

The Optimized Algorithm of Finding the Shortest Path in a Multiple Graph

A. V. Smirnov¹DOI: [10.18255/1818-1015-2023-1-6-15](https://doi.org/10.18255/1818-1015-2023-1-6-15)¹P. G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl 150003, Russia.

MSC2020: 05C38, 05C65

Research article

Full text in Russian

Received January 18, 2023

After revision February 20, 2023

Accepted February 22, 2023

In this paper, we study undirected multiple graphs of any natural multiplicity $k > 1$. There are edges of three types: ordinary edges, multiple edges and multi-edges. Each edge of the last two types is a union of k linked edges, which connect 2 or $(k + 1)$ vertices, correspondingly. The linked edges should be used simultaneously. If a vertex is incident to a multiple edge, it can be also incident to other multiple edges and it can be the common end of k linked edges of some multi-edge. If a vertex is the common end of some multi-edge, it cannot be the common end of another multi-edge.

As for an ordinary graph, we can define the integer function of the length of an edge for a multiple graph and set the problem of the shortest path joining two vertices. Any multiple path is a union of k ordinary paths, which are adjusted on the linked edges of all multiple and multi-edges. In the article, we optimize the algorithm of finding the shortest path in an arbitrary multiple graph, which was obtained earlier. We show that the optimized algorithm is polynomial. Thus, the problem of the shortest path is polynomial for any multiple graph.

Keywords: multiple graph; multiple path; shortest path; reachability set; polynomial algorithm

INFORMATION ABOUT THE AUTHORS

Alexander Valeryevich Smirnov
correspondence author

orcid.org/0000-0002-0980-2507. E-mail: alexander_sm@mail.ru
PhD, Associate Professor, Department of Theoretical Computer Science.

Funding: This work was supported by P. G. Demidov Yaroslavl State University Project № VIP-016.

For citation: A. V. Smirnov, "The Optimized Algorithm of Finding the Shortest Path in a Multiple Graph", *Modeling and analysis of information systems*, vol. 30, no. 1, pp. 6-15, 2023.

Оптимизированный алгоритм поиска кратчайшего пути в кратном графе

А. В. Смирнов¹

DOI: [10.18255/1818-1015-2023-1-6-15](https://doi.org/10.18255/1818-1015-2023-1-6-15)

¹Ярославский государственный университет им. П. Г. Демидова, ул. Советская, д. 14, г. Ярославль, 150003 Россия.

УДК 519.17

Научная статья

Полный текст на русском языке

Получена 18 января 2023 г.

После доработки 20 февраля 2023 г.

Принята к публикации 22 февраля 2023 г.

В статье рассматриваются неориентированные кратные графы произвольной натуральной кратности $k > 1$. Кратный граф содержит ребра трех типов: обычные, кратные и мультиребра. Ребра последних двух типов представляют собой объединение k связанных ребер, которые соединяют 2 или $(k + 1)$ вершину соответственно. Связанные ребра могут использоваться только согласованно. Если вершина инцидентна кратному ребру, то она может быть инцидентна другим кратным ребрам, а также она может быть общим концом k связанных ребер мультиребра. Если вершина является общим концом мультиребра, то она не может быть общим концом никакого другого мультиребра.

Как и для обычного графа, для кратного графа можно ввести целочисленную функцию длины ребра и поставить задачу о кратчайшем пути между двумя вершинами. Кратный путь является объединением k обычных путей, согласованных на связанных ребрах кратных и мультиребер. В статье оптимизирован полученный ранее алгоритм поиска кратчайшего пути в произвольном кратном графе. Показано, что оптимизированный алгоритм полиномиален. Таким образом, задача о кратчайшем пути является полиномиальной для любого кратного графа.

Ключевые слова: кратный граф; кратный путь; кратчайший путь; множество достижимости; полиномиальный алгоритм

ИНФОРМАЦИЯ ОБ АВТОРАХ

Александр Валерьевич Смирнов
автор для корреспонденции

orcid.org/0000-0002-0980-2507. E-mail: alexander_sm@mail.ru
канд. физ.-мат. наук, доцент, кафедра теоретической информатики.

Финансирование: Работа выполнена в рамках инициативной НИР ЯрГУ им. П. Г. Демидова № VIP-016.

Для цитирования: A. V. Smirnov, "The Optimized Algorithm of Finding the Shortest Path in a Multiple Graph", *Modeling and analysis of information systems*, vol. 30, no. 1, pp. 6-15, 2023.

