ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Сентябрь—октябрь 2014. Том 21, № 5. С. 23–39

УДК 519.8

ЗАДАЧА РАЗМЕЩЕНИЯ С ОГРАНИЧЕННЫМИ ОБЪЁМАМИ ПРОИЗВОДСТВА НА СЛУЧАЙНЫХ ВХОДНЫХ ДАННЫХ *)

А. А. Курочкин

Аннотация. Рассматривается задача размещения с ограничениями на объёмы производства на случайных входных данных. Предполагается, что элементы матрицы транспортных расходов заданы независимыми случайными величинами с равномерной функцией распределения на целочисленном сегменте $[1, r]$. Для задачи с одинаковыми ограничениями на объёмы производства построен приближённый алгоритм решения и проведён вероятностный анализ его работы. Представлены условия, при которых алгоритм является асимптотически точным и имеет временную трудоёмкость $O(n \ln m)$, где n — число потребителей, m — число возможных пунктов производства. Для задачи с произвольными ограничениями построен алгоритм с трудоёмкостью порядка $O(n^{2(1-\theta)}m)$ и определены условия его асимптотической точности для некоторого $\theta < \frac{1}{2}$.

Ключевые слова: задача размещения производства, полиномиальный алгоритм, вероятность, асимптотическая точность.

Введение

Задача размещения производства является известной NP-трудной проблемой даже в простейшей постановке без ограничений на объёмы производства [13]. Поэтому для задачи с ограничениями на объёмы производства наиболее актуальны исследования по построению приближённых полиномиальных алгоритмов с гарантированными оценками качества. Стоит отметить, что для некоторых специальных подзадач возможно построение точных полиномиальных алгоритмов решения. Однако в литературе известно не так много подобных примеров. Например,

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 12–01–00093, 12–01–00631, 13–07–00070 и 12–01–33028–мол-а-вед), целевой программы Президиума РАН (проект № 227) и междисциплинарного интеграционного проекта СО РАН № 7Б.

в [8] для задачи с одинаковыми ограничениями на объёмы производства на путевом графе (когда множества пунктов производства и спроса являются вершинами, лежащими на одном пути) предложен точный алгоритм с трудоёмкостью $O(m^5n^2 + m^3n^3)$. Позднее в [1] была предложена модификация этого алгоритма, имеющая на порядок меньшую трудоёмкость $O(m^4n^2)$.

В то же время существует достаточно много работ, посвящённых приближённым методам решения задачи. Для самой распространённой метрической постановки задачи размещения с ограничениями в [12] предложен алгоритм с оценкой точности $6(1 + \varepsilon)$. Он является модификацией алгоритма из [15], имеющего точность $8(1 + \varepsilon)$ и основанного на методе локального поиска. Достаточно широко исследовалось решение задачи размещения с помощью различных эвристических методов [9, 10, 16].

Интересна также постановка задачи на входных данных, заданных случайным образом, которая становится особенно актуальной с ростом размерности рассматриваемой задачи. Так, в [17] подробно исследуется метрическая постановка, в которой множество клиентов подчиняется некоторому вероятностному распределению в заданной области. Обзор по основным методам вероятностного подхода можно также найти в [14]. В данной работе исследуется задача размещения с ограничениями на объёмы производства, в которой транспортные расходы по доставке продукции задаются как значения независимых случайных величин, равновероятно принимающих значения из целочисленного сегмента $[1, r]$. Ранее подобная задача рассматривалась в [2], где проведён её вероятностный анализ и построен приближённый алгоритм решения. Показано, что предложенный алгоритм асимптотически точный, но при условии наложения достаточно жёстких ограничений на входные данные задачи. Такая задача рассматривалась и в [5], где предложен другой приближённый алгоритм решения, но и в ней условия асимптотической точности получены при введении серьёзных ограничений на входные данные: предполагалось, что функция спроса потребителей является кусочно-постоянной с точками разрыва, кратными определённой величине. В настоящей работе предлагается новый алгоритм, условия асимптотической точности которого не накладывают столь существенных ограничений на входные данные задачи, как в [2, 5].

В работе рассматриваются две постановки задачи размещения: с одинаковыми и произвольными ограничениями на объёмы производства. Для задачи с одинаковыми ограничениями построен приближённый алгоритм решения и проведён вероятностный анализ его работы. Представ-

лены условия, при которых алгоритм является асимптотически точным и имеет временну́ю трудоёмкость $O(n \ln m)$, где n — число потребителей, а m — число возможных пунктов производства. Для задачи с произвольными ограничениями построен алгоритм с трудоёмкостью порядка $O(n^{2(1-\theta)}m)$ и определены условия его асимптотической точности для некоторого значения $\theta < \frac{1}{2}$.

1. Задача размещения с одинаковыми ограничениями на объёмы производства

1.1. Описание задачи. Дадим общую формулировку задачи размещения с ограничениями на объёмы производства. Пусть $J = \{1, \ldots, n\}$ и $I = \{1, \ldots, m\}$ — множества пунктов потребления некоторого продукта и возможных пунктов размещения предприятий соответственно. Известны затраты g_i^0 на размещение предприятия и ограничения d_i на объёмы производства в пункте $i \in I$. Для каждого пункта потребления $j \in J$ задан требуемый объём b_j спроса продукта. Также заданы транспортные затраты g_{ij}, связанные с доставкой единицы продукции из предприятия $i \in I$ в пункт потребления $j \in J$.

