

Управление в социально-экономических системах

© 2019 г. Д.В. УЖЕГОВ (denis.uzhegov@phystech.edu),

А.А. АНАНЬЕВ (ananev.andrey@phystech.edu),

П.В. ЛОМОВИЦКИЙ (pavel.lomovitskiy@phystech.edu),

А.Н. ХЛЮПИН (khlyupin@phystech.edu)

(Инжиниринговый центр МФТИ по трудноизвлекаемым полезным ископаемым,
Московский физико-технический институт (государственный университет))

НОВЫЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ СПЕЦИАЛЬНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ С ФУНКЦИЕЙ СТОИМОСТИ ОБЩЕГО ВИДА ПРИ НАЛИЧИИ ОГРАНИЧЕНИЙ

Рассматривается задача о назначениях специальной структуры с функцией стоимости общего вида и запретами на некоторые паросочетания. В этом случае стоимость назначения может быть не определена, пока не будет найдена какая-либо подстановка. Задача формулируется в терминах теории графов и сводится к поиску пути минимальной стоимости в графе с нелокальными весами ребер. Предлагаемый метод решения является модификацией алгоритма Дейкстры поиска кратчайшего пути во взвешенном ориентированном графе. Исследования мотивированы приложениями к бурению скважин. Приведен анализ численных экспериментов.

Ключевые слова: квадратичная задача о назначениях, недопустимые паросочетания, кратчайший путь в графе, алгоритм Дейкстры.

DOI: 10.1134/S0005231019010070

1. Введение

Задача о назначениях является одной из фундаментальных задач в области исследований операций, которая на практике возникает при решении проблем снабжения предприятий, строительства магистралей, медицинского обслуживания, создания радиоэлектронной аппаратуры и т.п. [1]. В общем случае ее можно сформулировать следующим образом:

Даны два множества A и B одинаковой мощности и задана функция $C: A \times B \rightarrow \mathbf{R}$. Необходимо найти биекцию $\pi: A \rightarrow B$ такую, чтобы была минимальной целевая функция

$$\sum_{a \in A} C(a, \pi(a)).$$

Если функция стоимости задана матрицей, то целевую функцию можно записать в виде

$$\sum_{a \in A} C_{a\pi(a)}.$$

Другими словами, имеется некоторое одинаковое количество работ и исполнителей. Каждый исполнитель может быть назначен на выполнение только одной работы, но с различными затратами. Необходимо распределить работы по исполнителям, минимизируя суммарные затраты (рис. 1, а).

В данной постановке задача является линейной (*linear assignment problem, LAP*). Известно множество методов ее решения, которое можно поделить на две части: симплекс-методы (см. [2, 3]) и прямодвойственные (*primal-dual*) методы (см. [4, 5]). В основе первых лежит тот факт, что задача о назначениях является разновидностью транспортной задачи, которая может быть представлена как задача линейного программирования [6]:

$$\min \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij}$$

при условии

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= 1, \quad i = 1, \dots, n, \\ \sum_{i=1}^n x_{ij} &= 1, \quad j = 1, \dots, n, \\ x_{ij} &\in \{0, 1\}, \quad i, j = 1, \dots, n, \end{aligned}$$

где

C_{ij} — стоимость назначения исполнителя i на работу j , $i \in A$, $j \in B$;

$x_{ij} = 1$, если исполнитель i назначен на работу j ;

$x_{ij} = 0$ в противном случае.

К прямодвойственным методам относится широко используемый венгерский алгоритм [4] и схожие с ним, например аукционный алгоритм [5]. Основная идея методов заключается в приведении матрицы стоимости к определенному виду. Большинство методов решения LAP подробным образом разобраны в [7].

В случае, когда целевая функция квадратична, говорят о квадратичной задаче о назначениях (*Quadratic assignment problem, QAP*). В формулировке Купманса–Бекмана [8] она звучит следующим образом (рис. 1, б). Заданы два множества одинаковой мощности: A — предприятия и B — города, а также функция $f: A \times A \rightarrow \mathbf{R}$, характеризующая поток между предприятиями, расстояние между городами $d: B \times B \rightarrow \mathbf{R}$ и стоимость размещения предприятия в городе $C: A \times B \rightarrow \mathbf{R}$. Обычно все функции стоимостей задаются в виде матриц (f_{ij}) , (d_{kl}) , (C_{ik}) . Необходимо найти подстановку (трактуются как биекция) $\pi: A \rightarrow B$ такую, чтобы была минимальной целевая функция

$$(1) \quad \sum_{a \in A} \sum_{b \in A} f_{ab} d_{\pi(a)\pi(b)} + \sum_{a \in A} C_{a\pi(a)}.$$

