ISSN 1561-2430 (Print)
ISSN 2524-2415 (Online)
УДК 519.1
https://doi.org/10.29235/1561-2430-2022-58-2-155-168

Поступила в редакцию 25.11.2021
Received 25.11.2021

О. И. Дугинов, С. С. Химич

Белорусский государственный университет, Минск, Республика Беларусь

РАЗБИЕНИЕ РЕБЕР ДВУДОЛЬНОГО ГРАФА НА НАИМЕНЬШЕЕ ЧИСЛО ПОДГРАФОВ, ИЗОМОРФНЫХ ПОДГРАФАМ ПРОСТОГО ЦИКЛА ПОРЯДКА 4

Аннотация. Изучается вычислительная сложность задачи разбиения ребер двудольного графа на наименьшее число подграфов, изоморфных подграфам простого цикла порядка 4, в специальных классах графов. Задача относится к числу NP-трудных и находит применение при организации распределения сетевых пакетов по каналам связи в процессе передачи от одного маршрутизатора к другому. Разработан алгоритм, решающий задачу в классе деревьев порядка n за время $O(n \log n)$. Выделены трудноразрешимые случаи задачи.

Ключевые слова: разбиение множества ребер графа, деревья, подграфы простого цикла порядка 4

Для цитирования. Дугинов, О. И. Разбиение ребер двудольного графа на наименьшее число подграфов, изоморфных подграфам простого цикла порядка 4 / О. И. Дугинов, С. С. Химич // Вес. Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. – 2022. – Т. 58, № 2. – С. 155–168. https://doi.org/10.29235/1561-2430-2022-58-2-155-168

Oleg I. Duginov, Semyon S. Khimich

Belarusian State University, Minsk, Republic of Belarus

PARTITIONING THE EDGE SET OF A BIPARTITE GRAPH INTO THE MINIMAL NUMBER OF SUBGRAPHS ISOMORPHIC TO THOSE OF A SIMPLE 4 ORDER CYCLE

Abstract. In this paper, we study the computational complexity for a problem of partitioning the edge set of a bipartite graph into the minimal number of subgraphs isomorphic to those of a simple cycle of order 4 in special graph classes. This problem is NP-hard and finds application in organizing the distribution of network packets over communication channels in the process of transmission from one router to another. We develop an $O(n\log n)$ algorithm for solving that problem in a class of n order trees. Intractable cases of the problem are identified.

Keywords: partitioning the edge set of a graph, trees, subgraphs of a simple cycle of order 4

For citation. Duginov O. I., Khimich S. S. Partitioning the edge set of a bipartite graph into the minimal number of subgraphs isomorphic to those of a simple 4 order cycle. *Vestsi Natsyianal'nai akademii navuk Belarusi. Seryia fizika-matematychnykh navuk = Proceedings of the National Academy of Sciences of Belarus. Physics and Mathematics series*, 2022, vol. 58, no. 2, pp. 155–168 (in Russian). https://doi.org/10.29235/1561-2430-2022-58-2-155-168

Введение. В работе рассматриваются только конечные неориентированные графы $G = (V, E)$ без кратных ребер и петель со множеством вершин $V = V(G)$ и множеством ребер $E = E(G)$. Пусть S – это подмножество множества вершин графа G. Множество вершин графа G, смежных хотя бы с одной вершиной из множества S, обозначается через $N_G(S)$ или $N(S)$, если из контекста ясно о каком графе G идет речь. Подграф графа G, порожденный множеством S, обозначается через $G(S)$. Все неопределяемые в работе стандартные понятия и обозначения теории графов могут быть найдены в [1].

Пусть F – набор графов и S – семейство реберно непересекающихся подграфов графа G. Если каждый граф из семейства S изоморфен некоторому графу из набора F, то семейство S называется *F-упаковкой* в графе G; если к тому же каждое ребро графа G содержится в некотором графе, принадлежащем семейству S, то семейство S называется *F-разбиением ребер* графа G.

В настоящей работе рассматривается задача, которая в виде задачи распознавания формулируется следующим образом:

(C_4, P_4, P_3, $2P_2$, P_2)-РАЗБИЕНИЕ РЕБЕР ДВУДОЛЬНОГО ГРАФА.

Условие: заданы двудольный граф G и натуральное число k.

Вопрос: существует ли (C_4, P_4, P_3, $2P_2$, P_2)-разбиение S ребер графа G такое, что $|S| \leq k$?

В оптимизационной версии этой задачи требуется найти (C_4, P_4, P_3, $2P_2$, P_2)-разбиение ребер заданного графа G такое, что мощность разбиения минимальна. Известно, что рассматриваемая задача NP-полна [2]. В работе [2] разработана серия приближенных алгоритмов с гарантированными оценками точности для решения оптимизационной версии задачи (C_4, P_4, P_3, $2P_2$, P_2)-РАЗБИЕНИЕ РЕБЕР ДВУДОЛЬНОГО ГРАФА.

Отметим, что рассматриваемая задача является частным случаем задачи F-РАЗБИЕНИЕ РЕБЕР, в которой для фиксированного набора F графов и заданного натурального числа k требуется ответить на следующий вопрос: существует ли F-разбиение S ребер заданного графа такое, что $|S| \leq k$?

Основная цель настоящей работы – исследование вычислительной сложности задачи (C_4, P_4, P_3, $2P_2$, P_2)-РАЗБИЕНИЕ РЕБЕР ДВУДОЛЬНОГО ГРАФА в специальных подклассах двудольных графов. Мотивацией исследования служит слабая изученность рассматриваемой задачи и актуальность ее приложений при организации распределения сетевых пакетов по каналам связи волоконно-оптического кабеля [2].

Вычислительная сложность задачи для некоторых классов графов, определяемых в терминах циклических свойств графов. Установим сложностной статус рассматриваемой задачи для двудольных графов с обхватом, ограниченным снизу произвольной положительной константой.

Т е о р е м а 1. *Для любого фиксированного числа* $\ell \geq 2$ *задача* (C_4, P_4, P_3, $2P_2$, P_2)-РАЗБИЕНИЕ РЕБЕР ДВУДОЛЬНОГО ГРАФА *в классе* (C_4, C_6, …, $C_{2\ell}$)*-свободных двудольных графов является NP-полной.*

Д о к а з а т е л ь с т в о. Задача принадлежит классу NP. В качестве эталонной задачи выберем задачу P_4-РАЗБИЕНИЕ РЕБЕР, NP-полнота которой следует из доказанной Д. Дор и М. Тарси [3] гипотезы И. Холиера [4] о том, что задача H-РАЗБИЕНИЕ РЕБЕР для любого графа H, в котором есть связная компонента с не менее чем тремя ребрами, является NP-полной.

Пусть $G = (V, E)$ – произвольный граф. Заменим каждое ребро $\{u, v\}$ графа G на простую цепь порядка пять с множеством вершин $\{u, x_{uv}, y_{uv}, z_{uv}, v\}$ и множеством ребер $\{\{u,x_{uv}\},\{x_{uv},y_{uv}\},\{y_{uv},z_{uv}\},\{z_{uv},v\}\}$. Заметим, что полученный граф является двудольным. Будем применять операцию замены всех ребер в графе на простые цепи порядка пять ℓ раз и получим двудольный граф H, в котором нет циклов длины меньше $2\ell + 1$. Тогда $|E(H)| = 4^{\ell}|E(G)|$ и общее число замен ребер в ходе преобразования графа G в граф H составит $|E(G)| + 4|E(G)| + 4^2|E(G)| + \ldots + 4^{\ell-1}|E(G)| = (4^{\ell} - 1)\,|E(G)|\,/\,3$.

У т в е р ж д е н и е. *Существует* P_4*-разбиение* S *ребер графа* G *такое, что* $|S| \leq k$ *тогда и только тогда, когда существует* P_4*-разбиение* S' *графа* H *такое, что* $|S'| \leq k + (4^{\ell} - 1)\,|E(G)|\,/\,3$.

Д о к а з а т е л ь с т в о. Пусть G_1 – это граф, который получается из графа G заменой одного ребра $\{u, v\}$ на простую цепь порядка пять. Для того, чтобы доказать утверждение, достаточно показать, что наличие в графе G P_4-разбиения ребер мощности не больше, чем k, равносильно существованию в графе H P_4-разбиения ребер мощности не больше, чем $k + 1$.