Введение

В данной статье мы рассмотрим задачу о *кратчайшем пути* в кратном графе. *Кратные графы* содержат три типа ребер (обычные, кратные и мультиребра) и являются обобщением обычных графов – по сути, обычный граф имеет кратность $k = 1$. Определение кратного графа кратности $k > 1$ было сформулировано в статье [1]. Там же была поставлена задача о кратчайшем кратном пути между двумя вершинами, дано определение связности кратного графа, предложены полиномиальные алгоритмы ее проверки. В отличие от обычного графа, в связном кратном графе путь между двумя вершинами существует не всегда, и в работе [1] обоснован критерий существования кратного пути между двумя вершинами, а также получены быстрые полиномиальные алгоритмы для проверки условия критерия.

В статье [2] получен полиномиальный алгоритм поиска кратчайшего кратного пути в *делимом графе* – частном случае кратного графа. Особенностью делимых графов является возможность представления графа в виде объединения k частей. Каждая часть – это обычный граф, в котором присутствует ровно одно связанное ребро каждого кратного и мультиребра. При этом никакие две части не содержат одинаковых ребер. Кроме того, в работе [2] предложена модификация указанного алгоритма для случая произвольного кратного графа. Однако модифицированный алгоритм является экспоненциальным по параметру k в общем случае.

В данной статье мы оптимизируем общий алгоритм поиска кратчайшего кратного пути. В результате будет получен полиномиальный алгоритм для произвольного кратного графа.

1. Постановка задачи о кратчайшем кратном пути

Напомним несколько понятий, связанных с кратными графами и путями. Поскольку данная статья продолжает исследование, описанное в работе [2], здесь будут сформулированы только самые важные определения. Поясняющие примеры и ряд связанных определений были подробно рассмотрены в статьях [1, 2].

Определение 1. Кратный граф G произвольной натуральной кратности $k > 1$ – это граф, вершины которого могут соединяться ребрами одного из 3 видов:

1. Обычное ребро e^o ; множество обычных ребер обозначим через E^o .
2. Кратное ребро e^k между двумя вершинами, которое состоит из k одинаковых связанных ребер; связанные ребра кратного ребра могут использоваться только согласованно; множество кратных ребер обозначим через E^k .
3. Связанное ребро e между двумя вершинами, имеющее один общий конец с другим $(k - 1)$ ребром (у любых двух из k связанных ребер только один конец является общим); множество связанных общей вершиной ребер будем называть мультиребром e^m ; связанные ребра мультиребра могут использоваться только согласованно; множество мультиребер обозначим через E^m .

Если вершина инцидентна какому-либо кратному ребру, то она может быть инцидентна другим кратным ребрам, а также она может быть общим концом какого-либо мультиребра.

Если вершина является общим концом какого-либо мультиребра, то она не может быть общим концом никакого другого мультиребра.

Если вершина является отдельным концом мультиребра или инцидентна обычному ребру, то она не может быть общим концом мультиребра и не может быть инцидентна кратному ребру.

Множества вершин и ребер графа G обозначим через V и E соответственно. Заметим, что $E = E^o \cup E^k \cup E^m$.

В данной статье рассматриваются только неориентированные кратные графы.

Определение 2. Обычной вершиной назовем вершину, которая инцидентна обычному ребру или является отдельным концом мультиребра.

Кратной вершиной назовем вершину, которая инцидентна кратному ребру или является общим концом мультиребра.

Из определения 1 следует, что множества обычных и кратных вершин не пересекаются. При этом кратная вершина может быть соединена с обычными только посредством мультиребра.

Определение 3. $S(x, y) = \cup_{i=1}^k S^i(x, y)$ является кратным путем из вершины x в вершину y в графе $G(V, E)$, если выполнены следующие условия:

1. $S^i(x, y) = (\{x, v_1^i\}, \{v_1^i, v_2^i\}, \dots, \{v_{l_i-1}^i, v_{l_i}^i\}, \{v_{l_i}^i, y\})$, где $l_i \geq 0$, – последовательность ребер, представляющая собой обычный (некратный) путь из x в y , где каждое ребро $\{a, b\}$ является либо обычным ребром в графе $G(V, E)$, либо i -ым связанным ребром кратного или мультиребра. Значения l_i и l_j ($i \neq j$) не согласовываются и могут быть как равными, так и различными. Если в путь $S(x, y)$ не входит ни одного кратного или мультиребра, то $S^2(x, y) = S^3(x, y) = \dots = S^k(x, y) = \emptyset$.
2. Любая обычная вершина может встретиться в $S^i(x, y)$ несколько раз, то есть $S^i(x, y)$ может содержать циклы.
3. Никакая кратная вершина не может встретиться в $S^i(x, y)$ дважды.
4. Любое обычное ребро может встречаться в $S^i(x, y)$ несколько раз, причем направления, в которых оно проходится в разных вхождениях, могут не совпадать.
5. Обычное ребро, входящее в $S^i(x, y)$, может также входить в любой $S^j(x, y)$, $j \neq i$.
6. Все пути $S^i(x, y)$ согласованы (одинаковы) на общей части. Это условие означает, что если связанное ребро какого-то кратного или мультиребра входит в некоторый путь $S^i(x, y)$, то остальные связанные ребра должны входить во все $S^j(x, y)$, $j \neq i$ (по одному связанному ребру в каждый $S^i(x, y)$). При этом порядок вхождения всех кратных и мультиребер во все $S^i(x, y)$ одинаков.