Требуется определить пункты размещения производства и транспортные потоки между производителями и клиентами таким образом, чтобы удовлетворить потребности всех потребителей и минимизировать общие затраты на размещение производства и поставку продукции. Математическая формулировка задачи: минимизировать суммарные затраты

$$C(x) = \sum_{i=1}^{m} g_i^0 x_i + \sum_{i=1}^{m} \sum_{j=1}^{n} g_{ij} x_{ij} \tag{1}$$

по переменным x_i, x_{ij} при выполнении условий

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j \in J, \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} \leqslant d_i x_i, \quad i \in I, \tag{3}$$

$$x_{ij} \geqslant 0, \quad x_i \in \{0, 1\}, \quad i \in I,\ j \in J, \tag{4}$$

где x_i, $i \in I$, — переменные выбора предприятий (если $x_i = 1$, то в соответствующем решении предприятие $i \in I$ является открытым), x_{ij}, $i \in I$, $j \in J$, — транспортные переменные, задающие объём продукта,

поставляемый предприятием $i \in I$ в пункт спроса $j \in J$ в соответствующем решении.

Будем считать, что объёмы производства всех предприятий одинаковы:

$$d_i = d, \quad i \in I. \tag{5}$$

В данной работе исследуется задача (1)–(5) в предположении, что транспортные расходы g_{ij}, $i \in I$, $j \in J$, заданы независимыми случайными величинами из класса $U[1, r]$ (класс случайных величин, равновероятно принимающих значения из целочисленного сегмента $[1, r]$, r — целое).

Без ограничения общности всюду далее будем считать, что предприятия в I и пункты спроса в J перенумерованы таким образом, что $g_1^0 \leqslant \ldots \leqslant g_m^0$ и $b_1 \leqslant \ldots \leqslant b_n$.

1.2. Некоторые определения и вспомогательные утверждения. Будем опираться на определение асимптотической точности, введённое в [6]. Будем говорить, что алгоритм $\widetilde{\mathcal{A}}$ имеет оценки точности $(\varepsilon_n, \delta_n)$ в классе $\mathcal{K}_n$ задачи минимизации размерности n, если

$$\mathbb{P}\left\{ \frac{C(x^{\widetilde{\mathcal{A}}}) - C(x^*)}{C(x^*)} > \varepsilon_n \right\} \leqslant \delta_n,$$

где x^* — оптимальное решение для индивидуальной задачи I_n, $x^{\widetilde{\mathcal{A}}}$ — решение, полученное при помощи алгоритма $\widetilde{\mathcal{A}}$, $\mathbb{P}\{S\}$ — вероятность события S, ε_n — относительная погрешность, δ_n — вероятность несрабатывания алгоритма $\widetilde{\mathcal{A}}$.

Алгоритм называется *асимптотически точным в классе задач* $\mathcal{K} = \bigcup\limits_{n=1}^{\infty} \mathcal{K}_n$, если существуют оценки $(\varepsilon_n, \delta_n)$ такие, что $\varepsilon_n \to 0$, $\delta_n \to 0$ при $n \to \infty$.

Решение задачи будем называть *допустимым*, если оно удовлетворяет всем ограничениям задачи (2)–(5). Поскольку рассматриваемые алгоритмы не всегда приводят к допустимым решениям, для задачи I_n размерности n определим индикаторную функцию

$$\nu_n(x) = \begin{cases} 1, & \text{если } x \text{ является допустимым решением,} \\ 0 & \text{в противном случае.} \end{cases}$$

Оценку вероятности несрабатывания алгоритма можно провести таким образом:

$$\mathbb{P}\left\{\frac{C(x^{\widetilde{\mathcal{A}}})-C(x^*)}{C(x^*)}>\varepsilon_n\right\}$$
$$=\mathbb{P}\left\{\frac{C(x^{\widetilde{\mathcal{A}}})-C(x^*)}{C(x^*)}>\varepsilon_n \;\middle|\; \nu_n(x^{\widetilde{\mathcal{A}}})=1\right\}\mathbb{P}\{\nu_n(x^{\widetilde{\mathcal{A}}})=1\}$$
$$+\mathbb{P}\left\{\frac{C(x^{\widetilde{\mathcal{A}}})-C(x^*)}{C(x^*)}>\varepsilon_n \;\middle|\; \nu_n(x^{\widetilde{\mathcal{A}}})=0\right\}\mathbb{P}\{\nu_n(x^{\widetilde{\mathcal{A}}})=0\}$$
$$\leqslant\mathbb{P}\left\{\frac{C(x^{\widetilde{\mathcal{A}}})-C(x^*)}{C(x^*)}>\varepsilon_n \;\middle|\; \nu_n(x^{\mathcal{A}})=1\right\}+\mathbb{P}\{\nu_n(x^{\widetilde{\mathcal{A}}})=0\}. \quad (6)$$

Другими словами, для доказательства асимптотической точности алгоритма достаточно доказать, что относительная погрешность построенного допустимого решения и вероятность построения недопустимого решения стремятся к нулю при $n\to\infty$.

В дальнейшем для решения задачи понадобится быстрый алгоритм поиска совершенного паросочетания в случайном двудольном графе. А именно, пусть дан двудольный граф $G(n,p)$ с множеством вершин $V=S\cup T$, $|S|=|T|=n$. Каждое ребро, соединяющее произвольную вершину $s\in S$ с вершиной $t\in T$, появляется независимо от других рёбер с вероятностью p. Необходимо найти совершенное паросочетание в $G(n,p)$, т. е. определить подмножество попарно не смежных рёбер, покрывающих все вершины графа $G(n,p)$. В [5] представлен асимптотически точный алгоритм $\mathcal{G}K$, который позволяет находить совершенное паросочетание в случайном графе $G(n,p)$, а также доказана следующая оценка.

Теорема 1. *При $p\geqslant\frac{\lambda\ln n}{n}$, $\lambda>4$, алгоритм $\mathcal{G}K$ с временно́й сложностью $O(n\log n)$ находит совершенное паросочетание в случайном двудольном графе $G(n,p)$ с вероятностью, стремящейся к 1 при $n\to\infty$. Вероятность несрабатывания $\mathcal{G}K$ на графах такого вида равна $O(n^{-\frac{\lambda-4}{6}})$.*

Стоит отметить, что в [11] также предлагался приближённый алгоритм отыскания совершенного паросочетания с аналогичной временно́й трудоёмкостью. Однако, оценки точности данного алгоритма установлены только в неявном виде, что существенно ограничивает возможность его аналитического использования.