Пусть S – это P_4-разбиение ребер графа G такое, что $|S| \leq k$. Тогда существует цепь P из разбиения S, которая содержит ребро $\{u, v\}$. Возможны три случая.

С л у ч а й 1. Пусть v – это концевая вершина простой цепи P. Преобразуем P_4-разбиение S ребер графа G в P_4-разбиение S' графа G_1. В цепи P множества S заменим вершину v на вершину x_{uv}. Добавим в множество S цепь с вершинами x_{uv}, y_{uv}, z_{uv}, v. Получившееся в результате множество S' является P_4-разбиением ребер графа G_1, при этом $|S'| = |S| + 1$.

С л у ч а й 2. Пусть u – это концевая вершина простой цепи P. Аналогичным образом преобразуем множество S в множество S'. В цепи P множества S заменим вершину u на вершину z_{uv}. Добавим в множество S цепь с вершинами u, x_{uv}, y_{uv}, z_{uv}. Получившееся в результате множество S' является P_4-разбиением ребер графа G_1, при этом $|S'| = |S| + 1$.

С л у ч а й 3. Пусть концевые вершины ℓ, r цепи P отличны от вершин u, v. Удалим из S цепь P и добавим в S две цепи – цепь с вершинами ℓ, u, x_{uv}, y_{uv}, а также цепь с вершинами y_{uv}, z_{uv},

v, r. Получившееся в результате множество S' является P_4-разбиением ребер графа G_1, при этом $|S'| = |S| + 1$.

В каждом случае получили P_4-разбиение S' ребер графа G_1 такое, что $|S'| = |S| + 1 \leq k + 1$.

Пусть S' – это P_4-разбиение ребер графа G_1 такое, что $|S'| \leq k + 1$. Пусть P – это цепь, которая принадлежит S' и содержит ребро $\{y_{uv}, z_{uv}\}$. Рассмотрим три случая.

С л у ч а й А. Пусть y_{uv} – это концевая вершина цепи P. Обозначим вторую концевую вершину цепи P через w. Рассмотрим цепь Q, которая принадлежит S' и содержит ребро $\{x_{uv}, y_{uv}\}$. Одна концевая вершина цепи Q – это вершина y_{uv}. Другую концевую вершину цепи Q обозначим через w'. Удалим из множества S' цепи P и Q, а также добавим в S' цепь с вершинами w', u, v, w. В результате получим множество S, являющееся P_4-разбиением ребер графа G, при этом $|S| = |S'| - 1$.

С л у ч а й Б. Пусть z_{uv} – концевая вершина цепи P. Тогда вторая концевая вершина цепи P – это вершина u. Пусть Q – это цепь, которая принадлежит S' и содержит ребро $\{z_{uv}, v\}$. Удалим из множества S' цепь P, а также в цепи Q множества S' заменим вершину z_{uv} на вершину u. В результате получим множество S, которое является P_4-разбиением ребер графа G такое, что $|S| = |S'| - 1$.

С л у ч а й В. Пусть вершины y_{uv} и z_{uv} не являются концевыми вершинами цепи P. Тогда концевые вершины цепи P – это вершины x_{uv} и v. Пусть Q – это цепь, которая принадлежит множеству S' и содержит ребро $\{x_{uv}, u\}$. Удалим из S' цепь P, а также в цепи Q множества S' заменим вершину x_{uv} на вершину v. В результате получим P_4-разбиение S ребер графа G, при этом $|S| = |S'| - 1$.

В каждом случае получили P_4-разбиение S ребер графа G такое, что $|S| = |S'| - 1 \leq k$. Тем самым утверждение доказано.

Доказательство теоремы 1 завершает очевидный факт: существование P_4-разбиения ребер графа H равносильно наличию в графе H $(C_4, P_4, P_3, 2P_2, P_2)$-разбиения ребер мощности не больше $|E(H)|/3$.

Пусть Г – это класс двудольных графов, в которых каждая компонента связности является эйлеровым графом.

Т е о р е м а 2. *Задача* $(C_4, P_4, P_3, 2P_2, P_2)$-РАЗБИЕНИЕ РЕБЕР ДВУДОЛЬНОГО ГРАФА *в классе графов Г является NP-полной, а в классе графов* ($Г \cap C_4$-*свободные графы*) *решается за полиномиальное время.*

Д о к а з а т е л ь с т в о. Докажем первую часть утверждения. Задача принадлежит классу NP. Приведем полиномиальное сведение задачи $(C_4, P_4, P_3, 2P_2, P_2)$-РАЗБИЕНИЕ РЕБЕР ДВУДОЛЬНОГО ГРАФА к этой же задаче, ограниченной классом графов Г. Рассмотрим произвольный двудольный граф G. По графу G построим граф H, принадлежащий классу графов Г. Если граф G принадлежит Г, то в качестве графа H возьмем граф G. Пусть граф G не принадлежит классу графов Г. Стало быть, в графе G есть вершины нечетной степени, и поэтому нет C_4-разбиения ребер. Как следствие этого, нет $(C_4, P_4, P_3, 2P_2, P_2)$-разбиения ребер графа G мощности не больше $|E(G)|/4$. В этом случае возьмем любой граф H из класса графов Г, для которого нет $(C_4, P_4, P_3, 2P_2, P_2)$-разбиения ребер мощности не больше $|E(H)|/4$. Итак, в графе G существует $(C_4, P_4, P_3, 2P_2, P_2)$-разбиение ребер мощности не больше $|E(G)|/4$ тогда и только тогда, когда в графе H существует $(C_4, P_4, P_3, 2P_2, P_2)$-разбиение ребер мощности не больше $|E(H)|/4$.

Докажем вторую часть утверждения. Пусть G – произвольный C_4-свободный граф, принадлежащий классу Г. Пусть $C^1, C^2, \ldots, C^k$ – это эйлеровы циклы связных компонент графа G. В каждом цикле C^i выделим $\lfloor |E(C^i)|/3 \rfloor$ реберно непересекающихся простых цепей порядка 4, а также одну цепь, порядок которой равен остатку от деления $|E(C)|$ на 3, если число ребер в цикле C^i не делится на 3. Пусть S – множество выделенных цепей в циклах $C^1, C^2, \ldots, C^k$. Если в S имеется несколько подграфов, изоморфных P_2, то разобьем их на пары и образуем из них подграфы, изоморфные $2P_2$. При этом возможно, что для одного из подграфов, изоморфных P_2, не найдется пары. Нетрудно видеть, что после преобразования множество S является $(C_4, P_4, P_3, 2P_2, P_2)$-разбиением ребер графа G наименьшей мощности. Теорема 2 доказана.

Алгоритм решения задачи в классе деревьев. В этом разделе мы рассматриваем задачу в предположении, что графы являются деревьями. По определению в деревьях нет циклов и, в частности, простых циклов порядка 4. Поэтому мы можем ограничиться рассмотрением

(P_4, P_3, $2P_2$, P_2)-разбиений ребер дерева, исключив из списка допустимых графов простой цикл C_4. Установим важные свойства (P_4, P_3, $2P_2$, P_2)-разбиений ребер дерева. Мы сформулируем и докажем их для более общего класса графов, а именно, для C_4-свободных двудольных графов.

Лемма 1. *В наименьшем* (P_4, P_3, $2P_2$, P_2)*-разбиении ребер двудольного* C_4*-свободного графа G содержится не более чем один граф, изоморфный* P_2.

Доказательство. Пусть в наименьшем (P_4, P_3, $2P_2$, P_2)-разбиении ребер двудольного графа G найдутся два графа, изоморфные P_2. Тогда, объединив эти два графа в один, получим (P_4, P_3, $2P_2$, P_2)-разбиение ребер графа G меньшей мощности, что невозможно, поскольку исходное разбиение ребер графа G является наименьшим. Лемма 1 доказана.

Лемма 2. *Существует наименьшее* (P_4, P_3, $2P_2$, P_2)*-разбиение ребер двудольного* C_4*-свободного графа G, в котором число графов, изоморфных* P_4*, равно максимальному числу реберно непересекающихся простых цепей порядка* 4 *в графе G*.

