УДК 519.8

Э. Х. Гимади, А. А. Курочкин

ОДНА ЗАДАЧА РАЗМЕЩЕНИЯ С ОДИНАКОВЫМИ ОБЪЕМАМИ ПРОИЗВОДСТВА НА СЛУЧАЙНЫХ ВХОДНЫХ ДАННЫХ*

Рассматривается задача размещения с одинаковыми объемами производства при некоторых специальных ограничениях на объемы спроса клиентов и число открываемых предприятий. Предполагается, что элементы матрицы транспортных расходов (g_{ij}) — независимые случайные величины с равномерной функцией распределения на целочисленном сегменте $[1, r]$. Построен приближенный алгоритм решения задачи и проведен вероятностный анализ его работы. Представлены условия, при которых алгоритм является асимптотически точным и имеет временную сложность $O(n \ln m)$, где n — число потребителей, m — число возможных пунктов производства.

Ключевые слова: задача размещения, транспортная задача, граф со случайными ребрами, совершенное паросочетание, неравенство Чебышева, теорема Петрова, асимптотически точный алгоритм.

Введение

Известно, что задача размещения является NP-трудной даже без ограничений на объемы производства [1]. Для задачи размещения с ограничениями на объемы производства актуальны исследования в двух направлениях: построение приближенных полиномиальных алгоритмов с оценками качества и поиск специальных подклассов задач, когда возможно построение точных полиномиальных или псевдополиномиальных алгоритмов. В первом направлении достигнуты некоторые успехи в классе задач со случайными входными данными. Например, в работе [2] проведено исследование определенных задач из этого класса. Однако получить условия асимптотической точности представленного в этой работе алгоритма удалось при введении достаточно жестких ограничений на входные данные. Во втором направлении, например для задачи размещения на путевом графе с одинаковыми объемами производства, в работе [3] был построен полиномиальный алгоритм с трудоемкостью $O(m^4 n^2)$, где n — число потребителей, m — число возможных пунктов размещения производства. Для задач без ограничений на объемы производства удается построить алгоритмы гораздо меньшей трудоемкости. Так, например, для задачи размещения без ограничений на объемы производства на древовидных сетях в статье [4] представлен алгоритм с временной сложностью $O(nm)$.

В настоящей работе рассматривается специальный подкласс задачи размещения с одинаковыми объемами производства на случайных входных данных. Для решения рассматриваемой специальной задачи предлагается схема, в которой в качестве подзадачи

* Работа выполнена при поддержке РФФИ (проекты 08-01-00516, 09-01-00032 и 10-07-00195), целевой программы Президиума РАН (проект № 227), междисциплинарного интеграционного проекта СО РАН № 30, а также целевой программы СО РАН (интеграционный проект № 44)

ISSN 1818-7897. Вестник НГУ. Серия: Математика, механика, информатика. 2011. Т. 11, вып. 1. С. 15–34

требуется отыскивать план перевозок от предприятий к потребителям, что является предметом решения задачи транспортного типа. Для последней задачи известны точные полиномиальные алгоритмы. Например, в статье [5] предложен алгоритм, имеющий трудоемкость $O(mn^2 \log n + n^2 \log^2 n)$; в [6] представлен другой алгоритм с трудоемкостью $O(m \log m(K + n \log n))$ (K — число ребер графа). В данной работе предлагается метод отыскания плана перевозок, который, вообще говоря, не всегда находит решение, но является достаточно быстрым (имеет на порядки меньшую трудоемкость, чем перечисленные выше точные алгоритмы). Речь идет о построении совершенного паросочетания в случайном графе, ребрам которого соответствуют коммуникации минимальной стоимости. При нахождении такого паросочетания получаем минимальные затраты на перевозку. Стоит отметить, что задача поиска совершенного паросочетания в n-вершинном графе, между любыми двумя вершинами которого с вероятностью p существует ребро, исследовалась в [7], где был построен приближенный алгоритм решения с трудоемкостью $O(n \log n)$. В данной работе мы предлагаем другой алгоритм решения этой задачи, который имеет ту же трудоемкость, но является гораздо более простым в понимании и использовании.

При анализе алгоритма будут использованы следующие определения и обозначения из [8].

Алгоритм $\mathcal{A}$ имеет оценки $(\varepsilon_n, \delta_n)$ в классе $\mathcal{K}_n$ задачи минимизации размерности n, если для каждого n выполнено неравенство $\mathbb{P}\{f_{\mathcal{A}}(I) > (1 + \varepsilon_n) f^*(I)\} \leq \delta_n$, где $f^*(I)$ — оптимальное решение для индивидуальной задачи I; $f_{\mathcal{A}}(I)$ — решение, полученное при помощи алгоритма $\mathcal{A}$; $\mathbb{P}\{J\}$ — вероятность события J; ε_n — относительная погрешность; δ_n — вероятность несрабатывания алгоритма $\mathcal{A}$.

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

Основная цель работы состоит в построениии быстрого (с временной сложностью $O(n \ln m)$) асимптотически точного алгоритма решения задачи на случайных входных данных. Более определенно, такой алгоритм почти всегда находит точное решение (т. е. $\varepsilon_n = 0$) с вероятностью несрабатывания $\delta_n \to 0$ при $n \to \infty$.

1. Описание задачи

Пусть $J = \{1, \ldots, n\}$ и $I = \{1, \ldots, m\}$ — множество пунктов потребления некоторого продукта и множество возможных пунктов размещения предприятий соответственно. Известны затраты g_i^0 на размещение предприятия и ограничения a_i на объемы производства в пункте $i \in I$. Для каждого пункта потребления $j \in J$ задан требуемый объем 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:

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

$$\sum_{j=1}^{n} x_{ij} \le a_i x_i,\ i \in I, \tag{3}$$

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

Сделаем некоторые дополнительные предположения. Будем считать, что объемы производства всех предприятий одинаковые:

$$a_i = a,\ i \in I. \tag{5}$$

Кроме этого, предполагаем, что транспортные расходы g_{ij} — независимые случайные величины, равновероятно принимающие значения из целочисленного сегмента $[1, r]$, r – целое, $i \in I,\ j \in J$. Обозначим $p = 1/r$.

Считаем также, что существует такое целое число $m_0,\ 1 \le m_0 \le m$, что функция спроса кусочно постоянна с точками излома, кратными m_0:

$$\begin{aligned} &b_1 = b_2 = \ldots = b_{m_0} \ge b_{m_0+1} = b_{m_0+2} = \ldots = b_{2m_0} \ge \\ &\qquad \ge b_{2m_0+1} = \ldots = b_{km_0} \ge b_{km_0+1} = \ldots = b_n, \text{ где } k = \left\lfloor \frac{n}{m_0} \right\rfloor \end{aligned} \tag{6}$$

и

$$m_0 a = \sum_{j \in J} b_j. \tag{7}$$

2. Некоторые замечания

Из условия (7) рассматриваемой задачи (1)–(7) заключаем, что для удовлетворения потребностей всех клиентов понадобится как минимум m_0 пунктов производства. Без ограничения общности можно считать $g_1^0 \le g_2^0 \le \ldots \le g_m^0$. В силу того, что все $g_{ij},\ i \in I,\ j \in J$, независимые, одинаково распределенные случайные величины, и все заводы имеют одинаковую мощность, для минимизации общих затрат выгоднее строить те предприятия, которые имеют меньшую стоимость открытия. Таким образом, открываем первые m_0 предприятий, $I_0 = \{1, \ldots, m_0\}$.

Для дальнейшего упрощения выкладок будем считать $k = \left\lfloor \frac{n}{m_0} \right\rfloor = \frac{n}{m_0}$. Если это не так, т. е. $\left\lfloor \frac{n}{m_0} \right\rfloor < \frac{n}{m_0}$, то можно добавить $(k+1)m_0 - n$ фиктивных потребителей, имеющих нулевые объемы спроса $b_{n+1} = b_{n+2} = b_{(k+1)m_0} = 0$. Точное решение новой задачи будет совпадать с решением исходной.

Открыв предприятия в пунктах из I_0, мы получаем транспортную задачу по доставке продукта из пунктов множества I_0 в пункты множества J. Разобьем множество J на подмножества $J_1, J_2, \ldots, J_k$, где

$$J_l = \{(l-1)m_0+1, (l-1)m_0+2, \ldots, lm_0\},\ 1 \le l \le k.$$

В силу (6) для всех $j \in J_l,\ b_j = b_{(l-1)m_0+1}$. Рассмотрим транспортные задачи (I_0, J_l), $1 \le l \le k$, в каждой из которых требуется поставить продукт из I_0 в J_l. Как уже отмечалось, транспортная задача является полиномиально разрешимой. В данной работе предлагается приближенный алгоритм $\widetilde{\mathcal{A}}$ решения задачи (I_0, J), который, вообще говоря, не всегда находит решение, но является достаточно быстрым и имеет на порядки меньшую трудоемкость, чем известные точные алгоритмы. Решение каждой подзадачи (I_0, J_l) будем искать, используя только минимальные коммуникации единичной стоимости. Причем ищем такое решение, в котором каждый производитель из I_0 обслуживает ровно одного потребителя из $J_l,\ 1 \le l \le k$, и обслуживает его в полном объеме. Другими словами, в каждой строке и в каждом столбце таблицы (g_{ij}) для $(I_0, J_l),\ 1 \le l \le k$ должен быть выбран ровно 1 элемент (рис. 1).

