ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Сентябрь—октябрь 2009. Том 16, № 5. С. 3–18

УДК 519.8

ПОЛИНОМИАЛЬНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ РАЗМЕЩЕНИЯ НА ЦЕПИ С ОДИНАКОВЫМИ ПРОИЗВОДСТВЕННЫМИ МОЩНОСТЯМИ ПРЕДПРИЯТИЙ *)

А. А. Агеев, Э. Х. Гимади, А. А. Курочкин

Аннотация. Рассматривается задача размещения на путевом графе в случае одинаковых производственных мощностей предприятий. Ранее построен точный алгоритм, решающий задачу за время $O(m^5n^2 + m^3n^3)$, где m и n — число предприятий и пунктов спроса соответственно. Предлагается модификация этого алгоритма с меньшей на порядок по обоим параметрам временно́й сложностью $O(m^4n^2)$.

Ключевые слова: задача размещения, одинаковые производственные мощности, путевой граф, точный алгоритм, полиномиальная трудоёмкость.

Введение

Задачи размещения составляют один из наиболее актуальных и широких разделов исследования операций [19]. Их математический анализ в большинстве случаев приводит к рассмотрению труднорешаемых задач дискретной оптимизации [8]. Значительные успехи в этой области к настоящему времени достигнуты в исследовании задач размещения с неограниченными производственными мощностями (объёмами производства) (Uncapacitated Facility Location Problem, UFLP) [1–7, 9–17, 21–23].

Данная статья посвящена рассмотрению одного из подкласссов задачи размещения с ограничениями на производственные мощности предприятий (Capacitated Facility Location Problem, CFLP). Пусть $\mathcal{F} = \{i_1, \dots, i_m\}$ — множество возможных пунктов размещения производства некоторого продукта, $\mathcal{C} = \{j_1, \dots, j_n\}$ — множество пунктов потребления данного продукта. Тогда задача размещения предприятий с ограниченными

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 08–01–00516, 08–01–00370 и 07–07–00022).

производственными мощностями имеет следующую математическую постановку:

$$C(S,x) = \sum_{i\in S} f_i + \sum_{i\in S}\sum_{j\in \mathcal{C}} c_{ij}x_{ij} \to \min_{(S\subseteq \mathcal{F},\ x)}, \tag{1}$$

$$\sum_{i\in S} x_{ij} = b_j,\ j\in \mathcal{C}, \tag{2}$$

$$\sum_{j\in \mathcal{C}} x_{ij} \leqslant a_i,\ i\in \mathcal{F}, \tag{3}$$

$$x_{ij} \geqslant 0 \text{ целые}, \quad i\in\mathcal{F}, j\in\mathcal{C}, \tag{4}$$

где $S\subseteq\mathcal{F}$ — искомый набор открываемых пунктов производства (предприятий); f_i — стоимость открытия предприятия в пункте i; a_i — производственная мощность предприятия i; b_j — объём потребляемого пунктом спроса j (клиентом j) продукта; c_{ij} — стоимость доставки единицы продукции от возможного пункта производства i к клиенту j; x_{ij} — искомый объём продукта, поставляемого предприятием i клиенту j. Необходимо выбрать множество открываемых предприятий S и определить план перевозок $x = (x_{ij})$ таким образом, чтобы удовлетворить спрос каждого пункта потребления и минимизировать общие затраты на открытие предприятий и доставку продукта. Допустимое решение задачи с множеством S и величинами $x = (x_{ij})$ будем обозначать через (S, x).

Задача размещения производства является NP-трудной даже без ограничения на объёмы производства [13, 19]. Тем более для задачи с ограничениями (CFLP) актуальны исследования по построению приближённых полиномиальных алгоритмов с оценками качества и поиску специальных подклассов задач, когда возможно построение точных полиномиальных или псевдополиномиальных алгоритмов.

В практических приложениях довольно часто множества $\mathcal{F}$ и $\mathcal{C}$ являются множествами точек некоторого метрического пространства. Класс таких задач называется метрическим, и для него построен ряд полиномиальных (или псевдополиномиальных) алгоритмов, как точных [1, 2, 4–7], так и приближённых [3, 12, 16, 17, 20, 24].

Метрический вариант CFLP включает в себя достаточно важный сетевой класс задач, когда стоимость обслуживания c_{ij} задаётся длиной кратчайшего пути между вершинами i и j в связном графе $G = (V, E)$, $V = \mathcal{F}\cup\mathcal{C}$, с неотрицательными длинами рёбер d_{ij} между любыми двумя вершинами $i, j \in V$. В таком случае входные данные к задаче содержат значения длин всех рёбер d_{ij}.

Сетевая задача CFLP также является NP-трудной, поскольку содержит в качестве подзадачи вариант задачи о рюкзаке (достаточно рассмотреть случай, когда все рёбра графа имеют нулевую длину). В частности, CFLP является NP-трудной и на путевом графе.

В данной работе мы рассматриваем специальный подкласс CFLP на путевом графе, когда мощности всех предприятий равны между собой, т. е. $a_i = a,\ i \in \mathcal{F}$ (Uniform Capacitated Facility Location Problem, UCFLP). Для этого подкласса в [10] предложен точный алгоритм A решения задачи за время $O(m^5n^2 + m^3n^3)$. В разд. 1 данной статьи приведены основные определения, в разд. 2 — основные утверждения и их обоснование. В разд. 3 представлено описание модификации $\widetilde{A}$ алгоритма точного решения задачи UCFLP на путевом графе. В разд. 4 показано, что временна́я сложность этого алгоритма меньше на порядок (по сравнению с алгоритмом A) как по числу пунктов производства, так и по числу потребителей, а именно, она равна $O(m^4n^2)$.