Введём некоторые дополнительные обозначения. Обозначим через m_0 минимальное количество предприятий, необходимое для удовлетворения потребностей всех пунктов спроса. Другими словами, $m_0=\left\lceil\sum\limits_{i\in J}b_j/d\right\rceil$.

Если заданного подобным образом значения m_0 не существует, то очевидно, что задача является несбалансированной и решения не существует, т. е. суммарный спрос пунктов спроса больше суммарной мощности предприятий. Поэтому всюду далее будем считать, что $\sum\limits_{i\in I} d = md \geqslant \sum\limits_{j\in J} b_j$ и значение m_0 определено. Обозначим через I_0 первые m_0 предприятий множества I, $I_0 = \{1, \ldots, m_0\}$.

Положим $n = km_0 + t$ для некоторых целых $k > 0$, $m_0 > t \geqslant 0$, и разобьём множество J на $k+1$ подмножеств $J_1, \ldots, J_{k+1}$ следующим образом: $J_1 = \{1, \ldots, t\}$, $J_s = \{t + (s-2)m_0 + 1, \ldots, t + (s-1)m_0\}$, $2 \leqslant s \leqslant k+1$. Тем самым множество J_1 состоит из t элементов, а множества J_2, …, J_{k+1} состоят из m_0 элементов каждое.

1.3. Идея и описание алгоритма $\mathcal{A}$. Опишем общую идею алгоритма. Открыв предприятия в пунктах из I_0, получаем транспортную задачу по доставке продукта из пунктов множества I_0 в пункты множества J. Рассмотрим транспортные задачи (I_0, J_s), $1 \leqslant s \leqslant k+1$, в каждой из которых требуется поставить продукт из I_0 в J_s. Решение каждой задачи (I_0, J_s), $1 \leqslant s \leqslant k$, будем искать, используя только коммуникации единичной стоимости, причём ищем решение такое, что каждый производитель из I_0 обслуживает ровно одного потребителя из J_s, $1 \leqslant s \leqslant k$, и полностью удовлетворяет его спрос. Другими словами, в каждой строке и в каждом столбце таблицы (g_{ij}) для (I_0, J_s), $1 \leqslant s \leqslant k$, должен быть выбран ровно 1 элемент (рис. 1). Если таким образом построим решение для всех задач (I_0, J_s), $1 \leqslant s \leqslant k$, то построим оптимальное решение транспортной задачи $\left(I_0, \bigcup\limits_{s=1}^{k} J_s\right)$.

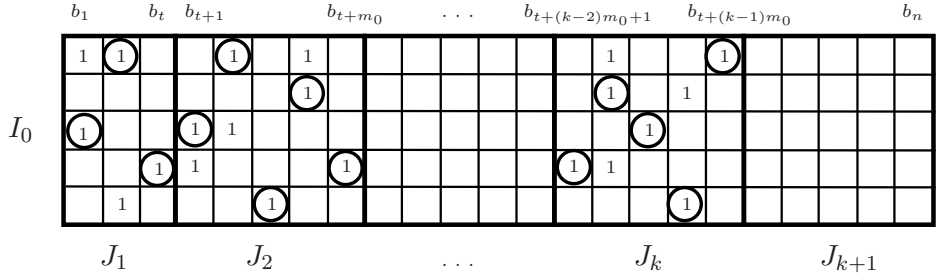

Рис. 1. Таблица $(g_{ij})_{i\in I_0,\, j\in J}$

Транспортную задачу на (I_0, J_{k+1}) будем решать, используя любой из известных полиномиальных алгоритмов. Объединение решений для задач (I_0, J_s), $1 \leqslant s \leqslant k+1$, даст решение задачи (I_0, J).

Теперь перейдём к более формальному описанию работы алгоритма $\mathcal{A}$.

Начало работы. Перед началом работы алгоритма имеем множества I и J, занумерованные так, что $g^0_1 \leqslant \ldots \leqslant g^0_m$ и $b_1 \leqslant \ldots \leqslant b_n$, числа m_0, k, t и множества $I_0, J_1, \ldots, J_{k+1}$, заданные следующим образом: $m_0 = \left\lceil \sum\limits_{i \in J} b_j/d \right\rceil$, $k = \lfloor n/m_0 \rfloor$, $t = n - km_0$, $I_0 = \{1, \ldots, m_0 \mid g^0_1 \leqslant \ldots \leqslant g^0_{m_0}\}$, $J_1 = \{1, \ldots, t\}$, $J_s = \{t + (s-2)m_0 + 1, \ldots, t + (s-1)m_0\}$, $2 \leqslant s \leqslant k+1$.

Шаг 1. Решаем транспортную задачу (I_0, J_1). Обозначим через $G_1 = (V_1; E_1)$ двудольный граф на множестве вершин $V_1 = I_0 \cup J_1$ с множеством рёбер $E_s = \{(i,j) \mid i \in I_0,\ j \in J_1,\ g_{ij} = 1\}$. Обозначим через J'_1 произвольное множество, состоящее из $m_0 - t$ вершин, и рассмотрим двудольный граф $G'_1 = (V'_1; E'_1)$ на множестве вершин $V'_1 = I_0 \cup (J_1 \cup J'_1)$ с множеством рёбер $E'_1 = E_1 \cup \{(i,j) \mid i \in I_0,\ j \in J'_1\}$, т. е. граф G'_1 получается из G_1 добавлением произвольных $m_0 - t$ вершин, каждая из которых соединяется рёбрами со всеми вершинами I_0. Поскольку в двудольном графе G'_1 обе доли состоят из одинакового числа вершин, применяем алгоритм $\mathcal{G}K$ для построения совершенного паросочетания в G'_1. Если паросочетание построить не удалось, то алгоритм останавливает свою работу с констатацией несрабатывания алгоритма. Если такое паросочетание удалось построить, то обозначим его через P'_1. Положим $P_1 = P'_1 \cap E_1$ и переходим на следующий шаг.

