

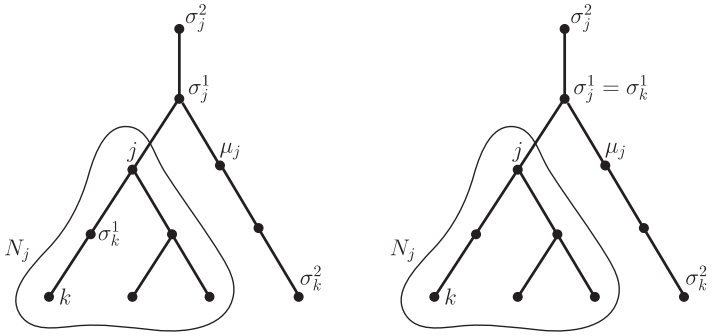
$$c_{\sigma_k^2 \sigma_k^1} = c_{\sigma_k^2 j} + c_{j\sigma_k^1} \geqslant c_{\mu_j j} + c_{j\sigma_k^1} \geqslant c_{\mu_j \sigma_k^1}.$$

Таким образом, в этом случае $c_{\mu_j \sigma_k^1} \leqslant c_{\sigma_k^2 \sigma_k^1}$. Добавив к обеим частям неравенства $c_{\sigma_k^1 k}$ получим

$$c_{\mu_j \sigma_k^1} + c_{\sigma_k^1 k} \leqslant c_{\sigma_k^2 \sigma_k^1} + c_{\sigma_k^1 k}. \tag{5}$$

Поэтому если в решении σ предприятие второго этапа, обслуживающее пункт потребления k (т. е. σ_k^2), заменить на μ_j, то стоимость решения не увеличится. Более конкретно, рассмотрим назначение π, заданное таким образом: $\pi_i^r = \sigma_i^r$ для $i \neq k,\ r = 1, 2$, и $\pi_k^1 = \sigma_k^1$, $\pi_k^2 = \mu_j(\sigma)$. В силу неравенства (5) стоимость решения, соответствующего назначению π, не превышает стоимости оптимального решения σ. Более того, в решении π вершина k хорошая для j и в силу замечания 1 она хорошая для всех остальных вершин дерева. Аналогично предыдущему пункту получаем,

что $\nu(\pi) < \nu(\sigma)$. Таким образом, π оптимально и имеет меньший индекс, что противоречит исходному предположению.



(a) (b)

Рис. 3. Иллюстрация случая 2

Подслучай 2.2: $\sigma_k^1 = \sigma_j^1 \notin N_j$ (рис. 3(b)). В этом случае справедливы равенства $c_{\sigma_k^1 j} = c_{\sigma_j^1 j}, c_{\sigma_k^1 k} = c_{\sigma_j^1 k}$. В силу оптимальности решения σ имеем $c_{\sigma_j^2 \sigma_j^1} + c_{\sigma_j^1 j} \leqslant c_{\sigma_k^2 \sigma_k^1} + c_{\sigma_k^1 j}$. Следовательно, $c_{\sigma_j^2 \sigma_j^1} \leqslant c_{\sigma_k^2 \sigma_k^1}$ и

$$c_{\sigma_j^2 \sigma_j^1} + c_{\sigma_j^1 k} \leqslant c_{\sigma_k^2 \sigma_k^1} + c_{\sigma_k^1 k}. \tag{6}$$

Рассмотрим назначение π: $\pi_i^r = \sigma_i^r$ для $i \neq k$, $r = 1, 2$, и $\pi_k^1 = \sigma_j^1,\ \pi_k^2 = \sigma_j^2$. В силу неравенства (6) стоимость решения, соответствующего назначению π, не превышает стоимости оптимального решения σ. Аналогично предыдущим пунктам в решении π вершина k хорошая для вершины j и, следовательно, для всех остальных вершин. Таким образом, назначение π оптимально и имеет меньший индекс, чем σ. Получаем противоречие с исходным предположением.