1. Основные определения

Без ограничения общности будем представлять сформулированную выше задачу UCFLP как задачу на прямой. Другими словами, можно полагать, что множества $\mathcal{F} = \{i_1, \ldots, i_m\}$ и $\mathcal{C} = \{j_1, \ldots, j_n\}$ представляют собой множества точек некоторой прямой, причём $i_1 \leqslant i_2 \leqslant \ldots \leqslant i_m$ и $j_1 \leqslant j_2 \leqslant ... \leqslant j_n$. Стоимость обслуживания c_{ij} $(i \in \mathcal{F},\ j \in \mathcal{C})$ в таком случае определяется расстоянием между точками i и j на прямой.

Для любых $i' \leqslant i''$ множество $\sigma \subseteq \mathcal{F}$ будем называть *сегментом предприятий* и писать $\sigma = [i', i'']$, если $\sigma = \{i', \ldots, i''\}$. Аналогично для любых $j' \leqslant j''$ множество $\tau \subseteq \mathcal{C}$ будем называть *сегментом клиентов* и писать $\tau = [j', j'']$, если $\tau = \{j', \ldots, j''\}$.

Пусть (S, x) – допустимое решение задачи (1)–(4). Будем говорить, что $i \in \mathcal{F}$ *обслуживает* клиента $j \in \mathcal{C}$, если $x_{ij} > 0$. Пункт производства $i \in \mathcal{F}$ будем называть *насыщенным* , если $\sum\limits_{j\in\mathcal{C}} x_{ij} = a$, и *ненасыщенным,* если $0 < \sum\limits_{j\in\mathcal{C}} x_{ij} < a$. Отметим, что насыщенные и ненасыщенные пункты производства обязательно должны быть открытыми.

Для обоснования предлагаемого в данной статье алгоритма решения задачи нам понадобится доказать ряд утверждений, анонсированных в [10].

2. Основные утверждения

В этом разделе будет показано, что множество всех оптимальных решений задачи UCFLP на прямой содержит специальные решения, удовлетворяющие некоторым свойствам, благодаря которым эти специальные решения могут быть построены с помощью полиномиального алгоритма.

Далее будем рассматривать только те задачи, для которых выполнено неравенство $\sum\limits_{j \in \mathcal{C}} b_j \leqslant ma$, поскольку в противном случае суммарной мощности всех заводов не хватит для удовлетворения общего спроса клиентов и задача не имеет решения.

Рассмотрим произвольный вход $\mathcal{I}$ задачи UCFLP на прямой. В дальнейшем будет удобно представлять себе множества пунктов производства и клиентов как множества точек, лежащих на двух параллельных копиях исходной прямой. На всех последующих рисунках верхней прямой будет соответствовать множество $\mathcal{F}$, а нижней — множество $\mathcal{C}$. Порядок точек на каждой из прямых совпадает с порядком на исходной (рис. 1).

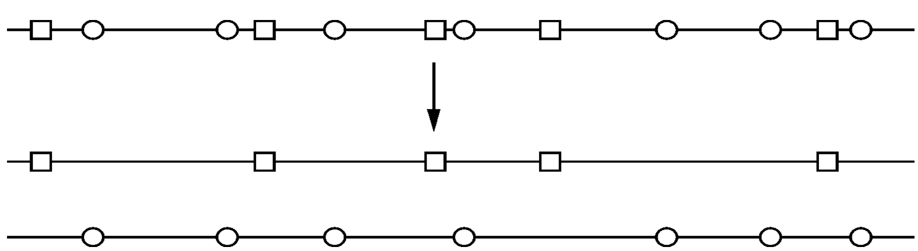

Рис. 1. Возможные пункты производства представлены квадратами на верхней прямой, пункты потребления — кругами на нижней

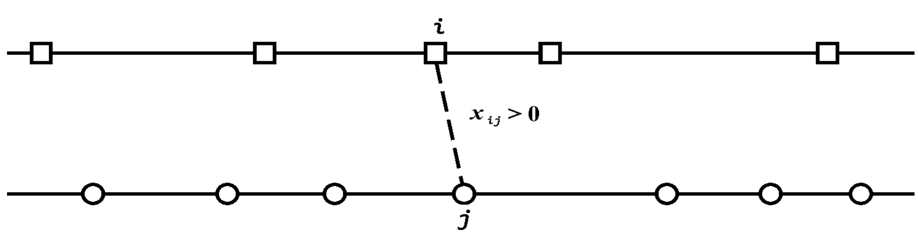

Рис. 2. Проведённое пунктиром ребро означает, что пункт производства i обслуживает пункт потребления j

Каждому возможному решению (S, x) задачи со входом $\mathcal{I}$ сопоставим двудольный граф, в котором между вершинами $i \in \mathcal{F}$ и $j \in \mathcal{C}$ существует

ребро, если и только если $x_{ij} > 0$, т. е. пункт потребления i обслуживает клиента j (рис. 2).

Обозначим произвольную индивидуальную задачу UCFLP на прямой через $P(\mathcal{I})$.

Лемма 1. *Задача $P(\mathcal{I})$ имеет оптимальное решение (S, x), удовлетворяющее свойству*

(1^0) *для любых $i', i'' \in \mathcal{F}$ и $j', j'' \in \mathcal{C}$ таких, что $i' < i''$ и $j' < j''$, либо $x_{i'j''} = 0$, либо $x_{i''j'} = 0$.*

Доказательство. В терминах графов докажем, что найдётся оптимальное решение задачи $P(\mathcal{I})$, для которого соответствующий двудольный граф не имеет пересекающихся рёбер (является планарным) (рис. 3).