Замечание. Для упрощения понимания этого шага можно представить, что решаем транспортную задачу (I_0, J_1) следующим образом: добавляем множество J'_1 из $m_0 - t$ фиктивных клиентов, каждый из которых имеет нулевой спрос ($b_{j'} = 0,\ j' \in J'_1$) и находится на минимальном расстоянии до любого пункта производства из I_0 ($g_{ij'} = 1,\ i \in I_0,\ j' \in J'_1$). Очевидно, что оптимальное решение транспортной задачи (I_0, J_1) совпадает с оптимальным решением задачи $(I_0, J_1 \cup J'_1)$.

Шаг s, $2 \leqslant s \leqslant k$. Решаем транспортную задачу (I_0, J_s). Обозначим через $G_s = (V_s; E_s)$ двудольный граф на множестве вершин $V_s = I_0 \cup J_s$ с множеством рёбер $E_s = \{(i,j) \mid i \in I_0,\ j \in J_s,\ g_{ij} = 1\}$. Используем алгоритм $\mathcal{G}K$ для построения совершенного паросочетания в графе G_s. Если такое паросочетание удалось построить, то обозначим его через P_s и переходим на следующий шаг. Если паросочетания построить не удалось, то алгоритм останавливает свою работу с констатацией несрабатывания алгоритма.

Шаг $k+1$. Вычисляем значения $d'_i = d - \sum\limits_{\{j \mid (i,j) \in \bigcup\limits_{s=1}^{k} P_s\}} b_j$ для всех $i \in I_0$. Неформально говоря, d'_i — мощность, которая осталась у предприятия $i \in I_0$ после прохождения первых k шагов алгоритма (удовлетворения спроса клиентов из $J_1, \ldots, J_k$). Рассмотрим транспортную задачу (I_0, J_{k+1}), в которой потребности клиентов J_{k+1} заданы входными данными исходной задачи размещения, а мощности предприятий заданы как $d'_i, i \in I_0$. Находим допустимое решение этой задачи методом северо-западного угла (см., например, [7]). Обозначим это решение через $(\widetilde{x}_{ij},\ i \in I_0,\ j \in J_{k+1})$.

Завершение алгоритма. После успешного прохождения предыдущих $k+1$ шагов алгоритма имеем паросочетания $P_1, \ldots, P_k$ и допустимое решение $\widetilde{x}_{ij}$ задачи (I_0, J_{k+1}). Построим решение $x_i^{\mathcal{A}},\ x_{ij}^{\mathcal{A}}$ исходной задачи размещения следующим образом:

$$x_i^{\mathcal{A}} = \begin{cases} 1 & \text{при } i \in I_0, \\ 0 & \text{в противном случае,} \end{cases} \qquad x_{ij}^{\mathcal{A}} = \begin{cases} b_j & \text{при } (i,j) \in \bigcup\limits_{s=1}^{k} P_s, \\ \widetilde{x}_{ij} & \text{при } i \in I_0,\ j \in J_{k+1}, \\ 0 & \text{в остальных случаях.} \end{cases}$$

1.4. Анализ алгоритма $\mathcal{A}$. Обозначим через p вероятность появления единичного ребра в нашей исходной задаче (т. е. вероятность события $g_{ij} = 1$ для произвольных $i \in I,\ j \in J$). В силу введённых ранее определений $p = 1/r$.

Оценку погрешности и вероятности несрабатывания алгоритма будем проводить, используя неравенство (6). А именно, найдём оценки для обоих слагаемых в правой части неравенства.

Лемма 1. *При $p \geqslant \frac{\lambda \ln m_0}{m_0}$, $\lambda > 4$, и $m_0 = n^\theta$ для $\theta \in (6/(\lambda+2), 1)$ алгоритм $\mathcal{A}$ строит допустимое решение задачи с вероятностью, стремящейся к 1 при $n \to \infty$.*

Доказательство. Докажем, что $\mathbb{P}\{\nu_n(x^{\mathcal{A}}) = 0\} \to 0$ при $n \to \infty$. Для начала покажем, что алгоритм работает корректно и мощности предприятий можно разделить указанным в алгоритме способом. Убедимся, что для любого решения $x^{\mathcal{A}}$, построенного алгоритмом, будут выполняться ограничения (3): $\sum\limits_{j \in J} x_{ij}^{\mathcal{A}} \leqslant d,\ i \in I_0$.

В терминах нашего алгоритма достаточно показать, что на $(k+1)$-м шаге $d'_i \geqslant 0$ для всех $i \in I_0$, т. е. $\sum\limits_{j \in \bigcup\limits_{s=1}^{k} J_s} x_{ij}^{\mathcal{A}} \leqslant d,\ i \in I_0$.

По построению алгоритма $\mathcal{A}$ для фиксированного $i \in I_0$ решение $x_{ij}^{\mathcal{A}}$ на каждом множестве $J_s,\ 2 \leqslant s \leqslant k$, принимает нулевые значения во всех точках J_s, кроме одной, в которой принимает значение из множества $\{b_{t+(s-2)m_0+1}, b_{t+(s-2)m_0+2}, \ldots, b_{t+(s-1)m_0}\}$. На множестве J_1 решение $x_{ij}^{\mathcal{A}}$ принимает также нулевые значения во всех точках J_1, кроме, может быть, одной, в которой принимает значение из множества $\{b_1, b_2, \ldots, b_t\}$. Поскольку $b_1 \leqslant \ldots \leqslant b_n$, имеем

$$\sum_{j \in \bigcup\limits_{s=1}^{k} J_s} x_{ij}^{\mathcal{A}} \leqslant b_t + b_{t+m_0} + \ldots + b_{t+(k-1)m_0}.$$

В силу согласованности задачи верно неравенство $\sum\limits_{j \in J} b_j \leqslant m_0 d$.