Доказательство. Пусть семейство P подграфов графа G является наименьшим (P_4, P_3, $2P_2$, P_2)-разбиением ребер графа G. Пусть также p_1, p_2, p_3 – это количества графов в семействе P с одним, двумя и тремя ребрами соответственно. Утверждается, что

$$|P| \geq p_3 + \frac{|E(G)| - 3p_3}{2}. \tag{1}$$

В самом деле, в разбиении P ребер графа G число графов, содержащих три ребра, равно p_3. Все остальные графы семейства P в совокупности содержат $|E(G)| - 3p_3$ ребер, при этом каждый из этих графов содержит не более чем два ребра. Поэтому остальных графов в семействе P не меньше, чем $(|E(G)| - 3p_3) / 2$. Пусть t_3 – это максимальное число реберно непересекающихся подграфов графа G, изоморфных P_4. Предположим, что $p_3 < t_3$, т. е.

$$p_3 + 1 \leq t_3. \tag{2}$$

Построим (P_4, P_3, $2P_2$, P_2)-разбиение ребер графа G, в котором подграфов графа G, изоморфных P_4, ровно t_3. Пусть S – наибольшая P_4-упаковка в графе G. Дополним семейство S до (P_4, P_3, $2P_2$, P_2)-разбиения ребер графа G. Из ребер графа G, которые не содержатся в графах из семейства S, произвольным образом составим пары ребер. Получим подграфы графа G, изоморфные графам P_3 или $2P_2$. Добавим эти подграфы графа G в S. При этом возможно, что для одного ребра графа G не найдется пары. Тогда это ребро включим в S как подграф графа G, изоморфный P_2. Нетрудно видеть, что результирующее семейство S подграфов графа G является (P_4, P_3, $2P_2$, P_2)-разбиением ребер графа G и

$$|S| = t_3 + \left\lceil \frac{|E(G)| - 3t_3}{2} \right\rceil.$$

Учитывая

$$\left\lceil \frac{|E(G)| - 3t_3}{2} \right\rceil \leq \frac{|E(G)| - 3t_3 + 1}{2},$$

получаем

$$|S| \leq t_3 + \frac{|E(G)| - 3t_3 + 1}{2}. \tag{3}$$

Имеем

$$|S| \leq t_3 + \frac{|E(G)| - 3t_3 + 1}{2} = \frac{|E(G)| - t_3 + 1}{2} \leq \frac{|E(G)| - (p_3 + 1) + 1}{2} = \frac{|E(G)| - p_3}{2} \leq |P|.$$

В этой цепочке равенств и неравенств первое неравенство – это неравенство (3); второе неравенство верно, так как верно неравенство (2); третье неравенство верно, так как верно неравенство (1). Итак, S является $(P_4, P_3, 2P_2, P_2)$-разбиением ребер графа G и $|S| \le |P|$, т. е. S – наименьшее $(P_4, P_3, 2P_2, P_2)$-разбиение ребер графа G, в котором число подграфов графа G, изоморфных P_4, равно t_3. Лемма 2 доказана.

Таким образом, для того чтобы получить наименьшее $(P_4, P_3, 2P_2, P_2)$-разбиение ребер дерева G, достаточно найти наибольшую P_4-упаковку S в дереве G, а затем из ребер дерева G, не вошедших в цепи упаковки S, составить пары.

Наша ближайшая задача – разработать алгоритм поиска наибольшей P_4-упаковки в дереве. Отметим, что задача нахождения наибольшей P_4-упаковки для планарных двудольных графов относится к числу NP-трудных [5]. Введем необходимые понятия. Вершина дерева G называется *листом*, если ее степень равна 1. Вершина степени два, смежная с листом, называется *основанием листа*. Пусть v – вершина дерева G. Обозначим через $L_G(v)$ множество листьев дерева G, смежных с вершиной v, а через $S_G(v)$ – множество листовых оснований, смежных с вершиной v.

Л е м м а 3. *Для любого дерева G порядка больше* 1 *существует вершина v такая, что выполняется хотя бы одно из следующих условий*:

(У1) $2|L_G(v)| > \deg_G(v)$;

(У2) $2|S_G(v)| > \deg_G(v)$;

(У3) $|L_G(v)| > 0$ и $|S_G(v)| > 0$;

(У4) $\deg_G(v) = 2$ и $|S_G(v)| > 0$.

Д о к а з а т е л ь с т в о. Рассмотрим в дереве G наидлиннейшую простую цепь P. Пусть концевыми вершинами этой цепи являются вершины u, w. Заметим, что $\deg_G(u) = \deg_G(w) = 1$ (иначе цепь P не является наидлиннейшей простой цепью в дереве G).

Пусть длина цепи P равна 1. Тогда дерево G состоит из двух вершин u, w и одного ребра $\{u, w\}$. В этом случае $L_G(w) = \{u\}$ и для вершины $v = w$ выполняется условие (У1).

Пусть длина цепи P равна 2. Тогда дерево G представляет собой звезду $K_{1,t}$, где $t \ge 2$. Обозначим центральную вершину звезды G символом v. Для вершины v выполняется условие (У1).

Пусть теперь длина цепи P больше, чем 2. Обозначим промежуточные вершины цепи P, ближайшие к вершине w, через w_1, w_2, w_3, при этом вершина w_1 располагается ближе к вершине w, чем вершина w_2, и вершина w_2 находится ближе к вершине w, чем вершина w_3. Отметим, что любая вершина дерева G, которая отлична от вершин w, w_2 и смежна с вершиной w_1, является листом в дереве G, поскольку в противном случае простая цепь P в дереве G не наидлиннейшая. Если $\deg_G(w_1) > 2$, то для вершины $v = w_1$ выполняется условие (У1). Предположим теперь, что $\deg_G(w_1) = 2$. Так как вершины w_1 и w смежны, то вершина w_1 является основанием листа. При этом предположении рассмотрим вершину w_2. Заметим, что $w_1 \in S_G(w_2)$. Если $\deg_G(w_2) = 2$, то для вершины $v = w_2$ выполняется условие (У4). Пусть теперь $\deg_G(w_2) > 2$. Если в дереве G существует листовая вершина z, смежная с вершиной w_2, то для вершины $v = w_2$ выполняется условие (У3), поскольку $z \in L_G(w_2)$ и $w_1 \in S_G(w_2)$. Пусть теперь любая вершина, смежная с вершиной w_2, имеет степень не меньше, чем 2. Пусть все вершины, смежные с вершиной w_2, за исключением, быть может, вершины w_3, являются основаниями листьев. В этом случае для вершины $v = w_2$ выполняется условие (У2). Предположим, что существует вершина $x \ne w_3$, смежная с вершиной w_2 и степень которой больше, чем 2. Заметим, что любая вершина, смежная с вершиной x и отличная от вершины w_2, является листом, иначе в дереве G простая цепь P не наидлиннейшая. В этом случае для вершины $v = x$ выполняется условие (У1). Лемма 3 доказана.

Рассмотрим вопрос о том, как устроена наибольшая P_4-упаковка в окрестности вершины v, удовлетворяющей условиям из леммы 3.

Л е м м а 4. *Пусть для вершины v дерева G выполняется условие* (У1) *из леммы* 3. *Тогда удаление из дерева G любой вершины, принадлежащей множеству $L_G(v)$, не изменяет мощность наибольшей P_4-упаковки.*

Д о к а з а т е л ь с т в о. Рассмотрим наибольшую P_4-упаковку S дерева G. Так как $2|L_G(v)| >$ $> \deg_G(v)$, то существует лист x дерева G, смежный с вершиной v и не содержащийся ни в какой цепи упаковки S. Пусть w – произвольная вершина, принадлежащая множеству $L_G(v)$. Если