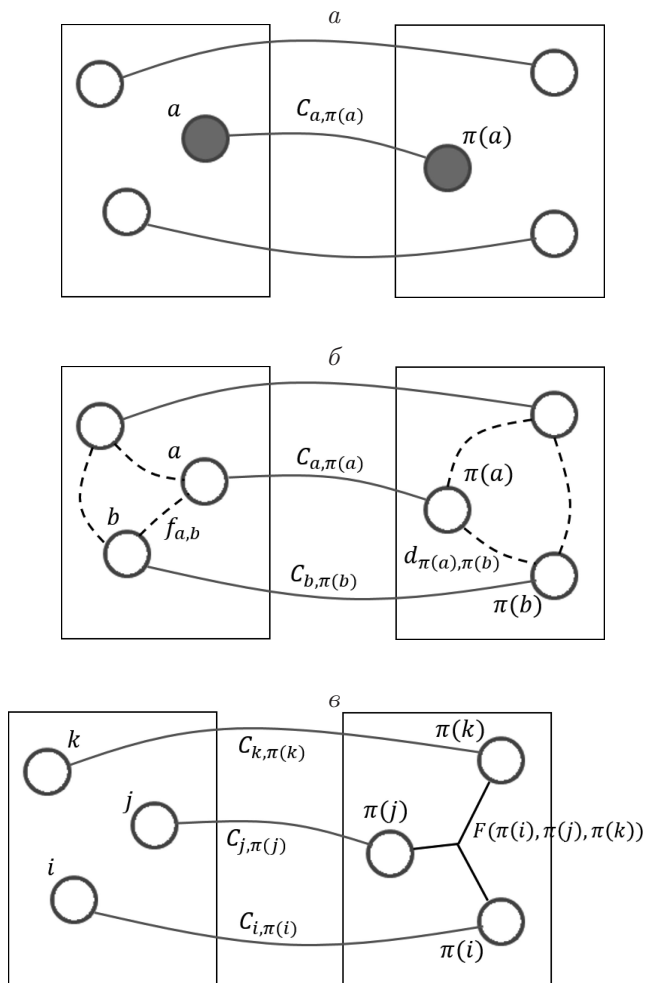


Рис. 1. Иллюстрация задач о назначениях: *a* – линейная задача о назначениях; *б* – квадратичная задача о назначениях; *в* – специальная задача о назначениях с функцией стоимости общего вида.

Будучи NP-сложной, квадратичная задача о назначениях не может быть решена за полиномиальное время [9]. Однако при выполнении специальных условий на матрицы задача является полиномиально разрешимой на подстановке заданного вида. Так называемые условия сильной разрешимости впервые были сформулированы в [10] и исследованы затем в [11, 12]. Тем не менее, для задач размерности более 25 нахождение точного решения не гарантировано [13]. Существуют эвристические алгоритмы, такие как генетический [14], муравьиный [15], имитирующие природные процессы, или алгоритм поиска с запретами [16]. Применение эвристических алгоритмов позволяет найти близкое к оптимальному решение задач большой размерности за разумное время.

В работе рассматривается некоторое обобщение квадратичной задачи. Можно представить ситуацию, когда необходимо учесть не расстояния между каждым предприятиями, а их совокупное расположение, которое может

быть задано условием на все предприятия сразу (рис. 1, в). В таком случае получить значение целевой функции невозможно без предположения о подстановке. Еще одно обобщение связано с существованием недопустимых паросочетаний, если размещение предприятия в каком-либо городе запрещено. В статье предлагается решение данной проблемы для случая равенства потоков между предприятиями $f_{ij} = \delta_{ij}$. Метод основан на представлении задачи о назначениях в терминах теории графов, который описан в [17]. Также показывается, что предлагаемый алгоритм позволяет решить квадратичную задачу о назначениях для симметричных матриц стоимостей. В первой главе приводится математическая формулировка задачи. Вторая глава содержит описание алгоритма решения. В третьей главе представлены результаты практического применения алгоритма.

2. Математическая формулировка задачи

С математической точки зрения задача отличается от классической более общим видом функции стоимости,

$$(2) \quad Z_{\pi} = \sum_i \{C_{i\pi(i)} + F(\pi(1), \dots, \pi(i))\},$$

где F — произвольная функция, зависящая от подстановки π . Если речь идет о квадратичной задаче о назначениях, данная функция имеет следующий вид:

$$(3) \quad F(\pi(1), \dots, \pi(i)) = d_{\pi(i)\pi(1)} + d_{\pi(i)\pi(2)} + \dots + d_{\pi(i)\pi(i)} = \sum_{j \leq i} d_{\pi(i)\pi(j)}.$$

Тогда в случае равенства потоков между предприятиями $f_{ij} = \delta_{ij}$ и симметрии матрицы $d_{ij} = d_{ji}$ задача о назначениях (1) эквивалентна (2) с функцией F вида (3). Решение задачи представимо в виде матрицы

$$X_{\pi} = (x_{ij}), \quad \text{где } x_{ij} = \begin{cases} 1, & \text{если } j = \pi(i), \\ 0, & \text{если } j \neq \pi(i). \end{cases}$$

Это матрица, в которой каждая строка и столбец содержат ровно один отличный от нуля элемент, а его положение задает искомую биекцию

$$P = \begin{pmatrix} i_1 & i_2 & \dots & i_n \\ \pi(i_1) & \pi(i_2) & \dots & \pi(i_n) \end{pmatrix}.$$

Например, решению X_{ϕ} соответствует биекция ϕ

$$X_{\phi} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \phi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}.$$

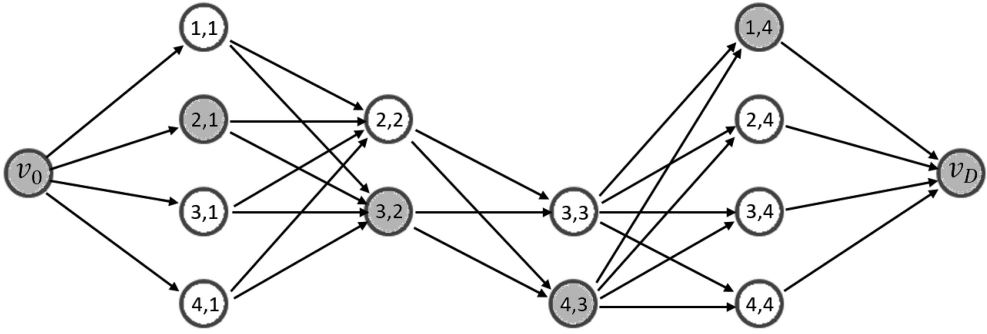


Рис. 2. Граф возможных назначений. Выделенный путь соответствует биекции ϕ , задающей матрицу X_ϕ .