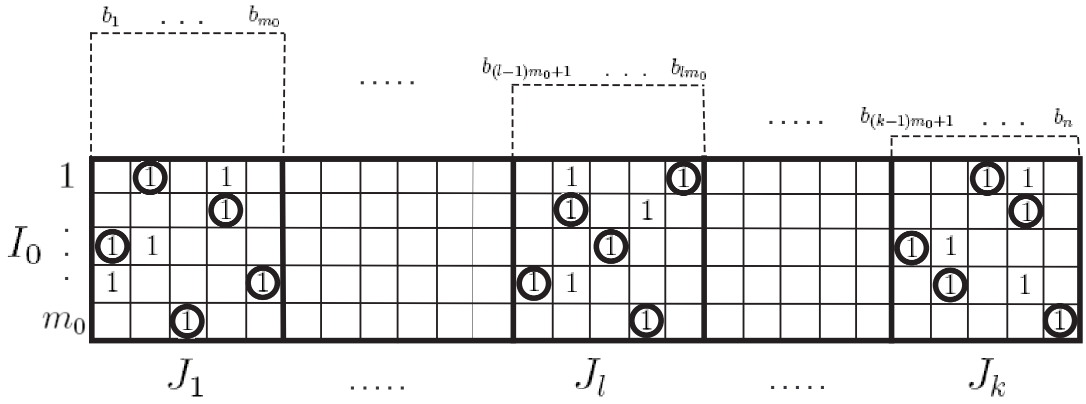

Рис. 1. Таблица $(g_{ij})_{i \in I_0, j \in J}$

Для каждого $l,\ 1 \le l \le k$ рассмотрим случайный двудольный граф с множеством вершин $V_l = I_0 \cup J_l,\ |I_0| = m_0,\ |J_l| = m_0$. Каждое ребро, соединяющее вершину $i \in$ $\in I_0$ с вершиной $j \in J_l$, появляется независимо от других ребер с вероятностью p (это есть вероятность того, что $g_{ij} = 1$). Если найдено совершенное паросочетание данного графа, то получено решение задачи (I_0, J_l). Если найдены совершенные паросочетания для всех $l,\ 1 \le l \le k$, то автоматически получаем решение исходной задачи. Каждый пункт спроса $j \in J_l,\ 1 \le l \le k$, обслуживается в полном объеме из пункта производства по ребру единичной стоимости, входящему в одно из построенных совершенных паросочетаний. Причем в силу условия (6) в искомом решении каждое предприятие $i \in I_0$ должно удовлетворять спрос, равный $b_1 + b_{m_0+1} + \ldots + b_{(k-1)m_0+1}$, т. е. обслуживать в полном объеме по одному потребителю из каждого $J_l,\ l = 1, \ldots, k$. Согласно (7) мощность каждого предприятия $i \in I_0$ в точности совпадает с этой суммой. Итак построенное описанным выше образом решение будет удовлетворять всем ограничениям (2)–(7) и будет доставлять минимум целевой функции (1).

Далее опишем приближенный алгоритм $\mathcal{A}$ поиска совершенного паросочетания в произвольном случайном графе. Данная задача интересна сама по себе и имеет множество применений. Также будут найдены условия, при которых алгоритм является асимптотически точным. Кроме того, рассмотрим работу построенного алгоритма на ориентированном графе и двудольном случайном графе, найдем вероятностные оценки качества его работы.

3. Поиск совершенного паросочетания в произвольном случайном графе. Описание алгоритма. Трудоемкость

Дан неориентированный граф G со множеством вершин V, $|V| = n$. Ребро между любыми двумя вершинами G независимо от остальных появляется с вероятностью p.

Опишем приближенный алгоритм $\mathcal{A}$ отыскания совершенного паросочетания графа G. Алгоритм состоит из двух этапов:

Этап 1. Выделяем в G путь L длины $n - m - 1$, здесь m — вспомогательная величина, значение которой определим позже. В таком пути содержится $n - m$ вершин. Считаем, что изначально все вершины графа покрашены в черный цвет и $L = \varnothing$.

Шаг 1. Выбираем произвольную вершину v_1 графа G и первое ребро из ее списка смежности. Обозначим его (v_1, v_2). Тогда полагаем $L = \{v_1, v_2\}$ и красим вершины v_1, v_2 в белый цвет. Передвигаемся в вершину v_2 и переходим к следующему шагу.

Шаг i $(1 < i < n - m)$. Находимся в белой вершине v_i пути L длины $i-1$. Берем первое ребро из списка смежности v_i, вторая вершина которого не принадлежит L, т. е. не является белой. Обозначим его (v_i, v_{i+1}). Если такого ребра нет, то алгоритм прекращает свою работу и выдает ошибку, поскольку нам не удалось построить путь заданной длины и перейти к следующему этапу. Полагаем $L = L \cup v_{i+1}$, красим вершину v_{i+1} в белый цвет и передвигаемся в нее.

Шаг $(n - m)$. Имеем путь $L = \{v_1, v_2, \ldots, v_{n-m}\}$ длиной $n - m - 1$, все вершины которого покрашены в белый цвет. Работа первого этапа алгоритма окончена. Пример правильной работы этапа 1 приведен на рис. 2.

Этап 2. Строим совершенное паросочетание P. Сначала полагаем $P = \varnothing$. После первого этапа имеем путь L из $n - m$ белых вершин и m незадействованных вершин $V \setminus L$ черного цвета.

Шаг 1. Находимся в 1-й вершине пути (v_1). Просматривая ее список смежности, ищем ребро, инцидентное какой-либо из вершин $V \setminus L$. Возможны два варианта.

(1) Такое ребро существует. Тогда добавляем его в паросочетание P и красим вторую вершину ребра в белый цвет. Ребро паросочетания, соединя-

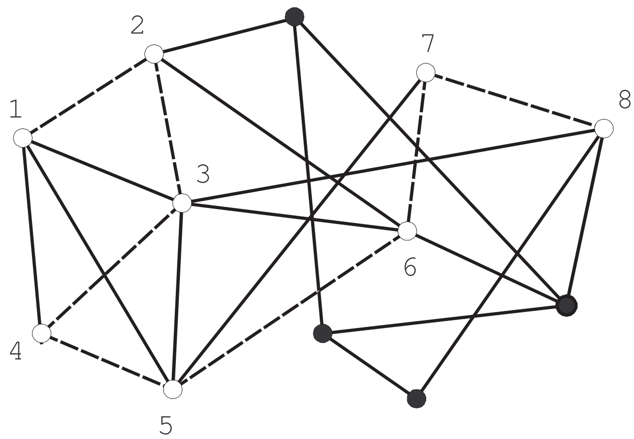

Рис. 2. В данном примере $n = 12$, $m = 4$. Вершины пути L покрашены в белый цвет.

ющее вершину из L с вершиной, не принадлежащей L, будем называть β-ребром. Берем вторую вершину L и переходим к следующему шагу.

(2) Такого ребра не существует. Тогда добавляем в P ребро (v_1, v_2). Ребро паросочетания, соединяющее две последовательные вершины пути L, будем называть α-ребром. Берем третью вершину L и переходим к следующему шагу.

Шаг i. Находимся в j-й вершине пути L. Просматривая ее список смежности, ищем ребро, инцидентное какой-либо из черных вершин. Возможны два варианта.

(1) Такое ребро существует. Тогда к P добавляем данное β-ребро и красим вторую вершину в красный цвет. Берем $(j+1)$-ю вершину пути L и переходим к следующему шагу.

(2) Такого ребра не существует. Тогда к P добавляем α-ребро (v_j, v_{j+1}). Берем $(j+2)$-ю вершину пути L и переходим к следующему шагу.

Действуем таким образом, пока не дойдем до конца пути L. Другими словами, как только вершина v_{n-m} попадает в P, останавливаем работу алгоритма. Если при этом все вершины графа окажутся покрашены в белый, т. е. все $V \setminus L$ незадействованных на первом этапе вершин будут участвовать в P, то второй этап, а с ним и весь алгоритм, сработал успешно, и совершенное паросочетание P построено. В противном случае считаем, что алгоритм не сработал, и выдаем ошибку. Пример успешной работы алгоритма приведен на рис. 3, 4.