Во всех возможных случаях получено противоречие с исходным предположением. Следовательно, существует решение с нулевым индексом. Теорема 1 доказана.

3. Описание алгоритма

Обозначим через $M = \langle M_1, M_2; N\rangle$ исходную двухэтапную задачу размещения. Рассмотрим семейство подзадач

$$M_j(i,k,k') = \big\{\langle M_1, M_2; N_j \mid \pi_j^1 = i,\ \pi_j^2 = k,\ \mu_j(\pi) = k'\rangle, \\ i \in M_1,\ k, k' \in M_2,\ 1 \leqslant j \leqslant n\big\}.$$

Обозначим оптимальное решение задачи $M_j(i,k,k')$, удовлетворяющее свойству $(I1)$, через $F_j(i,k,k')$ и введём дополнительные обозначения:

$$F_j(k,k') = \min_{i\in M_1} F_j(i,k,k'), \quad F_j(k') = \min_{k\in M_2} F_j(k,k'), \quad F_j = \min_{k'\in M_2} F_j(k').$$

Отметим, что в силу теоремы 1 оптимальное решение исходной задачи $M = \langle M_1, M_2; N\rangle$ (обозначим его через F^*) совпадает с F_1. Обозначим

$$g^2_{kk'} = g^2_k + g^2_{k'},\ k \neq k', \quad g^2_{kk} = g^2_k,$$

$$G_j = \min_{i,k,k'\in N_j} F_j(i,k,k'), \quad G_j(i,k) = \min_{k'\in N_j} F_j(i,k,k'),$$

$$G_j(k,k') = \min_{i\in N_j} F_j(i,k,k'), \quad G_j(k) = \min_{i,k'\in N_j} F_j(i,k,k'),$$

$$G_j(k') = \min_{i,k\in N_j} F_j(i,k,k').$$

Рекуррентное соотношение для подсчёта $F_j(i,k,k')$ даёт

Теорема 2. *Для любых $j \in N$, $i \in M_1$ и $k,k' \in M_2$ справедливо равенство*

$$F_j(i,k,k') = g^1_i + g^2_{kk'} + b_j(c_{ki} + c_{ij}) + \sum_{l\in S_j} \min\big\{G_l,\ G_l(k') - g^2_{k'},$$
$$G_l(k) - g^2_k,\ G_l(k,k') - g^2_{kk'},\ G_l(i,k) - g^1_i - g^2_k,\ F_l(i,k,k') - g^1_i - g^2_{kk'}\big\}$$

(*здесь и далее S_j — множество сыновей вершины j*).

Доказательство проведём индукцией по высоте h дерева N_j. База индукции $h = 0$ очевидна, поскольку $N_j = \{j\}$ и справедливо равенство

$$F_j(i,k,k') = g^1_i + g^2_{kk'} + b_j(c_{ki} + c_{ij}).$$

Пусть для h утверждение теоремы верно. Докажем индукционный переход, т. е. что оно верно и для деревьев высоты $h+1$.

Пусть дерево N_j имеет высоту $h+1$. Обозначим через σ подстановку, соответствующую $F_j(i,k,k')$, т. е. на данной подстановке достигается минимум задачи $M_j(i,k,k')$. По определению $\sigma^1_j = i,\ \sigma^2_j = k,\ \mu_j(\sigma) = k'$. Таким образом,

$$F_j(i,k,k') = g^1_i + g^2_{kk'} + b_j(c_{ki} + c_{ij}) + \sum_{l\in S_j} C_l(\sigma), \tag{7}$$

где $C_l(\sigma)$ — стоимость, связанная с обслуживанием пунктов спроса из N_l в назначении σ, когда открыты предприятия $i \in M_1$, $k, k' \in M_2$. По определению $F_j(i,k,k')$ решение σ удовлетворяет свойству $(I1)$, поэтому для каждого $l \in S_j$ вершина σ_l^1 лежит в множестве $N_l \cup \{i\}$, а σ_l^2 — в $N_l \cup \{k,k'\}$. При этом в силу замечания 1 $\mu_l(\sigma) = k'$ или $\mu_l(\sigma) \in N_l$. Рассмотрим все возможные случаи.

