Моделирование и анализ информационных систем. Т. 24, № 6 (2017), с. 788–801
Modeling and Analysis of Information Systems. Vol. 24, No 6 (2017), pp. 788–801

DOI: 10.18255/1818-1015-2017-6-788-801

УДК 519.17

Задача о кратчайшем пути в кратном графе

Смирнов А. В.[1]

получена 23 августа 2017

Аннотация.

В статье вводится определение неориентированного кратного графа 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 какому-либо кратному ребру, то она может быть инцидентна другим кратным ребрам, а также она может быть общим концом k связанных ребер какого-либо мультиребра. Если вершина является общим концом какого-либо мультиребра, то она не может быть общим концом никакого другого мультиребра.

Отдельно рассматривается класс делимых кратных графов, основной особенностью которых является возможность выделения k частей, согласованных на всех связанных ребрах и не содержащих общих ребер. Каждая из частей является обычным графом.

Для кратного графа обобщаются понятия степени вершины, связности графа, пути, цикла, веса ребра и длины пути.

Вводится понятие множества достижимости по обычным и по кратным ребрам, определяется свойство смежности двух множеств достижимости. Показано, что проверка связности кратного графа может быть выполнена за полиномиальное время с помощью алгоритма, основанного на поиске множеств достижимости и проверки их смежности.

Рассматривается критерий существования кратного пути между двумя вершинами и ставится задача о кратчайшем кратном пути. Строится алгоритм поиска кратчайшего пути в кратном графе, который использует алгоритм Дейкстры для поиска кратчайших путей в подграфах, соответствующих отдельным множествам достижимости.

Ключевые слова: кратный граф, делимый граф, множество достижимости, связность, кратный путь, кратчайший путь

Для цитирования: Смирнов А. В., "Задача о кратчайшем пути в кратном графе", *Моделирование и анализ информационных систем*, **24**:6 (2017), 788–801.

Об авторах:
Смирнов Александр Валерьевич, orcid.org/0000-0002-0980-2507, канд. физ.-мат. наук, доцент,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: alexander_sm@mail.ru

Благодарности:
[1] Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 17-07-00823 А).

Введение

В данной статье мы введем понятие *кратного графа* кратности $k > 1$ и рассмотрим задачу о кратчайшем пути в нем. Кратные графы содержат три типа ребер (обычные, кратные и мультиребра) и являются обобщением обычных графов – по сути, обычный граф имеет кратность $k = 1$.

Среди других известных обобщений графов стоит выделить мультиграфы, гиперграфы (см., например, [1] – [2]), а также метаграфы (см. [3] – [4]), как наиболее близкие нам концепции. Действительно, как и в мультиграфах, в кратных графах допускается наличие нескольких ребер между парой вершин (набор таких ребер мы будем в дальнейшем называть *кратным ребром*), однако в случае кратного графа количество таких ребер должно быть строго равным k. В кратных графах присутствуют *мультиребра*, соединяющие между собой $k + 1$ вершину. Но в отличие от гиперребер гиперграфа, мультиребро представляется в виде k связанных ребер, имеющих один общий конец, причем все эти k ребер должны использоваться согласованно. По сути, понятие мультиребра близко понятию ребра между вершиной и метавершиной в метаграфе. При этом в метаграфе, напомним, метапуть между двумя метавершинами фактически моделирует причинно-следственные связи в некоторой предметной области. Однако в кратном графе используется принципиально иной подход к определению пути: *кратный путь* должен состоять ровно из k обычных путей, проходящих по обычным ребрам, а также по связанным ребрам кратных и мультиребер; при этом пути должны быть согласованы (одинаковы) на кратных и мультиребрах. Поэтому кратный граф нельзя считать частным случаем метаграфа.

Ранее (см. [5] – [6]) мы разработали обобщение теории потоков Форда–Фалкерсона (см. [7]), названное кратными сетями и кратными потоками. В статьях [8] – [10] кратные сети и потоки используются для поиска решения задачи целочисленного сбалансирования трех- и четырехмерной матрицы. Задача целочисленного сбалансирования является частным случаем многоиндексных задач целочисленного линейного программирования (см., например, [11] – [13]), при этом она NP-полна (см. [14]). В работе [8] показано, что сведение задачи целочисленного сбалансирования трехмерной матрицы к поиску наибольшего кратного потока в сети кратности 2 в большинстве случаев оказывается более эффективным, нежели применение стандартных методов решения задач целочисленного линейного программирования. Заметим, что задача целочисленного сбалансирования имеет ряд приложений в сфере экономики, управления, финансов.

Следует отметить, что кратные сети являются частным случаем кратных графов, рассматриваемых в настоящей статье, а понятие кратного пути в кратном графе близко понятию обобщенного пути прорыва, на построении которого основаны алгоритмы поиска наибольшего потока в кратной сети.