Фактически это значит, что если e_1 и e_2 – это два ребра пути $S(x, y)$, каждое из которых либо кратное, либо мультиребро, и в проекции $S^i(x, y)$ связанное ребро из e_1 проходится раньше связанного ребра из e_2 , то во всех остальных проекциях $S^j(x, y)$ связанные ребра из e_2 могут проходиться только после связанных ребер из e_1 .

7. Если $S(x, y)$ содержит мультиребро $\{x_0, \{x_1, \dots, x_k\}\}$, проходимое в направлении от общего конца, то он не может содержать никакого другого мультиребра $\{y_0, \{x_1, \dots, x_k\}\}$, проходимого в том же направлении. Аналогичное условие должно выполняться и в случае движения к общему концу.

Определение 4. Множеством достижимости по кратным ребрам для некоторой кратной вершины x назовем множество R_x^k всех вершин y таких, что существует путь из x в y , проходящий только по кратным ребрам.

Определение 5. Множеством достижимости по обычным ребрам для некоторой обычной вершины x назовем множество R_x^o всех вершин y таких, что существует путь из x в y , проходящий только по обычным ребрам.

Очевидно, что $x \in R_x^k$, $x \in R_x^o$. Если $y \in R_x^k$, то $R_y^k = R_x^k$. Если $y \in R_x^o$, то $R_y^o = R_x^o$.

Определение 6. Целочисленная функция $l(e)$, определенная для всех ребер $e \in E$, является длиной (весом) ребра в кратном графе $G(V, E)$, если выполнено следующее:

1. $l(e) > 0$ для любого ребра e .
2. Если e является кратным или мультиребром, то $l(e_1) = l(e_2) = \dots = l(e_k)$ и $l(e) = k \cdot l(e_1)$, где $e_1, \dots, e_k$ – это связанные ребра данного ребра e .

Тогда длина кратного пути $S(x, y)$ будет определяться по формуле

$$l(S(x, y)) = \sum_{i=1}^k l(S^i(x, y)) = \sum_{i=1}^k \sum_{e_j \in S^i(x, y)} l(e_j),$$

при этом может оказаться $e_j = e_p$ ($j \neq p$), то есть в сумме учитывается каждое повторное вхождение обычного ребра в $S^i(x, y)$.

Задача 1 (кратчайший кратный путь). В кратном графе $G(V, E)$ требуется найти кратчайший путь из вершины x в вершину y , то есть такой путь $S_{\min}(x, y)$, что для любого пути $S(x, y)$ выполнено

$$l(S_{\min}(x, y)) \leq l(S(x, y)).$$

2. Полиномиальный алгоритм поиска кратчайшего кратного пути в произвольном кратном графе

Пусть имеется произвольный взвешенный кратный граф $G(V, E)$ кратности k . Получим полиномиальный алгоритм решения задачи 1 для него. Для этого возьмем алгоритм 2 из статьи [2] и модифицируем его.

Как и ранее, на отдельных шагах алгоритма для нахождения минимальных участков пути, содержащих только обычные ребра, мы будем использовать известный алгоритм Дейкстры (см. [3]). Кроме того, для поиска подмножества минимальных участков пути на шаге 4.4 алгоритма мы будем применять венгерский алгоритм (см. [4]).

Алгоритм ищет кратчайший кратный путь $S_{\min}(x, y)$ между двумя выбранными вершинами x и y . Через e_a^m обозначим мультиребро, инцидентное кратной вершине a . В алгоритме мы будем использовать следующие структуры данных:

- множества достижимости R_a^k и R_b^o ;
- множества индексов I_a , ассоциированные с каждым мультиребром e_a^m ;
- $G^{ord}(V^{ord}, E^{ord})$ – обычный граф, минимальному пути $S_{\min}^{ord}(x, y)$ в котором будет соответствовать кратчайший кратный путь $S_{\min}(x, y)$ той же длины в исходном графе;
- $S_{mo}(a, b)$ – кратный путь между двумя кратными вершинами, состоящий из мультиребер e_a^m и e_b^m , а также из минимальных обычных путей между соответствующими парами обычных вершин – концов этих мультиребер;
- рейтинговая матрица R^j , используемая при поиске $S_{mo}(a, b)$.