Для дальнейшего изложения необходимо ввести матрицу Φ , которая отражает информацию о том, возможно ли размещение предприятия в городе:

$$\Phi = (f_{ij}), \quad \text{где } f_{ij} = \begin{cases} 1, & \text{если } C_{ij} < \infty, \\ 0, & \text{если } C_{ij} = \infty. \end{cases}$$

Таким образом, матрица X_ϕ получается из матрицы Φ выбором элементов, соответствующих биекции ϕ .

Все возможные назначения в соответствии с матрицей Φ можно представить в качестве множества вершин некоторого графа $\tilde{G}(\tilde{V}, \tilde{E})$:

$$\begin{aligned} \tilde{V} &= \{v_{(i,j)} : 1 \leq i, j \leq n\}, \\ \tilde{E} &= \{[v_{(i,j)}, v_{(k,l)}] : i \neq k, j \neq l\}. \end{aligned}$$

Если в матрице есть нули, то соответствующие вершины в графе отсутствуют. Как и квадратичную задачу о назначениях, (2) можно рассматривать как задачу поиска максимальной клики в графе $\tilde{G}(\tilde{V}, \tilde{E})$ с минимальным общим весом вершин и ребер [17]. В качестве веса вершины $\omega(v_{(i,j)})$ принимается значение стоимости C_{ij} ; вес ребра $\omega[v_{(i,j)}, v_{(k,l)}]$ равен в общем виде $F(j, l)$, а в эквивалентной квадратичной задаче — d_{jl} .

Пусть теперь есть несколько иной граф $G(V, E)$ с теми же вершинами $V = \tilde{V}$, но с ребрами $E = \{[v_{(i,j)}, v_{(k,j+1)}] : i \neq k\}$, соединяющими вершины, соответствующие соседним столбцам матрицы Φ . Веса ребер априори неизвестны, но могут быть вычислены в соответствии с (2) как значения известной функции $F(\pi^{(j)}(i_1), \pi^{(j)}(i_2), \dots, j, j+1)$, где $\pi^{(j)}$ — некоторая подстановка, известная на момент вычисления веса. Если добавить к нему две вершины с нулевым весом и нулевым весом смежных с ними ребер

$$\begin{aligned} v_0 &: \{[v_0, (i, 1)] : 1 \leq i \leq n\} \subset E, \\ v_D &: \{[(i, n), v_D] : 1 \leq i \leq n\} \subset E, \end{aligned}$$

то подстановка π будет задавать матрицу X_π , единичные элементы которой соответствуют пути минимальной длины (с минимальным суммарным весом

ребер и вершин) из корневой вершины v_0 в конечную вершину v_D (рис. 2). Матрица Φ , соответствующая данному графу, выглядит следующим образом:

$$\Phi = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}.$$

В таком случае задача поиска максимальной клики в графе $\tilde{G}(\tilde{V}, \tilde{E})$ сводится к задаче нахождения кратчайшего пути в графе $G(V, E)$. Дана матрица Φ , которая задает граф $G(V, E)$, где V — описанное выше множество вершин, веса которых равны C_{ij} , а E — ребра графа. Веса ребер, выходящих из корневой вершины v_0 , и ребер, входящих в вершину v_D , полагаются равными нулю. Веса остальных ребер априори не определены, но известна функция F . Необходимо найти кратчайший путь из v_0 в v_D , содержащий ровно одну вершину, соответствующую каждому столбцу и строке матрицы Φ .

3. Алгоритм решения

$G(V, E)$ является взвешенным ориентированным графом с неотрицательными весами ребер. Для поиска кратчайшего расстояния между двумя вершинами v_0 и v_D модифицирован широко известный алгоритм Дейкстры. Прежде чем приводить предлагаемую модификацию, стоит остановиться на основных моментах алгоритма Дейкстры. Его цель — поиск кратчайшего расстояния от одной вершины **a** до всех остальных во взвешенном ориентированном графе без дуг отрицательного веса. Каждой вершине графа сопоставляется метка — минимальное расстояние от нее до заданной вершины. Изначально значение меток всех вершин равно бесконечности, а метка начальной вершины равна нулю. Все вершины помечаются как непосещенные. Шаг алгоритма состоит в выборе среди еще непосещенных вершины **u** с минимальной меткой и посещении всех соседних с ней вершин. При этом вычисляются расстояния до них как сумма метки вершины **u** и веса ребра, соединяющего каждую соседнюю вершину с **u**. Если это значение меньше текущей метки вершины, то новым значением метки становится полученное расстояние до **a**. Вершина **u** помечается посещенной, и шаг алгоритма повторяется до тех пор, пока все вершины не будут посещены.

В графе $G(V, E)$ метка конечной вершины v_D будет равна минимальному значению функции стоимости (2). Чтобы получить весь путь, необходимо в каждой вершине хранить информацию о *родительской* вершине, из которой совершен шаг. Далее подробно рассматривается модификация алгоритма Дейкстры.

На начальном этапе метка корневой вершины полагается равной 0, родительская вершина для нее отсутствует, метки остальных вершин равны бесконечности, а их родители не определены, множество непосещенных вершин U совпадает с V . Шаг алгоритма начинается с выбора среди множества U вершины $u_{(k,l)}$ с минимальной меткой (на первом шаге это вершина v_0). Далее среди всех соседних вершин определяется множество Q , в которые возможно сделать шаг. Здесь есть две причины, по которым необходима эта проверка.