Оценим трудоемкость работы нашего алгоритма. Рассмотрим произвольную вершину v графа G. По условию задачи, в соответствие вершине v можно поставить $(n-1)$ независимых случайных величин $\varphi_{v,v_i}, 1 \le i \le n-1$, где φ_{v,v_i} принимает значение 1 с вероятностью p. Случайная величина φ_{v,v_i} принимает значение 1, если между вершинами v и v_i есть ребро, и 0 в противном случае. Тогда величина $h_v = \sum_{i=1}^{n-1} \varphi_{v,v_i}$ соответствует числу вершин, инцидентных вершине v. Согласно закону больших чисел, представленному в [9], с увеличением n значение случайной величины h становится

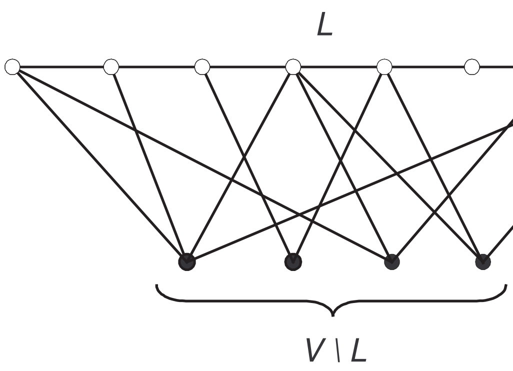

Рис. 3. Перед началом второго этапа $n-m$ вершин (вершины L) покрашены в белый цвет, m вершин (вершины $V \setminus L$) покрашены в черный цвет.

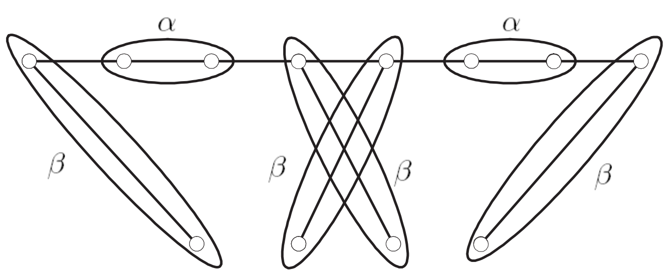

Рис. 4. Результат успешной работы второго этапа. Все вершины участвуют в P, т. е. покрашены в белый цвет.

ближе к математическому ожиданию Eh, равному

$$Eh = E\sum_{i=1}^{n-1} \varphi_{v,v_i} = \sum_{i=1}^{n-1} E\varphi_{v,v_i} = \sum_{i=1}^{n-1} p = (n-1)p.$$

Таким образом, с ростом n количество ребер, инцидентных одной вершине графа G, становится ближе к математическому ожиданию, равному $(n-1)p$. Тогда математическое ожидание числа ребер, рассматриваемых на этапе 1, равняется $(n-m)(n-1)p$. С учетом этого можно заключить, что трудоемкость первого этапа есть величина $O(n^2p)$. Трудоемкость второго этапа имеет тот же порядок. Таким образом, трудоемкость всего алгоритма есть $O(n^2p)$.

4. Анализ алгоритма

4.1. *Оценка этапа 1*

Лемма 1. *При* $p \geq \frac{\lambda_1 \ln n}{n}$, $m = \frac{n}{\lambda}$, *где* λ_1, λ — *постоянные, вероятность несрабатывания первого этапа алгоритма не превышает величины* $\frac{1}{\lambda_1 n^{\frac{\lambda_1}{\lambda}-1} \ln n}$. *В частности, при* $\lambda_1 \geq \lambda$ *эта вероятность стремится к* 0, *при* $n \to \infty$.

ДОКАЗАТЕЛЬСТВО. Оценим вероятность того, что на этапе 1 нам удастся построить путь длиной $n-m-1$. Обозначив $q = 1-p$, получим:

$$\mathbb{P}\{\text{построен путь длины } n-m-1\} = \prod_{i=1}^{n-m-1}[1-q^{n-i}] \geq$$
$$\geq 1 - \sum_{i=1}^{n-m-1} q^{n-i} = 1 - q^{m+1}(1+q+\ldots+q^{n-m-2}) =$$
$$= 1 - q^{m+1}\frac{1-q^{n-m-1}}{1-q} \geq 1 - \frac{q^{m+1}}{p}.$$

Воспользовавшись известным оценочным неравенством,

$$1 - x \leq e^{-x}, x \in [0,1],$$

получаем такую цепочку неравенств:

$$1-p \leq e^{-p}, \quad q \leq e^{-p}, \quad q^{m+1} \leq q^m \leq e^{-mp}, \quad -q^{m+1} \geq -e^{-mp}.$$

Применяем последнее неравенство к полученной ранее оценке:

$$\mathbb{P}\{\text{построен путь длины } n-m-1\} \geq 1 - \frac{q^{m+1}}{p} \geq 1 - \frac{e^{-mp}}{p}.$$

Для заданных в условии p и m получим оценку сверху для вероятности несрабатывания первого этапа:

$$\mathbb{P}\{\text{несрабатывание первого этапа}\} = 1 - \mathbb{P}\{\text{построен путь длины } n-m-1\},$$
$$\mathbb{P}\{\text{несрабатывание первого этапа}\} \leq \frac{e^{-mp}}{p} \leq \frac{ne^{-\frac{n}{\lambda}\frac{\lambda_1 \ln n}{n}}}{\lambda_1 \ln n} = \frac{1}{\lambda_1 n^{\frac{\lambda_1}{\lambda}-1}\ln n}.$$

При $\lambda_1 \geq \lambda$ имеем $\frac{\lambda_1}{\lambda} - 1 \geq 0$, и, следовательно, вероятность несрабатывания алгоритма стремится к 0 с ростом n.

Лемма доказана. □

4.2. *Оценка этапа 2*

Перед началом работы второго этапа алгоритма имеем путь L, состоящий из $n-m$ белых вершин, и m черных незадействованных в пути вершин $V \setminus L$. Для оценки вероятности несрабатывания второго этапа алгоритма введем в рассмотрение путь L', полученный добавлением бесконечного числа белых вершин к пути L. При этом считаем, что между любой добавленной вершиной v и черной вершиной u с вероятностью p существует ребро. Другими словами, добавленные вершины обладают теми же вероятностными характеристиками, что и остальные. Применим второй этап алгоритма к пути L' (вместо L) и оставшимся m вершинам. Так как путь L' бесконечен, то на каком-то шаге все вершины $V \setminus L$ окажутся покрашены в белый цвет, т. е. все m незадействованных в L вершин будут участвовать в паросочетании, — на этом шаге останавливаем работу алгоритма для L'.

Введем следующие m случайных величин:
ξ_1 — число α-ребер, вошедших в путь L' до появления первого β-ребра;
ξ_k — число α-ребер, вошедших в путь L' между $(k-1)$-м и k-м появлением β-ребер $(1 < k \leq m)$.

В силу бесконечности пути L' все $\xi_1, \ldots, \xi_m$ являются независимыми случайными величинами. Число вершин пути L', задействованных в построенном на этапе 2 паросочетании, в точности равно $2\sum_{k=1}^{m} \xi_k + m$, поскольку в каждом α-ребре участвуют две вершины L' и в каждом β-ребре — одна вершина.

Второй этап алгоритма для исходной задачи на пути L сработает тогда и только тогда, когда

$$2\sum_{k=1}^{m} \xi_k + m \leq n - m,$$

т. е. когда удастся найти пару для каждой из m вершин $V \setminus L$, не выходя за пределы пути L. Другими словами, когда при работе алгоритма на пути L' не будут задействованы вершины из $L' \setminus L$. Наглядно это показано на рис. 5.

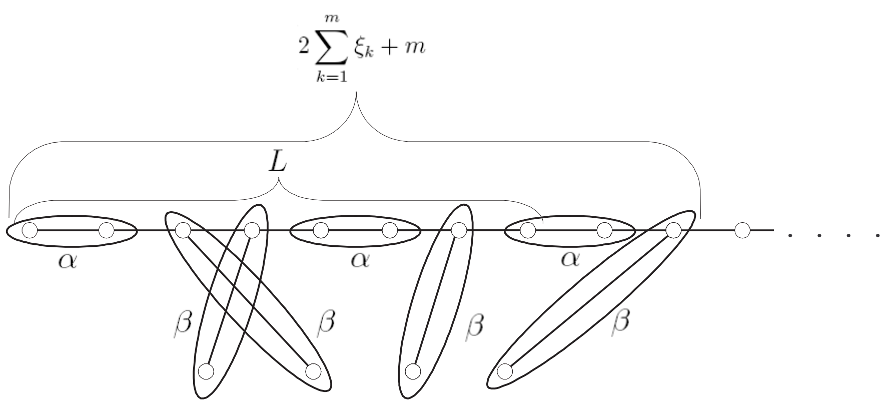

Рис. 5. Второй этап алгоритма не сработал для пути L, так как при работе алгоритма на L' задействованы вершины из $L' \setminus L$.

Таким образом, вероятность несрабатывания второго этапа есть

$$\mathbb{P}\left\{2\sum_{k=1}^{m} \xi_k + m > n - m\right\},$$

или, обозначив $S = \xi_1 + \xi_2 + \cdots + \xi_m$,

$$\mathbb{P}\left\{S > n/2 - m\right\}. \tag{8}$$

Для оценки вероятности (8) далее нам понадобится оценить величины $\mathbb{E}S$ и $\mathbb{D}S$.