Многие шаги алгоритма будут полностью совпадать с аналогичными шагами алгоритма 2 из статьи [2], поэтому такие шаги мы будем формулировать тезисно за исключением случаев, когда полное описание необходимо для понимания дальнейших действий.

Алгоритм (кратчайший путь в произвольном кратном графе).

1. Найдем все множества достижимости по кратным и обычным ребрам R_a^k и R_b^o . Пронумеруем все множества R_b^o в произвольном порядке от 1 до n и обозначим их через $R_1, \dots, R_n$.

2. Проверим выполнение критерия существования кратного пути между вершинами x и y . Если критерий не выполнен, выходим из алгоритма.

3. Для каждого мультиребра $e_a^m = \{a, \{a_1, \dots, a_k\}\}$ сформируем мультимножество индексов $I_a = \{i_1, \dots, i_k\}$ таким образом, что каждый i_p равен номеру множества достижимости, в которое попадает a_p : $a_p \in R_{i_p}$. Пусть вершины $a_1, \dots, a_k$ мультиребра e_a^m пронумерованы в порядке возрастания значений i_p (иначе их можно быстро перенумеровать).

4. Будем строить обычный граф $G^{ord}(V^{ord}, E^{ord})$ следующим образом.

4.1. Для каждой кратной вершины $v \in V$ создадим обычную вершину $v \in V^{ord}$.

4.2. Для каждого кратного ребра $\{u, v\} \in E$ создадим обычное ребро $\{u, v\} \in E^{ord}$ той же длины.

4.3. Для каждой кратной вершины a , инцидентной мультиребру e_a^m , создадим дополнительную обычную вершину $a' \in V^{ord}$ и обычное ребро $\{a, a'\} \in E^{ord}$ длины 1.

4.4. Рассмотрим все пары мультиребер из E^m . Для каждой такой пары $\{e_a^m, e_b^m\}$ ($e_a^m = \{a, \{a_1, \dots, a_k\}\}$, $e_b^m = \{b, \{b_1, \dots, b_k\}\}$) проверим равенство $I_a = I_b$. Если оно выполнено, будем поочередно рассматривать все подмножества I_a^j совпадающих индексов из I_a .

Если подмножество I_a^j содержит только один элемент i_p ($I_a^j = \{i_p\}$), с помощью алгоритма Дейкстры найдем кратчайший обычный путь $S_{\min}(a_p, b_p)$, проходящий только по вершинам из R_{i_p} . Обозначим $c_p = b_p$.

Если подмножество I_a^j содержит q одинаковых элементов ($I_a^j = \{i_p, i_{p+1}, \dots, i_{p+q-1}\}$, $i_s = i_t$ для всех $s \in \overline{p, p+q-1}$, $t \in \overline{p, p+q-1}$), найдем с помощью алгоритма Дейкстры q^2 кратчайших обычных путей $S_{\min}(a_s, b_t)$ ($s \in \overline{p, p+q-1}$, $t \in \overline{p, p+q-1}$), проходящих только по вершинам из R_{i_p} . Среди них нужно выбрать q путей, попарно непересекающихся в начальных и конечных вершинах и имеющих при этом минимальную суммарную длину.

Это можно сделать следующим образом. Сопоставим каждому a_s работника с номером s , а каждому b_t – работу с номером t ($s \in \overline{p, p+q-1}$, $t \in \overline{p, p+q-1}$). Далее сформируем *рейтинговую матрицу* R^j размерности $q \times q$, где

$$r_{st} = l(S_{\min}(a_s, b_t)).$$

Задача выбора q путей требуемого вида свелась к задаче о назначениях с рейтинговой матрицей R^j , а эта задача может быть решена при помощи венгерского алгоритма (см. [4]). Классический венгерский алгоритм ищет решение максимального веса, однако в статье [5] показано, что тот же алгоритм можно использовать и при поиске решения минимального веса. В результате в каждой строке и каждом столбце указанной матрицы будет выбран ровно один элемент, а сумма этих элементов будет минимально возможной. Выбранные элементы соответствуют искомым q путям.

Упорядочим выбранные пути по возрастанию номера начальной вершины, конечные вершины обозначим через c_r ($r \in \overline{p, p+q-1}$).