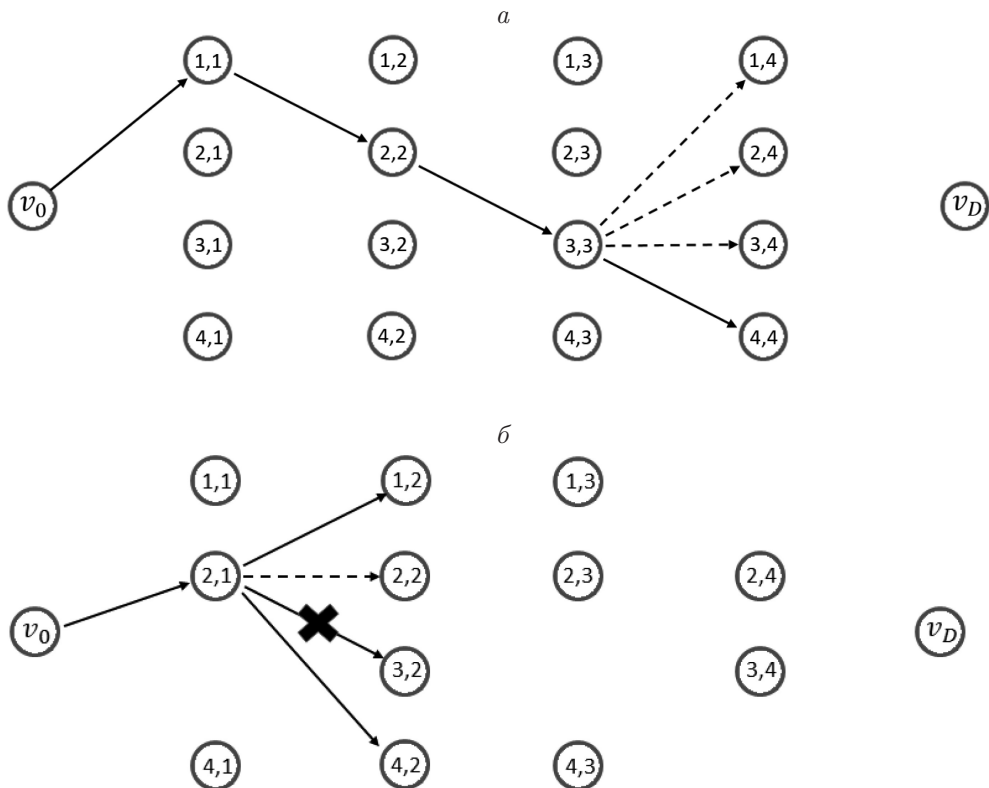


Рис. 3. *а* – Единственной допустимой вершиной, куда можно сделать шаг из вершины $v_{(3,3)}$, является $v_{(4,4)}$. *б* – Выбор вершины $v_{(3,2)}$ на данном шаге невозможен, так как из нее нельзя построить путь до v_D в силу причин, проиллюстрированных на рис. 3, *а*.

Каждой вершине, кроме корневой и конечной, сопоставлена пара (k, l) , которая соответствует элементу ϕ_{kl} матрицы Φ , и она имеет вес $\omega(v_{(k,l)}) = C_{kl}$. Чтобы добиться взаимной однозначности, вершина $q_{(k,l)} \in Q$ должна иметь пару индексов, отличную по обоим значениями от всех уже посещенных вершин (рис. 3, *а*). Поскольку в матрице возможных назначений Φ есть нулевые элементы, которые не задают вершин графа (рис. 2), то существуют вершины, из которых невозможно построить путь до конечной точки (рис. 3, *б*).

Чтобы определить, принадлежит ли вершина $q_{(k,l)}$ множеству Q , из матрицы Φ вычеркиваются строки и столбцы, соответствующие родительским вершинам, т.е. которые составляют путь до $q_{(k,l)}$. Если в матрице Φ не появляются нулевые строки или столбцы, то производятся те же действия с оставшейся матрицей и так далее, пока либо останется одна единица — $q_{(k,l)} \in Q$, либо обнаружится столбец или строка из одних нулей — $q_{(k,l)}$ признается недопустимой.

Данная процедура проверки — крайне важная модификация алгоритма. Нельзя не учитывать факт наличия недопустимых паросочетаний и отбросить проверку — это приведет к неверным результатам. Далее разобран слу-