В то же время

$$\sum_{j \in J} b_j = \sum_{j \in J_1} b_j + \sum_{j \in J_2} b_j + \ldots + \sum_{j \in J_{k+1}} b_j \geqslant \sum_{j \in J_1} b_1 + \sum_{j \in J_2} b_{t+1} + \ldots + \sum_{j \in J_{k+1}} b_{t+(k-1)m_0+1} = tb_1 + m_0 b_{t+1} + \ldots + m_0 b_{t+(k-1)m_0+1} \geqslant m_0(b_t + b_{t+m_0} + \ldots + b_{t+(k-1)m_0}).$$

Таким образом, $b_t + b_{t+m_0} + \ldots + b_{t+(k-1)m_0} \leqslant d$, следовательно,

$$\sum_{j \in \bigcup\limits_{s=1}^{k} J_s} x_{ij}^{\mathcal{A}} \leqslant d, \quad i \in I_0.$$

Поэтому ограничения (3) не нарушаются ни на одном из шагов и алгоритм работает корректно.

Получение недопустимого решения возможно только на этапе решения подзадач $(I_0, J_s),\ 1 \leqslant s \leqslant k$, при использовании алгоритма $\mathcal{G}K$. Поскольку выполнены все условия теоремы 1, вероятность несрабатывания алгоритма $\mathcal{G}K$ для каждой задачи (I_0, J_s) оценивается величиной $O(m_0^{-(\lambda-4)/6})$. В силу независимости событий успеха решения каждой из k подзадач $(I_0, J_s),\ 1 \leqslant s \leqslant k$, напрямую следующей из независимости

элементов входной матрицы транспортных затрат, имеет место оценка

$$\mathbb{P}\{\nu_n(x^{\mathcal{A}})=0\} = kO(m_0^{-\frac{\lambda-4}{6}}) \leqslant O\left(\frac{n}{m_0^{\frac{\lambda+2}{6}}}\right) = O(n^{1-\theta\frac{\lambda+2}{6}}).$$

Замечание. Очевидно, что вероятность несрабатывания алгоритма $\mathcal{GK}$ на первом шаге алгоритма $\mathcal{A}$ не превосходит вероятности несрабатывания на последующих шагах, поскольку из m_0 вершин в $J_1 \cup J_1'$ у $m - t_0$ вершин с вероятностью 1 существует ребро, соединяющее их с вершинами из I_0, в то время как на последующих шагах s, $2 \leqslant s \leqslant k$, все рёбра между m_0 вершинами J_s и I_0 существуют лишь с вероятностью $p \leqslant 1$.

Таким образом, поскольку $1-\theta\frac{\lambda+2}{6} < 0$ в силу условия леммы, имеет место

$$\mathbb{P}\{\nu_n(x^{\mathcal{A}})=0\} \to 0 \quad \text{при } n \to \infty.$$

Лемма 1 доказана.

Докажем, что $\mathbb{P}\left\{\frac{C(x^{\mathcal{A}})-C(x^*)}{C(x^*)} > \varepsilon_n \mid \nu_n(x^{\mathcal{A}})=1\right\}$ также стремится к 0 при $n \to \infty$. Обозначим через $B(n)$ максимальную потребность пунктов спроса, $B(n) = \max\limits_{j\in J} b_j$.

Лемма 2. *При $p \geqslant \frac{\lambda \ln m_0}{m_0}$, $\lambda > 10$, $m_0 = n^\theta$ для $\theta \in \left(\frac{6}{\lambda+2}, \frac{1}{2}\right)$ и $B(n) = o(n^{1-2\theta})$ относительная погрешность построенного алгоритмом $\mathcal{A}$ допустимого решения стремится к 0 при $n \to \infty$.*

Доказательство. Отметим, что по построению алгоритма $\mathcal{A}$ верна следующая оценка стоимости решения:

$$C(x^{\mathcal{A}}) = \sum_{i=1}^{m} g_i^0 x_i^{\mathcal{A}} + \sum_{i=1}^{m} \sum_{j\in \bigcup\limits_{s=1}^{k} J_s} x_{ij}^{\mathcal{A}} g_{ij} + \sum_{i=1}^{m} \sum_{j\in J_{k+1}} x_{ij}^{\mathcal{A}} g_{ij}$$

$$\leqslant \sum_{i=1}^{m_0} g_i^0 + \sum_{j\in \bigcup\limits_{s=1}^{k} J_s} b_j + \sum_{j\in J_{k+1}} b_j r.$$

Оптимальное решение, очевидно, оценивается снизу таким образом:

$$C(x^*) = \sum_{i=1}^{m} g_i^0 x_i^* + \sum_{i=1}^{m}\sum_{j\in J} x_{ij}^* g_{ij} \geqslant \sum_{i=1}^{m_0} g_i^0 + \sum_{j\in J} b_j.$$

Тогда для относительной погрешности построенного решения имеем

$$\frac{C(x^{\mathcal{A}})-C(x^*)}{C(x^*)} \leqslant \frac{\sum\limits_{j\in J_{k+1}} b_j r - \sum\limits_{j\in J_{k+1}} b_j}{\sum\limits_{j\in J} b_j} \leqslant r\frac{\sum\limits_{j\in J_{k+1}} b_j}{\sum\limits_{j\in J} b_j}.$$

Учитывая, что $b_j \geqslant 1,\ j \in J$, последнее неравенство можно переписать в виде

$$r\frac{\sum\limits_{j\in J_{k+1}} b_j}{\sum\limits_{j\in J} b_j} \leqslant r\frac{m_0 B(n)}{n} \leqslant \frac{m_0}{\lambda \log m_0}\frac{m_0 B(n)}{n} \leqslant \frac{m_0^2 B(n)}{n} \leqslant \frac{B(n)}{n^{1-2\theta}}.$$

В силу условия леммы $1-2\theta > 0$. Таким образом, получаем

$$\frac{C(x^{\mathcal{A}})-C(x^*)}{C(x^*)} \leqslant \frac{B(n)}{n^{1-2\theta}} = \frac{o(n^{1-2\theta})}{n^{1-2\theta}} \to 0 \quad \text{при } n\to\infty.$$

Лемма 2 доказана.