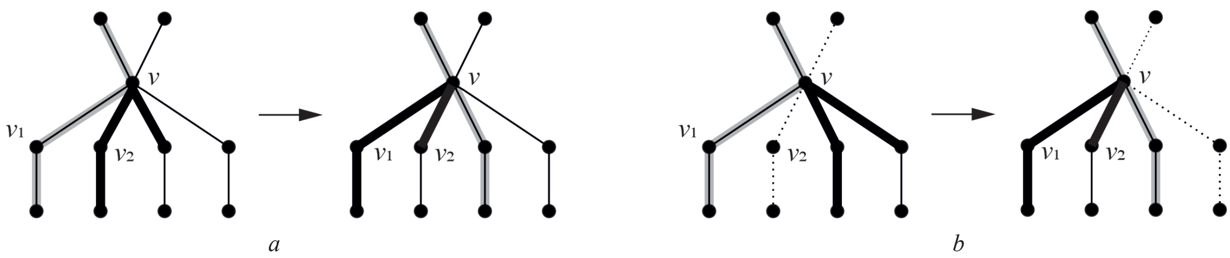

Рис. 1. Слева (*a*) изображен фрагмент дерева *G* с двумя выделенными цепями порядка 4, которые принадлежат упаковке *S*, а справа (*a*) – тот же фрагмент дерева *G* с результатом преобразования этих двух цепей (остальные цепи упаковки *S* остаются без изменения); слева (*b*) изображен фрагмент дерева *G* с тремя выделенными цепями порядка 4, которые содержатся в упаковке *S*, а справа (*b*) – тот же фрагмент дерева *G* с результатом преобразования этих трех цепей (остальные цепи упаковки *S* остаются без изменения)

Fig. 1. (*a*) The fragment of the tree *G* with two emphasized 4 order paths that belong to packing *S* (left), and this fragment after the transformation of two paths, the other paths of *S* remain without changing (right). (*b*) The fragment of the tree with three emphasized paths of the 4 order that belong to *S* (left), and this fragment after the modification of these three paths (right)

лист w не содержится ни в одной цепи упаковки S, то упаковка S является наибольшей P_4-упаковкой в дереве $G - w$. Предположим, что вершина w содержится в некоторой цепи P упаковки S. Вершины цепи P – вершины w, v, a, b. Заменим в цепи P вершину w на вершину x. Результирующая упаковка S является наибольшей P_4-упаковкой в графе $G - w$. Лемма 4 доказана.

Л е м м а 5. *Пусть G – граф с не менее чем четырьмя вершинами, и для вершины v дерева G выполняется условие* (У2) *из леммы* 3. *Тогда для любых двух различных вершин v_1 и v_2, принадлежащих множеству $S_G(v)$, существует наибольшая P_4-упаковка, в которой есть цепь, содержащая ребра $\{v, v_1\}$ и $\{v, v_2\}$.*

Д о к а з а т е л ь с т в о. Пусть для вершины v дерева G выполняется условие (У2). Тогда в множестве $S_G(v)$ или одна вершина, или хотя бы две вершины. Если $|S_G(v)| = 1$, то $\deg_G(v) = 1$ и, как следствие, в дереве G три вершины, что не так. Поэтому гарантируется существование хотя бы двух различных вершин, принадлежащих множеству оснований $S_G(v)$. Пусть в дереве G вершины v_1 и v_2 – это основания листьев из множества $S_G(v)$. Рассмотрим наибольшую P_4-упаковку S в дереве G. Возможны следующие два случая.

С л у ч а й 1. Пусть в упаковке S есть цепь P такая, что $|V(P) \cap S_G(v)| = 2$. Можно перестроить цепи упаковки S без изменения ее мощности таким образом, чтобы в упаковке S оказалась цепь, содержащая ребра $\{v, v_1\}$ и $\{v, v_2\}$. Не будем доказывать это общее утверждение. Ограничимся рассмотрением примеров преобразования цепей в S, из которых нетрудно уяснить, как действовать в общем случае. В левой части рис. 1, *a* ребро $\{v, v_2\}$ содержится в цепи P, а ребро $\{v, v_1\}$ содержится в другой цепи Q. В упаковке S цепи P и Q заменим на две другие цепи, изображенные на правой части рис. 1, *a*. В левой части рисунка 1, *b* ребра $\{v, v_1\}$ и $\{v, v_2\}$ содержатся в цепях Q, R упаковки S, отличных от цепи P. В упаковке S цепи P, Q и R заменим на три цепи, выделенные в правой части рис. 1, *b*.

С л у ч а й 2. Пусть теперь в упаковке S все цепи P содержат не больше одной вершины из $S_G(v)$. Этот случай сведем к предыдущему следующим образом. Так как $2|S_G(v)| > \deg_G(v)$, то найдется основание $u \in S_G(v)$ такое, что ребро $\{v, u\}$ не содержится ни в одной цепи упаковки S. Пусть l – это лист дерева G, смежный с основанием u. Для любого другого основания $w \in S_G(v)$, $w \neq u$, ребро $\{v, w\}$ содержится в некоторой цепи P_w упаковки S (иначе S – это не наибольшая P_4-упаковка, поскольку ее можно расширить до P_4-упаковки большего размера). В упаковке S выберем одну из таких цепей P_w. Удалим из цепи P_w все вершины, кроме вершин w и v, и дополним ее до цепи порядка 4, присоединив к ней вершины u и l. Доказательство леммы 5 завершено.

С л е д с т в и е 1. *Пусть для вершины v дерева G выполняется условие* (У2) *из леммы* 3. *Пусть v_1 и v_2 – произвольные различные вершины, принадлежащие множеству $S_G(v)$, u_1 – лист дерева G, смежный с вершиной v_1, и u_2 – лист дерева G, смежный с вершиной v_2. Тогда мощ-*

ность наибольшей P_4-упаковки дерева $G - v_1 - u_1 - v_2 - u_2$ на единицу меньше, чем мощность наибольшей P_4-упаковки дерева G.

Лемма 6. *Пусть для вершины v дерева G выполняется условие* (У3) *леммы* 3. *Тогда для любой вершины $v_1 \in S_G(v)$ и для любой вершины $v_2 \in L_G(v)$ существует наибольшая P_4-упаковка дерева G, в которой есть цепь, содержащая ребра $\{v, v_1\}$ и $\{v, v_2\}$.*

Доказательство. Пусть S – наибольшая P_4-упаковка в дереве G. Заметим, что хотя бы одно из ребер $\{v, v_1\}$ и $\{v, v_2\}$ содержится в цепи из упаковки S (иначе P_4-упаковка S не является наибольшей в дереве G). Возможны только два случая.

Случай 1. Пусть оба ребра $\{v, v_1\}$ и $\{v, v_2\}$ содержатся в цепях из упаковки S. Если эти ребра содержатся в одной и той же цепи из S, то утверждение доказано. Пусть указанные ребра содержатся в разных цепях P и Q упаковки S. Нетрудно видеть, что эти две цепи всегда можно перестроить таким образом, чтобы одна из них содержала ребра $\{v, v_1\}$ и $\{v, v_2\}$. В самом деле, разобьем каждую из цепей P и Q вершиной v на две части и соединим части от разных цепей. При этом возможно, что какая-то из результирующих цепей будет короткой или длинной. В этом случае «нарастим» или уменьшим цепь с помощью добавления листа, смежного с вершиной v_1, или удаления «лишних» вершин.

Случай 2. Пусть ровно одно из ребер $\{v, v_1\}$ или $\{v, v_2\}$ содержится в цепи упаковки S. Тогда удалим эту цепь из S и добавим в S цепь порядка 4, содержащую ребра $\{v, v_1\}$ и $\{v, v_2\}$. Лемма 6 доказана.

Следствие 2. *Пусть для вершины v дерева G выполняется условие* (У3) *из леммы* 3. *Тогда размер наибольшей P_4-упаковки в графе G на единицу больше, чем размер наибольшей P_4-упаковки в графе $G - v_2 - v_1 - l$, где v_1 – произвольный лист дерева, принадлежащий $L_G(v)$, v_2 – это произвольное основание листа дерева, принадлежащее $S_G(v)$, и l – это единственный лист дерева G, смежный с основанием v_2.*

Лемма 7. *Пусть для вершины v дерева G выполняется условие* (У4) *из леммы* 3. *Тогда для вершины $u \in S_G(v)$ существует наибольшая P_4-упаковка S в дереве G, в которой есть цепь, содержащая вершины v, u, а также лист, смежный с вершиной u.*

Лемма 7 тривиальна. Поэтому мы не приводим ее доказательство.