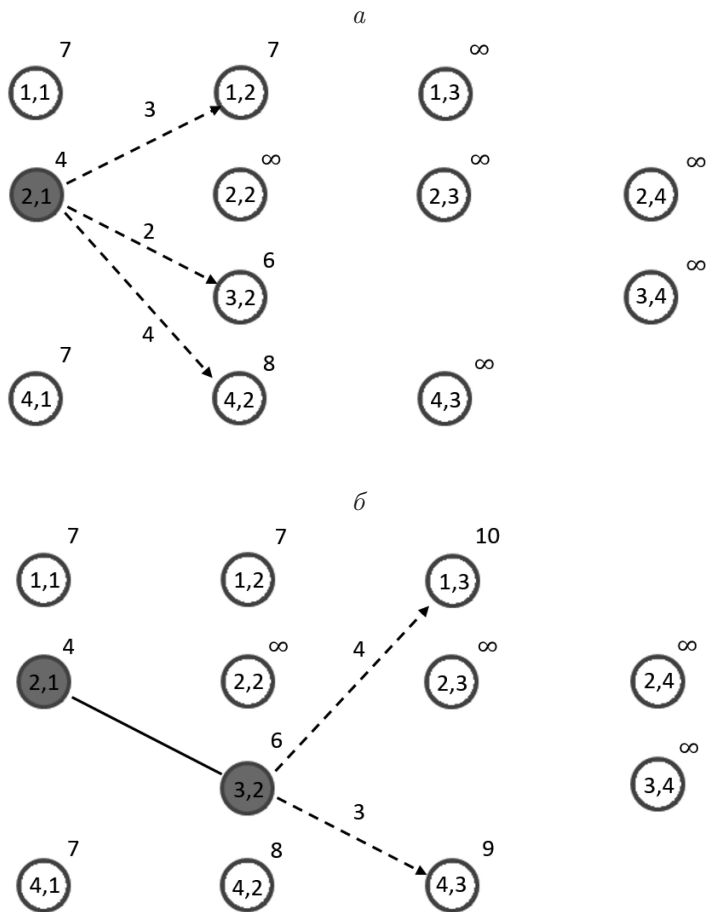


Рис. 4. Иллюстрация шага алгоритма. Закрашены посещенные вершины. Числа над стрелками соответствуют весам ребер. *a* – Изменение маркеров соседних вершин; *б* – вершина с маркером 6 посещена и изменяются маркеры соседних для нее вершин

чай, когда отсутствие проверки допустимости вершины приводит к неверному результату.

Если на шаге алгоритма не осуществлять проверку на возможность дойти до конечной вершины, то вершине $v_{(3,2)}$ присвоится значение маркера, равное 6 (рис. 4,*a*). На следующем шаге она будет отмечена посещенной, так как ее маркер будет минимальным среди всех непосещенных вершин (рис. 4,*б*). Далее алгоритм не сможет изменить маркер этой вершины, поскольку все остальные маркеры меньше 6, а веса ребер положительны (рис. 5,*a*). Родительской вершиной для нее навсегда останется вершина $v_{(2,1)}$.

Спустя несколько шагов окажется, что из нее нельзя дойти до конечной вершины (рис. 5,*б*). Причина в том, что самый первый шаг (рис. 4,*a*) был недопустим, из-за чего родителя этой вершины никак не изменить. При этом оптимальным окажется путь, изображенный сплошной линией на рис. 6, что

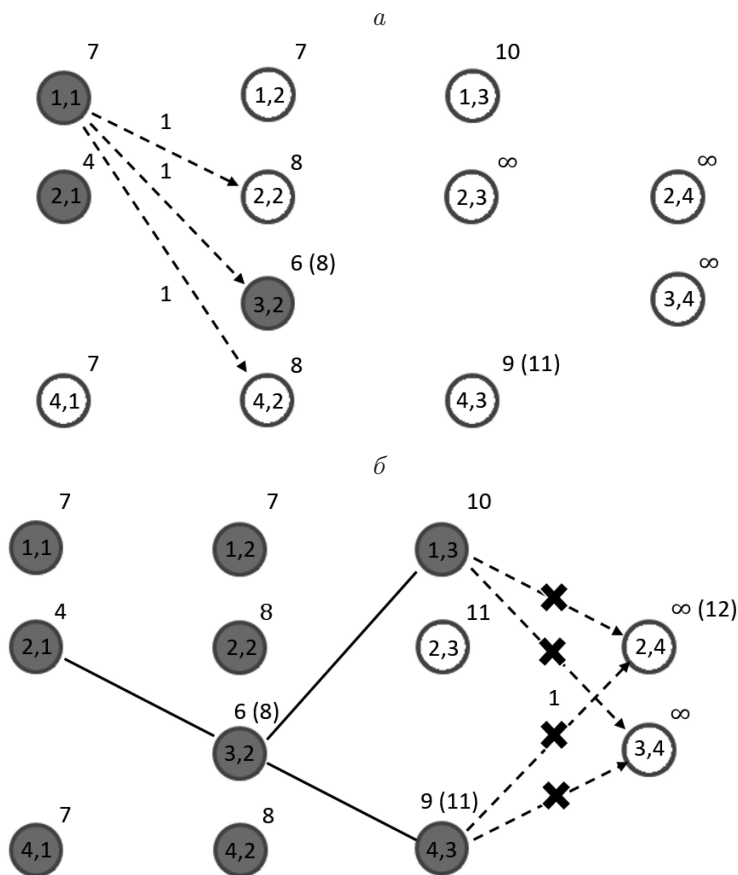


Рис. 5. Иллюстрация шага алгоритма. *a* – Значение маркера вершины $v_{(3,2)}$ не меняется. В скобках указаны возможные значения маркеров. *б* – Шаги, обозначенные пунктиром, запрещены. Сплошными линиями выделены связи с родительскими вершинами.

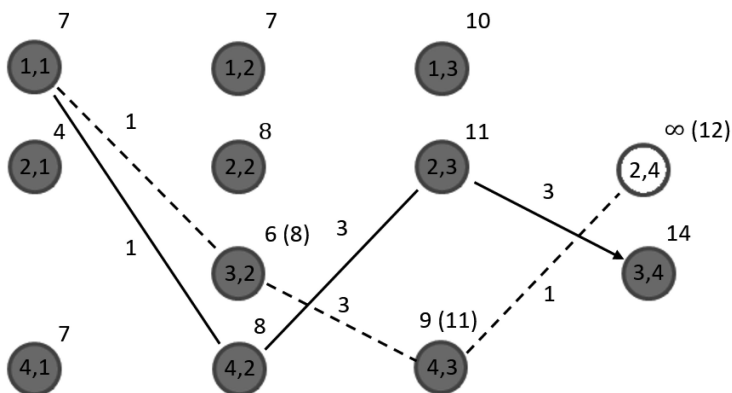


Рис. 6. Возможные пути в графе. Выбранный путь (сплошная линия) дорожке.