Просмотрев все подмножества, сформируем и запомним кратный путь

$$S_{mo}(a, b) = e_a^m \cup S_{\min}(a_1, c_1) \cup \dots \cup S_{\min}(a_k, c_k) \cup e_b^m,$$

который будет кратчайшим кратным путем без кратных ребер между вершинами a и b . Добавим в E^{ord} обычное ребро $\{a', b'\}$ длины $l(\{a', b'\}) = l(S_{mo}(a, b)) - 2$.

4.5. Если x – обычная вершина, скопируем ее в V^{ord} . Для всех мультиребер e_a^m , для которых это возможно, находим кратчайший кратный путь $S_{mo}(x, a)$ без кратных ребер как в алгоритме 2 из статьи [2]. Добавим в E^{ord} обычное ребро $\{x, a'\}$ длины $l(\{x, a'\}) = l(S_{mo}(x, a)) - 1$.

4.6. Если y – обычная вершина, скопируем ее в V^{ord} . Для всех мультиребер e_a^m , для которых это возможно, находим кратчайший кратный путь $S_{mo}(a, y)$ без кратных ребер как в алгоритме 2 из статьи [2]. Добавим в E^{ord} обычное ребро $\{a', y\}$ длины $l(\{a', y\}) = l(S_{mo}(a, y)) - 1$.

4.7. Если обе вершины x и y обычные и $y \in R_x^o$, находим кратчайший путь $S'_{\min}(x, y)$, проходящий только по вершинам из R_x^o . Добавим в E^{ord} обычное ребро $\{x, y\}$ длины $l(S'_{\min}(x, y))$.

5. Найдем кратчайший путь $S_{\min}^{ord}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$ с помощью алгоритма Дейкстры.

6. Преобразуем путь $S_{\min}^{ord}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$ в искомым кратчайший кратный путь $S_{\min}(x, y)$ в графе $G(V, E)$ как в алгоритме 2 из статьи [2].

Применимость полученного алгоритма для задачи 1 в случае произвольного кратного графа следует из применимости алгоритма 2 из статьи [2], которая была обоснована в указанной статье. Действительно, в новом алгоритме меняется лишь принцип выбора путей при построении графа $G^{ord}(V^{ord}, E^{ord})$, условия выбора остаются прежними.

Алгоритм 2 из статьи [2] был экспоненциальным из-за того, что на шаге 4.4 могла возникнуть ситуация перебора $q!$ (а в худшем случае – $k!$) вариантов подмножеств кратчайших путей. Однако в новом алгоритме мы для этой цели производим полиномиальное сведение подзадачи выбора к задаче о назначениях и решаем задачу о назначениях с помощью венгерского алгоритма. В статье [5] показана полиномиальность венгерского алгоритма, откуда и следует полиномиальность нашего алгоритма равно как и полиномиальность задачи 1 для произвольного кратного графа.

Отметим, что алгоритм универсален и применим для любого кратного графа, в том числе для делимого графа. Однако для делимого графа алгоритм 1 из статьи [2] будет более эффективным, поскольку он учитывает специфику построения такого графа.

3. Пример работы алгоритма

Продemonстрируем работу алгоритма. Для этого рассмотрим граф кратности 3, показанный на рис. 1. Для лучшего восприятия на рисунке подписаны только номера вершин без буквы “х”. На ребрах шрифтом Courier New отмечены длины. Кратные ребра показаны жирными линиями, а мультиребра – расщепляющимися на 3 части линиями. Обычные вершины показаны серым.