1. Кратный граф

В [5] – [6] было введено понятие *кратной сети* и рассмотрена задача о наибольшем кратном потоке в такой сети. Так же, как обычная сеть является частным случаем обычного графа, кратная сеть является частным случаем кратного графа. Введем соответствующие определения.

Определение 1. *В качестве* кратного графа *произвольной натуральной кратности* $k > 1$ *рассматривается граф* $G(X, E)$, *между вершинами которого могут быть ребра 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 другими* $k-1$ *ребрами (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.

Определение 2. *Кратный граф является* ориентированным, *если хотя бы на одном обычном, кратном или мультиребре задано направление. При этом ориентация всех связанных ребер кратного или мультиребра совпадает.*

В данной работе мы ограничимся рассмотрением неориентированных кратных графов.

Определение 3. Петлей *в кратном графе является обычное или кратное ребро вида* $\{x, x\}$, *где* $x \in X$.

Отметим, что петля $\{x, x\}$ в кратном графе обязательно является кратным ребром, если вершина x инцидентна какому-либо кратному ребру или является общим концом мультиребра. В противном случае петля $\{x, x\}$ обязательно является обычным ребром.

Определение 4. *Для любой вершины* $x \in X$ *определена ее* степень $\deg x$ – *количество обычных или связанных ребер, инцидентных* x.

Таким образом, каждое обычное ребро вида $\{x, y\}$ добавляет 1 к $\deg x$, каждое кратное ребро вида $\{x, y\}$ добавляет k к $\deg x$, а каждое мультиребро вида $\{x, \{y_1, \ldots, y_k\}\}$ добавляет k к $\deg x$ и 1 к каждому из $\deg y_i$.

Определение 5. *Вершина является* висячей, *если она инцидентна только одному ребру.*

Следует отметить, что степень висячей вершины равна 1, если она инцидентна обычному ребру либо является концом одного связанного ребра какого-то мультиребра. Если же висячая вершина инцидентна кратному ребру либо является общим концом мультиребра, то ее степень будет равна k.

Теорема 1. $\sum\limits_{x\in X} \deg x = 2\left(|E^o| + k\left|E^k\right| + k\left|E^m\right|\right).$

Справедливость теоремы 1 следует непосредственно из определения степени вершины в кратном графе.

Следствие 1. *Количество вершин нечетной степени четно.*

Определение 6. Делимым кратным графом *назовем такой граф, для каждого мультиребра которого не существует пути из конца одного связанного ребра в конец другого связанного ребра того же мультиребра, проходящего только по обычным ребрам.*

Очевидно, что при удалении всех мультиребер делимый граф распадется на n компонент связности (связность здесь понимается в том же смысле, что и для обычных графов), каждая из которых содержит только кратные ребра либо только обычные ребра. При этом связанные ребра каждого мультиребра можно пронумеровать от 1 до k таким образом, что каждой компоненте связности, содержащей только обычные ребра, будут инцидентны связанные ребра мультиребер с одинаковыми номерами.

Определение 7. Частью G_i *($i \in \overline{1,k}$) делимого графа $G(X,E)$ назовем подграф, содержащий связанные ребра с номером i всех кратных и мультиребер, а также компоненты связности, состоящие из обычных ребер и инцидентные i-м связанным ребрам всех мультиребер.*

Очевидно, что каждая часть G_i является обычным графом. При этом возможность выделения частей G_i является особенностью делимых графов. В общем случае получить части G_i не удастся.

2. Кратный путь и задача о наименьшем кратном пути

Дадим теперь определение кратного пути. Основное отличие кратного пути от пути в обычном графе состоит в том, что связанные ребра каждого кратного и мультиребра должны проходиться в этом пути согласованно.

Определение 8. $S(x,y) = \cup_{i=1}^{k} S^i(x,y)$ *является* кратным путем из вершины x в вершину y *в графе $G(X,E)$, если выполнены следующие условия:*

1) $S^i(x,y) = \left(\{x, v_1^i\}, \{v_1^i, v_2^i\}, \ldots, \{v_{p-1}^i, v_p^i\}, \{v_p^i, y\}\right)$, где $p \geq 0$, – последовательность ребер, представляющая собой обычный (некратный) путь из x в y, где каждое ребро $\{a,b\}$ является либо обычным ребром в графе $G(X,E)$, либо i-м связанным ребром кратного или мультиребра. Если в путь $S(x,y)$ не входит ни одного кратного или мультиребра, то $S^2(x,y) = S^3(x,y) = \ldots = S^k(x,y) = \varnothing$;

2) вершины, не являющиеся общим концом связанных ребер мультиребра и не инцидентные кратным ребрам, могут встречаться в $S^i(x,y)$ несколько раз, то есть $S^i(x,y)$ может содержать циклы;