Лемма 2. *Для математического ожидания и дисперсии случайных величин ξ_k, $k = 1, 2, \ldots, m$, справедливы формулы*

$$\mathbb{E}\xi_k = \frac{q_k}{1-q_k},$$
$$\mathbb{D}\xi_k = \frac{q_k}{(1-q_k)^2},$$

где $q = 1 - p$, $q_k = q^{m-k+1}$.

ДОКАЗАТЕЛЬСТВО. Вероятность того, что на первом шаге второго этапа появилось β-ребро (т. е. первая вершина L инцидентна хотя бы одной вершине из $V \setminus L$), равна $\mathbb{P}\{\xi_1 = 0\} = 1 - q^m$.

Вероятность того, что на первом шаге появилось α-ребро, а на втором — β-ребро, равна $\mathbb{P}\{\xi_1 = 1\} = q^m(1 - q^m)$.

Вероятность того, что на первых i шагах были построены α-ребра, а на $(i+1)$-м — β-ребро, равна $\mathbb{P}\{\xi_1 = i\} = q^{mi}(1 - q^m),\ \ i = 0, 1, \ldots$

Вероятность того, что вслед за 1-м β-ребром сразу появилось 2-е β-ребро, равна $\mathbb{P}\{\xi_2 = 0\} = (1 - q^{m-1})\mathbb{P}\{\text{появилось первое } \beta\text{-ребро}\}$. В силу бесконечности пути L', первое β-ребро гарантированно появится на одном из шагов алгоритма, поэтому $\mathbb{P}\{\text{появилось первое } \beta\text{-ребро}\} = 1$.

Введем дополнительное обозначение

$$q_k = q^{m-k+1}.$$

С учетом этого обозначения имеет место формула

$$\mathbb{P}\{\xi_k = i\} = q_k^i(1 - q_k), \quad i = 0, 1, \ldots,\ k = 1, \ldots, m.$$

Оценим математическое ожидание и дисперсию величин ξ_k, $k = 1, \ldots, m$.

$$\mathbb{E}\xi_k = \sum_{i=1}^{\infty} i\mathbb{P}\{\xi_k = i\} = (1 - q_k)\sum_{i=1}^{\infty} iq_k^i = \sum_{i=1}^{\infty} iq_k^i - \sum_{i=1}^{\infty} iq_k^{i+1} =$$
$$= \sum_{i=1}^{\infty} iq_k^i - (\sum_{i=2}^{\infty} iq_k^i - \sum_{i=2}^{\infty} q_k^i) = \sum_{i=1}^{\infty} q_k^i = \frac{q_k}{1 - q_k};$$
$$\mathbb{E}\xi_k^2 = \sum_{i=1}^{\infty} i^2\mathbb{P}\{\xi_k = i\} = (1 - q_k)\sum_{i=1}^{\infty} i^2q_k^i = \sum_{i=1}^{\infty} i^2q_k^i - \sum_{i=1}^{\infty} i^2q_k^{i+1} =$$
$$= \sum_{i=1}^{\infty} i^2q_k^i - \sum_{i=1}^{\infty}(i-1)^2q_k^i = 2\sum_{i=1}^{\infty} iq_k^i - \sum_{i=1}^{\infty} q_k^i = \frac{2q_k}{(1 - q_k)^2} - \frac{q_k}{(1 - q_k)} = \frac{q_k + q_k^2}{(1 - q_k)^2};$$
$$\mathbb{D}\xi_k = \mathbb{E}\xi_k^2 - (\mathbb{E}\xi_k)^2 = \frac{q_k + q_k^2}{(1 - q_k)^2} - \Big(\frac{q_k}{1 - q_k}\Big)^2 = \frac{q_k}{(1 - q_k)^2}\,.$$

Отметим, что при работе с бесконечными рядами мы существенно опирались на сходимость рядов вида $\sum\limits_{i=1}^{\infty} i^a b^i$, где a — произвольная постоянная, $0 < b < 1$.

Лемма доказана. □

Оценку вероятности (8) проведем двумя методами — с помощью неравенства Чебышева и с помощью теоремы Петрова.

4.3. *Оценка второго этапа с помощью неравенства Чебышева*

Сформулируем неравенство Чебышева, приведенное в [9].

Неравенство Чебышева. *Если случайная величина X имеет конечную дисперсию $\mathbb{D}X$, то для любого $x > 0$*

$$\mathbb{P}\{|X - \mathbb{E}X| > x\} \leq \frac{\mathbb{D}X}{x^2}.$$

Лемма 3. *Верны следующие оценки:*

$$\mathbb{E}S \le \frac{1-\ln p}{p}, \quad \mathbb{D}S \le \frac{2}{p^2}.$$

ДОКАЗАТЕЛЬСТВО. С учетом обозначений $q_k = q^{m-k+1}$ и формул для $\mathbb{E}\xi_k$ и $\mathbb{D}\xi_k$, $k = \overline{1,m}$, полученных в лемме 2, имеем

$$\mathbb{E}S = \sum_{k=1}^{m} \mathbb{E}\xi_k = \sum_{k=1}^{m} \frac{q^k}{1-q^k} = \frac{q}{p} + \sum_{k=2}^{m} \frac{q^k}{1-q^k} \le \frac{q}{p} + \int\limits_1^m \frac{q^x}{1-q^x}\,dx =$$
$$= \frac{1-p}{p} - \frac{\ln(1-q^m)-\ln p}{\ln(1-p)} \le \frac{1-p}{p} + \frac{\ln(1-q^m)-\ln p}{p} \le \frac{1-\ln p}{p}.$$
$$\mathbb{D}S = \sum_{k=1}^{m} \mathbb{D}\xi_k = \sum_{k=1}^{m} \frac{q^k}{(1-q^k)^2} \le \frac{q}{p^2} + \int\limits_1^m \frac{q^x}{(1-q^x)^2}\,dx =$$
$$= \frac{q}{p^2} + \Big(-\frac{1}{(q^m-1)\ln q} + \frac{1}{(q-1)\ln q}\Big) = \frac{q}{p^2} - \frac{1}{p(q^m-1)}\frac{q^m-q}{\ln(1-p)}.$$

Используя тот факт, что $-\frac{1}{\ln(1-p)} \le \frac{1}{p}$, имеем:

$$\mathbb{D}S \le \frac{q}{p^2} + \frac{q-q^m}{p^2(1-q^m)} \le \frac{q}{p^2} + \frac{q}{p^2} \le \frac{2}{p^2}$$

Лемма доказана. □

Лемма 4. *При* $p \ge \frac{\lambda_1 \ln n}{n}$, $m = \frac{n}{\lambda}$, $\lambda_1, \lambda > 4$ *вероятность несрабатывания второго этапа алгоритма равна* $O(\ln^{-2} n)$.

ДОКАЗАТЕЛЬСТВО. Оценим вероятность наступления нежелательного события (8) с помощью неравенства Чебышева:

$$\mathbb{P}\left\{S > \frac{n}{2} - m\right\} = \mathbb{P}\left\{S - \mathbb{E}S > \frac{n}{2} - m - \mathbb{E}S\right\} \le$$
$$\le \mathbb{P}\left\{\big|S - \mathbb{E}S\big| > \frac{n}{2} - m - \mathbb{E}S\right\} \le \frac{\mathbb{D}S}{(\frac{n}{2} - m - \mathbb{E}S)^2}. \tag{9}$$

Проверим выполнение необходимого для применения неравенства Чебышева условия $\frac{n}{2} - m - \mathbb{E}S \ge 0$:

$$\frac{n}{2} - m - \mathbb{E}S \ge \frac{n}{2} - m - \left(\frac{1-\ln p}{p}\right) = n\left(\frac{1}{2} - \frac{1}{\lambda} + \frac{\ln(\lambda_1 \ln n) - \ln n - 1}{\lambda_1 \ln n}\right) =$$
$$= \frac{n}{2}\left(1 - \frac{2}{\lambda} - \frac{2}{\lambda_1} + 2\frac{\ln(\lambda_1 \ln n) - 1}{\lambda_1 \ln n}\right) > 0, \quad \text{при } \lambda_1,\ \lambda > 4.$$

Применяем к (9) полученные в лемме 3 оценки:

$$\mathbb{P}\left\{S > \frac{n}{2} - m\right\} \le \frac{\mathbb{D}S}{\left(\frac{n}{2} - m - \mathbb{E}S\right)^2} \le$$
$$\le \frac{2/p^2}{\left(\frac{n}{2} - \frac{n}{\lambda} - \frac{1-\ln p}{p}\right)^2} = \frac{8}{\lambda_1^2(\ln n)^2}\frac{1}{\left(1 - \frac{2}{\lambda} - \frac{2}{\lambda_1} + 2\frac{\ln(\lambda_1 \ln n) - 1}{\lambda_1 \ln n}\right)^2}.$$

Таким образом, с использованием неравенства Чебышева для вероятности несрабатывания второго этапа получена оценка $O(\ln^{-2} n)$. Лемма доказана. □

4.4. *Оценка второго этапа с помощью теоремы Петрова*

В данном разделе будет получена улучшенная оценка вероятности несрабатывания второго этапа алгоритма (8). Нам понадобится следующая теорема из [10].