Следствие 3. *Пусть для вершины v дерева G выполняется условие* (У4) *из леммы* 3. *Тогда размер наибольшей P_4-упаковки в графе G на единицу больше, чем размер наибольшей P_4-упаковки в графе $G - v - u - l$, где $u \in S_G(v)$ и l – это лист дерева G, смежный с основанием u.*

Лемма 3 гарантирует, что в дереве G существует вершина v, для которой выполняется хотя бы одно из условий (У1)–(У4). Согласно лемме 4 и следствиям 1–3, из дерева можно удалить несколько вершин таким образом, чтобы число цепей в наибольшей P_4-упаковке дерева уменьшилось на единицу или не изменилось. Идея алгоритма состоит в том, чтобы на каждой итерации искать вершину v, удовлетворяющую хотя бы одному из условий (У1)–(У4), а затем удалять вершины из дерева в зависимости от того, какое условие имеет место. Выполняем эту итерационную процедуру до тех пор, пока в дереве не останется меньше восьми вершин. Для дерева порядка меньше 8 наибольшую P_4-упаковку находим за константное время полным перебором. Отметим, что при удалении вершины из дерева, у некоторых вершин u дерева могут измениться множества $L_G(u)$ и $S_G(u)$. Рассмотрим, в каких случаях и как изменяются множества $L_G(u)$ и $S_G(u)$.

Пусть для вершины v дерева G выполняется условие (У1) леммы 3. Из того, что $|V(G)| \geq 8$ и $2|L_G(v)| > \deg_G(v)$, следует $|L_G(v)| \geq 2$. Так как $|L_G(v)| \geq 2$ и $|V(G)| \geq 8$, то $\deg_G(v) \geq 3$. Отметим, что вершина v не является ни листом, ни основанием листа. Рассмотрим произвольный лист u, принадлежащий $L_G(v)$. Удалим из дерева G лист u. Тогда $L_{G-u}(v) = L_G(v) \setminus \{u\}$. После удаления степень изменится только у вершины v. В дереве $G - u$ вершина v не является листом, так как до удаления $|L_G(v)| \geq 2$, но может быть основанием листа. Причем последнее возможно только в одном случае, когда $|L_G(v)| = 2$ и $\deg_G(v) = 3$. Предположим, что $|L_G(v)| = 2$ и $\deg_G(v) = 3$. Пусть также $N_G(v) = \{u, l, w\}$ и $L_G(v) = \{u, l\}$. Тогда в дереве $G - u$ вершина v является основанием листа, смежным с вершиной w. Стало быть, $S_{G-u}(w) = S_G(w) \cup \{v\}$.

Пусть для вершины v дерева G выполняется условие (У2) леммы 3. Так как $2|S_G(v)| > \deg_G(v)$ и $|V(G)| \geq 8$, то $|S_G(v)| \geq 2$. Так как $|S_G(v)| \geq 2$ и $|V(G)| \geq 8$, то $\deg_G(v) \geq 3$. Рассмотрим произвольные

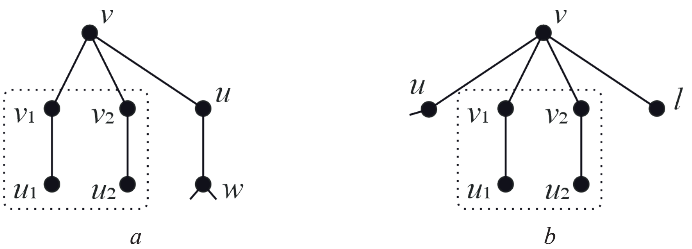

Рис. 2. Фрагмент дерева G такой, что после удаления вершин v_1, v_2, u_1, u_2 вершина v становится листом (*a*) и основанием листа (*b*)

Fig. 2. A fragment of the tree G has such a form that after deleting vertices v_1, v_2, u_1, u_2 the v vertex becomes a leaf (*a*), and a base of a leaf (*b*)

вершины $v_1 \in S_G(v)$ и $v_2 \in S_G(v)$. Пусть u_1 – это лист дерева G, смежный с вершиной v_1, и u_2 – это лист дерева G, смежный с вершиной v_2. Так как $\deg_G(v) \geq 3$, то вершина v не является ни листом, ни основанием листа. Удалим из дерева G вершины v_1, v_2, u_1, u_2. Тогда

$$S_{G-v_1-v_2-u_1-u_2}(v) = S_G(v) \setminus \{v_1, v_2\}.$$

После удаления вершин из дерева вершина v может быть листом или основанием листа. Вершина v становится листом только в том случае, когда (рис. 2, *a*) $\deg_G(v) = 3$. Пусть в дереве $G - v_1 - v_2 - u_1 - u_2$ вершина v – лист, смежный с некоторой вершиной u. Следовательно,

$$L_{G-v_1-v_2-u_1-u_2}(u) = L_G(u) \cup \{v\}.$$

Если, кроме этого, $\deg_G(u) = 2$, то в дереве $G - v_1 - v_2 - u_1 - u_2$ вершина u является листовым основанием, смежным с некоторой вершиной w. В этом случае

$$S_{G-v_1-v_2-u_1-u_2}(w) = S_G(w) \cup \{u\}.$$

Наконец, рассмотрим ситуацию, когда в дереве $G - v_1 - v_2 - u_1 - u_2$ вершина v является основанием листа. Такое возможно только в том случае, когда $\deg_G(v) = 4$ и $|L_G(v)| = 1$ (рис. 2, *b*). Пусть $N_G(v) = \{u, v_1, v_2, l\}$ и $\deg_G(l) = 1$. Заметим, что $\deg_G(u) > 1$, поскольку $|V(G)| \geq 8$. Пусть в дереве $G - v_1 - v_2 - u_1 - u_2$ вершина v является листовым основанием, смежным с вершиной u. Тогда

$$S_{G-v_1-v_2-u_1-u_2}(u) = S_G(u) \cup \{v\}.$$

Пусть для вершины v дерева G выполняется условие (У3) из леммы 3. Рассмотрим произвольные вершины $v_1 \in S_G(v)$, $v_2 \in L_G(v)$. Пусть u_1 – это лист дерева G, смежный с основанием v_1. Так как $|S_G(v)| > 0$, $|L_G(v)| > 0$ и $|V(G)| \geq 8$, то $\deg_G(v) \geq 3$. Значит, вершина v не является ни листом, ни основанием листа. Удалим из дерева G вершины v_1, v_2, u_1. Тогда

$$S_{G-v_1-v_2-u_1}(v) = S_G(v) \setminus \{v_1\}, \quad L_{G-v_1-v_2-u_1}(v) = L_G(v) \setminus \{v_2\}.$$

Степень изменится только у вершины v. В дереве $G - v_1 - v_2 - u_1$ вершина v может быть листом или основанием листа. Пусть вершина v – лист в дереве $G - v_1 - v_2 - u_1$. Это возможно только в одном случае, когда $\deg_G(v) = 3$ (рис. 3, *a*). Вершина v смежна с некоторой вершиной $u \in N_G(v) \setminus \{v_1, v_2\}$. Следовательно,

$$L_{G-v_1-v_2-u_1}(u) = L_G(u) \cup \{v\}.$$

Если $\deg_G(u) = 2$, то в дереве $G - v_1 - v_2 - u_1$ вершина u является листовым основанием, смежным с некоторой вершиной w. В этом случае необходимо обновить множество оснований для вершины w:

$$S_{G-v_1-v_2-u_1}(w) = S_G(w) \cup \{u\}.$$

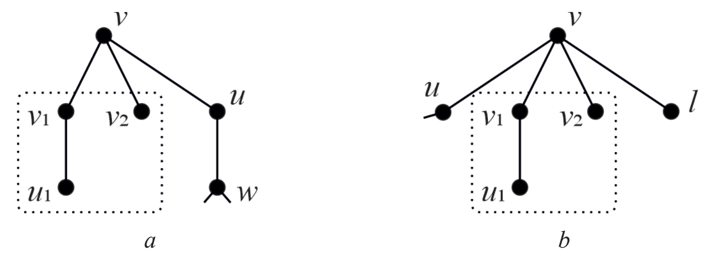

Рис. 3. Фрагмент дерева G такой, что после удаления вершин v_1, v_2 и u вершина v становится листом (a) и основанием листа (b)

Fig. 3. A fragment of the tree G has such a form that after deleting vertices v_1, v_2 and u the v vertex becomes a leaf (a), and a base of a leaf (b)