Случай 1: $\sigma_l^1 = i,\ \sigma_l^2 = k,\ \mu_l(\sigma) = k'$. В силу оптимальности решения σ и теоремы 1 стоимость $C_l(\sigma)$ обслуживания предприятий из N_l в назначении σ совпадает с оптимальным решением задачи $M_l(i,k,k')$ за вычетом стоимостей открытия пунктов производства i,k,k', поскольку они уже учтены в формуле (7). Другими словами,

$$C_l(\sigma) = F_l(i,k,k') - g_i^1 - g_{kk'}^2.$$

Случай 2: $\sigma_l^1 = i$, $\sigma_l^2 = k$, $\mu_l(\sigma) \neq k'$. В силу оптимальности решения σ и теоремы 1 стоимость $C_l(\sigma)$ совпадает со значением $\min\limits_{t \in N_l} F_l(i,k,t)$ за вычетом стоимостей открытия пунктов производства i,k, поскольку они уже учтены в формуле (7). Таким образом,

$$C_l(\sigma) = G_l(i,k) - g_i^1 - g_k^2.$$

Случай 3: $\sigma_l^1 \neq i,\ \sigma_l^2 = k,\ \mu_l(\sigma) = k'$. Тогда $C_l(\sigma) = G_l(k,k') - g_{kk'}^2$.

Случай 4: $\sigma_l^1 \neq i,\ \sigma_l^2 = k,\ \mu_l(\sigma) \neq k'$. Тогда $C_l(\sigma) = G_l(k) - g_k^2$.

Случай 5: $\sigma_l^1 \neq i,\ \sigma_l^2 \neq k,\ \mu_l(\sigma) = k'$. Тогда $C_l(\sigma) = G_l(k') - g_{k'}^2$.

Случай 6: $\sigma_l^1 \neq i,\ \sigma_l^2 \neq k,\ \mu_l(\sigma) \neq k'$. Тогда $C_l(\sigma) = G_l$.

Отметим, что случай $\sigma_l^1 = i,\ \sigma_l^2 \neq k$ невозможен, так как по постановке двухэтапной задачи размещения каждое предприятие первого этапа получает продукцию только от одного предприятия второго этапа.

В силу оптимальности решения, соответствующего назначению σ, $C_l(\sigma)$ принимает минимальное значение для каждого $l \in S_j$ из всех возможных вариантов, приведённых выше. Другими словами,

$$C_l(\sigma) = \min\big\{F_l(i,k,k') - g_i^1 - g_{kk'}^2,\ G_l(i,k) - g_i^1 - g_k^2,\ G_l(k,k') - g_{kk'}^2,\\ G_l(k) - g_k^2,\ G_l(k') - g_{k'}^2; G_l\big\}.$$

Таким образом,

$$F_j(i,k,k') = g_i^1 + g_{kk'}^2 + b_j(c_{ki} + c_{ij}) + \sum_{l \in S_j} \min\big\{G_l,\ G_l(k') - g_{k'}^2,$$

$$G_l(k) - g_k^2,\ G_l(k,k') - g_{kk'}^2,\ G_l(i,k) - g_i^1 - g_k^2,\ F_l(i,k,k') - g_i^1 - g_{kk'}^2\big\}.$$

Теорема 2 доказана.

4. Алгоритм

Из рекуррентных соотношений теоремы 2 вытекает построение алгоритма $\mathcal{A}$ решения задачи. Он состоит из h этапов, где h — высота исходного дерева N.

Алгоритм $\mathcal{A}$.