Из лемм 1 и 2 непосредственно вытекает справедливость следующей теоремы, которая указывает условия асимптотической точности алгоритма.

Теорема 2. *При* $p \geqslant \frac{\lambda \ln m_0}{m_0}$, $\lambda > 10$, $m_0 = n^\theta$ *для* $\theta \in \left(\frac{6}{\lambda+2}, \frac{1}{2}\right)$ *и* $B(n) = o(n^{1-2\theta})$, *алгоритм* $\mathcal{A}$ *решения задачи размещения* (1)–(5) *на входных данных, заданных случайными величинами из класса* $U[1,r]$, *имеет трудоёмкость* $O(n\log m_0)$, *относительную погрешность* $\frac{B(n)}{n^{1-2\theta}}$ *и вероятность несрабатывания* $O(n^{1-\theta\frac{\lambda+2}{6}})$.

Доказательство. Действительно, в силу леммы 1 существует последовательность $\delta_n^1 \to 0$ при $n\to\infty$ такая, что $\mathbb{P}\{\nu_n(x^{\mathcal{A}})=0\} \leqslant \delta_n^1$. Более конкретно, согласно доказательству леммы 1 в качестве оценки δ_n^1 можно взять $O(n^{1-\theta\frac{\lambda+2}{6}})$.

В силу леммы 2 имеем

$$\mathbb{P}\left\{\frac{C(x^{\mathcal{A}})-C(x^*)}{C(x^*)} > \frac{B(n)}{n^{1-2\theta}} \ \mid \nu_n(x^{\mathcal{A}})=1\right\} = 0.$$

Выберем в качестве оценок относительной погрешности и вероятности несрабатывания алгоритма $\mathcal{A}$ последовательности $\varepsilon_n = B(n)/n^{1-2\theta}$ и $\delta_n = \delta_n^1$. В силу условия теоремы и предыдущих рассуждений $\varepsilon_n \to 0$, $\delta_n \to 0$ при $n\to\infty$, а также выполняется оценка

$$\mathbb{P}\left\{\frac{C(x^{\mathcal{A}})-C(x^*)}{C(x^*)}>\varepsilon_n\right\}\leqslant\mathbb{P}\left\{\frac{C(x^{\mathcal{A}})-C(x^*)}{C(x^*)}>\varepsilon_n\ \mid\nu_n(x^{\mathcal{A}})=1\right\}$$
$$+\,\mathbb{P}\{\nu_n(x^{\mathcal{A}})=0\}\leqslant 0+\delta_n^1=\delta_n.$$

Тем самым алгоритм $\mathcal{A}$ асимптотически точный с оценками (ε_n,δ_n).

Трудоёмкость каждого из k первых шагов алгоритма $\mathcal{A}$ по теореме 1 не превышает $O(m_0\log m_0)$, трудоёмкость $(k+1)$-го шага не превышает $O(m_0)$, и трудоёмкость всего алгоритма оценивается величиной $kO(m_0\log m_0)\leqslant O(n\log m_0)$. Теорема 2 доказана.

2. Задача с произвольными ограничениями на объёмы производства

В предыдущих разделах рассмотрена задача размещения (1)–(5), где условие (5) накладывает одинаковое ограничение на производственные мощности всех предприятий. Большой интерес представляет и общая постановка задачи, когда предприятия могут иметь различные мощности. В этом разделе приведём модификацию алгоритма $\mathcal{A}$, которая будет находить решение задачи с произвольными мощностными ограничениями.

Таким образом, рассматриваем задачу размещения (1)–(4)

$$C(x)=\sum_{i=1}^m g_i^0x_i+\sum_{i=1}^m\sum_{j=1}^n g_{ij}x_{ij}\longrightarrow\min_{x_i,x_{ij}},$$

$$\sum_{i=1}^m x_{ij}=b_j,\quad j\in J,$$

$$\sum_{j=1}^n x_{ij}\leqslant d_ix_i,\quad i\in I,$$

$$x_{ij}\geqslant 0,\quad x_i\in\{0,1\},\quad i\in I,\ j\in J,$$

с тем же предположением относительно вероятностного распределения значений транспортных расходов, что и ранее: $g_{ij},\ i\in I,\ j\in J,$ — значения, заданные независимыми случайными величинами из класса $U[1,r]$.

Без ограничения общности будем считать, что предприятия и пункты спроса занумерованы так, что $d_1\leqslant\ldots\leqslant d_m$ и $b_1\leqslant b_2\leqslant\ldots\leqslant b_n$.

2.1. Описание алгоритма. Дадим описание алгоритма $\widetilde{\mathcal{A}}$ решения данной задачи.

Начало работы. Найдём множество I_0, заданное следующим образом:

$$\sum_{i\in I_0} g_i^0 = \min_{I'\subset I}\Big\{\sum_{i\in I'} g_i^0 \mid \sum_{i\in I'} d_i \geqslant \sum_{j\in J} b_j\Big\}.$$

Это частный случай обратной задачи о ранце и для её решения будем использовать метод динамического программирования (см., например, [4]). Без ограничения общности перенумеруем предприятия таким образом, что $I_0 = \{1,\dots,m_0\}$ для некоторого $m_0 \leqslant m$ и $d_1 \leqslant \dots \leqslant d_{m_0}$.