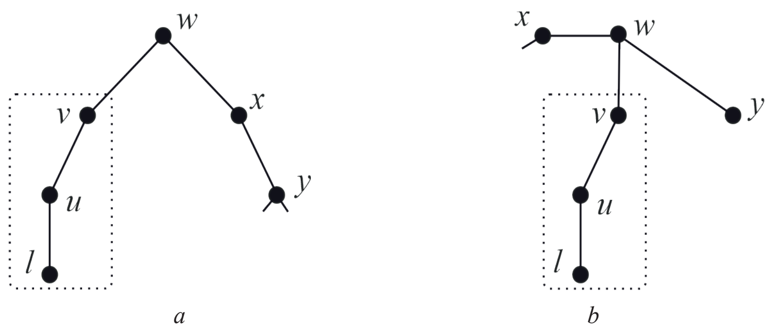

Рис. 4. Фрагмент дерева G в случае, в котором после удаления вершин v, u и l вершина w становится листом (a) и основанием листа (b)

Fig. 4. A fragment of the tree G has such a form that after deleting vertices v, u and l the w vertex becomes a leaf (a), and a base of a leaf (b)

Рассмотрим теперь ситуацию, в которой вершина v в дереве $G - v_1 - v_2 - u_1$ является основанием листа. Это возможно только тогда, когда $\deg_G(v) = 4$ и $|L_G(v)| = 2$ (рис. 3, b). Пусть $N_G(v) = \{u, v_1, v_2, l\}$ и $L_G(v) = \{v_2, l\}$. Тогда $S_{G-v_1-v_2-u_1}(u) = S_G(u) \cup \{v\}$.

Пусть вершина v дерева G удовлетворяет условию (У4) из леммы 3. Рассмотрим произвольную вершину $u \in S_G(v)$. Пусть l – это лист дерева G, смежный с вершиной u. Так как $|V(G)| \geq 8$, то существует вершина $w (w \neq u)$, смежная с вершиной v, степени $\deg_G(w) \geq 2$. Удалим из дерева G вершины v, u, l. В дереве $G - v - u - l$ вершина w может быть листом или основанием листа.

Вершина w в дереве $G - v - u - l$ является листом только в случае, когда $\deg_G(w) = 2$, т. е. когда в дереве G существует вершина $x (x \neq v)$, смежная с вершиной w (рис. 4, a). В этом случае

$$L_{G-v-u-l}(x) = L_G(x) \cup \{w\}.$$

Если $\deg_G(x) = 2$, то в дереве $G - v - u - l$ существует вершина $y (y \neq w)$, смежная с вершиной x, и вершина x – листовое основание, смежное с вершиной y. Стало быть,

$$S_{G-v-u-l}(y) = S_G(y) \cup \{x\}.$$

Пусть теперь вершина w в дереве $G - v - u - l$ является основанием листа. Это возможно только тогда, когда $\deg_G(w) = 3$ и $|L_G(w)| = 1$ (рис. 4, b). Пусть $N_G(w) = \{x, v, y\}$ и $L_G(w) = \{y\}$. Тогда

$$S_{G-v-u-l}(x) = S_G(x) \cup \{w\}.$$

Переходим к описанию алгоритма нахождения наибольшей P_4-упаковки в дереве G порядка n. Предполагаем, что дерево представляется в виде списков смежности. Алгоритм:

Шаг 1. Находим для каждой вершины v дерева G множества $S_G(v)$ и $L_G(v)$, а также ее степень $\deg(v)$. При этом $S_G(v)$ и $L_G(v)$ представляются в виде очереди $S(v)$ и $L(v)$ соответственно.

Шаг 2. Создаем очередь Q, которая содержит все вершины дерева G в произвольном порядке, а также пустое множество S.

Шаг 3. Если в дереве число вершин меньше 8, то находим в нем наибольшую P_4-упаковку полным перебором, поместим все цепи найденной упаковки в множество S, и алгоритм завершает свою работу.

Шаг 4. Извлечем вершину v из очереди Q.

Шаг 5. Если вершины v нет в дереве или для вершины v не выполняется ни одно из условий (У1)–(У4), то переходим на шаг 4.

Шаг 6. Если для вершины v выполняется условие (У1), то:

1) извлечем лист u из очереди $L(v)$;

2) если $|L(v)| = 1$ и $\deg(v) = 3$, то для единственной вершины $w \in N(v) \setminus \left(L(v) \cup \{u\}\right)$ добавим в очередь $S(w)$ вершину v, а саму вершину w добавим в очередь Q;

3) удалим из дерева G лист u и уменьшим степень вершины v на единицу;

4) добавим вершину v в очередь Q;

переходим на шаг 3.

Шаг 7. Если для вершины v выполняется условие (У2), то:

1) извлечем два листовых основания v_1, v_2 из очереди $S(v)$; пусть $u_1 \in L(v_1)$ и $u_2 \in L(v_2)$; добавим в S цепь с вершинами v_2, v, v_1, u_1;

2) если $\deg(v) = 3$, то для единственной вершины $u \in N(v) \setminus \{v_1, v_2\}$ добавим в $L(u)$ вершину v, а также вершину u добавим в очередь Q;

3) если $\deg(v) = 3$ и степень единственной вершины $u \in N(v) \setminus \{v_1, v_2\}$ равна 2, то для единственной вершины $w \in N(u) \setminus \{v\}$ добавим в очередь $S(w)$ вершину u, а также вершину w добавим в очередь Q;

4) если $\deg(v) = 4$ и $|L(v)| = 1$, то для единственной вершины $u \in N(v) \setminus \left(L(v) \cup \{v_1, v_2\}\right)$ добавим в очередь $S(u)$ вершину v, а также вершину u добавим в Q;

5) удалим из дерева G вершины v_1, v_2 и u_1, u_2, а также уменьшим степень вершины v на два;

6) вершину v добавим в очередь Q;

переходим на шаг 3.

Шаг 8. Если для вершины v выполняется условие (У3), то:

1) извлечем вершину v_1 из очереди $S(v)$ и вершину v_2 из очереди $L(v)$. Пусть $u_1 \in L(v_1)$; добавим в S цепь дерева с вершинами v_2, v, v_1, u_1;

2) если $\deg(v) = 3$, то для единственной вершины $u \in N(v) \setminus \{v_1, v_2\}$ добавим в $L(u)$ вершину v, а также добавим вершину u в очередь Q;

3) если $\deg(v) = 3$ и степень единственной вершины $u \in N(v) \setminus \{v_1, v_2\}$ равна 2, то для единственной вершины $w \in N(u) \setminus \{v\}$ добавим в очередь $S(w)$ вершину u, а также вершину w добавим в очередь Q;

4) если $\deg(v) = 4$ и $|L(v)| = 2$, то для единственной вершины $u \in N(v) \setminus \left(L(v) \cup \{v_1, v_2\}\right)$ добавим в $S(u)$ вершину v, а также вершину u добавим в очередь Q;

5) удалим из дерева G вершины v_1, v_2 и u_1, а также уменьшим степень вершины v на два;

6) добавим в очередь Q вершину v;

переходим на шаг 3.

Шаг 9. Если для вершины v выполняется условие (У4), то:

1) извлечем вершину u из очереди $S(v)$; пусть $l \in L(u)$ и $w \in N(v) \setminus \{u\}$; добавим в S цепь с вершинами w, v, u, l;

2) если $\deg(w) = 2$, то для единственной вершины $x \in N(w) \setminus \{v\}$ добавим в очередь $L(x)$ вершину w, а также добавим в очередь Q вершину x;

3) если $\deg(w) = 2$ и степень единственной вершины $x \in N(w) \setminus \{v\}$ равна 2, то для единственной вершины $y \in N(x) \setminus \{w\}$ добавим в очередь $S(y)$ вершину x, а также добавим в очередь Q вершину y;

4) если $\deg(w) = 3$ и $|L(w)| = 1$, то для единственной вершины $x \in N(w) \setminus \left\{L(w) \cup \{v\}\right\}$ добавим в очередь $S(x)$ вершину w, а также добавим в очередь Q вершину x;

5) вершину v добавим в очередь Q;
переходим на шаг 3.

Конец алгоритма.