3) вершины, инцидентные кратным ребрам либо являющиеся общим концом мультиребра, не могут 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 $S^j(x, y)$). При этом порядок вхождения всех кратных и мультиребер во все $S^i(x, y)$ одинаков;

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 9. *Кратный путь $S(x, y)$ является* кратным циклом*, если $x = y$ и $S(x, y) \neq \varnothing$.*

Условие 6 в определении 8 фактически является требованием синхронности использования связанных ребер (при движении по кратному пути соответствующие связанные ребра проходятся параллельно). Такой подход к определению кратного пути является наиболее естественным. При этом условия 2 и 4 допускают наличие циклических участков в частях пути $S^i(x, y)$, совокупность которых, однако, не образует кратного цикла. Возможность добавления таких циклов в путь вполне оправдана даже для делимых кратных графов, что можно проиллюстрировать следующим примером.

Пример 1. Пусть кратность $k = 2$ и делимый граф имеет структуру, представленную на рис. 1 (жирными линиями обозначены кратные ребра, раздваивающимися линиями обозначены мультиребра).

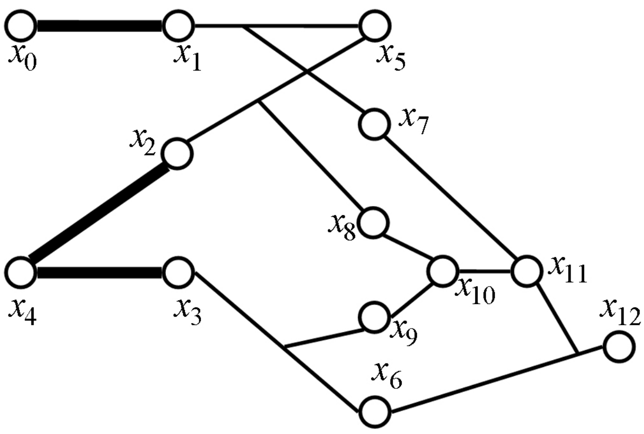

Рис. 1. Делимый граф кратности 2
Fig. 1. Divisible graph of multiplicity 2

Нетрудно убедиться, что путь $S(x_0, x_{12})$ в этом графе можно построить единственным образом (связанные ребра кратных ребер выделены однократным подчеркиванием, мультиребер – двойным подчеркиванием):

$$S^1(x_0, x_{12}) = \Big(\underline{\{x_0, x_1\}}, \underline{\underline{\{x_1, x_5\}}}, \underline{\underline{\{x_5, x_2\}}}, \underline{\{x_2, x_4\}}, \underline{\{x_4, x_3\}}, \underline{\underline{\{x_3, x_6\}}}, \underline{\underline{\{x_6, x_{12}\}}}\Big);$$

$$S^2(x_0, x_{12}) = \Big(\underline{\{x_0, x_1\}}, \underline{\underline{\{x_1, x_7\}}}, \{x_7, x_{11}\}, \{x_{11}, x_{10}\}, \{x_{10}, x_8\}, \underline{\underline{\{x_8, x_2\}}}, \underline{\{x_2, x_4\}}, \\ \underline{\{x_4, x_3\}}, \underline{\underline{\{x_3, x_9\}}}, \{x_9, x_{10}\}, \{x_{10}, x_{11}\}, \underline{\underline{\{x_{11}, x_{12}\}}}\Big).$$

При этом участок пути $(\{x_{10}, x_8\}, \{x_8, x_2\}, \{x_2, x_4\}, \{x_4, x_3\}, \{x_3, x_9\}, \{x_9, x_{10}\})$ в части $S^2(x_0, x_{12})$ является циклическим. Более того, ребро $\{x_{10}, x_{11}\}$ проходится в этой части дважды (в противоположных направлениях).

Отметим также, что совокупность условий 2–5 и 7 из определения 8 ограничивает наличие кратных циклов внутри пути $S(x, y)$: циклическим (в кратном смысле) участком может быть только участок пути, начинающийся и заканчивающийся в вершине z, инцидентной обычным ребрам. Если дополнительно потребовать минимальность пути, такой участок очевидно будет отброшен. Запрет всех вариантов кратного цикла внутри пути (за исключением отмеченного) также вполне оправдан, так как возможность добавления кратных циклов в путь существенно увеличит количество операций, необходимых для отыскания этого пути.

Кроме того, заметим, что для делимых графов условие 5 из определения 8 не нужно, так как каждый из путей $S^i(x, y)$ будет проходить по своей части графа G_i, при этом части G_i не имеют общих обычных ребер. По этой же причине в пути в делимом графе не может возникнуть никакого кратного цикла.

Теперь мы можем ввести понятие связности для кратного графа.