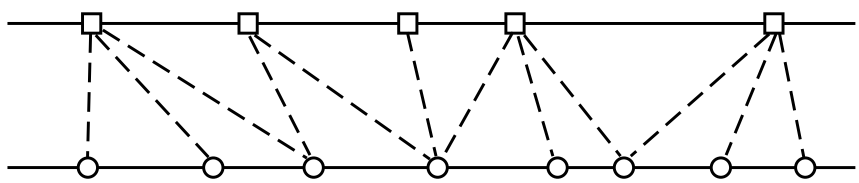

Рис. 3. Решение удовлетворяет свойству (1^0)

Для начала отметим, что для любых $i', i'' \in \mathcal{F}$ и $j', j'' \in \mathcal{C}$ таких, что $i' < i''$ и $j' < j''$, верно

$$c_{i'j''} + c_{i''j'} \geqslant c_{i'j'} + c_{i''j''}. \tag{5}$$

Это неравенство может быть легко доказано проверкой трёх возможных случаев: $i' < i'' < j' < j''$, $i' < j' < i'' < j''$ и $j' < i' < i'' < j''$.

Предположим противное. Пусть для задачи $P(\mathcal{I})$ не существует оптимального решения, удовлетворяющего (1^0). Выберем произвольное оптимальное решение (S, x) задачи. Четвёрку $(i', i''; j', j'')$ назовём *пересекающейся* , если $x_{i'j''} > 0$, $x_{i''j'} > 0$. Пару (i, j) назовём *дефектной,* если найдётся пересекающаяся четверка $(i', i; j', j)$ для некоторых $i' < i$ и $j' < j$. Отметим, что пара (i', j') не является дефектной. Также заметим, что если $i' < i''$, $j' < j''$ и пары (i', j''), (i'', j') являются дефектными, то пара (i'', j'') также является дефектной. Таким образом, множество всех дефектных пар содержит самую правую пару (i, j) такую, что для любой другой дефектной пары (i', j') выполняется $i' \leqslant i$ и $j' \leqslant j$. Соответствующее число $k = i + j$ назовём *индексом* решения (S, x).

Пусть (S, x) — оптимальное решение $P(\mathcal{I})$ с минимальным индексом. Для каждой пересекающейся четвёрки $(i', i; j', j)$ решения (S, x) проделаем ряд преобразований, в результате которых получим новое допустимое

решение задачи. А именно, определим новое решение $(S,\overline{x})$ следующим образом:

$$\begin{aligned}
&\overline{x}_{i'j'} = x_{i'j'} + \min\{x_{i'j''}, x_{i''j'}\},\\
&\overline{x}_{i''j''} = x_{i''j''} + \min\{x_{i'j''}, x_{i''j'}\},\\
&\overline{x}_{i'j''} = x_{i'j''} - \min\{x_{i'j''}, x_{i''j'}\},\\
&\overline{x}_{i''j'} = x_{i''j'} - \min\{x_{i'j''}, x_{i''j'}\},\\
&\overline{x}_{ij} = x_{ij}
\end{aligned}$$

для всех $i \neq i', i''$ и $j \neq j', j''$. По построению новое решение $(S,\overline{x})$ является допустимым, и в силу (5) значение целевой функции на этом решении не превышает значения целевой функции для (S,x). Более того, четвёрка $(i', i; j', j)$ не является пересекающейся в решении $(S,\overline{x})$. Таким образом, проделав подобные преобразования для всех пересекающихся четвёрок оптимального решения (S,x) вида $(i', i; j', j)$, мы получим новое допустимое решение (S,x^*), которое также является оптимальным и в котором пара (i,j) не является дефектной. Следовательно, либо в решении (S,x^*) больше нет дефектных пар и поэтому оно удовлетворяет свойству (1^0), либо индекс решения (S,x^*) строго меньше индекса (S,x). В обоих случаях получаем противоречие со сделанными предположениями. Лемма 1 доказана.

Следующая лемма непосредственно следует из предыдущей и при этом описывает структуру возможного решения, удовлетворяющего свойству (1^0).

Лемма 2. *Пусть (S,x) — допустимое решение задачи $P(\mathcal{I})$, удовлетворяющее свойству (1^0). Тогда*

(i) *для любых $j', j'', j''' \in \mathcal{C}$ таких, что $j' < j'' < j'''$ и $i \in \mathcal{F}$, если $x_{ij'} > 0$ и $x_{ij'''} > 0$, то $x_{ij''} = d_{j''}$;*

(ii) *для любого сегмента предприятий $\sigma \subseteq \mathcal{F}$ множество клиентов, обслуживаемых пунктами производства из σ, является сегментом (будем называть его сегментом, обслуживаемым σ);*

(iii) *для любых непересекающихся сегментов предприятий σ_1 и σ_2 сегменты клиентов, обслуживаемые σ_1 и σ_2, имеют не более одного общего клиента.*

Пусть (S,x) — некоторое допустимое решение задачи $P(\mathcal{I})$. Будем называть сегмент предприятий σ *простым,* если он содержит открытые пункты производства и максимум одно из них является ненасыщенным. Два открытых пункта производства $i_1, i_2 \in \mathcal{F}$, $i_1 \leqslant i_2$, будем называть *соседними,* если сегмент $[i_1 + 1, i_2 - 1]$ не содержит открытых пунктов

производства. Также будем говорить, что открытые пункты производства i_1, i_2 являются *смежными,* если сегменты клиентов, обслуживаемые пунктами i_1 и i_2, имеют общего клиента, и *несмежными* — в противном случае.

Следующая лемма является центральным утверждением этого раздела статьи.