Теорема (Петров). *Пусть $\xi_1, \ldots, \xi_n$ независимые случайные величины и существуют положительные постоянные $g_1, \ldots, g_n$ и T такие, что для всех $0 \le t \le T$:*

$$\mathbb{E}e^{t\xi_k} \le e^{\frac{g_k t^2}{2}}, \quad 1 \le k \le n.$$

Положим $S = \sum\limits_{k=1}^{n} \xi_k$, $G = \sum\limits_{k=1}^{n} g_k$. Тогда

$$\mathbb{P}\left\{S > x\right\} \le \begin{cases} \exp\left(-\frac{x^2}{2G}\right), & \text{при } \ 0 \le x \le GT, \\ \\ \exp\left(-\frac{Tx}{2}\right), & \text{при } \ x \ge GT. \end{cases}$$

Найдем значения $\mathbb{E}e^{t\xi_k}$ для введенных нами случайных величин:

$$\mathbb{E}e^{t\xi_k} = \sum_{i=1}^{\infty} e^{ti} q^{(m-k+1)i}(1 - q^{m-k+1}) = (1 - q^{m-k+1}) \frac{e^t q^{m-k+1}}{1 - e^t q^{m-k+1}}, \quad k = \overline{1, m}.$$

Отметим, что $\mathbb{E}e^{t\xi_k}$, $k = \overline{1, m}$ существуют при условии $e^t q^k < 1$, $k = \overline{1, m}$. Таким образом, для существования величин $\mathbb{E}e^{t\xi_k}$, $k = \overline{1, m}$ должно выполняться необходимое условие $e^T q < 1$, или $T < -\ln(1 - p)$.

Для удобства проведения дальнейших выкладок перенумеруем случайные величины ξ_k, $k = \overline{1, m}$ таким образом, что

$$\mathbb{E}e^{t\xi_k} = \left(1 - q^k\right) \frac{e^t q^k}{1 - e^t q^k}, \quad k = \overline{1, m}.$$

Введем в рассмотрение центрированные случайные величины $\xi_k - \mathbb{E}\,\xi_k$, $k = \overline{1, m}$,

$$\mathbb{E}e^{t(\xi_k - \mathbb{E}\xi_k)} = e^{-t\mathbb{E}\xi_k} \mathbb{E}e^{t\xi_k} = (1 - q^k) \frac{e^t q^k}{1 - e^t q^k} \exp\left(-t \frac{q^k}{1 - q^k}\right).$$

Применим теорему Петрова к набору $\xi_k - \mathbb{E}\,\xi_k$, $k = \overline{1, m}$. Для этого достаточно найти такие постоянные T, g_k, $k = \overline{1, m}$, что следующее неравенство выполняется для всех $k = \overline{1, m}$ и $0 \le t \le T$:

$$(1 - q^k) \exp\left(-\frac{g_k t^2}{2}\right) \le (1 - e^t q^k) \exp\left(t \frac{q^k}{1 - q^k}\right) = \\ = \exp\left(t \frac{q^k}{1 - q^k}\right) - q^k \exp\left(\frac{t}{1 - q^k}\right).$$

Лемма 5. *При $T = \frac{p}{2}$ и $g_k = \frac{8}{(1-q^k)^2}$, $k = \overline{1, m}$, для всех $0 \le t \le T$ и $k = \overline{1, m}$, верно*

$$(1 - q^k) \exp\left(-\frac{g_k t^2}{2}\right) \le \exp\left(t \frac{q^k}{1 - q^k}\right) - q^k \exp\left(\frac{t}{1 - q^k}\right). \tag{10}$$

ДОКАЗАТЕЛЬСТВО. Рассмотрим данное неравенство для $k=1$ и $g_1 = g$ (для остальных значений k выкладки будут аналогичными с точностью до замены q на q^k и g на g_k). Разложим обе части неравенства (10) в ряд Тейлора. Левая часть:

$$\exp\left(-\frac{gt^2}{2}\right)(1-q) = (1-q)\left(1 - gt^2\left(\frac{1}{2} - \frac{gt^2}{8} + \ldots + \frac{(-1)^n g^n t^{2n}}{2^{n+1}(n+1)!} + \ldots\right)\right).$$

Правая часть:

$$\exp\left(t\frac{q}{1-q}\right) - q\exp\left(\frac{t}{1-q}\right) = 1 + t\frac{q}{1-q} + \frac{t^2}{2}\left(\frac{q}{1-q}\right)^2 + \ldots + \frac{t^n}{n!}\left(\frac{q}{1-q}\right)^n + \ldots$$
$$- q - t\frac{q}{1-q} - \frac{t^2}{2}\frac{q}{(1-q)^2} - \ldots - \frac{t^n}{n!}\frac{q}{(1-q)^n} - \ldots =$$
$$= (1-q)\left(1 - \frac{t^2 q}{(1-q)^2}\left(\frac{1}{2} + \ldots + \frac{t^{n-2}(q^{n-2} + \ldots + q + 1)}{n!(1-q)^{n-2}} + \ldots\right)\right).$$

Таким образом, (10) преобразуется к виду

$$g\left(\frac{1}{2} - \frac{gt^2}{8} + \ldots + \frac{(-1)^n g^n t^{2n}}{2^{n+1}(n+1)!} + \ldots\right) \geq$$
$$\geq \frac{q}{(1-q)^2}\left(\frac{1}{2} + \frac{t(1+q)}{6(1-q)} + \ldots + \frac{t^{n-2}(q^{n-2} + \ldots + q + 1)}{n!(1-q)^{n-2}} + \ldots\right).$$

Подставим $g = \frac{c}{(1-q)^2}$, $c = \mathrm{const} \leq 32$:

$$\frac{c}{(1-q)^2}\left(\frac{1}{2} - \frac{ct^2}{8(1-q)^2} + \ldots + \frac{(-1)^n c^n t^{2n}}{(1-q)^{2n} 2^{n+1}(n+1)!} + \ldots\right) \geq$$
$$\geq \frac{q}{(1-q)^2}\left(\frac{1}{2} + \frac{t(1+q)}{6(1-q)} + \ldots + \frac{t^{n-2}(q^{n-2} + \ldots + q + 1)}{n!(1-q)^{n-2}} + \ldots\right). \quad (11)$$

Отметим, что сумма знакопеременного ряда в левой части неравенства неотрицательна:

$$\frac{c^2 t^4}{(1-q)^4 2^3 3!} + \ldots + \frac{(-1)^n c^n}{(1-q)^{2n}}\frac{t^{2n}}{2^{n+1}(n+1)!} + \ldots \geq 0$$

Действительно, рассмотрим сумму двух соседних членов, соответствующих $n = 2l$ и $n = 2l+1$ для $l \geq 1$ и покажем, что она неотрицательна:

$$\frac{(-1)^{2l} c^{2l}}{(1-q)^{4l}}\frac{t^{4l}}{2^{2l+1}(2l+1)!} + \frac{(-1)^{2l+1} c^{2l+1}}{(1-q)^{4l+2}}\frac{t^{4l+2}}{2^{2l+2}(2l+2)!} \geq 0,$$
$$\frac{ct^2}{(1-q)^2 2(2l+2)} \leq 1.$$

Достаточно показать, что неравенство верно для $t = \frac{p}{2}$:

$$\frac{cp^2}{p^2 8(2l+2)} \leq 1, \quad c \leq 8(2l+2).$$

В силу нашего предположения $c \leq 32$ последнее неравенство верно для всех $l \geq 1$.

Усилим неравенство (11), учитывая неотрицательность суммы знакопеременного ряда в правой части:

$$c\left(\frac{1}{2} - \frac{c}{(1-q)^2}\frac{t^2}{8}\right) \geq \frac{q}{2} + \frac{qt(1+q)}{6(1-q)} + \ldots + \frac{qt^{n-2}(q^{n-2} + \ldots + q + 1)}{n!(1-q)^{n-2}} + \ldots$$

Это должно выполняться для всех $t \leq \frac{p}{2} = \frac{1-q}{2}$. Усиливаем неравенство слева и справа:

$$c\left(\frac{1}{2} - \frac{c}{32}\right) \geq \frac{q}{2} + \frac{q(1+q)}{12} + \ldots + \frac{q(q^{n-2} + \ldots + q + 1)}{n!2^{n-2}} + \ldots$$

Еще усилим неравенство, подставив $q = 1$:

$$\frac{c}{2} - \frac{c^2}{32} \geq \frac{1}{2} + \frac{1}{6} + \ldots + \frac{n-1}{n!2^{n-2}} + \ldots$$

Сумма ряда в правой части неравенства легко оценивается

$$\sum_{n=2}^{\infty} \frac{n-1}{n!2^{n-2}} \leq \sum_{n=2}^{\infty} \frac{1}{2^{n-1}} = 1.$$

Таким образом, для любого c, удовлетворяющего неравенству $c^2 - 16c + 32 \leq 0$, будет выполнено последнее неравенство, а следовательно, и (10). В частности, все проведенные выкладки будут верны и для $c = 8$. Лемма доказана. $\square$

Проверим выполнение необходимого для существования $\mathbb{E}e^{t\xi_k}$, $k = \overline{1,m}$ условия $T < -\ln(1-p)$. Действительно, для $p < 1$ верно $-p > \ln(1-p)$. Следовательно, $-p > 2\ln(1-p)$ и $T = \frac{p}{2} < -\ln(1-p)$.