Определение 10. *Кратный граф $G(X, E)$ является* связным, *если одновременно выполнены два условия:*

1) для любых двух вершин $x \in X$, $y \in X$, каждая из которых либо инцидентна кратному ребру, либо является общим концом мультиребра, существует кратный путь $S(x, y)$;

2) невозможно выделить такой подграф $G' \subset G$, который будет содержать только обычные ребра, и при этом подграфы G' и $G \setminus G'$ не будут соединены ни одним ребром (обычным ребром или связанным ребром мультиребра).

В отличие от обычных графов связность кратного графа не предполагает наличие кратных путей из каждой вершины в каждую. Фактически в связном кратном графе между каждой парой вершин должен существовать обычный (некратный) путь, использующий связанные ребра кратных и мультиребер несогласованно, а кратные пути обязательно должны существовать только для пар вершин, каждая из которых инцидентна кратным ребрам или является общим концом мультиребра. Рассмотрим пример.

<u>Пример 2.</u> Пусть граф кратности $k = 2$ состоит из 4 вершин и 2 мультиребер (рис. 2).

Граф очевидно является связным, но при этом в нем не существует путей $S(x_1, x_2)$, $S(x_1, x_3)$, $S(x_4, x_2)$ и $S(x_4, x_3)$, поскольку их невозможно построить согласованно относительно связанных ребер.

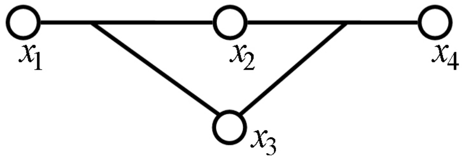

Рис. 2. Граф кратности 2
Fig. 2. Graph of multiplicity 2

Алгоритм проверки связности кратного графа будет рассмотрен в разделе 3.

Отметим, что для делимого кратного графа определение 10 может быть переписано в более простой форме, что обусловлено структурой графа.

Определение 11. *Делимый кратный граф $G(X, E)$ является* связным*, если одновременно выполнены два условия:*

1) для любых двух вершин $x \in X$, $y \in X$, каждая из которых либо инцидентна кратному ребру, либо является общим концом мультиребра, существует кратный путь $S(x, y)$;

2) каждая из частей G_i является связным графом.

Поставим задачу о наименьшем кратном пути. Сначала определим длину ребра.

Определение 12. *Целочисленная функция $l(e)$, определенная для всех ребер $e \in E$, является* длиной (весом) ребра *в кратном графе $G(X, E)$, если выполнено следующее:*

1) $l(e) > 0$ для любого ребра e;

2) если e является кратным или мультиребром, то $l(e_1) = l(e_2) = \ldots = l(e_k)$ и $l(e) = k \cdot l(e_1)$, где $e_1, \ldots, e_k$ – это связанные ребра данного ребра e.

Тогда *длина 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. *В кратном графе $G(X, E)$ 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*

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

Условия существования кратного пути и алгоритм решения задачи 1 будут рассмотрены в разделе 4.

3. Множества достижимости и связность кратного графа

В данном разделе мы получим полиномиальный алгоритм для проверки связности кратного графа.

Определение 13. Множеством достижимости по кратным ребрам *для некоторой вершины x назовем множество R^k_x всех вершин y таких, что существует путь из x в y, проходящий только по кратным ребрам.*

Определение 14. Множеством достижимости по обычным ребрам *для некоторой вершины x назовем множество R^o_x всех вершин y таких, что существует путь из x в y, проходящий только по обычным ребрам.*

Очевидно, что $x \in R^k_x$, $x \in R^o_x$. Если $y \in R^k_x$, то $R^k_y = R^k_x$. Если $y \in R^o_x$, то $R^o_y = R^o_x$.

Мы будем определять множества R^k_x только для вершин x, инцидентных кратным ребрам либо являющихся общим концом мультиребра, а множества R^o_x мы будем определять только для вершин x, инцидентных обычным ребрам либо являющихся концом одного связанного ребра какого-то мультиребра.

Определение 15. *Множества достижимости R^k_x и R^k_y являются* смежными, *если для произвольных вершин $a \in R^k_x$, $b \in R^k_y$ существует соединяющий их кратный путь $S(a, b)$.*

Очевидно, что отношение смежности является *рефлексивным*, *симметричным* и *транзитивным*.

Мы будем использовать множества достижимости и свойство смежности этих множеств для проверки связности кратного графа. Нетрудно убедиться, что будет справедлив следующий критерий связности.

Теорема 2. *Кратный граф является связным тогда и только тогда, когда все R^k_x являются попарно смежными и в каждом из R^o_y найдется вершина, инцидентная одному связанному ребру какого-либо мультиребра.*

Алгоритм 1 (*построение множества достижимости по кратным ребрам для вершины x*).

1. Все вершины графа являются непросмотренными. $R^k_x = \{x\}$.