Этап s, $1 \leqslant s < h$. Для всех вершин j, находящихся на расстоянии $h - s + 1$ от корня дерева и для всех вершин $i \in M_1,\ k, k' \in M_2$ вычисляем $F_j(i,k,k')$, $G_j(i,k)$, $G_j(k,k')$, $G_j(k)$, $G_j(k')$.

Этап h. Для всех вершин $i \in M_1,\ k, k' \in M_2$ вычисляем $F_1(i,k,k')$ и находим значение F_1, совпадающее с оптимальным решением задачи.

С помощью алгоритма $\mathcal{A}$ можем найти оптимальное значение целевой функции исходной задачи. Для поиска оптимальной подстановки π понадобится обратный алгоритм $\widetilde{\mathcal{A}}$, который также вытекает из теоремы 2.

Алгоритм $\widetilde{\mathcal{A}}$.

Этап 1. Для всех вершин $j \in N,\ i \in M_1,\ k, k' \in M_2$ алгоритмом $\mathcal{A}$ вычисляем значения $F_j(i,k,k'), G_j(i,k), G_j(k,k'), G_j(k), G_j(k'), G_j$ и F_1, которое совпадает с $F_1(i,k,k')$ для некоторых i,k,k'. Полагаем $\pi_1^1 = i,\ \pi_1^2 = k,\ \mu_1(\pi) = k'$ и переходим на следующий этап.

Этап s, $1 < s \leqslant h$, применяется к каждой вершине $u \in N$, находящейся на расстоянии s от корня дерева. Обозначим через y отца вершины u. На этапе $s-1$ находятся значения $\pi_y^1,\ \pi_y^2,\ \mu_y(\pi)$. Выбираем минимальное из значений, вычисленных алгоритмом $\mathcal{A}$:

$$\min\big\{F_u\big(\pi_y^1, \pi_y^2, \mu_y(\pi)\big) - g^1_{\pi_y^1} - g^2_{\pi_y^2 \mu_y(\pi)},\ G_u(\pi_y^1, \pi_y^2) - g^1_{\pi_y^1} - g^2_{\pi_y^2},$$

$$G_u\big(\pi_y^2, \mu_y(\pi)\big) - g^2_{\pi_y^2 \mu_y(\pi)},\ G_u\big(\pi_y^2\big) - g^2_{\pi_y^2},\ G_u\big(\mu_y(\pi)\big) - g^2_{\mu_y(\pi)},\ G_u\big\}.$$

Возможны варианты.

(i): $F_u\big(\pi_y^1, \pi_y^2, \mu_y(\pi)\big) - g^1_{\pi_y^1} - g^2_{\pi_y^2 \mu_y(\pi)}$. Тогда

$$\pi_u^1 = \pi_y^1, \quad \pi_u^2 = \pi_y^2, \quad \mu_u(\pi) = \mu_y(\pi).$$

(ii): $G_u\big(\pi_y^1, \pi_y^2\big) - g^1_{\pi_y^1} - g^2_{\pi_y^2}$. Тогда

$$\pi_u^1 = \pi_y^1, \quad \pi_u^2 = \pi_y^2, \quad \mu_u(\pi) = \arg\min_{k' \in N_u} F_u\big(\pi_y^1,\ \pi_y^2, k'\big).$$

(iii): $G_u\big(\pi_y^2, \mu_y(\pi)\big) - g^2_{\pi_y^2 \mu_y(\pi)}$. Тогда

$$\pi_u^1 = \arg\min_{i \in N_u} F_u\big(i, \pi_y^2, \mu_y(\pi)\big), \quad \pi_u^2 = \pi_y^2, \quad \mu_u(\pi) = \mu_y(\pi).$$

(iv): $G_u\big(\pi_y^2\big) - g^2_{\pi_y^2}$. Тогда

$$\pi_u^2 = \pi_y^2, \quad \pi_u^1 = i, \quad \mu_u(\pi) = k',$$

причём величины $i \in N_u,\ k' \in N_u$ доставляют минимум $F_u\big(i, \pi_y^2, k'\big)$.