Получим еще одну необходимую оценку.

Лемма 6. *Для* $g_k = \frac{8}{1-q^2}$, $p = \frac{\lambda \ln n}{n}$ *и* $m = \frac{n}{\lambda}$ *верна оценка* $G = \sum_{k=1}^{m} g_k \leq \frac{16}{p^2} + 16m$.

ДОКАЗАТЕЛЬСТВО.

$$G = \sum_{k=1}^{m} g_k = \sum_{k=1}^{m} \frac{8}{(1-q^k)^2} \leq \frac{8}{p^2} + 8\int_1^m \frac{dx}{(1-q^x)^2} = \frac{8}{p^2} + 8\int_1^m \frac{(1-q^x)dx}{(1-q^x)^2} +$$

$$+ 8\int_1^m \frac{q^x dx}{(1-q^x)^2} \leq \frac{8}{p^2} + 8\int_1^m \frac{dx}{(1-q^x)} + \frac{8q}{p^2} \leq \frac{16}{p^2} - \frac{8}{\ln q}\int_{1-q}^{1-q^m} \frac{dy}{(1-y)y} =$$

$$= \frac{16}{p^2} - \frac{8}{\ln q}(\ln(1-q^m) - \ln(q^m) - \ln(1-q) + \ln q) =$$

$$= \frac{16}{p^2} - \frac{8}{\ln q}\Big(\ln\left(\frac{1-q^m}{1-q}\right) - m\ln q + \ln q\Big) =$$

$$= \frac{16}{p^2} - 8\frac{\ln(1 + \ldots + q^{m-1})}{\ln q} + 8m - 8 \leq (\text{используем, что } \ln q \leq -p)$$

$$\leq \frac{16}{p^2} + \frac{8}{p}\ln(1 + \ldots + q^{m-1}) + 8m \leq (\text{используем, что } q \leq 1) \leq \frac{16}{p^2} + \frac{8\ln m}{p} + 8m \leq$$

$$\leq \frac{16}{p^2} + \frac{8\ln\frac{n}{\lambda}}{\frac{\lambda \ln n}{n}} + 8m \leq \frac{16}{p^2} + \frac{8n}{\lambda} + 8m = \frac{16}{p^2} + 16m.$$

Лемма доказана. $\square$

Для доказательства следующей леммы мы применим теорему Петрова к центрированным случайным величинам $\xi_k - \mathbb{E}\,\xi_k$, $1 \leq k \leq m$ с постоянными $T = \frac{p}{2}$, $g_k = \frac{8}{(1-q^k)^2}$, $1 \leq k \leq m$.

Лемма 7. *При* $p \geq \frac{\lambda_1 \ln n}{n}$, $m = \frac{n}{\lambda}$, $\lambda_1 \geq \lambda > 4$ *вероятность несрабатывания второго этапа алгоритма оценивается величиной* $O(n^{-(\lambda_1\lambda - 2\lambda_1 - 2\lambda)/8\lambda})$ *при достаточно больших* n. *В частности, эта вероятность стремится к* 0 *при* $n \to \infty$.

ДОКАЗАТЕЛЬСТВО. Отметим, что чем больше значение вероятности p появления ребра между любыми двумя вершинами графа, тем меньше вероятность несрабатывания второго этапа алгоритма. Данное свойство очевидно следует из описания алгоритма, поскольку значение вероятности несрабатывания второго этапа алгоритма убывает с ростом числа ребер рассматриваемого графа. Поэтому достаточно доказать заявленную в условии леммы оценку для $p = \frac{\lambda_1 \ln n}{n}$.

Вероятность несрабатывания второго этапа (8) представима в виде

$$\mathbb{P}\left\{S \geq \frac{n}{2} - m\right\} = \mathbb{P}\left\{S - \mathbb{E}S \geq \frac{n}{2} - m - \mathbb{E}S\right\} = \mathbb{P}\left\{\sum_{i=1}^{m}(\xi_i - \mathbb{E}\xi_i) \geq \frac{n}{2} - m - \mathbb{E}S\right\}.$$

По лемме 3

$$\mathbb{E}S \leq \frac{1 - \ln p}{p} = \frac{1 - \ln(\frac{\lambda_1 \ln n}{n})}{\frac{\lambda_1 \ln n}{n}} = \frac{n}{\lambda_1 \ln n} + \frac{n}{\lambda_1} - \frac{\ln(\lambda_1 \ln n)}{\lambda_1 \ln n} \leq \frac{1}{p} + \frac{n}{\lambda_1}.$$

Получаем

$$\mathbb{P}\left\{\sum_{i=1}^{m}(\xi_i - \mathbb{E}\xi_i) \geq \frac{n}{2} - m - \mathbb{E}S\right\} \leq \mathbb{P}\left\{\sum_{i=0}^{m}(\xi_i - \mathbb{E}\xi_i) \geq \frac{n}{2} - \frac{n}{\lambda} - \frac{n}{\lambda_1} - \frac{1}{p}\right\}.$$

Применяем теорему Петрова к центрированным случайным величинам $\xi_k - \mathbb{E}\xi_k$, $k = \overline{1,m}$, с указанными в лемме 4 постоянными $T,\ g_k$, $k = \overline{1,m}$.

В терминах теоремы Петрова

$$x = \frac{n}{2} - \frac{n}{\lambda} - \frac{n}{\lambda_1} - \frac{1}{p} = n\left(\frac{1}{2} - \frac{1}{\lambda} - \frac{1}{\lambda_1}\right) - \frac{1}{p}.$$

По лемме 6 имеем

$$GT \leq \frac{8}{p} + 8mp = \frac{8n}{\lambda_1 \ln n} + 8\frac{\lambda_1}{\lambda}\ln n.$$

При достаточно больших n имеем $x > GT$. Действительно, при достаточно больших n справедливо

$$\frac{9}{\lambda_1}\frac{1}{\ln n} + \frac{8\lambda_1}{\lambda}\frac{\ln n}{n} \leq \frac{1}{2} - \frac{1}{\lambda} - \frac{1}{\lambda_1},$$

и, следовательно, выполняется второе неравенство теоремы Петрова

$$\mathbb{P}\left\{\sum_{i=1}^{m}(\xi_i - \mathbb{E}\xi_i) \geq x\right\} \leq \exp\left(-\frac{Tx}{2}\right) =$$

$$= \exp\left(-\frac{p}{4}\left(n\left(\frac{1}{2} - \frac{1}{\lambda} - \frac{1}{\lambda_1}\right) - \frac{1}{p}\right)\right) = \frac{e^{1/4}}{n^{(\lambda_1\lambda - 2\lambda_1 - 2\lambda)/8\lambda}}.$$

Заметим, что при $\lambda_1 \geq \lambda > 4$

$$\lambda_1\lambda - 2\lambda_1 - 2\lambda > 4\lambda_1 - 2\lambda_1 - 2\lambda \geq 0.$$

Таким образом,

$$\mathbb{P}\left\{S > \frac{n}{2} - m\right\} \leq \frac{e^{1/4}}{n^c},$$

где $c = (\lambda_1\lambda - 2\lambda_1 - 2\lambda)/8\lambda$ — константа. Поскольку $c > 0$, то вероятность несрабатывания стремится к 0 с ростом n. Лемма доказана. □

Теперь перейдем к оценке вероятности несрабатывания всего алгоритма $\mathcal{A}$. Обозначим через δ_1 и δ_2 вероятности несрабатывания первого и второго этапов соответственно (значения δ_1, δ_2 найдены в леммах 1, 4, 7). Тогда вероятность $\delta_{\mathcal{A}}$ несрабатывания алгоритма $\mathcal{A}$ выражается следующим образом:

$$\delta_{\mathcal{A}} = 1 - (1 - \delta_1)(1 - \delta_2) = \delta_1 + \delta_2 - \delta_1\delta_2.$$

Следовательно, $\delta_{\mathcal{A}}$ есть величина порядка $\delta_1 + \delta_2$. С помощью неравенства Чебышева получена следующая оценка вероятности несрабатывания алгоритма $\mathcal{A}$ (см. леммы 1 и 4):

$$\delta_{\mathcal{A}} = O(\ln^{-2} n),$$

т. е. $\delta_{\mathcal{A}}$ определяется вероятностью несрабатывания второго этапа алгоритма δ_2 (δ_1 экспоненциально убывает с ростом n).