2. Берем произвольную непросмотренную вершину $z \in R^k_x$. Если таких вершин нет, множество R^k_x сформировано, выход.

3. Для каждого ребра $\{z, y\} \in E^k$ включаем вершину y в R^k_x, если она является непросмотренной.

4. Вершина z становится просмотренной. Переходим на шаг 2.

Алгоритм 2 (*построение множества достижимости по обычным ребрам для вершины x*).

Аналогично алгоритму 1, но рассматриваем ребра из E^o.

Алгоритм 3 (*поиск всех пар смежных множеств достижимости по кратным ребрам*).

1. Выбираем из множества E^m пару мультиребер $\{a, \{a_1, \ldots, a_k\}\}$ и $\{b, \{b_1, \ldots, b_k\}\}$.

2. Если для всех $i \in \overline{1,k}$ выполнено $a_i \in R^o_{b_i}$, то включаем $\{R^k_a, R^k_b\}$ в список пар смежных множеств достижимости. Если остались непросмотренные пары мультиребер, переходим на шаг 1.

3. Пользуясь свойством транзитивности отношения смежности, дополняем список пар смежных множеств достижимости, пока это возможно.

Алгоритм 4 (*проверка связности кратного графа*).

1. С помощью алгоритма 2 ищем все R^o_y. Если хотя бы одно из R^o_y не содержит вершины, инцидентной связанному ребру какого-то мультиребра, то граф несвязен, выход.

2. С помощью алгоритма 1 ищем все R^k_x. После этого применяем алгоритм 3 для определения пар смежных множеств достижимости. Если хотя бы одна пара $\{R^k_a, R^k_b\}$ не вошла в этот список, то граф несвязен, выход.

3. Граф связен.

Отметим, что все приведенные в данном разделе алгоритмы выполняются за полиномиальное время.

4. Алгоритм поиска кратчайшего пути в кратном графе

Прежде чем сформулировать алгоритм поиска минимального кратного пути, определим условия, при которых этот путь существует.

Теорема 3. *В кратном графе $G(X, E)$ существует путь из вершины $x \in X$ в вершину $y \in X$ тогда и только тогда, когда выполнено одно из условий:*

1) $x \in R^o_y$;

2) $x \in R^k_y$;

3) каждая из вершин x и y инцидентна кратным ребрам либо является общим концом связанных ребер мультиребра, и при этом $R^k_x \neq R^k_y$, но R^k_x и R^k_y смежны;

4) x и y не инцидентны кратным ребрам и не являются общим концом связанных ребер мультиребра, при этом $R^o_x \neq R^o_y$, но в E^m есть два мультиребра $\{a, \{a_1, \ldots, a_k\}\}$ и $\{b, \{b_1, \ldots, b_k\}\}$ такие, что для всех $i \in \overline{1,k}$ выполнено $a_i \in R^o_x$ и $b_i \in R^o_y$, а R^k_a и R^k_b смежны;

5) x инцидентен кратным ребрам либо является общим концом связанных ребер мультиребра, а y нет, но при этом в E^m есть мультиребро $\{a, \{a_1, \ldots, a_k\}\}$ такое, что для всех $i \in \overline{1,k}$ выполнено $a_i \in R^o_y$, а R^k_a и R^k_x смежны;

6) y инцидентен кратным ребрам либо является общим концом связанных ребер мультиребра, а x нет, но при этом в E^m есть мультиребро $\{a, \{a_1, \ldots, a_k\}\}$ такое, что для всех $i \in \overline{1,k}$ выполнено $a_i \in R^o_x$, а R^k_a и R^k_y смежны.

Справедливость данной теоремы вытекает непосредственно из свойств множеств достижимости и отношения смежности.

Следует отметить, что для любых двух вершин кратного графа выполнение условия теоремы можно проверить за полиномиальное время, используя алгоритмы 1–3 из предыдущего раздела. При этом для делимых кратных графов достаточно проверять только первые три условия, поскольку структура таких графов делает невозможным выполнение оставшихся условий.

Сформулируем теперь алгоритм решения задачи 1. На отдельных шагах этого алгоритма для нахождения минимальных участков пути без мультиребер мы будем использовать известный алгоритм Дейкстры (см. [15]).

В алгоритме мы будем использовать следующие структуры данных:

– множества достижимости R_z^k и R_z^o;

– минимальный на текущий момент путь S_0 длины $L_{\min}$;

– множество путей-кандидатов C (в это множество будут добавляться строящиеся пути S_i, которые могут в итоге оказаться минимальными; в конце выполнения алгоритма это множество окажется пустым);

– M_i – множество запрещенных вершин для строящегося пути S_i (согласно определению кратного пути вершины, инцидентные кратным ребрам либо являющиеся общим концом мультиребра, не могут использоваться дважды).