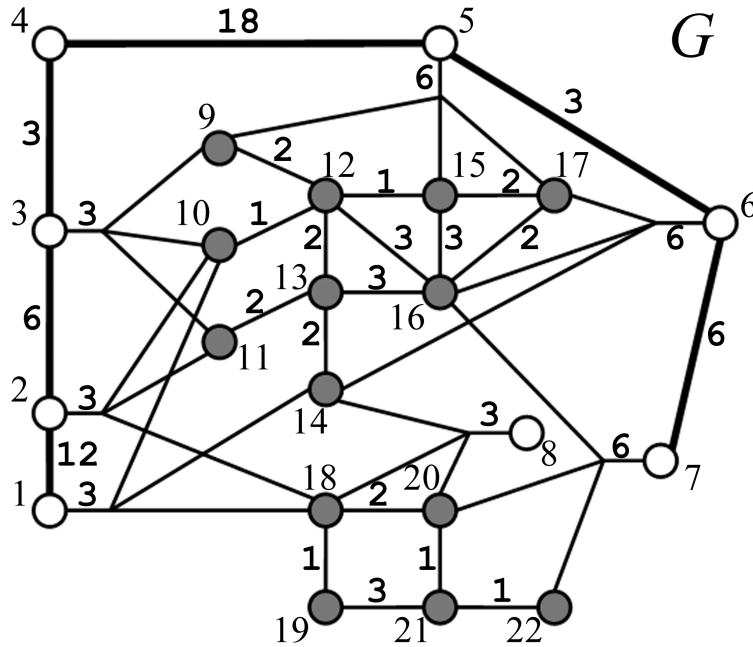


Fig. 1. Weighted graph of multiplicity 3

Рис. 1. Взвешенный граф кратности 3

Будем искать кратчайший кратный путь $S_{\min}(x_1, x_8)$. Вершины x_1 и x_8 кратные и граф $G(V, E)$ связан, поэтому такой путь существует (следует из критерия существования кратного пути – теорема 3 из статьи [1]). Определим и пронумеруем множества достижимости по обычным ребрам:

$$R_1 = R_{x_9}^o = \{x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}\}, \quad R_2 = R_{x_{18}}^o = \{x_{18}, x_{19}, x_{20}, x_{21}, x_{22}\}.$$

Множества индексов I_a для мультиребер тогда определяются так:

$$I_{x_1} = \{1, 1, 2\}, \quad I_{x_2} = \{1, 1, 2\}, \quad I_{x_3} = \{1, 1, 1\}, \quad I_{x_5} = \{1, 1, 1\}, \quad I_{x_6} = \{1, 1, 1\}, \quad I_{x_7} = \{1, 2, 2\}, \quad I_{x_8} = \{1, 2, 2\}.$$

Будем теперь строить обычный граф $G^{ord}(V^{ord}, E^{ord})$ (шаг 4 алгоритма). Шаги 4.1–4.3 реализуются тривиальным образом, шаги 4.5–4.7 не потребуются, так как путь строится между двумя кратными вершинами. Наибольшую сложность представляет шаг 4.4, рассмотрим его подробно. Здесь основные вычисления связаны с поиском кратных путей $S_{mo}(a, b)$ для подходящих пар мультиребер. Таких пар будет пять.

Сначала рассмотрим $e_{x_1}^m = \{x_1, \{x_{10}, x_{14}, x_{18}\}\}$ и $e_{x_2}^m = \{x_2, \{x_{10}, x_{11}, x_{18}\}\}$. Для подмножества $I_{x_1}^1 = \{1, 1\}$ найдем кратчайшие пути между всеми парами обычных вершин, соответствующих этому

подмножеству, и получим рейтинговую матрицу (выделенные элементы являются решением задачи о назначениях):

$$R^1 = \begin{array}{c|cc} & x_{10} & x_{11} \\ \hline x_{10} & \boxed{0} & 5 \\ x_{14} & 5 & \boxed{4} \end{array}$$

Таким образом, в $S_{mo}(x_1, x_2)$ нужно включить $S_{\min}(x_{10}, x_{10})$ и $S_{\min}(x_{14}, x_{11})$. Для подмножества $I_{x_1}^2 = \{2\}$ достаточно взять пустой кратчайший путь $S_{\min}(x_{18}, x_{18})$ и добавить его в $S_{mo}(x_1, x_2)$. Получаем:

$$S_{mo}(x_1, x_2) = e_{x_1}^m \cup \emptyset \cup (\{x_{14}, x_{13}\}, \{x_{13}, x_{11}\}) \cup \emptyset \cup e_{x_2}^m, \quad l(S_{mo}(x_1, x_2)) = 10.$$

Далее рассмотрим $e_{x_3}^m = \{x_3, \{x_9, x_{10}, x_{11}\}\}$ и $e_{x_5}^m = \{x_5, \{x_9, x_{15}, x_{17}\}\}$. Для единственного подмножества $I_{x_3}^1 = \{1, 1, 1\}$ найдем кратчайшие пути между всеми парами обычных вершин, соответствующих этому подмножеству, и получим рейтинговую матрицу:

$$R^1 = \begin{array}{c|ccc} & x_9 & x_{15} & x_{17} \\ \hline x_9 & \boxed{0} & 3 & 5 \\ x_{10} & 3 & \boxed{2} & 4 \\ x_{11} & 6 & 5 & \boxed{7} \end{array}$$

В $S_{mo}(x_3, x_5)$ нужно включить $S_{\min}(x_9, x_9)$, $S_{\min}(x_{10}, x_{15})$ и $S_{\min}(x_{11}, x_{17})$. Получаем:

$$S_{mo}(x_3, x_5) = e_{x_3}^m \cup \emptyset \cup (\{x_{10}, x_{12}\}, \{x_{12}, x_{15}\}) \cup (\{x_{11}, x_{13}\}, \{x_{13}, x_{16}\}, \{x_{16}, x_{17}\}) \cup e_{x_5}^m, \quad l(S_{mo}(x_3, x_5)) = 18.$$