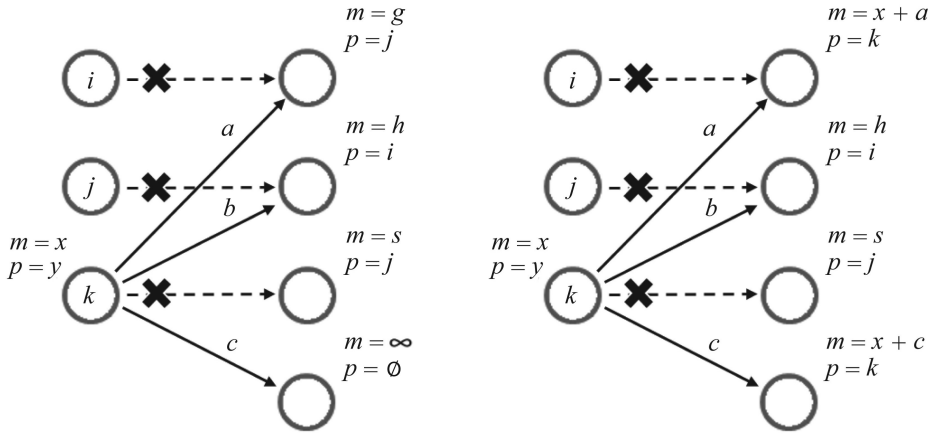


Рис. 7. Иллюстрация шага алгоритма; p – родительская вершина, m – метка вершины, $x + a > g$; $x + b > h$; $x + c < \infty$.

неправильно. Если же предусмотреть все, то алгоритм найдет оптимальный путь, изображенный штрихом на рис. 6 (в скобках указаны значения маркеров, которые бы имели вершины в таком случае).

После определения множества Q допустимых вершин вычисляется вес ω для каждого ребра, соединяющего u и все $q \in Q$, а также метки m всех допустимых вершин. Чтобы правильно вычислить стоимость назначений для каждой вершины, необходимо хранить множество всех ее родительских вершин $P[u_{(i,j)}] = \{v_{(i(j),j)}, v_{(i(j-1),j-1)}, \dots, v_{(i(1),1)}\}$. Это множество составляет текущий путь до нее от корневой вершины и задает подстановку $\pi^{(P)}$ для вычисления веса ребра:

$$(4) \quad \omega[u_{(i,j)}, q_{(k,j+1)}] = F\left(\pi^{(P)}(i_1), \pi^{(P)}(i_2), \dots, j, j+1\right),$$

где $k = 1, \dots, |Q|$, $j = \pi^{(P)}(k)$,

$$(5) \quad m(q_{(k,j+1)}) = m(u_{(i,j)}) + C_{kj+1} + \omega(u_{(i,j)}, q_{(k,j+1)}).$$

Затем в соответствии с классическим шагом алгоритма Дейкстры происходит сравнение с существующей меткой $m^*(q_{k,j+1})$. Если значение новой метки меньше существующей, то метка вершины изменяется, а родителем помечается вершина, из которой произведен шаг (рис. 7).

Когда все допустимые вершины рассмотрены, $u_{(i,j)}$ помечается посещенной и удаляется из множества U . После чего шаг алгоритма повторяется до тех пор, пока вершиной с минимальной меткой не окажется $u_{(i,n)}$. В таком случае единственной вершиной, куда возможно сделать шаг, является конечная вершина. Цепочка всех родителей составит кратчайший путь из v_0 в v_D . Их индексы определяют финальную подстановку (биекцию) π и соответствующее решение задачи о назначении X_π . Значение метки $m(v_D)$ (см. (4), (5))

определит минимальную стоимость в соответствии с функцией стоимости (2):

[illegible]

Таким образом, метка конечной вершины равна сумме всех весов ребер, составляющих путь из v_0 в v_D :

$$(7) \quad m(v_D) = C_{n,\pi(n)} + C_{n-1,\pi(n-1)} + \dots + C_{2,\pi(2)} + C_{1,\pi(1)} + \\ + F(\pi(1), \pi(2)) + \dots + F(\pi(1), \dots, \pi(n-1)) + F(\pi(1), \dots, \pi(n)).$$

Полученное выражение (7) совпадает с целевым функционалом (2), а значит, значение метки $m(v_D)$ равно минимуму стоимости назначений, что иллюстрирует оптимальность решения π задачи о назначении.

4. Результаты и практическое применение алгоритма

В качестве демонстрации результатов работы предлагается рассмотреть применение данного алгоритма для решения задачи о назначениях, возникающей в нефтяной индустрии при планировании бурения на месторождении. Задача состоит в следующем: требуется пробурить скважины так, чтобы каждая скважина вскрывала нефтяной пласт в определенном месте. При этом есть множество устьев (начало скважин на поверхности) и множество целей в пласте. Необходимо пробурить скважину из устья до цели так, чтобы обеспечить минимальную суммарную стоимость бурения. Кроме того, на геометрию скважин накладываются некоторые ограничения, из-за которых существует риск пересечения траекторий. Поэтому необходимо не допустить опасного сближения скважин.

В терминах задачи о назначениях необходимо установить взаимно однозначное соответствие между двумя данными множествами. Заданы стоимости C_{ij} строительства скважины из j -го устья, вскрывающего пласт в i -й точке. Функция F выражает зависимость от расстояния между скважинами в пространстве. Если эта функция зависит от положения только двух скважин и нет разницы в выборе устья, то задача становится в точности квадратичной задачей, описанной в главе 2.