Алгоритм 5 (*поиск минимального кратного пути из x в y*).

0. С помощью алгоритмов 1–3 проверяем существование пути из x в y. Если условие выполнено, устанавливаем $L_{\min} = \infty$ и переходим на шаг 1, иначе выход.

1. Если $x \in R_y^o$, то с помощью алгоритма Дейкстры находим кратчайший путь $S(x, y)$, проходящий только по вершинам из множества R_y^o (этот путь будет содержать только обычные ребра).

Запоминаем $S(x, y)$ как путь S_0 и устанавливаем $L_{\min} = l(S(x, y))$.

Если при этом граф является делимым, найденный путь S_0 является минимальным, выход.

2. Если $x \in R_y^k$, то с помощью алгоритма Дейкстры находим кратчайший путь $S(x, y)$, проходящий только по вершинам из множества R_y^k (этот путь будет содержать только кратные ребра).

Запоминаем $S(x, y)$ как путь S_0 и устанавливаем $L_{\min} = l(S(x, y))$.

3. Если x инцидентен кратным ребрам или является общим концом мультиребра, переходим на шаг 4.

Иначе для каждого мультиребра вида $e_b^m = \{\{b_1, \ldots, b_k\}, b\}$, где все $b_p \in R_x^o$ $(p \in \overline{1, k})$, находим с помощью алгоритма Дейкстры кратчайшие пути $S(x, b_p)$, проходящие только по вершинам из множества R_x^o. Если при этом

$$\sum_{p=1}^{k} l(S(x, b_p)) + l(e_b^m) < L_{\min},$$

то участок кратного пути $S_i = S(x, b_1) \cup \ldots \cup S(x, b_p) \cup e_b^m$ добавляем в множество путей кандидатов C (i – это некоторый уникальный в пределах множества C идентификатор).

Связываем с S_i множество запрещенных вершин $M_i = \varnothing$.

Переходим на шаг 5.

4. Для каждой вершины $a \in R_x^k$, являющейся началом какого-либо мультиребра, находим с помощью алгоритма Дейкстры кратчайший путь $S(x, a)$, проходящий только по вершинам из множества R_x^k (если $a = x$, этот путь будет иметь нулевую длину). Если

$$l(S(x, a)) + l(e_a^m) + k \le L_{\min},$$

где e_a^m – мультиребро с началом в a, то участок кратного пути $S_i = S(x, a) \cup e_a^m$

добавляем в множество путей-кандидатов C (i – уникальный в пределах множества C идентификатор).

Связываем с S_i множество запрещенных вершин M_i, содержащее все вершины пути $S(x,a)$.

5. Если множество C пусто, то запомненный путь S_0 длины $L_{\min}$ является минимальным, выход.

6. Извлекаем произвольный участок пути S_i из множества C. Если последняя вершина S_i инцидентна кратному ребру или является общим концом мультиребра, переходим на шаг 7.

Иначе переходим на шаг 9.

7. Пусть a – последняя вершина пути S_i. Рассмотрим множество $R^k_a \setminus M_i$.

Если $y \in R^k_a \setminus M_i$, то с помощью алгоритма Дейкстры ищем кратчайший путь $S(a,y)$, проходящий только по вершинам из множества $R^k_a \setminus M_i$. Если такой путь существует и

$$l(S_i) + l(S(a,y)) < L_{\min},$$

то запоминаем кратный путь $S_i \cup S(a,y)$ как S_0 и устанавливаем $L_{\min} = l(S_i) + l(S(a,y))$.

8. Рассмотрим все вершины $b \in R^k_a \setminus M_i$, каждая из которых является началом мультиребра $e^m_b = \{b, \{b_1, \dots, b_k\}\}$, если при этом в путь S_i ранее не было включено мультиребро $\{c, \{b_1, \dots, b_k\}\}$, проходимое в том же направлении (от общего конца). Будем с помощью алгоритма Дейкстры искать кратчайший путь $S(a,b)$, проходящий только по вершинам из множества $R^k_a \setminus M_i$. Если такой путь существует и

$$l(S_i) + l(S(a,b)) + l(e^m_b) + k \le L_{\min},$$

то добавляем в C участок кратного пути $S_j = S_i \cup S(a,b) \cup e^m_b$ (j – уникальный в пределах множества C идентификатор).

Множество M_j состоит из всех вершин множества M_i и всех вершин пути $S(a,b)$.

Переходим на шаг 5.

9. Последними вершинами S_i являются вершины $\{a_1, \dots, a_k\}$ – k концов некоторого мультиребра.