С помощью теоремы Петрова получена экспоненциальная оценка вероятности несрабатывания второго этапа (см. лемму 7). Поскольку значение постоянной λ_1 является элементом входных данных алгоритма, то рассмотрим $\delta_1(\lambda)$, $\delta_2(\lambda)$ при фиксированном значении $\lambda_1 > 4$. Значение параметра λ не задается входными данными, поэтому необходимо выбрать его таким образом, чтобы вероятность несрабатывания алгоритма принимала минимальное значение. Ранее доказали, что

$$\delta_1(\lambda) = O\left(\frac{1}{n^{\frac{\lambda_1}{\lambda}-1}\ln n}\right), \quad \delta_2(\lambda) = O(n^{-(\lambda_1\lambda - 2\lambda_1 - 2\lambda)/8\lambda}).$$

Ограничившись более грубой оценкой $\widetilde{\delta_1}(\lambda) = O\Big(\frac{1}{n^{\frac{\lambda_1}{\lambda}-1}}\Big)$, имеем

$$\delta_1(\lambda) + \delta_2(\lambda) \le \widetilde{\delta_1}(\lambda) + \delta_2(\lambda) \le 2\max\{\widetilde{\delta_1}(\lambda), \delta_2(\lambda)\}.$$

Найдем значение λ, при котором правая часть неравенства достигает наименьшего значения

$$\min_{4<\lambda\le\lambda_1} \max\{\widetilde{\delta_1}(\lambda), \delta_2(\lambda)\}.$$

Иными словами, ищем точку минимума мажоранты функций $\widetilde{\delta_1}(\lambda)$, $\delta_2(\lambda)$ на $(4, \lambda_1]$. Заметим, что $\widetilde{\delta_1}(\lambda)$ является возрастающей по λ функцией, а $\delta_2(\lambda)$ — убывающей, причем $\widetilde{\delta_1}(4) < \delta_2(4)$ (т. е. показатель степени при n у $\widetilde{\delta_1}(4)$ меньше, чем у $\delta_2(4)$). В силу этих свойств в качестве искомого значения λ можно взять точку пересечения графиков функций $\widetilde{\delta_1}(\lambda)$ и $\delta_2(\lambda)$ (точку, в которой они имеют одинаковые порядки по n). Приравняем степени

$$\frac{\lambda_1\lambda - 2\lambda_1 - 2\lambda}{8\lambda} = \frac{\lambda_1}{\lambda} - 1,$$

и получаем

$$\lambda = \frac{10\lambda_1}{\lambda_1 + 6}. \tag{12}$$

Отметим, что при $\lambda_1 > 4$ из (12) следует $\lambda_1 > \lambda > 4$, т. е. $\lambda \in (4, \lambda_1]$. При таком выборе λ вероятность несрабатывания алгоритма есть величина

$$\delta_{\mathcal{A}} = O\left(n^{-\frac{\lambda_1 - 4}{10}}\right).$$

Таким образом, с помощью теоремы Петрова получена более точная оценка вероятности несрабатывания алгоритма $\mathcal{A}$, чем с помощью неравенства Чебышева, и мы можем сформулировать центральное утверждение.

Теорема 1. *При* $p \geq \frac{\lambda_1 \ln n}{n}$, $\lambda_1 > 4$ *алгоритм* $\mathcal{A}$ *отыскания совершенного паросочетания в случайном графе является асимптотически точным и имеет вероятность несрабатывания* $O\left(n^{-\frac{\lambda_1-4}{10}}\right)$.

ДОКАЗАТЕЛЬСТВО. Определим внутренние параметры λ, m алгоритма $\mathcal{A}$ следующим образом: $\lambda = \frac{10\lambda_1}{\lambda_1+6}$ и $m = \frac{n}{\lambda}$. Тогда $\lambda_1 > \lambda > 4$ (тем самым выполнены условия необходимых лемм), и по сделанным выше замечаниям теорема верна.

4.5. *Задача на ориентированном графе*

Дан ориентированный граф G со множеством вершин V, $|V| = n$. Между любыми двумя вершинами графа с вероятностью p существует ребро в каждом из направлений. Необходимо найти совершенное паросочетание в графе G.

Для решения задачи на ориентированном случайном графе мы можем применить алгоритм $\mathcal{A}$, описанный ранее. На первом этапе мы строим произвольную цепь из $n-m$ вершин без учета направлений ребер, а на каждом шаге второго этапа ищем ребро в любом из направлений. Хотя бы в одном направлении ребро существует уже с вероятностью $2p$, поэтому итоговая оценочная теорема будет иметь такой вид.

Теорема 2. *При* $p \geq \frac{\lambda_1 \ln n}{2n}$, $\lambda_1 > 4$ *алгоритм* $\mathcal{A}$ *является асимптотически точным и имеет вероятность несрабатывания* $O\left(n^{-\frac{\lambda_1-4}{10}}\right)$ *на случайном ориентированном графе.*

5. Поиск совершенного паросочетания в случайном двудольном графе

Дан двудольный граф G со множеством вершин $V = S \cup T$, $|S| = n$, $|T| = n$. Каждое ребро, соединяющее вершину $s \in S$ с вершиной $t \in T$, появляется независимо от других ребер с вероятностью p.

Для построения совершенного паросочетания применим к графу G описанный алгоритм $\mathcal{A}$. Найдем вероятностные оценки и условия асимптотической точности работы алгоритма $\mathcal{A}$ на двудольном случайном графе. Доказательства соответствующих утверждений опустим, так как они строятся аналогичным образом, как и в случае произвольного случайного графа. Для удобства при проведении оценок будем считать, что на первом этапе алгоритма строим путь L, состоящий из $2n - 2m$ вершин.

Лемма 8. *При* $p \geq \frac{\lambda_1 \ln n}{n}$, $m = \frac{n}{\lambda}$, *вероятность несрабатывания первого этапа алгоритма на двудольном графе не превышает величины* $\frac{2}{\lambda_1 n^{\frac{\lambda_1}{\lambda}-1} \ln n}$. *В частности, при* $\lambda_1 \geq \lambda$ *эта вероятность стремится к* 0, *при* $n \to \infty$.

Лемма 9. *При* $p \geq \frac{\lambda_1 \ln n}{n}$, $m = \frac{n}{\lambda}$, $\lambda_1, \lambda > 4$ *вероятность несрабатывания второго этапа алгоритма на двудольном графе есть* $O(\ln^{-2} n)$ *(с помощью неравенства Чебышева).*

Замечание. Дадим некоторые комментарии к доказательству этой леммы. На первом этапе алгоритма построен путь L из $2n - 2m$ вершин, в котором $n - m$ вершин лежат в одной доле графа, и $n-m$ вершин — во второй. По m вершин из каждой доли исходного графа остались незадействованными в пути L и составляют множество $V \backslash L$. Запишем вероятностные характеристики случайных величин ξ_i, $i = 1, \ldots, 2m$, аналогичных величинам ξ_i, $i = 1, \ldots, m$, введенным в лемме 2:

$\mathbb{P}\{\xi_1 = 0\} = 1 - q^m,$
$\mathbb{P}\{\xi_1 = k\} = q^{km}(1 - q^m),$
$\mathbb{P}\{\xi_2 = 0\} = 1 - q^m,$
$\mathbb{P}\{\xi_2 = k\} = q^{km}(1 - q^m),$
$\mathbb{P}\{\xi_3 = 0\} = 1 - q^{m-1},$
$\mathbb{P}\{\xi_3 = k\} = q^{k(m-1)}(1 - q^{m-1}),$
$\mathbb{P}\{\xi_{2s} = k\} = q^{k(m-s+1)}(1 - q^{m-s+1}),$
$\mathbb{P}\{\xi_{2s+1} = k\} = q^{k(m-s)}(1 - q^{m-s}).$

Лемма 10. *При* $p \geq \frac{\lambda_1 \ln n}{n}$, $m = \frac{n}{\lambda}$, $\lambda_1 \geq \lambda > 4$ *вероятность несрабатывания второго этапа алгоритма оценивается величиной* $O(n^{-(\lambda_1\lambda - 2\lambda_1 - 2\lambda)/4\lambda})$ *при достаточно больших* n. *В частности, эта вероятность стремится к* 0 *при* $n \to \infty$ *(с помощью теоремы Петрова).*

Теорема 3. *При* $p \geq \frac{\lambda_1 \ln n}{n}$, $\lambda_1 > 4$ *алгоритм* $\mathcal{A}$ *находит совершенное паросочетание в случайном двудольном графе с вероятностью, стремящейся к* 1 *при* $n \to \infty$. *Вероятность несрабатывания алгоритма на графах такого вида есть* $O\Big(n^{-\frac{\lambda_1 - 4}{6}}\Big)$.