Лемма 3. *Любая задача $P(\mathcal{I})$ имеет оптимальное решение, удовлетворяющее свойствам* (1^0) *и* (2^0)*: для любых ненасыщенных пунктов производства i' и i'' сегмент предприятий $[i', i'']$ содержит открытые предприятия i^* и i^{**}, которые являются соседними и несмежными.*

Доказательство. В терминах графов данное утверждение формулируется так. Задача $P(\mathcal{I})$ имеет оптимальное решение (S, x), для которого соответствующий двудольный граф является планарным (свойство (1^0)) и любые два ненасыщенных пункта производства лежат в разных компонентах связности данного графа (свойство (2^0)).

Предположим противное. Пусть (S, x) — оптимальное решение задачи $P(\mathcal{I})$ такое, что выполнено свойство (1^0), а число пар ненасыщенных пунктов производства, не удовлетворяющих свойству (2^0), минимально. Пусть i', i'', $i' < i''$, — одна из таких пар. Без ограничения общности можно полагать, что сегмент $[i'+1, i''-1]$ не содержит ненасыщенных пунктов производства.

Положим $w' = a - \sum\limits_{j=1}^{n} x_{i'j}$ и $w'' = a - \sum\limits_{j=1}^{n} x_{i''j}$. Заметим, что $0 < w' < a$ и $0 < w'' < a$.

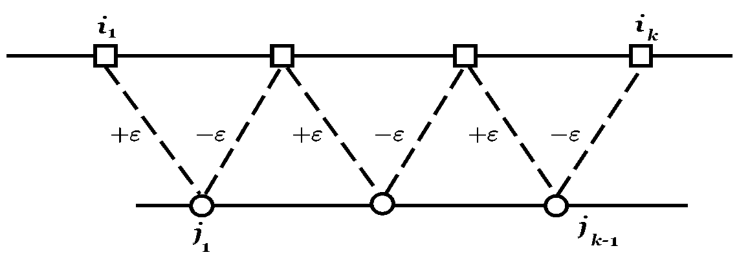

Рис. 4

Обозначим через $\{i_1, \ldots, i_k\}$, где $i' = i_1 < i_2 < \ldots < i_k = i''$, множество открытых пунктов производства сегмента $[i', i'']$ таких, что каждый пункт производства i_t, $2 \leqslant t < k$, обслуживает не менее двух клиентов. В случае $k = 2$ в силу предположения существует клиент, обслуживаемый i_1 и i_2. Напомним, что пункты производства i_t являются насыщенными для всех $t = 2, \ldots, k-1$.

Для всех значений $t = 1, \ldots, k-1$ обозначим через $j_t \in \mathcal{C}$ клиента, которого одновременно обслуживают пункты производства i_t и i_{t+1}. В силу леммы 2 имеем $j_1 < \ldots < j_{k-1}$.

Отметим, что $\mathcal{P} = (i_1, j_1, \ldots, j_{k-1}, i_k)$ образует путь между вершинами i_1 и i_k в графе, соответствующем решению (S, x) (рис. 4).

Для $t = 2, \ldots, k-1$ положим $x_{i_t j_{t-1}}(\varepsilon) = x_{i_t j_{t-1}} - \varepsilon$, $x_{i_t j_t}(\varepsilon) = x_{i_t j_t} + \varepsilon$. А также $x_{i_1 j_1}(\varepsilon) = x_{i_1 j_1} + \varepsilon$, $x_{i_k j_{k-1}}(\varepsilon) = x_{i_k j_{k-1}} - \varepsilon$ и $x_{ij}(\varepsilon) = x_{ij}$ для всех оставшихся пар i и j. Определим

$$\varepsilon_1 = \min\{\min_{t \in \{2,\ldots,k-1\}} x_{i_t j_{t-1}}, x_{i_k j_{k-1}}, w'\},$$

$$\varepsilon_2 = -\min\{\min_{t \in \{2,\ldots,k-1\}} x_{i_t j_t}, x_{i_1 j_1}, w''\}.$$

По построению решение $(S, x(\varepsilon))$ является допустимым и удовлетворяет свойству (1^0) для всех значений $\varepsilon \in [\varepsilon_2, \varepsilon_1]$.

Так как целевая функция задачи линейно зависит от ε, то стоимость одного из решений $(S, x(\varepsilon_1))$ или $(S, x(\varepsilon_2))$ не превышает стоимости решения (S, x). В силу свойства (1^0) в графе (S, x) путь $\mathcal{P}$ является единственным, соединяющим вершины i_1 и i_k. Это означает, что графы, соответствующие решениям $(S, x(\varepsilon_1))$ и $(S, x(\varepsilon_2))$, удовлетворяют следующему свойству: или вершины i_1 и i_k лежат в разных компонентах связности, или одна из них является насыщенной. Таким образом, для обоих графов число пар ненасыщенных пунктов производства, соединённых путём (или, что то же самое, не удовлетворяющих (2^0)), уменьшается по крайней мере на одну, что противоречит выбору решения (S, x). Лемма 3 доказана.

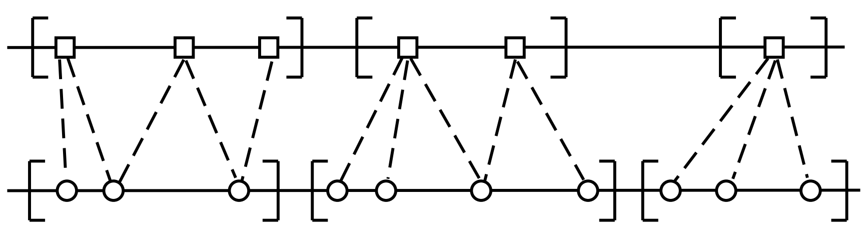

Рис. 5. Графическое представление решения задачи, удовлетворяющего свойствам (1^0) и (2^0)

Лемма 4. *Если оптимальное решение задачи $P(\mathcal{I})$ удовлетворяет свойствам (1^0) и (2^0), то множество возможных пунктов производства $\mathcal{F}$ представляет собой объединение непересекающихся простых сегментов предприятий, никакие два из которых не имеют общего клиента (рис. 5).*

Доказательство. Непосредственное следствие из леммы 3.