На каждой итерации алгоритма из очереди Q извлекается вершина v. Если вершина v не удовлетворяет условиям (У1)–(У4), то текущая итерация завершается и совершается переход на следующую итерацию. Если извлеченная вершина v удовлетворяет хотя бы одному из условий (У1)–(У4), то в зависимости от условия, во-первых, из дерева удаляются некоторые вершины и, во-вторых, обновляются множества $S_G(u)$, $L_G(u)$, а также степень $\deg_G(u)$ для всех вершин u, для которых эти атрибуты изменились. Вершины с обновленными атрибутами добавляются в очередь Q. После этого совершается переход на следующую итерацию. Итерационный процесс завершается в тот момент, когда в дереве меньше 8 вершин. Далее покажем, что на каждой итерации очередь Q включает в себя все вершины дерева, удовлетворяющие хотя бы одному условию (У1)–(У4). Лемма 3 гарантирует существование на каждой итерации в очереди Q хотя бы одной вершины v, удовлетворяющей условию (У1), (У2), (У3) или (У4). Отметим, что выполнение или не выполнение условий (У1)–(У4) относительно вершины v зависит только от множеств $S_G(v)$, $L_G(v)$ и степени $\deg_G(v)$.

Покажем, что на каждой итерации очередь Q содержит все вершины дерева, удовлетворяющие хотя бы одному из условий (У1)–(У4). На первой итерации очередь Q содержит все вершины дерева и, в частности, все вершины дерева, удовлетворяющие по крайней мере одному условию (У1)–(У4). Пусть выполнена $(i-1)$-я итерация. По индуктивному предположению все вершины текущего дерева со свойствами (У1)–(У4) есть в очереди Q. На i-й итерации из очереди Q извлекается вершина v. Если вершина v не удовлетворяет условиям (У1)–(У4), то Q, по-прежнему, содержит все вершины, удовлетворяющие условиям (У1), (У2), (У3) или (У4). Пусть теперь вершина v удовлетворяет хотя бы одному из четырех условий. В зависимости от условия из дерева удаляются некоторые вершины. Затем осуществляется обновление множеств $S(u)$, $L(u)$ и степени $\deg(u)$ для тех вершин u, для которых эти характеристики изменяются. Заметим, что для самой вершины v изменяется ее степень.

Рассмотрим вершины дерева, для которых указанные характеристики не изменились. Если одна из рассматриваемых вершин не содержится в очереди Q, т. е. она не удовлетворяет условиям (У1)–(У4) на предыдущей $(i-1)$-й итерации, то эта же вершина не будет удовлетворять этим условиям и на i-й итерации. Теперь рассмотрим те вершины дерева, для которых характеристики изменились. Заметим, что число таких вершин не превосходит 3. Потенциально каждая такая вершина может удовлетворять хотя бы одному из условий (У1)–(У4). Поскольку все такие вершины добавляются в очередь Q, то в конце i-й итерации очередь Q также будет содержать все вершины, удовлетворяющие условиям (У1), (У2), (У3) или (У4).

Переходим к анализу трудоемкости представленного алгоритма. Утверждается, что временная сложность алгоритма в худшем случае $O(n \log n)$, где n – число вершин во входном дереве. Нетрудно видеть, что шаг 1 можно реализовать таким образом, чтобы он работал за время $O(n)$. Мы предполагаем, что списки смежности вершин дерева реализованы в виде деревьев поиска. Поиск, вставка вершины в список смежности и удаление вершины из списка смежности можно осуществить за время $O(\log n)$. Проверка условий (У1)–(У4) для конкретной вершины v занимает $O(1)$ времени. Удаление вершины из дерева (шаги 6–9) реализуется за время $O(\log n)$. Итак, каждая итерация занимает $O(\log n)$ времени. Остается оценить число итераций. Заметим, что каждый раз в очередь Q добавляется не более трех вершин. При добавлении в очередь Q ненулевого числа вершин хотя бы одна вершина удаляется из дерева. Стало быть, общее число вершин, добавляемых в очередь Q на шагах 6–9, не больше $3n$. Число вершин в очереди Q не превосходит $4n$. Следовательно, число итераций, совершаемых алгоритмом, не больше $4n$. Таким образом, число итераций $O(n)$ и общее время работы алгоритма в худшем случае $O(n \log n)$.

NP-полнота задачи в классе квазихордальных двудольных графов. Пусть $G = (V, E)$ – двудольный граф. Ребро $e = \{u, v\}$ этого графа называется *бисимплициальным*, если подграф графа G, порожденный множеством вершин $N_G(u) \cup N_G(v)$, изоморфен некоторому полному двудольному графу. Двудольный граф G называется *квазихордальным* [6], если существует упорядоченный набор $(e_1, e_2, \ldots, e_k)$ его ребер такой, что

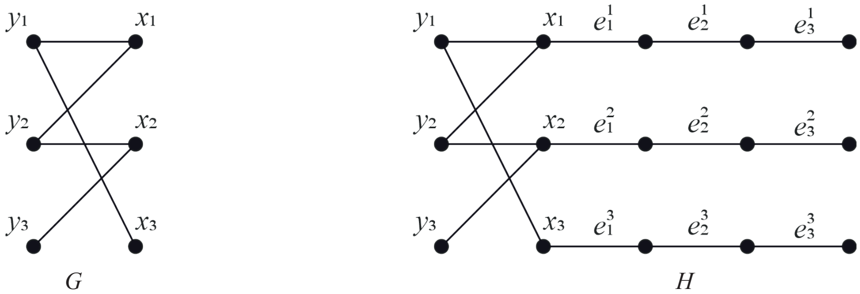

Рис. 5. Пример двудольного графа G и граф H, который получается в результате преобразования

Fig 5. An example of a bipartite graph G and a graph H that is a result of the transformation of the G graph

а) для каждого индекса $i \in \{1, 2, ..., k-1\}$ ребро e_i является бисимплициальным в графе $G(V \setminus S_{i-1})$, где $S_0 = \varnothing$ и S_{i-1} – множество концевых вершин ребер $e_1, e_2, \ldots, e_{i-1}$ $(i > 1)$;

б) $G(V \setminus S_k)$ – пустой граф, где S_k – множество концевых вершин ребер $e_1, e_2, \ldots, e_k$.

Существуют полиномиальные алгоритмы распознавания квазихордальных двудольных графов [7].

Т е о р е м а 3. *Задача* НАИМЕНЬШЕЕ $(C_4, P_4, P_3, 2P_2, P_2)$-РАЗБИЕНИЕ РЕБЕР ДВУДОЛЬНОГО ГРАФА *для квазихордальных двудольных графов NP-полна.*

Д о к а з а т е л ь с т в о. Задача принадлежит классу NP. Построим полиномиальное сведение NP-полной задачи НАИМЕНЬШЕЕ $(C_4, P_4, P_3, 2P_2, P_2)$-РАЗБИЕНИЕ РЕБЕР ДВУДОЛЬНОГО ГРАФА для всех двудольных графов к этой же задаче, ограниченной квазихордальными двудольными графами.

Пусть $G = (X \cup Y, E)$ – двудольный граф, где $X = \{x_1, x_2, \ldots, x_m\}$ и $Y = \{y_1, y_2, \ldots, y_n\}$. К каждой вершине x_i, $i \in \{1, 2, ..., m\}$, присоединим простую цепь P^i длины 3 с ребрами e_1^i, e_2^i, e_3^i. Результирующий граф обозначим через H. На рис. 5 приведен пример двудольного графа G и соответствующего ему графа H. Граф H двудольный квазихордальный. Соответствующий набор ребер имеет вид $\left(e_3^1, e_1^1, e_3^2, e_1^2, ..., e_3^m, e_1^m\right)$.

Утверждается, что существует $(C_4, P_4, P_3, 2P_2, P_2)$-разбиение S ребер графа G такое, что $|S| \le k$ тогда и только тогда, когда существует $(C_4, P_4, P_3, 2P_2, P_2)$-разбиение R ребер графа H такое, что $|R| \le k + m$.

Для того чтобы по $(C_4, P_4, P_3, 2P_2, P_2)$-разбиению $S(|S| \le k)$ ребер графа G получить $(C_4, P_4, P_3, 2P_2, P_2)$-разбиение $R(|R| \le k + m)$ ребер графа H, достаточно в S добавить цепи $P^1, P^2, \ldots, P^m$.