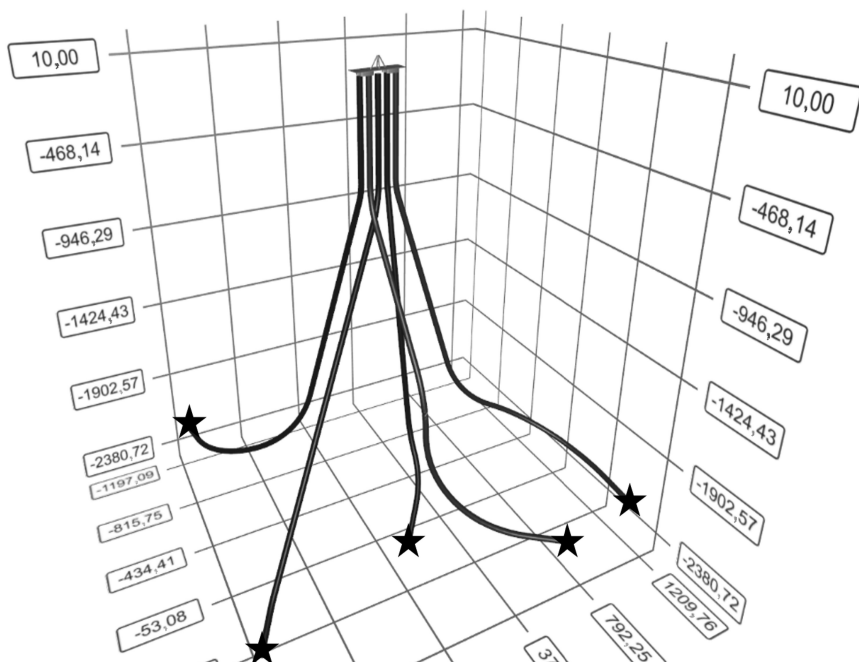


Рис. 8. Результат применения алгоритма при решении задачи бурения пяти скважин.

Решение π задачи о назначениях с такими функциями стоимостей, полученное в результате работы рассматриваемого алгоритма, определит оптимальное множество скважин. На рис. 8 изображены траектории скважин, соответствующие решению для пяти устьев и соответственно пяти целей.

В приведенном примере для каждой скважины дополнительно подбиралась ее геометрия, чтобы протяженность была минимальной. Также вычисление веса ребра (функции F) на каждом шаге алгоритма может быть крайне ресурсозатратно в зависимости от специфики прикладной задачи и связано с наличием недопустимых паросочетаний. В данном случае не все скважины возможно пробурить из-за технологических ограничений на бурение. Поэтому необходимо провести анализ трудоемкости самого алгоритма.

Далее приведены зависимости временных затрат от размерности задачи. На рис. 9 представлена зависимость времени работы алгоритма, реализованного на языке MATLAB, от количества элементов в множествах n . Значения $C_{(ij)}$ заданы и не меняются в ходе решения. Элементами множеств являются координаты точек на поверхности и в пласте. Функция F задана в виде (3), где

$$(8) \quad d_{\pi(i)\pi(j)} = |S(i, \pi(i)) - S(j, \pi(j))| ,$$

S – длина отрезка, соединяющего точки i и $\pi(i)$.

Кривая «А» соответствует случаю, когда матрица Φ имеет нули. На каждом шаге алгоритма необходимо производить проверку для определения мно-

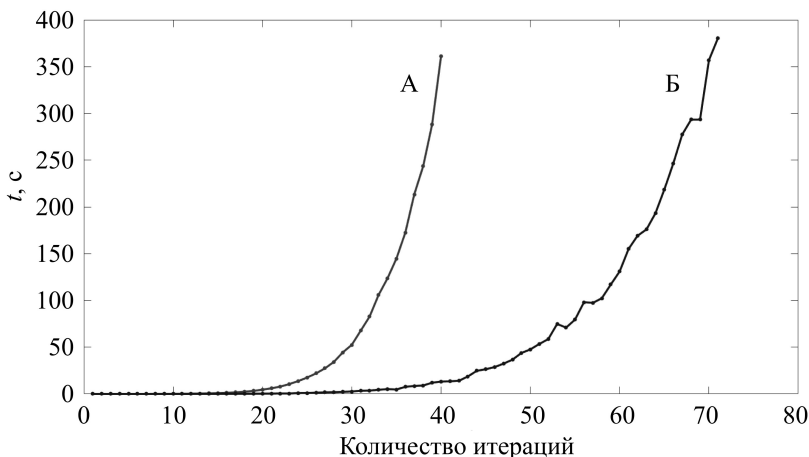


Рис. 9. Зависимость времени работы алгоритма от размерности задачи в случае наличия нулей в матрице Φ (кривая «А»), и их отсутствия (кривая «Б»).

жества допустимых вершин Q (см. главу 3). Временные затраты резко возрастают при размерности задачи более 25–30. Если же в матрице нет нулей, что соответствует отсутствию каких-либо ограничений на назначения, то нет необходимости в этой проверке (кривая «Б» на графике).