3. Описание алгоритма $\widetilde{A}$

В данном разделе воспользуемся доказанными выше утверждениями о существовании оптимального решения специального вида и построим полиномиальный алгоритм решения задачи UCFLP на прямой.

С целью упрощения выкладок вместо $\mathcal{F} = \{i_1, \ldots, i_m\}, \mathcal{C} = \{j_1, \ldots, j_n\}$ будем работать с номерами предприятий и клиентов, т. е. считаем, что $\mathcal{F} = \{1, \ldots, m\}, \mathcal{C} = \{1, \ldots, n\}$.

Обозначим через $R(i, j)$ стоимость оптимального решения, удовлетворяющего свойствам (1^0) и (2^0), в подзадаче, когда множество заводов — это $\{1, \ldots, i\}$, а множество клиентов — $\{1, \ldots, j\}$; положим $R(0, 0) = 0$.

Из леммы 3 следует, что $R(m, n) = C(S^*, x^*)$, где (S^*, x^*) — оптимальное решение исходной задачи.

Пусть $Q_{i',i''}(j', j'')$ – оптимум решения в подзадаче с множествами заводов $[i', i'']$ и клиентов $[j', j'']$, удовлетворяющий свойству (1^0), и при этом сегмент предприятий $[i', i'']$ является простым, т. е. содержит не более одного ненасыщенного пункта производства. Тогда в силу леммы 4 справедлива рекуррентная формула для вычисления $R(i, j)$ для всех $i = \overline{1, m}$ и $j = \overline{1, n}$:

$$R(i, j) = \min_{i',j'} \{R(i' - 1, j' - 1) + Q_{i',i}(j', j) \ : \ i' = \overline{1, i}, \ j' = \overline{1, j}\}. \qquad (6)$$

Таким образом, для определения $R(i, j)$ достаточно найти значения всех величин $Q_{i',i''}(j', j'')$. Уменьшение на порядок трудоёмкости алгоритма решения сетевой задачи UCFLP, по сравнению с описанным в [10], достигается именно за счёт более быстрого метода подсчёта данных величин. Дальнейшее изложение посвящено описанию этого метода.

Всюду далее величину $\sum\limits_{j=j'}^{j''} b_j$ будем называть *совокупным спросом* сегмента клиентов $[j', j'']$.

Введём дополнительные обозначения:

$B_j = \sum\limits_{k=1}^{j} b_k$ — совокупный спрос сегмента клиентов $[1, j], \ j = \overline{1, n}$;

$C_{ij} = \sum\limits_{k=1}^{j} c_{ik} b_k$ — транспортные расходы, связанные с удовлетворением i-м предприятием совокупного спроса $[1, j]$; $\quad C_{i0} = 0$;

$r = \lceil (B_{j''} - B_{j'-1})/a \rceil$ — число открываемых предприятий в сегменте $[i', i'']$ для удовлетворения совокупного спроса сегмента $[j', j'']$ при условии, что $[i', i'']$ является простым сегментом.

Фактически нам необходимо определить, какое предприятие из сегмента $[i', i'']$ будет являться ненасыщенным. Оставшиеся открытые предприятия (их число равно $r-1$) будут насыщенными, т. е. каждое из них будет производить продукт в максимальном объёме a. Для наглядности удобно рассмотреть отрезок прямой длины $\sum\limits_{j=j'}^{j''} b_j$, который соответствует совокупному спросу сегмента $[j', j'']$. Наша задача — покрыть его набором отрезков, состоящим из $(r-1)$-го отрезка длины a и одного отрезка длины $\sum\limits_{j=j'}^{j''} b_j - (r-1)a$. Таким образом, «передвигая» по участку совокупного спроса отрезок длины $\sum\limits_{j=j'}^{j''} b_j - (r-1)a$, соответствующий возможно ненасыщенному предприятию, мы сможем эффективно вычислить значение $Q_{i',i''}(j', j'')$ (рис. 6).

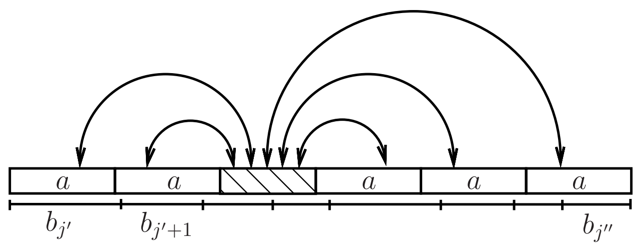

Рис. 6. Заштрихованный отрезок соответствует мощности ненасыщенного предприятия. Передвигая его по участку совокупного спроса, сможем эффективно вычислить значение $Q_{i',i''}(j', j'')$

Определим вспомогательное множество клиентов $T = \{j_1^*, \ldots, j_{r-1}^*\}$ следующим образом:

$$B_{j_t^*-1} - B_{j'-1} < ta \leqslant B_{j_t^*} - B_{j'-1},\ 1 \leqslant t < r.$$

По определению T (рис. 7) если $\{i_1, i_2, \ldots, i_k\}$, $k \leqslant r$, — первые k открытых предприятий сегмента $[i', i'']$, причём $i_1, \ldots, i_{k-1}$ являются насыщенными, то для любого $1 \leqslant s \leqslant k-1$ либо пересечение областей обслуживания соседних предприятий i_s и i_{s+1} происходит по клиенту j_s^*, либо если области обслуживания не пересекаются, предприятие i_s обслуживает клиента j_s^*, а предприятие i_{s+1} — клиента j_s^*+1 (следующего за j_s^* клиента из $[j', j'']$).