Для пары $e_{x_3}^m = \{x_3, \{x_9, x_{10}, x_{11}\}\}$ и $e_{x_6}^m = \{x_6, \{x_{14}, x_{16}, x_{17}\}\}$ то же подмножество $I_{x_3}^1 = \{1, 1, 1\}$ приводит к рейтинговой матрице:

$$R^1 = \begin{array}{c|ccc} & x_{14} & x_{16} & x_{17} \\ \hline x_9 & 6 & \boxed{5} & 5 \\ x_{10} & 5 & 4 & \boxed{4} \\ x_{11} & \boxed{4} & 5 & 7 \end{array}$$

В $S_{mo}(x_3, x_6)$ нужно включить $S_{\min}(x_9, x_{16})$, $S_{\min}(x_{10}, x_{17})$ и $S_{\min}(x_{11}, x_{14})$. Получаем:

$$S_{mo}(x_3, x_6) = e_{x_3}^m \cup (\{x_9, x_{12}\}, \{x_{12}, x_{16}\}) \cup (\{x_{10}, x_{12}\}, \{x_{12}, x_{15}\}, \{x_{15}, x_{17}\}) \cup (\{x_{11}, x_{13}\}, \{x_{13}, x_{14}\}) \cup e_{x_6}^m,$$

$$l(S_{mo}(x_3, x_6)) = 22.$$

Стоит отметить, что $S_{mo}(x_3, x_6)$ может быть получен и в альтернативном виде — с использованием $S_{\min}(x_9, x_{17})$, $S_{\min}(x_{10}, x_{16})$ и $S_{\min}(x_{11}, x_{14})$.

Еще одна пара $e_{x_5}^m = \{x_5, \{x_9, x_{15}, x_{17}\}\}$ и $e_{x_6}^m = \{x_6, \{x_{14}, x_{16}, x_{17}\}\}$ с тем же множеством индексов дает рейтинговую матрицу:

$$R^1 = \begin{array}{c|ccc} & x_{14} & x_{16} & x_{17} \\ \hline x_9 & \boxed{6} & 5 & 5 \\ x_{15} & 5 & \boxed{3} & 2 \\ x_{17} & 7 & 2 & \boxed{0} \end{array}$$

В $S_{mo}(x_5, x_6)$ нужно включить $S_{\min}(x_9, x_{14})$, $S_{\min}(x_{15}, x_{16})$ и $S_{\min}(x_{17}, x_{17})$. Получаем:

$$S_{mo}(x_5, x_6) = e_{x_5}^m \cup (\{x_9, x_{12}\}, \{x_{12}, x_{13}\}, \{x_{13}, x_{14}\}) \cup (\{x_{15}, x_{16}\}) \cup \emptyset \cup e_{x_6}^m, \quad l(S_{mo}(x_5, x_6)) = 21.$$

Рассмотрим последнюю пару $e_{x_7}^m = \{x_7, \{x_{16}, x_{20}, x_{22}\}\}$ и $e_{x_8}^m = \{x_8, \{x_{14}, x_{18}, x_{20}\}\}$. Для подмножества $I_{x_7}^1 = \{1\}$ нужно взять кратчайший путь $S_{\min}(x_{16}, x_{14})$ и добавить его в $S_{mo}(x_7, x_8)$. Для подмножества $I_{x_7}^2 = \{2, 2\}$, как и выше, выполним сведение к задаче о назначениях:

$$R^2 = \begin{array}{c|cc} & x_{18} & x_{20} \\ \hline x_{20} & 2 & \boxed{0} \\ x_{22} & \boxed{4} & 2 \end{array}$$

Таким образом, в $S_{mo}(x_7, x_8)$ нужно включить $S_{\min}(x_{20}, x_{20})$ и $S_{\min}(x_{22}, x_{18})$ (но есть и альтернативный вариант с двумя путями длины 2). Получаем:

$$S_{mo}(x_7, x_8) = e_{x_7}^m \cup (\{x_{16}, x_{13}\}, \{x_{13}, x_{14}\}) \cup \emptyset \cup (\{x_{22}, x_{21}\}, \{x_{21}, x_{20}\}, \{x_{20}, x_{18}\}) \cup e_{x_8}^m, \quad l(S_{mo}(x_7, x_8)) = 18.$$

В итоге выполнения шага 4 алгоритма будет получен обычный граф $G^{ord}(V^{ord}, E^{ord})$, показанный на рис. 2. Вершины кратчайшего пути $S_{\min}^{ord}(x_1, x_8)$ (шаг 5) показаны серым. Длина этого пути $l(S_{\min}^{ord}(x_1, x_8)) = 61$.