Полученный результат обосновывает применимость алгоритма в прикладных задачах, в которых количество объектов невелико. В задаче бурения, например, число скважин, которые бурят с одной площадки, редко превышает 20–25. Кроме того, зависимость, полученная при отсутствии недопустимых паросочетаний, иллюстрирует разрешимость за разумное время рассматриваемой в главе 2 квадратичной задачи о назначениях при размерности 60–70. Отбрасывая условие равенства потоков $f_{ij} = \delta_{ij}$, задавшись вместо этого симметричной матрицей $b_{ij} = |\beta_i - \beta_j|$, функцию F можно представить в следующем виде:

$$(9) \quad F(\pi(1), \dots, \pi(i)) = d_{\pi(i)\pi(1)} b_{i,1} + d_{\pi(i)\pi(2)} b_{i,2} + \dots + d_{\pi(i)\pi(i-1)} b_{i,i-1} = \sum_{j < i} d_{\pi(i)\pi(j)} |\beta_i - \beta_j|.$$

Тогда задача (2) с функцией (9) представляет собой специальный случай квадратичной задачи о назначении, описанный в [18], и может быть также решена с помощью предлагаемого алгоритма. Таким образом, подбирая функцию F , возможно разрешить различные специальные случаи квадратичной задачи о назначениях.

5. Заключение

Разработан алгоритм для решения задачи о назначениях специальной структуры с функцией стоимости общего вида при наличии ограничений на допустимые паросочетания. При таком обобщении затраты на назначения невозможно определить без предположения о подстановке или задать их в

виде матрицы с известными коэффициентами. Кроме некоторых постоянных коэффициентов стоимости, необходимо учесть зависимость затрат на одно назначение от всех остальных назначений. В терминах теории графов данная проблема может быть сформулирована как задача поиска кратчайшего пути в графе, построенном по всем возможным назначениям.

Предлагаемый метод решения основан на модификации алгоритма Дейкстры для нахождения кратчайшего пути во взвешенном ориентированном графе и не требует задания определенного вида подстановки.

Разработанный метод является эмпирическим, поскольку авторы не касались вопроса строгого математического обоснования эффективности работы алгоритма. Единственным обоснованием является хорошая работа на практике. Алгоритм был применен для решения прикладной задачи, связанной с бурением скважин. В результате получены зависимости времени работы от размерности задачи. Отмечено, что на время оказывает сильное влияние наличие недопустимых паросочетаний. Однако для характерных задачи бурения размерностей алгоритм эффективно находит оптимальные назначения. Другим практическим применением алгоритма может быть его использование в задачах размещения радиолокационного оборудования различной мощности в определенном множестве позиций для покрытия области наибольшей площади.

СПИСОК ЛИТЕРАТУРЫ

1. Ловас Л., Пламмер М. Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии. М.: Мир, 1998.
2. Dantzig G.B. Linear Programming and Extensions, Princeton University Press, Princeton, N.J. 1963 / Linear Programm. Extensions 1963. MLA.
3. Barr R.S., Glover F., Klingman D. The alternating basis algorithm for assignment problems // Math. Programm. 1977. Т. 13. No. 1. С. 1–13.
4. Kuhn H.W. The Hungarian method for the assignment problem // Naval Res. Logist. (NRL). 1955. Т. 2. No. 1–2. С. 83–97.
5. Bertsekas D.P. A new algorithm for the assignment problem // Math. Programm. 1981. Т. 21. No. 1. С. 152–171.
6. Хемди А. Таха Введение в исследование операций. 7-е изд М.: Вильямс, 2007.
7. Burkard R.E., Cela E. Linear assignment problems and extensions / Handbook Combinat. Optim. Springer US, 1999. С. 75–149.
8. Koopmans T.C., Beckmann M. Assignment problems and the location of economic activities // Econometrica: j. Econometric Soc. 1957. Т. 25. No. 1. С. 53–76.
9. Sahni S., Gonzalez T. P-complete approximation problems // J. ACM (JACM). 1976. Т. 23. No. 3. С. 555–565.
10. Hardy G.H., Littlewood J.E., Polya G. The maximum of a certain bilinear form // Proc. London Math. Soc. 1926. Т. 2. No. 1. С. 265–282.
11. Burkard R.E., Cela E., Rote G., Woeginger G.J. The quadratic assignment problem with a monotone anti-Monge and a symmetric Toeplitz matrix: Easy and hard cases // Math. Programm. 1998. Т. 82. No. 1. С. 125–158.

12. *Demidenko V.M., Finke G., Gordon V.S.* Well solvable cases of the quadratic assignment problem with monotone and bimonotone matrices // J. Math. Modell. Algorithms. 2006. T. 5. No. 2. C. 167–187.
13. *Burkard R., Dell'Amico M., Martello S.* Assignment problems: revised reprint. Siam, 2012. T. 125.
14. *Tate D.M., Smith A.E.* A genetic approach to the quadratic assignment problem // Comput. Oper. Res. 1995. T. 22. No. 1. C. 73–83.
15. *Maniezzo V., Colorni A.* The ant system applied to the quadratic assignment problem // IEEE Transact. Knowledge Data Engineer. 1999. T. 11. No. 5. C. 769–778.
16. *Taillard E.* Robust taboo search for the quadratic assignment problem // Parallel Comput. 1991. T. 17. No. 4. C. 443–455.
17. *Jünger M., Kaibel V.* A basic study of the QAP polytope // Technical Report 96.215. Institut für Informatik, Universität zu Köln, Germany. 1996.
18. *Çela E., Schmuck N.S., Wimer S., Woeginger G.J.* The Wiener maximum quadratic assignment problem [Электронный ресурс] // arXiv.org. 2011. Дата обновления: 20.04.2011. URL: <https://arxiv.org/abs/1102.3030v3> (дата обращения: 15.02.2017)

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 10.03.2017

После доработки 11.01.2018

Принята к публикации 08.11.2018