Найдём затраты $g_{i,t,j'}$, связанные с открытием и обслуживанием насыщенного предприятия $i \in [i', i'']$, при условии, что среди предприятий

из $[i', i-1]$ ровно $t-1$ открытых и все они являются насыщенными (считаем, что $t < r$).

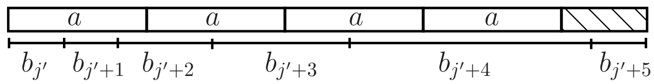

Рис. 7. Пример наглядного построения T для $r = 5$ и j', $j'' = j' + 5$. В данном случае $j_1^* = j' + 2$, $j_2^* = j' + 3$, $j_3^* = j' + 4$, $j_4^* = j' + 4$

Пересечение области обслуживания предприятия i с соседними открытыми происходит в пунктах спроса j_{t-1}^* и j_t^*. Возможны два случая.

1. $j_{t-1}^* = j_t^*$. В таком случае насыщенное предприятие i обслуживает только одного клиента j_t^* и искомые затраты $g_{i,t,j'}$ равны $f_i + c_{ij_t^*}a$.

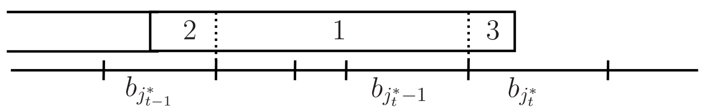

Рис. 8. Область обслуживания t-го открытого насыщенного предприятия

2. $j_{t-1}^* < j_t^*$. В таком случае i-е насыщенное предприятие обслуживает в полном объёме все пункты спроса из интервала (j_{t-1}^*, j_t^*) (участок 1 на рис. 8). Стоимость обслуживания данного интервала составляет

$$C_{ij_t^*-1} - C_{ij_{t-1}^*}.$$

Транспортные затраты по обслуживанию крайнего левого пункта спроса j_{t-1}^* (участок 2 на рис. 8) равны

$$c_{ij_{t-1}^*}(B_{j_{t-1}^*} - B_{j'-1} - (t-1)a).$$

Для крайнего правого пункта спроса j_t^* (участок 3 на рис. 8) аналогично получаем

$$c_{ij_t^*}(ta - (B_{j_t^*-1} - B_{j'-1})).$$

Таким образом, находим искомые затраты

$$\begin{aligned} g_{i,t,j'} = f_i &+ (C_{ij_t^*-1} - C_{ij_{t-1}^*}) \\ &+ c_{ij_{t-1}^*}(B_{j_{t-1}^*} - B_{j'-1} - (t-1)a) + c_{ij_t^*}(ta - (B_{j_t^*-1} - B_{j'-1})). \end{aligned}$$

Обозначим через $G_{i,t,j'}^L$ оптимальные затраты, связанные с открытием и обслуживанием t насыщенных предприятий в сегменте $[i', i]$, причём

предприятие i обязательно открыто (считаем, что $|\{i',\ldots,i\}| = i'-i+1 \geqslant t$). Также введём $\widetilde{G}^L_{i,t,j'}$ — оптимальные затраты, связанные с открытием и обслуживанием первых t насыщенных предприятий в сегменте $[i',i]$, когда предприятие i необязательно должно быть открыто.

Значения введённых величин для всех значений параметров вычисляются посредством следующих рекуррентных соотношений:

$$G^L_{i,1,j'} = g_{i,1,j'},;\ G^L_{i,t,j'} = g_{i,t,j'} + \widetilde{G}^L_{i-1,t-1,j'},\quad i-i'+1 \geqslant t \geqslant 2;$$
$$\widetilde{G}^L_{i,1,j'} = \min_{i'\leqslant k\leqslant i} G^L_{k,1,j'} = \min_{i'\leqslant k\leqslant i} g_{k,1,j'};$$
$$\widetilde{G}^L_{i,t,j'} = \min_{i'\leqslant k\leqslant i,\ (k-i'+1)>t} G^L_{k,t,j'} = \min\left\{G^L_{i,t,j'}, \widetilde{G}^L_{i-1,t,j'}\right\},\quad i-i'+1 \geqslant t \geqslant 2.$$

Таким образом можно эффективно вычислить значения $\widetilde{G}^L_{i,t,j'}$.

По идентичной схеме определим величины $\widetilde{G}^R_{i,t,j''}$, которые обозначают затраты, связанные с открытием и обслуживанием t насыщенных предприятий в сегменте $[i,i'']$. Разница состоит в том, что в предыдущих рассуждениях рассматривались открытые насыщенные предприятия, начиная от левого конца сегмента $[i',i'']$, и эти предприятия обслуживали некоторый сегмент клиентов $[j',j_s] \subseteq [j',j'']$. Теперь мы рассматриваем открытые насыщенные предприятия, начиная от правого конца сегмента $[i',i'']$. При этом они будут обслуживать некоторый сегмент клиентов $[j_s,j''] \subseteq [j',j'']$. Через $g'_{i,t,j''}$ обозначим стоимость открытия и обслуживания насыщенного предприятия $i \in [i',i'']$ при условии, что среди предприятий $[i+1,i'']$ ровно $t-1$ открытых и все они насыщенные. Величины, имеющие значения, аналогичные $B_j, T = \{j^*_1,\ldots,j^*_{r-1}\}$, будем обозначать соответственно $B'_j, T' = \{j^{**}_1,\ldots,j^{**}_{r-1}\}$. Нетрудно понять, что формулы для их вычисления строятся тем же образом:

$$B'_j = \sum_{k=j}^{n} b_k,\ j=\overline{1,n};\quad r = \lceil (B'_{j'} - B'_{j''+1})/a \rceil;\quad T' = \{j^{**}_1,\ldots,j^{**}_{r-1}\};$$
$$B'_{j^{**}_t+1} - B'_{j''+1} < ta \leqslant B'_{j^{**}_t} - B'_{j''+1},\quad t=1,\ldots,r-1;$$
$$g'_{i,t,j''} = f_i + (C_{ij^{**}_t+1} - C_{ij^{**}_{t-1}})$$
$$\qquad + c_{ij^{**}_{t-1}}(B'_{j^{**}_{t-1}} - B'_{j''+1} - (t-1)a) + c_{ij^{**}_t}(ta - (B'_{j^{**}_t+1} - B'_{j''+1}));$$
$$G^R_{i,1,j''} = g'_{i,1,j''};\quad G^R_{i,t,j''} = g'_{i,t,j''} + \widetilde{G}^R_{i+1,t-1,j''},\quad i''-i+1 \geqslant t \geqslant 2;$$
$$\widetilde{G}^R_{i,1,j''} = \min_{i\leqslant k\leqslant i''} G^R_{k,1,j''} = \min_{i\leqslant k\leqslant i''} g'_{k,1,j''};$$

$$\widetilde{G}^R_{i,t,j''} = \min_{i \leqslant k \leqslant i'',\ (i''-k+1)>t} G^R_{k,t,j''} = \min\bigl\{G^R_{i,t,j''}, \widetilde{G}^R_{i+1,t,j''}\bigr\},$$
$$i'' - i + 1 \geqslant t \geqslant 2.$$

Осталось сделать последний шаг для вычисления величин $Q_{i',i''}(j',j'')$, а именно, пусть единственное ненасыщенное предприятие $i \in [i',i'']$ является t-м по счёту среди открытых предприятий, если считать от i'. Также оно является $(r-t+1)$-м открытым предприятием, если считать открытые от i''. Обозначим через S_{it} затраты, связанные с открытием этого ненасыщенного предприятия, включая затраты на удовлетворение этим предприятием своей области спроса. Тогда нетрудно получить формулу

$$S_{it} = f_i + (C_{ij^{**}_{r-t}-1} - C_{ij^*_{t-1}}) + c_{ij^*_{t-1}}(B_{j^*_{t-1}} - B_{j'-1} - (t-1)a) +$$
$$+ c_{ij^{**}_{r-t}}(B'_{j^{**}_{r-t}} - B'_{j''+1} - (r-t)a), \quad i - i' + 1 \geqslant t.$$

Таким образом (рис. 9), слева от ненасыщенного предприятия i находятся ровно $t-1$ открытых насыщенных предприятий, оптимальные затраты на открытие и обслуживание которых составляют $\widetilde{G}^L_{i-1,t-1,j'}$, а справа от i находятся ровно $r-t$ открытых насыщенных предприятий, оптимальные затраты на открытие и обслуживание которых составляют $\widetilde{G}^R_{i+1,r-t,j''}$.

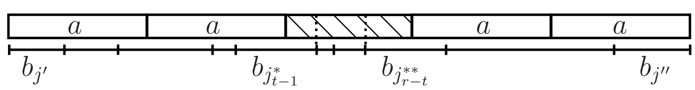

Рис. 9. Единственное ненасыщенное предприятие сегмента $[i',i'']$ имеет номер t среди открытых, если считать от i', и $r-t+1$ среди открытых, если считать от i''

Теперь можем выписать общую формулу для нахождения $Q_{i',i''}(j',j'')$:

$$Q_{i',i''}(j',j'') = \min_{1 \leqslant t \leqslant r,\ i' \leqslant i \leqslant i''} \bigl\{\widetilde{G}^L_{i-1,t-1,j'} + S_{it} + \widetilde{G}^R_{i+1,r-t,j''}\bigr\}. \qquad (7)$$

С учётом всего вышесказанного формула для подсчёта $Q_{i',i''}(j',j'')$ имеет вид

$$Q_{i',i''}(j',j'') = \min_{1 \leqslant t \leqslant r,\ i' \leqslant i \leqslant i''} \bigl\{\widetilde{G}^L_{i-1,t-1,j'} + S_{it} + \widetilde{G}^R_{i+1,r-t,j''}\bigr\}. \qquad (8)$$

Описание алгоритма $\widetilde{A}$ закончено.

4. Трудоёмкость алгоритма $\widetilde{A}$

Теорема 1. *Алгоритм $\widetilde{A}$ находит оптимальное решение сетевой задачи UCFLP за время $O(m^4n^2)$.*

Доказательство. Оценим временну́ю сложность нахождения всех необходимым величин.

На подсчёт всех значений B_j, C_{ij} потребуется $O(n)$, $O(mn)$ операций соответственно.

Построение множества T для каждого фиксированного j' требует не больше $O(m)$ действий; следовательно, всего потребуется $O(mn)$ операций.

Величины $g_{i,t,j'}$ вычисляются за время $O(nm^2)$.

Подсчёт значений $\widetilde{G}^L_{i,t,j'}$ для всех $1 \leqslant i', i, t \leqslant m$ и $1 \leqslant j' \leqslant n$ осуществляется за время $O(m^3n)$.

Вычисление величины S_{it} для фиксированных j', j'' потребует $O(m^2)$ операций. Тогда вычисление этих величин для всех значений параметров потребует не более $O(n^2m^2)$ действий.

Таким образом, на вычисление всех необходимых для $Q_{i',i''}(j', j'')$ величин будет затрачено не более $O(n^2m^2)$ действий.

Сложность вычислений одной величины $Q_{i',i''}(j', j'')$, как видно из формулы (8), не превосходит $O(m^2)$. Вычисление этих величин для всех значений параметров потребует не более $O(m^4n^2)$ операций.

Из вида рекуррентной формулы (6) заключаем, что при найденных $Q_{i',i''}(j', j'')$ вычисление всех величин $R(i, j)$ займёт время $O(m^2n^2)$.