ДОКАЗАТЕЛЬСТВО. Определим значение параметров λ, m алгоритма $\mathcal{A}$ следующим образом: $\lambda = \frac{6\lambda_1}{\lambda_1 + 2}$ и $m = \frac{n}{\lambda}$. Тогда $\lambda_1 > \lambda > 4$, и вероятности несрабатывания первого и второго этапов имеют одинаковый порядок $O\Big(n^{-\frac{\lambda_1 - 4}{6}}\Big)$.

5.1. *Применение алгоритма отыскания совершенного паросочетания в двудольном графе к задаче размещения* (1)–(7)

Дадим описание алгоритма $\widetilde{\mathcal{A}}$ решения задачи размещения (1)–(7).

Перед началом работы алгоритма имеем множество $J = \{1, \ldots, n\}$ пунктов потребления некоторого продукта и множество $I = \{1, \ldots, m\}$ возможных пунктов размещения предприятий. При этом транспортные расходы g_{ij} — независимые случайные величины, равновероятно принимающие значения из целочисленного сегмента $[1, r]$, r — целое, $i \in I$, $j \in J$. Мы обозначили $p = 1/r$. Приведем также условия (6)–(7): существует такое целое число m_0, $1 \leq m_0 \leq m$, что функция спроса кусочно постоянна с точками излома, кратными m_0:

$$b_1 = b_2 = \ldots = b_{m_0} \geq b_{m_0+1} = b_{m_0+2} = \ldots = b_{2m_0} \geq$$
$$\geq b_{2m_0+1} = \ldots = b_{km_0} \geq b_{km_0+1} = \ldots = b_n, \text{ где } k = \left\lfloor \frac{n}{m_0} \right\rfloor$$

и

$$m_0 a = \sum_{j \in J} b_j.$$

Мы уже отмечали, что без ограничения общности можно считать открытыми пункты производства множества $I_0 = \{1, \ldots, m_0\}$, а также можно считать $k = \left\lfloor \frac{n}{m_0} \right\rfloor = \frac{n}{m_0}$.

Каждой задаче размещения (1)–(7) поставим в соответствие случайный двудольный граф $G = (I_0, J)$ со множеством вершин $V = I_0 \cup J$. Считаем, что между вершинами i и j есть ребро тогда и только тогда, когда $g_{ij} = 1$.

Опишем алгоритм $\widetilde{\mathcal{A}}$ решения задачи размещения (1)–(7).

Этап 0. Строим случайный двудольный граф $G = (I_0, J)$, соответствующий задаче размещения (1)–(7). Полученный граф G разбиваем на k случайных двудольных подграфов $G_l = (I_0, J_l), 1 \le l \le k$, где $J_l = \{(l-1)m_0 + 1, \ldots, lm_0\}, 1 \le l \le k$.

Этап $\boldsymbol{l}, 1 \le l \le k$. Применяем алгоритм $\mathcal{A}$ к графу G_l. Если алгоритм $\mathcal{A}$ построил совершенное паросочетание графа G_l, то для каждого ребра (i, j) данного паросочетания полагаем $x_{ij} = b_j$ и переходим к следующему этапу. Иначе говоря, в построенном нами решении задачи размещения (1)–(7) предприятие i будет обслуживать пункт спроса j в полном объеме b_j. Если алгоритм $\mathcal{A}$ не построил совершенное паросочетание графа G_l, то алгоритм $\widetilde{\mathcal{A}}$ выдает ошибку и завершает свою работу.

Этап $\boldsymbol{k+1}$. Построено совершенное паросочетание для каждого графа $G_l, 1 \le l \le k$. Таким образом, мы построили решение задачи размещения (1)–(7), в котором для каждого пункта спроса $j \in J$ найдено предприятие $i \in I_0$, обслуживающее j в полном объеме b_j.

Оптимальность решения, построенного алгоритмом $\widetilde{\mathcal{A}}$, очевидна, поскольку в решении алгоритм использует только коммуникации минимальной стоимости, т. е. для любого $x_{ij} \neq 0$ выполнено $g_{ij} = 1$.

Теорема 4. *При $p \ge \frac{\lambda_1 \ln m_0}{m_0}$, $\lambda_1 > 4$, и условии, что $(n/m_0^{\frac{\lambda_1+2}{6}}) \to 0$, при $n,\ m \to \infty$, алгоритм $\widetilde{\mathcal{A}}$ является асимптотически точным алгоритмом решения задачи размещения* (1)–(7).

ДОКАЗАТЕЛЬСТВО. Алгоритм $\widetilde{\mathcal{A}}$ состоит в $k = \frac{n}{m_0}$-кратном применении алгоритма $\mathcal{A}$ для множества графов $G_l, 1 \le l \le k$. Согласно теореме 3, при $p \ge \frac{\lambda_1 \ln m_0}{m_0}$, $\lambda_1 > 4$, вероятность несрабатывания $\mathcal{A}$ на G_l есть величина $O\Big(m_0^{-\frac{\lambda_1-4}{6}}\Big)$. Тогда вероятность несрабатывания алгоритма $\widetilde{\mathcal{A}}$ для исходной задачи размещения есть величина $O\Big(\frac{n}{m_0} m_0^{-\frac{\lambda_1-4}{6}}\Big) = O\Big(n/m_0^{\frac{\lambda_1+2}{6}}\Big)$. Таким образом, если $(n/m_0^{\frac{\lambda_1+2}{6}}) \to 0$, при $n,\ m \to \infty$, то алгоритм $\widetilde{\mathcal{A}}$ является асимптотически точным алгоритмом решения задачи размещения (1)–(7).

Трудоемкость работы алгоритма $\mathcal{A}$ на $G_l, 1 \le l \le k$ есть $O(m_0^2 p)$. Учитывая условие асимптотической точности, эта величина есть $O(m_0 \ln m_0)$. Тогда трудоемкость алгоритма $\widetilde{\mathcal{A}}$ решения задачи задачи размещения (1)–(7) есть $O(km_0 \ln m_0)$, что не превосходит $O(n \ln m)$.

Заключительные замечания

Представляет интерес развитие вероятностного анализа применительно к данной задаче с ослаблением специально введенных ограничений на объемы спроса клиентов (6), (7), мощности предприятий (5) и распределение элементов матрицы транспортных расходов (g_{ij}).

Список литературы

1. *Garey M. R., Johnson D. S.* Computers and Intractablity. San Franciskco: Freeman, 1979.

2. *Вознюк И. П., Гимади Э. Х., Филатов М. Ю.* Асимптотически точный алгоритм для решения задачи размещения с ограниченными объемами производства // Дискретный анализ и исследование операций. Сер. 2. Новосибирск: Изд-во ИМ СО РАН, 2001. Т. 8, № 2. С. 3–16.

3. *Агеев А. А, Гимади Э. Х., Курочкин А. А.* Полиномиальный алгоритм решения задачи размещения на цепи с одинаковыми производственными мощностями предприятий // Дискретный анализ и исследование операций. Новосибирск: Изд-во ИМ СО РАН. 2009. Т. 16, № 5. С. 3–18.

4. *Гимади Э. Х.* Эффективный алгоритм решения задачи размещения с областями обслуживания, связными относительно ациклической сети // Управляемые системы: Сб. науч. тр. Новосибирск: Изд-во ИМ СО АН СССР, 1983. Вып. 23. С. 12–23.

5. *Tokuyama T., Nakano J.* Efficient algorithms for the Hitchcock Transportation Problem // Symposium on Discrete Algorithms: Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms. Orlando, Florida, United States, 1992. P. 175–184.

6. *Kleinschmidt P., Schannath H.* A Strongly Polynomial Algorithm for the Transportation Problem // Mathematical Programming. 1995. No. 68. P. 1–13.

7. *Angluin D., Valiant L. G.* Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings // J. of Computer and System Sciences. 1979. Vol. 18. No. 2. P. 155–193.

8. *Гимади Э. Х., Перепелица В. А.* Асимптотический подход к решению задачи коммивояжера // Сб. научн. тр. Новосибирск: Изд-во ИМ СО АН СССР, 1974. Вып. 12. С. 35–45.

9. *Боровков А. А.* Теория вероятностей. М.: Наука, 1986.

10. *Петров В. В.* Предельные теоремы для сумм независимых случайных величин. М.: Наука, 1987.

Материал поступил в редколлегию 21.06.2009

Адреса авторов

ГИМАДИ Эдуард Хайрутдинович
Новосибирский государственный университет
ул. Пирогова, 2, Новосибирск, 630090, Россия
Институт математики им. С. Л. Соболева СО РАН
пр. Акад. Коптюга, 4, Новосибирск, 630090, Россия
e-mail: gimadi@math.nsc.ru

КУРОЧКИН Александр Александрович
Институт математики им. С. Л. Соболева СО РАН
пр. Акад. Коптюга, 4, Новосибирск, 630090, Россия
e-mail: alkurochkin@ngs.ru