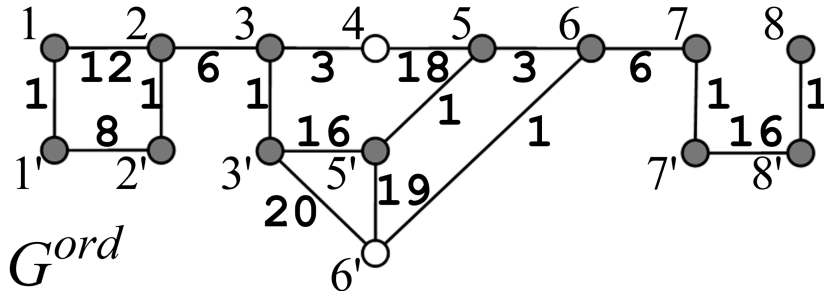


Fig. 2. Ordinary graph G^{ord} and the shortest path

Рис. 2. Обычный граф G^{ord} и кратчайший путь

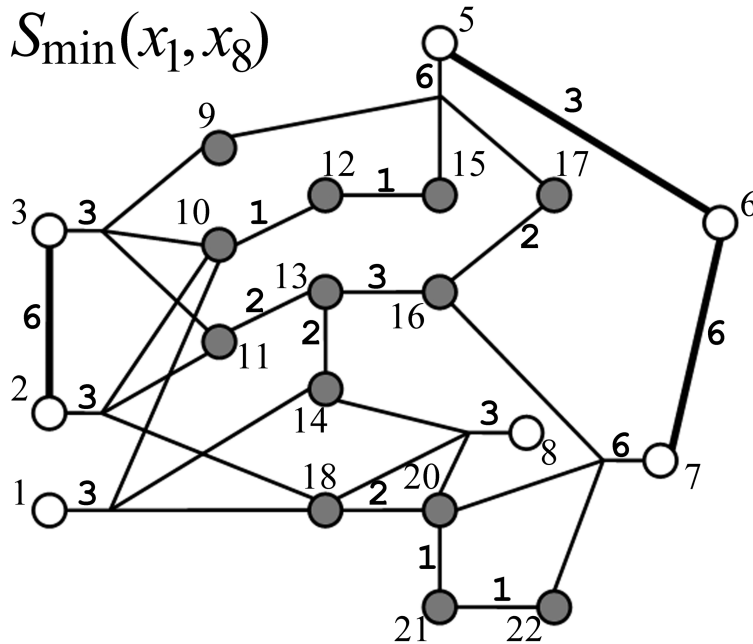


Fig. 3. The shortest path in the multiple graph G

Рис. 3. Кратчайший путь в кратном графе G

Применяя шаг 6, преобразуем путь $S_{\min}^{ord}(x_1, x_8)$ в искомый кратчайший кратный путь $S_{\min}(x_1, x_8)$ (рис. 3):

$$S_{\min}(x_1, x_8) = (S_{mo}(x_1, x_2), \{x_2, x_3\}, S_{mo}(x_3, x_5), \{x_5, x_6\}, \{x_6, x_7\}, S_{mo}(x_7, x_8)),$$

$$l(S_{\min}(x_1, x_8)) = l(S_{\min}^{ord}(x_1, x_8)) = 61.$$

Заметим, что каждое из обычных ребер $\{x_{11}, x_{13}\}$, $\{x_{13}, x_{14}\}$ и $\{x_{13}, x_{16}\}$ проходится дважды на разных участках полученного кратного пути, причем в противоположных направлениях, однако для кратных путей повторение обычных ребер допустимо.

References

- [1] A. V. Smirnov, "The shortest path problem for a multiple graph", *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 625–633, 2018. DOI: [10.3103/S0146411618070234](https://doi.org/10.3103/S0146411618070234).
- [2] A. V. Smirnov, "The polynomial algorithm of finding the shortest path in a divisible multiple graph", *Modeling and Analysis of Information Systems*, vol. 29, no. 4, pp. 372–387, 2022. DOI: [10.18255/1818-1015-2022-4-372-387](https://doi.org/10.18255/1818-1015-2022-4-372-387).
- [3] E. W. Dijkstra, "A note on two problems in connexion with graphs", *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959. DOI: [10.1007/BF01386390](https://doi.org/10.1007/BF01386390).
- [4] H. W. Kuhn, "The hungarian method for the assignment problem", *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. DOI: [10.1002/nav.3800020109](https://doi.org/10.1002/nav.3800020109).
- [5] J. Munkres, "Algorithms for the assignment and transportation problems", *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957. DOI: [10.1137/0105003](https://doi.org/10.1137/0105003).