Таким образом, в результате работы алгоритма получаем решение с оптимумом, равным $R(m, n)$, за время, ограниченное величиной $O(m^4n^2)$.

ЛИТЕРАТУРА

1. **Агеев А. А.** Полиномиальный алгоритм решения задачи размещения на последовательно-параллельной сети // Управляемые системы. Сб. науч. тр. Вып. 30. — Новосибирск: Ин-т математики СО РАН, 1990. — С. 3–16.
2. **Береснев В. Л., Гимади Э. Х., Дементьев В. Т.** Экстремальные задачи стандартизации. — Новосибирск: Наука, 1978. — 333 с.
3. **误нюк И. П., Гимади Э. Х., Филатов М. Ю.** Асимптотически точный алгоритм для решения задачи размещения с ограниченными объёмами производства // Дискрет. анализ и исслед. операций. Сер. 2. — 2001. — Том 8, № 2. — С. 3–16.
4. **Гимади Э. Х.** Эффективный алгоритм решения задачи размещения с областями обслуживания, связными относительно ациклической сети // Управляемые системы. Сб. науч. тр. Вып. 23. — Новосибирск: Ин-т математики СО АН СССР, 1983. — С. 12–23.

5. **Гимади Э. Х.** Задача размещения на сети с центрально-связными областями обслуживания //Управляемые системы. Сб. науч. тр. Вып. 25. — Новосибирск: Ин-т математики СО АН СССР, 1984. — С. 38–47.
6. **Гимади Э. Х.** Задача стандартизации с данными произвольного знака и связными, квазивыпуклыми и почти квазивыпуклыми матрицами // Управляемые системы. Сб. науч. тр. Вып. 27. — Новосибирск: Ин-т математики СО АН СССР, 1987. — С. 3–11.
7. **Гимади Э. Х.** Эффективные алгоритмы для решения многоэтапной задачи размещения на цепи // Дискрет. анализ и исслед. операций. — 1995. Том 2, № 4. — С. 13–31.
8. **Гэри M. R., Джонсон D. S.** Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. — 416 с.
9. **Ageev A. A.** A polynomial-time solvability for the network location problem // Integer programming and combinatorial optimization. Proc. IPCO II Conf. Campus Printing. — Pittsburg: Carnegi Mellon University, 1992. — P. 237–245.
10. **Ageev A. A.** A polynomial-time algorithm for the facility location problem with uniform hard capacities on path graph // Proc. of the 2nd Intern. workshop «Discrete optimization methods in production and logistics», DOM'2004. Omsk: Nasledie–Dialog Sibir Pbs, 2004. — P. 28–32.
11. **Billionet A., Costa M.-C.** Solving the uncapacitated plant location problem on trees // Discr. Appl. Math. — 1994. — Vol. 49, N 1–3. — P. 51–59.
12. **Chudak F. A., Williamson D.** Improved approximation algorithms for capacitated facility location problems // Lect. Notes in Comp. Sci. — 1999. — Vol. 1610. — P. 99–113.
13. **Cornuéjols G., Nemhauser G. L., Wolsey L. A.** The uncapacitated facility location problem // Discrete location theory. — New York: Wiley, 1990. — P. 119–171.
14. **Gimadi E. Kh.** Exact algorithms for some multi-level location problems on a chain and a tree // Oper. Res. Proc. 1996 (SOR96). — Berlin, Heidelberg: Springer–Verl., 1997. — P. 72–77.
15. **Hassin R., Tamir A.** Efficient algorithms for optimization on series-parallel graphs // SIAM J. Algebraic Discrete Methods. — 1986. — V. 7, N 3. — P. 379–386.
16. **Korupolu M. R., Plaxton C. G., Rajaraman R.** Analysis of a local search heuristic // Proc. of the 9th Ann. ACM-SIAM Symp. on discrete algorithms. — San Francisco, CA: SODA, 1998. — P. 1–10.
17. **Mahdian M., Pál M.** Universal facility location // Lect. Notes in Comp. Sci. — 2003. — Vol. 2832. — P. 409–421.
18. **Mirchandani P., Kohli R., Tamir A.** Capacitated location problems on a line // Transportation Sci. — 1996. — Vol. 30, N 1. — P. 75–80.
19. **Mirchandani P. B., Francis R. L.** Discrete location theory. Wiley-

Interscience series in discrete mathematics and optimization. — New York: Wiley, 1990. — 576 p.

20. **Pál M., Tardos E., Wexler T.** Facility location with hard capacities // Proc. of the 42nd Ann. Symp. on foundations of computer science. — Las Vegas, Nevada, USA : FOCS, 2001. — P. 320–328.
21. **Shah D.** An unified limited column generation approach for facility location problem on trees // Ann. Oper. Research. — 1999. — Vol. 87. — P. 363–382.
22. **Shah R., Farach-Colton V.** Undiscretized dynamic programming: faster algorithms for facility location and related problems on trees // Proc. of the 13th Ann. ACM-SIAM Symp. on discrete algorithms. — San Francisco, California: SODA, 2002. — P. 108–115.
23. **Tamir A., Lowe T.** The generalized p-forest problem on a tree network // Networks. — 1992. — Vol. 22. — P. 217–230.
24. **Zhang J., Chen B., Ye Y.** A multi-exchange local search algorithm for the capacitated facility location problem // Mathematics on Operation Research. — May 2005. — Vol. 30, N 2. — P. 389–403.

Агеев Александр Александрович,
e-mail: ageev@math.nsc.ru

Гимади Эдуард Хайрутдинович,
e-mail: gimadi@math.nsc.ru

Курочкин Александр Александрович,
e-mail: alkurochkin@ngs.ru

Статья поступила
25 июня 2009 г.