Шаг 1. Находим значения $d'_i = d_i - d_1,\ i \in I_0$,

$$n' = \arg\max_s \left\{ s \leqslant n \mid \sum_{i\in I_0} d'_i \leqslant \sum_{j=s}^{n} b_j \right\}.$$

Обозначим через J' множество пунктов спроса $\{n', n'+1,\dots,n\}$. Рассмотрим транспортную задачу (I_0, J'), в которой потребности клиентов заданы как $b'_j = b_j,\ j \in J'$, а мощности предприятий заданы как $d'_i, i \in I_0$. Находим допустимое решение этой задачи методом северо-западного угла, в результате чего останется только один клиент из J', чья потребность не была удовлетворена в полном объёме. Обозначим это решение через $x'_{ij},\ i \in I_0,\ j \in J'$.

Шаг 2. Рассмотрим транспортную задачу (I_0, J''), заданную следующим образом: $I_0 = \{1,\dots,m_0\}$, $J'' = \{1,\dots,n'\}$, $d''_i = d_1,\ i \in I_0$,

$$b''_j = \begin{cases} b_j & \text{при } j < n', \\ \sum\limits_{i=n'}^{n} b_j - \sum\limits_{i\in I_0} d'_i & \text{при } j = n'. \end{cases}$$

Задача (I_0, J'') является задачей размещения с одинаковыми ограничениями на объёмы производства. Поэтому для её решения применяем алгоритм $\mathcal{A}$, построенный в разд. 1. Обозначим полученное решение через x''_{ij}, $i \in I_0,\ j \in J''$.

Завершение алгоритма. После успешного прохождения предыдущих шагов алгоритма имеем допустимые решения соответствующих подзадач $x'_{ij},\ i \in I_0,\ j \in J'$, и $x''_{ij},\ i \in I_0,\ j \in J''$.

Построим теперь решение $x_i^{\widetilde{\mathcal{A}}}, x_{ij}^{\widetilde{\mathcal{A}}}$ исходной задачи размещения следующим образом:

$$x_i^{\widetilde{\mathcal{A}}} = \begin{cases} 1 & \text{при } i \in I_0, \\ 0 & \text{в противном случае,} \end{cases}$$

$$x_{ij}^{\widetilde{\mathcal{A}}} = \begin{cases} x''_{ij} & \text{при } i \in I_0,\ j < n', \\ x'_{ij} + x''_{ij} & \text{при } i \in I_0,\ j = n', \\ x'_{ij} & \text{при } i \in I_0,\ j > n', \\ 0 & \text{в остальных случаях.} \end{cases}$$

На этом работа алгоритма завершена.

2.2. Анализ алгоритма. Определим условия, при которых построенный алгоритм будет асимптотически точным. Для этого введём дополнительные обозначения: обозначим через $D_{\max}(n)$ и $D_{\min}(n)$ максимальную и минимальную мощность предприятий из I соответственно (в силу нашего предположения $D_{\max}(n) = d_n$ и $D_{\min}(n) = d_1$).

Теорема 3. *При* $p \geqslant \frac{\lambda \ln m_0}{m_0}$, $\lambda > 10$, $m_0 = n^\theta$ *для* $\theta \in \left(\frac{6}{\lambda+2}, \frac{1}{2}\right)$ *и* $B(n) = o(n^{1-2\theta})$, $D_{\max}(n) - D_{\min}(n) = o(n^{1-2\theta})$ *алгоритм* $\widetilde{\mathcal{A}}$ *решения задачи размещения* (1)–(4) *на входных данных, заданных случайными величинами из класса* $U[1, r]$, *имеет трудоёмкость* $O(n^{2(1-\theta)}m)$, *относительную погрешность* $\frac{B(n)+(D_{\max}(n)-D_{\min}(n))}{n^{1-2\theta}}$ *и вероятность несрабатывания* $O(n^{1-\theta\frac{\lambda+2}{6}})$.

Доказательство. Очевидно, что вероятность построения недопустимого решения стремится к 0 (рассуждения полностью аналогичны доказательству леммы 1). Покажем, что относительная погрешность алгоритма также стремится к 0 с ростом n.

По построению алгоритмов $\widetilde{\mathcal{A}}$ и $\mathcal{A}$ верна следующая оценка стоимости решения:

$$C(x^{\widetilde{\mathcal{A}}}) = \sum_{i=1}^{m} g_i^0 x_i^{\widetilde{\mathcal{A}}} + \sum_{i=1}^{m}\sum_{j=1}^{n} g_{ij} x_{ij}^{\widetilde{\mathcal{A}}} = \sum_{i=1}^{m_0} g_i^0 + \sum_{i=1}^{m_0}\sum_{j=1}^{n'} g_{ij} x_{ij}^{\widetilde{\mathcal{A}}}$$
$$+ \sum_{i=1}^{m}\sum_{j=n'+1}^{n} g_{ij} x_{ij}^{\widetilde{\mathcal{A}}} \leqslant \sum_{i=1}^{m_0} g_i^0 + \sum_{j=1}^{n'-m_0} b_j + \sum_{j=n'-m_0+1}^{n'} r b_j + \sum_{j=n'+1}^{n} r b_j.$$

Оптимальное решение оценивается снизу:

$$C(x^*) = \sum_{i=1}^{m} g_i^0 x_i^* + \sum_{i=1}^{m}\sum_{j=1}^{n} g_{ij} x_{ij}^* \geqslant \sum_{i=1}^{m_0} g_i^0 + \sum_{j=1}^{n} b_j.$$