Пусть теперь R – $(C_4, P_4, P_3, 2P_2, P_2)$-разбиение ребер графа H такое, что $|R| \le k + m$. Преобразуем это разбиение таким образом, чтобы в R содержались цепи $P^1, P^2, \ldots, P^m$. Рассмотрим цепь P^i. Эта цепь состоит из ребер e_1^i, e_2^i, e_3^i. Ребро e_3^i содержится в некотором графе из R. Обозначим этот граф через P. Рассмотрим четыре случая в зависимости от количества ребер в P.

С л у ч а й 1. Пусть граф P содержит в точности одно ребро, т. е. P – это цепь длины 1. В этом случае «нарастим» цепь P, добавив к ней ребра e_1^i, e_2^i. В результате цепь P превратится в цепь P^i. При этом оба ребра e_1^i, e_2^i необходимо удалить из остальных графов разбиения R.

С л у ч а й 2. Пусть граф P содержит в точности два ребра. Одно из ребер – ребро e_3^i. Второе ребро P является ребром e_2^i или отлично от него. Рассмотрим два подслучая.

П о д с л у ч а й 2.1. Пусть второе ребро графа P – это ребро e_2^i. Тогда P является цепью длины 2. «Нарастим» цепь P, добавив к ней ребро e_1^i. В результате цепь P превратится в цепь P^i. При этом необходимо удалить ребро e_1^i из других графов разбиения R.

П о д с л у ч а й 2.2. Пусть второе ребро графа P – ребро e, отличное от ребра e_2^i. Сведем этот подслучай к предыдущему. Пусть $Q \in R$ – это граф, содержащий ребро e_2^i. Если граф Q содержит не более двух ребер, то в графах P и Q поменяем местами ребра e и e_2^i. Теперь граф P содержит ребро e_2^i. Далее переходим к подслучаю 2.1. Если граф Q содержит в точности три ребра, то Q – это простая цепь, которая содержит ребра e_1^i, e_2^i и некоторое ребро графа G. В этом случае перенесем ребра e_1^i, e_2^i из Q в P, а ребро e из P в Q. Снова получаем подслучай 2.1. Наконец, Q не может содержать четыре ребра, так как в противном случае Q – это простой цикл порядка 4, содержащий ребро e_2^i, что невозможно.

Случай 3. Пусть граф P содержит в точности три ребра. Тогда P – простая цепь длины 3. Нетрудно видеть, что цепь P совпадает с цепью P^i. Таким образом, в этом случае нет необходимости корректировать разбиение R.

Случай 4. Пусть граф P содержит в точности четыре ребра. Тогда P – простой цикл порядка 4, содержащий ребро e_3^i. Этот случай не реализуется.

Итак, мы всегда можем скорректировать разбиение R без увеличения мощности таким образом, чтобы в R находились цепи $P^1, P^2, \ldots, P^m$. Удалим из R цепи $P^1, P^2, \ldots, P^m$ и получим требуемое разбиение исходного графа G. Теорема 3 доказана.

Заключение. В настоящей работе рассматривается задача нахождения наименьшего набора реберно непересекающихся подграфов двудольного графа при условии, что каждое ребро графа содержится в подграфе из набора и каждый подграф из набора изоморфен простому циклу порядка 4 или графу порядка не больше 4, в котором каждая компонента связности – простая цепь. Установлено, что задача остается NP-трудной в классе двудольных графов с эйлеровыми компонентами связности и в классе квазихордальных двудольных графов. Предложены полиномиальные алгоритмы решения задачи в классе деревьев и в классе двудольных C_4-свободных графов с эйлеровыми компонентами связности. Известно, что хордальные двудольные графы содержатся в классе квазихордальных двудольных графов. Одно из возможных направлений дальнейшего изучения задачи состоит в установлении сложностного статуса задачи в классе хордальных двудольных графов.

Благодарности. Исследование первым автором выполнено при финансовой поддержке Белорусского республиканского фонда фундаментальных исследований (проект № Ф21РМ-001).

Acknowledgements. The research was carried out by the first author under the financial support of the Belarusian Republican Foundation for Fundamental Research (project no. Ф21РМ-001).

Список использованных источников

1. Лекции по теории графов / В. А. Емеличев [и др.]. – М.: УРСС, 2019. – 390 с.

2. Hochbaum, D. S. Covering the edges of bipartite graphs using $K_{2,2}$ graphs / D. S. Hochbaum, A. Levin // Theor. Comput. Sci. – 2010. – Vol. 411. – P. 1–9.

3. Dor, D. Graph Decomposition is NP-Complete: A Complete Proof of Holyer's Conjecture / D. Dor, M. Tarsi // SIAM J. Comput. – 1997. – Vol. 26, № 4. – P. 1166–1187. https://doi.org/10.1137/s0097539792229507

4. Holyer, I. The NP-completeness of some edge partition problems / I. Holyer // SIAM SIAM J. Comput. – 1981. – Vol. 10, № 4. – P. 713–717. https://doi.org/10.1137/0210054

5. Dyer, M. E. On the complexity of partitioning graphs into connected subgraphs / M. E. Dyer, A. M. Frieze // Discrete Appl. Math. – 1985. – Vol. 10, № 2. – P. 139–153. https://doi.org/10.1016/0166-218x(85)90008-3

6. Golumbic, M. C. Perfect elimination and chordal bipartite graphs / M. C. Golumbic, G. F. Goss // J. Graph Theory. – 1978. – Vol. 2, № 2. – P. 155–163. https://doi.org/10.1002/jgt.3190020209

7. Goh, L. Recognition of perfect elimination bipartite graphs / L. Goh, D. Rotem // Inf. Process. Lett. – 1982. – Vol. 15, № 4. – P. 179–182. https://doi.org/10.1016/0020-0190(82)90101-6

References

1. Emelichev V. A., Mel'nikov O. I., Sarvanov V. I., Tyshkevich R. I. *Lectures on Graph Theory.* Manncheim, Vena, Zurich, B. I. Wissenschaftsferlfg, 1994. 364 p.

2. Hochbaum D. S., Levin A. Covering the edges of bipartite graphs using $K_{2,2}$ graphs. *Theoretical Computer Science*, 2010, vol. 411, pp. 1–9.

3. Dor D., Tarsi M. Graph Decomposition is NP-Complete: A Complete Proof of Holyer's Conjecture. *SIAM Journal on Computing*, 1997, vol. 26, no. 4, pp. 1166–1187. https://doi.org/10.1137/s0097539792229507

4. Holyer I. The NP-completeness of some edge partition problems. *SIAM Journal on Computing*, 1981, vol. 10, no. 4, pp. 713–717. https://doi.org/10.1137/0210054

5. Dyer M. E., Frieze A. M. On the complexity of partitioning graphs into connected subgraphs. *Discrete Applied Mathematics*, 1985, vol. 10, no. 2, pp. 139–153. https://doi.org/10.1016/0166-218x(85)90008-3

6. Golumbic M. C., Goss G. F. Perfect elimination and chordal bipartite graphs. *Journal of Graph Theory*, 1978, vol. 2, no. 2, pp. 155–163. https://doi.org/10.1002/jgt.3190020209

7. Goh L., Rotem D. Recognition of perfect elimination bipartite graphs. *Information Processing Letters*, 1982, vol. 15, no. 4, pp. 179–182. https://doi.org/10.1016/0020-0190(82)90101-6

Информация об авторах

Дугинов Олег Иванович – кандидат физико-математических наук, доцент кафедры дискретной математики и алгоритмики, Белорусский государственный университет (пр. Независимости, 4, 220030, Минск, Республика Беларусь). E-mail: oduginov@gmail.com

Химич Семен Сергеевич – студент, факультет прикладной математики и информатики, Белорусский государственный университет (пр. Независимости, 4, 220030, Минск, Республика Беларусь).
E-mail: semyonximich@gmail.com

Information about the authors

Oleg I. Duginov – Ph. D. (Physics and Mathematics), Associate Professor of the Department of Discrete Mathematics and Algorithmics, Belarusian State University (4, Nezavisimosti Ave., 220030, Minsk, Republic of Belarus).
E-mail: oduginov@gmail.com

Semyon S. Khimich – Student at Faculty of Applied Mathematics and Computer Science, Belarusian State University (4, Nezavisimosti Ave., 220030, Minsk, Republic of Belarus). E-mail: semyonximich@gmail.com