Если все $a_p \in R^o_y$ ($p \in \overline{1,k}$) (для делимого графа это условие заведомо не будет выполнено, так как в делимом графе $R^o_{a_q} \cap R^o_{a_s} = \varnothing$, если $q \ne s$), то с помощью алгоритма Дейкстры находим кратчайшие пути $S(a_p, y)$, проходящие только по вершинам из множества R^o_y. Если

$$l(S_i) + \sum_{p=1}^{k} l(S(a_p, y)) < L_{\min},$$

то запоминаем получившийся кратный путь $S_i \cup S(a_1, y) \cup \ldots \cup S(a_k, y)$ как S_0 и устанавливаем $L_{\min} = l(S_i) + \sum\limits_{p=1}^{k} l(S(a_p, y))$.

10. Рассмотрим все мультиребра вида $e^m_b = \{\{b_1, \dots, b_k\}, b\}$ такие, что $b_p \in R^o_{a_p}$ ($p \in \overline{1,k}$) и в путь S_i не было ранее включено мультиребро $\{\{b_1, \dots, b_k\}, c\}$, проходимое в том же направлении (к общему концу). При этом мультиребра, отличающиеся только порядком следования вершин b_p в перечислении концов, мы считаем одинаковыми и повторно не рассматриваем (эта ситуация возникает, когда

граф не является делимым и $R^o_{b_q} = R^o_{b_s}$, $q \neq s$; тогда одно и то же мультиребро e^m_b можно рассматривать как в виде $\{\{b_1, \ldots, b_q, \ldots, b_s, \ldots, b_k\}, b\}$, так и в виде $\{\{b_1, \ldots, b_s, \ldots, b_q, \ldots, b_k\}, b\}$).

Для каждого мультиребра e^m_b указанного вида найдем с помощью алгоритма Дейкстры кратчайшие пути $S(a_p, b_p)$, проходящие только по вершинам соответствующих множеств $R^o_{a_p}$ $(p \in \overline{1,k})$. Если

$$l(S_i) + \sum_{p=1}^{k} l(S(a_p, b_p)) + l(e^m_b) < L_{\min},$$

то добавляем в C участок кратного пути $S_j = S_i \cup S(a_1, b_1) \cup \ldots \cup S(a_k, b_k) \cup e^m_b$ (j – уникальный в пределах множества C идентификатор).

Устанавливаем $M_j = M_i$.

Переходим на шаг 5.

Отметим, что на шагах 4 и 8 алгоритма 4 строится участок пути, оканчивающийся k различными вершинами-концами мультиребра. Поэтому для достижения вершины y нам заведомо потребуется увеличить длину этого пути не менее чем на $k-1$ для произвольного кратного графа (если y – один из концов мультиребра, а остальные $k-1$ концов мультиребра соединены с y обычными ребрами длины 1) и не менее чем на k для делимого графа (мультиребро минимальной длины). Соответственно, при проверке условия на шагах 4, 8 появляется слагаемое k. Для делимого графа указанное условие можно представить как строгое неравенство.

Каждый из шагов алгоритма 4 полиномиален, однако шаги могут повторяться экспоненциальное количество раз. В худшем случае алгоритм переберет все возможные варианты пути из x в y. Однако следует отметить, что задача 1 скорее всего является NP-полной. Это можно предположить, исходя из того, что NP-полной является задача о нахождении максимального потока в кратной сети (см. [5] – [6]), решение которой получается с помощью последовательного нахождения и насыщения обобщенных путей прорыва, определение которых близко определению кратного пути.

Количество операций в алгоритме 4 можно существенно сократить, если изначально задать какой-то путь S_0 между x и y. Также при обновлении S_0 и значения $L_{\min}$ можно просматривать множество C и исключать оттуда пути S_i, длина которых стала больше $L_{\min}$, если последняя вершина S_i является общим концом мультиребра, или больше $L_{\min} - k + 1$ в противном случае.

Заключение

В данной статье мы ввели понятие кратного графа, который содержит обычные, кратные и мультиребра. При этом для кратных графов мы обобщили ряд понятий, используемых для обычных графов (степень вершины, связность, путь), поставили задачу о минимальном кратном пути между двумя вершинами.

Также мы построили полиномиальный алгоритм проверки связности кратного графа и экспоненциальный алгоритм поиска наименьшего кратного пути, который является обобщением алгоритма Дейкстры.

Список литературы / References

[1] Cormen T. H., Leiserson C. E., Rivest R. L., Stein C., *Introduction to Algorithms*, 3rd ed., The MIT Press, McGraw-Hill Book Company, 2009.

[2] Berge C., *Graphs and Hypergraphs*, North-Holland Publishing Company, 1973.

[3] Basu A., Blanning R. W., "Metagraphs in workflow support systems", *Decision Support Systems*, **25**:3 (1999), 199–208.

[4] Basu A., Blanning R. W., *Metagraphs and Their Applications*, Integrated Series in Information Systems, **15**, Springer US, 2007.