Для относительной погрешности построенного решения имеем следующую цепочку соотношений:

$$\frac{C(x^{\widetilde{\mathcal{A}}}) - C(x^*)}{C(x^*)} \leqslant \frac{\sum\limits_{j=1}^{n'-m_0} b_j + \sum\limits_{j=n'-m_0+1}^{n'} rb_j + \sum\limits_{j=n'+1}^{n} rb_j - \sum\limits_{j=1}^{n} b_j}{\sum\limits_{j=1}^{n} b_j}$$

$$= \frac{(r-1)\sum\limits_{j=n'-m_0+1}^{n'} b_j + (r-1)\sum\limits_{j=n'+1}^{n} b_j}{\sum\limits_{j=1}^{n} b_j} \leqslant (r-1)\frac{m_0 B(n) + \sum\limits_{i=1}^{m_0} d'_i}{\sum\limits_{j=1}^{n} b_j}$$

$$\leqslant \frac{m_0}{\lambda \log m_0}\frac{m_0 B(n) + m_0(D_{\max}(n) - D_{\min}(n))}{\sum\limits_{j=1}^{n} b_j}$$

$$\leqslant \frac{m_0^2 B(n)}{n} + \frac{m_0^2(D_{\max}(n) - D_{\min}(n))}{n}.$$

В силу условий теоремы верно неравенство

$$\frac{C(x^{\widetilde{\mathcal{A}}}) - C(x^*)}{C(x^*)} \leqslant \frac{B(n) + (D_{\max}(n) - D_{\min}(n))}{n^{1-2\theta}} \leqslant \frac{o(n^{1-2\theta})}{n^{1-2\theta}}.$$

Таким образом, относительная погрешность построенного алгоритма стремится к 0 с ростом n.

Трудоёмкость $\widetilde{\mathcal{A}}$ определяется начальным шагом алгоритма, когда находим множество I_0 и значение m_0 (трудоёмкость остальных шагов алгоритма не превышает трудоёмкости $\mathcal{A}$). Применяем метод динамического программирования, который имеет трудоёмкость порядка $O\big(\sum\limits_{j\in J} b_j m\big)$ (см., например, [4]). В силу условий теоремы

$$\sum_{j\in J} b_j m \leqslant nB(n)m \leqslant o(n^{2-2\theta}m).$$

Таким образом, трудоёмкость алгоритма $\widetilde{\mathcal{A}}$ не превышает $O(n^{2(1-\theta)}m)$. Теорема 3 доказана.

Заключение.

В данной работе рассматривалась задача размещения производства на входных данных, заданных независимыми случайными величинами, имеющими равномерное распределение на целочисленном сегменте. Проведены исследования двух постановок задачи — с одинаковыми и произвольными ограничениями на объёмы производства предприятий. Для

обеих постановок построены быстрые приближённые алгоритмы решения и определены условия их асимптотической точности. Дальнейший интерес представляет рассмотрение более общих случаев распределения входных данных. В частности, равномерное распределение на отрезке, нормальное распределение и произвольное распределение с заданными вероятностными характеристиками.

ЛИТЕРАТУРА

1. **Агеев А. А., Гимади Э. Х., Курочкин А. А.** Полиномиальный алгоритм решения задачи размещения на цепи с одинаковыми производственными мощностями предприятий // Дискрет. анализ и исслед. операций. — 2009. — Т. 16, № 5. — С. 3–18.
2. **Вознюк 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. — Т. 8, № 2. — С. 3–16.
3. **Гимади Э. Х.** Эффективный алгоритм решения задачи размещения с областями обслуживания, связными относительно ациклической сети // Управляемые системы. — 1983. — Вып. 23. — С. 12–23.
4. **Гимади Э. Х.** Математические модели и методы исследования операций. Учебн. пособие. — Новосибирск: СибГУТИ, 2009. — 122 с.
5. **Гимади Э. Х., Курочкин А. А.** Одна задача размещения с одинаковыми объёмами производства на случайных входных данных // Вестн. Новосиб. гос. ун-та. — 2011. — Т. 11, вып. 1. — С. 15–34.
   Gimadi E. Kh., Kurochkin A. A. Uniform capacitated facility location problem with random input data // J. Math. Sci. — 2013. — Vol. 188, N 4. — P. 359–377.
6. **Гимади Э. Х., Перепелица В. А.** Асимптотический подход к решению задачи коммивояжера // Сб. научн. тр. — Новосибирск: Ин-т математики СО АН СССР, 1974. — Вып. 12. — С. 35–45.
7. **Гольштейн Е. Г., Юдин Д. Б.** Задачи линейного программирования транспортного типа. — М.: Наука, 1969. — 494 с.
8. **Ageev A. A.** A polynomial-time algorithm for the facility location problem with uniform hard capacities on path graphs // Proc 2nd Int. Workshop Discrete Optimization Methods in Production and Logistics (DOM'2004, Omsk–Irkutsk, July 20–27, 2004). — Omsk: Nasledie; Dialog-Sibir, 2004. — P. 28–32.
9. **Aikens C. H.** Facility location models for distribution planning // Eur. J. Oper. Res. — 1985. — Vol. 22, N 3. — P. 263–279.
10. **Akinc U., Khumawala B. M.** An efficient branch and bound algorithm for the capacitated warehouse location problem // Manage. Sci. — 1977. — Vol. 23, N 6. — P. 585–594.

11. Angluin D., Valiant L. G. Fast probabilistic algorithms for Hamiltonian circuits and matchings // J. Comput. Syst. Sci. — 1979. — Vol. 18, N 2. — P. 155–193.

12. Chudak F., Williamson D. Improved approximation algorithms for capacitated facility location problems // Math. Program. — 2005. — Vol. 102, N 2. — P. 207–222.

13. Garey M. R., Johnson D. S. Computers and intractablity. — San Francisco: Freeman, 1979. — 340 p.

14. Karp R. M. The probabilistic analysis of some combinatorial algorithms // Algorithms and complexity: new directions and recent results. — New York: Acad. Press, 1976. — P. 1–19.

15. Korupolu M., Plaxton C., Rajaraman R. Analysis of a local search heuristic for facility location problems // J. Algorithms. — 2000. — Vol. 37, N 1. — P. 146–188.

16. Kuehn A. A., Hamburger M. J. A heuristic program for locating warehouses // Manage. Sci. — 1963. — Vol. 9, N 4. — P. 643–66.

17. Piersma N. A probabilistic analysis of the capacitated facility location problem // J. Comb. Optim. — 1999. — Vol. 3, N 1. — P. 31–50.

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

Статья поступила
28 октября 2013 г.

Переработанный вариант —
5 мая 2014 г.