(v): $G_u\big(\mu_y(\pi)\big) - g^2_{\mu_y(\pi)}$. Тогда

$$\mu_u(\pi) = \mu_y(\pi), \quad \pi_u^1 = i, \quad \pi_u^2 = k,$$

причём $i \in N_u,\ k \in N_u$ минимизируют $F_u\big(i, k, \mu_y(\pi)\big)$.

(vi): G_u. Тогда

$$\pi_u^1 = i, \quad \pi_u^2 = k, \quad \mu_u(\pi) = k',$$

причём $i \in N_u,\ k \in N_u,\ k' \in N_u$ минимизируют $F_u(i, k, k')$.

Построенное таким образом назначение π оптимально в силу теоремы 2.

Оценку трудоёмкости построенных алгоритмов даёт

Теорема 3. *Трудоёмкость алгоритмов $\mathcal{A}$ и $\widetilde{\mathcal{A}}$ не превосходит $O(nm^3)$.*

Доказательство. Для вычисления каждой величины $F_j(i, k, k')$ потребуется не более $O(\text{const})$ действий, всего же таких величин не более nm^3, где m — максимальное число возможных пунктов производства на каждом из этапов. Для вычисления всех величин $G_j(i, k)$, $G_j(k, k')$, $G_j(k)$, $G_j(k')$, G_j также понадобится не более $O(nm^3)$ действий. Таким образом, общая трудоёмкость алгоритма $\mathcal{A}$ не превосходит значения $O(nm^3)$. Трудоёмкость алгоритма $\widetilde{\mathcal{A}}$, очевидно, также не превосходит указанной величины. Теорема 3 доказана.

ЛИТЕРАТУРА

1. **Береснев В. Л., Гимади Э. Х., Дементьев В. Т.** Экстремальные задачи стандартизации. — Новосибирск: Наука, 1978. — 333 с.
2. **Гимади Э. Х.** Эффективные алгоритмы для решения многоэтапной задачи размещения на цепи // Дискрет. анализ и исслед. операций. Сер. 1. — 1995. — Т. 2, № 4. — С. 13–31.

3. **Ageev A., Ye Yi., Zhang J.** Improved combinatorial approximation algorithms for the k-level facility location problem // SIAM J. Discrete Math. — 2004. — Vol. 18, N 1. — P. 207–217.
4. **Gimadi E. Kh.** Exact algorithm for some multi-level location problems on a chain and a tree // Oper. Res. Proc. SOR'96 (Braunschweig, Germany, September 3–6, 1996). — Berlin: Springer-Verl., 1997. — P. 72–77.
5. Discrete location theory // Wiley-Intersci. Ser. Discrete Math. Optimization. — New York; Chichester; Brisbane; Toronto; Singapour: Wiley & Sons Inc., 1990. — 555 p.
6. **Zhang J.** Approximating the two-level facility location problem via a quasi-greedy approach // Proc. 15th ACM-SIAM Symp. Discrete Algorithms SODA (New Orleans, Louisiana, USA, January 11–14, 2004). — Philadelphia: SIAM, 2004. — P. 808–817.
7. **Bumb A. F., Kern W.** A simple dual ascent algorithm for the multilevel facility location problem // 4th Int. Workshop Approximation Algorithms Comb. Optimization. APPROX 2001 (Berkeley, CA, USA, August 18–20, 2001). — London: Springer-Verl., 2001. — P. 55–62. (Lect. Notes Comput. Sci.; Vol. 2129).

Гимади Эдуард Хайрутдинович,
e-mail: gimadi@math.nsc.ru
Курочкин Александр Александрович,
e-mail: alkurochkin@ngs.ru

Статья поступила
8 декабря 2011 г.
Переработанный вариант —
22 апреля 2012 г.