[5] Рублев В. С., Смирнов А. В., "Потоки в кратных сетях", *Ярославский педагогический вестник*, **3**:2 (2011), 60–68; [Rublev V. S., Smirnov A. V., "Flows in Multiple Networks", *Yaroslavsky Pedagogichesky Vestnik*, **3**:2 (2011), 60–68, (in Russian).]

[6] Smirnov A. V., "The Problem of Finding the Maximum Multiple Flow in the Divisible Network and its Special Cases", *Automatic Control and Computer Sciences*, **50**:7 (2016), 527–535.

[7] Ford L. R., Fulkerson D. R., *Flows in Networks*, Princeton University Press, 1962.

[8] Рублев В. С., Смирнов А. В., "Задача целочисленного сбалансирования трехмерной матрицы и алгоритмы ее решения", *Моделирование и анализ информационных систем*, **17**:2 (2010), 72–98; [Roublev V. S., Smirnov A. V., "The Problem of Integer-Valued Balancing of a Three-Dimensional Matrix and Algorithms of Its Solution", *Modeling and Analysis of Information Systems*, **17**:2 (2010), 72–98, (in Russian).]

[9] Smirnov A. V., "Some Solvability Classes for the Problem of Integer Balancing of a Three-Dimensional Matrix with Constraints of the Second Type", *Automatic Control and Computer Sciences*, **48**:7 (2014), 543–553.

[10] Смирнов А. В., "Сетевая модель для задачи целочисленного сбалансирования четырехмерной матрицы", *Моделирование и анализ информационных систем*, **23**:4 (2016), 466–478; [Smirnov A. V., "Network Model for The Problem of Integer Balancing of a Four-dimensional Matrix", *Modeling and Analysis of Information Systems*, **23**:4 (2016), 466–478, (in Russian).]

[11] Корбут А. А., Финкельштейн Ю. Ю., *Дискретное программирование*, Наука, М., 1969; [Korbut A. A., Finkelstein J. J., *Diskretnoe programmirovanie*, Nauka, Moscow, 1969, (in Russian).]

[12] Раскин Л. Г., Кириченко И. О., *Многоиндексные задачи линейного программирования*, Радио и связь, М., 1982; [Raskin L. G., Kirichenko I. O., *Mnogoindeksnye zadachi lineynogo programmirovaniya*, Radio i svyaz, Moscow, 1982, (in Russian).]

[13] Spieksma F. C. R., "Multi index assignment problems: complexity, approximation, applications", *Nonlinear Assignment Problems. Algorithms and Applications*, eds. P. M. Pardalos, L. S. Pitsoulis, Kluwer Academic Publishers, 2000, 1–11.

[14] Рублев В. С., Смирнов А. В., "NP-полнота задачи целочисленного сбалансирования трехмерной матрицы", *Доклады Академии Наук*, **435**:3 (2010), 314–316; English transl.: Roublev V. S., Smirnov A. V., "NP-Completeness of the Integer Balancing Problem for a Three-Dimensional Matrix", *Doklady Mathematics*, **82**:3 (2010), 912–914.

[15] Dijkstra E. W., "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, **1**:1 (1959), 269–271.

Smirnov A. V., "The Shortest Path Problem for a Multiple Graph", *Modeling and Analysis of Information Systems*, **24**:6 (2017), 788–801.

DOI: 10.18255/1818-1015-2017-6-788-801

Abstract. In the article, the definition of an undirected multiple graph of any natural multiplicity $k > 1$ is stated. There are edges of three types: ordinary edges, multiple edges and multi-edges. Each edge of the last two types is the 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 ending vertex to 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 any other multi-edge. Also, a class of the divisible multiple graphs is considered. The main peculiarity of them is a possibility to divide the graph into k parts, which are adjusted on the linked edges and which have no common edges. Each part is an ordinary graph. The following terms are generalized: the degree of a vertex, the connectedness of a graph, the path, the cycle, the weight of an edge, and the path length. There is stated the definition of the reachability set for the ordinary and multiple edges. The adjacency property is defined for a pair of reachability sets. It is shown, that we can check the connectedness of some multiple graph with the polynomial algorithm based on the search for the reachability sets and testing their adjacency. There is considered a criterion of the existence of a multiple path between two given vertices. The shortest multiple path problem is stated. Then we suggest an algorithm of finding the shortest path in a multiple graph. It uses Dijkstra's algorithm of finding the shortest paths in subgraphs, which correspond to different reachability sets.

Keywords: multiple graph, divisible graph, reachability set, connectedness, multiple path, shortest path

On the authors:
Alexander V. Smirnov, orcid.org/0000-0002-0980-2507, PhD, Associate Professor
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl, 150003 Russia, e-mail: alexander_sm@mail.ru

Acknowledgments:
[1] This work was supported by the Russian Foundation for Basic Research under the Grant No 17-07-00823 A.