Информационные процессы, Том 16, № 2, 2016, стр. 223–236.

Линейный алгоритм кратчайшей перестройки графов при разных ценах операций [1]

К. Ю. Горбунов, В. А. Любецкий

Институт проблем передачи информации им. А. А. Харкевича РАН, г. Москва, Россия
Поступила в редколлегию 16.06.2016

Аннотация—Предлагается новый эффективный по времени и памяти алгоритм решения задачи о наиболее экономном (т.е. с наименьшей суммарной ценой) преобразовании любого ориентированного графа, являющегося дизъюнктным объединением цепей и циклов, в любой другой граф этого вида. Доказаны корректность алгоритма (т.е. он всегда выдаёт минимум функционала суммарной цены) и линейная оценка на время и память его работы.

КЛЮЧЕВЫЕ СЛОВА: линейный алгоритм, ориентированный граф, цепь, цикл, преобразование графа, цена операции, комбинаторная оптимизация.

1. Постановка задачи. Приводится полное доказательство корректности линейного алгоритма решения следующей комбинаторно-оптимизационной задачи; до сих пор задача не была решена строго и при общих условиях, которые рассматриваются ниже.

Дан ориентированный граф, связные компоненты которого (без учёта ориентации) — циклы и цепи; рёбра графа помечены натуральными числами без повторений (*именами* рёбер). Задача наиболее экономного преобразования одного такого орграфа в другой на эвристическом уровне активно изучается в последние два десятилетия. Фиксирован список операций, которые преобразуют такие графы друг в друга. Можно рассматривать графы и более общего вида, чем указанный, и любой список операций, но в длительной истории исследований (по разным, прежде всего, прикладным причинам) сформировались указанные ограничения на граф и список операций, который приведён в следующем пункте 2. Каждой операции приписано число, которое называется её *ценой.* В прикладных задачах цены — строго положительные рациональные числа, но теоретически их можно считать натуральными числами. Любой последовательности операций, применяемых друг за другом, начиная с данного графа a, и заканчивая некоторым результирующим графом b, приписывается *суммарная цена*, сумма цен всех операций в этой последовательности.

Формулировка задачи. Пусть даны два графа a и b; найти минимальную по функционалу суммарной цены последовательность операций, которая преобразует a в b. Такую последовательность называют *кратчайшей*, а её цену - *кратчайшей ценой.*

Предполагается, хотя это не доказано, что задача нахождения кратчайшей последовательности или кратчайшей цены для переменных a, b и переменных цен операций является NP-трудной. Она остаётся таковой, если фиксировать произвольные (случайные) цены операций. Поскольку прикладной интерес представляют линейные или, во всяком случае, полиномиальные алгоритмы низкой степени, приходится накладывать какое-то условие на цены; простейший пример такого условия — все цены равны. Задачу с этим ограничением назовём задачей *с равными ценами.*

[1] Работа выполнена за счёт гранта Российского научного фонда (проект №14–50–00150).

В этой задаче известно ограничение и другого типа: в последовательности операций, которая преобразует a в b, включая и сами a и b, присутствует один и тот же постоянный набор имён. Задачу с этим ограничением назовём с *постоянным составом* (имён рёбер). В отсутствии этого ограничения задачу назовём с *переменным составом.*

Краткий обзор предшествующих работ содержится в [1, 2], одно из возможных численных приложений предложенного нами алгоритма описано в [2]. В [1] авторы предложили линейный по времени алгоритм построения кратчайшей последовательности в случае задачи с равными ценами; в этом случае искомая последовательность называется *минимальной.*

2. Список операций и вспомогательный результат. Для преобразования графа a в граф b разрешены следующие шесть операций. *Разрез* вершины, рис. 1a, разбивает цепь на две цепи или размыкает цикл по её/его вершине; обратная операция *склейки* (отождествления) двух свободных краёв разных цепей или одной цепи. *Полуторная переклейка*, рис. 1b, разбивает цепь или цикл и склеивает один из образовавшихся краёв с каким-то свободным краем. *Двойная переклейка*, рис. 1c, разбивает две цепи (или цикла) или одну цепь (цикл) в двух местах и склеивает четыре образовавшихся края по-другому. *Удаление* участка (из нескольких рёбер) цепи или цикла, рис. 1d, если номера рёбер участка не упоминаются в графе b и склейка образовавшихся концов; обратная операция *вставки* участка в цепь или цикл, если номера его рёбер не упоминаются в графе a. Ориентация рёбер не используется в операциях. Первые четыре операции называются *стандартными*, а последние две *дополнительными.*

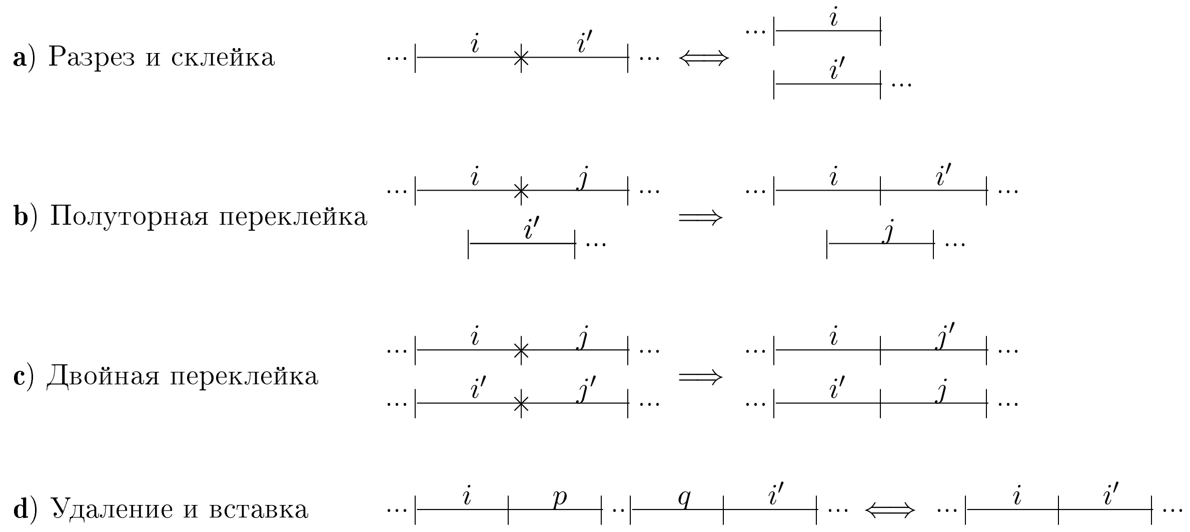

Рис. 1. Операции, которые преобразуют один граф, указанного вида, в другой того же вида.

Напомним схему алгоритма из [1]. Вводится понятие *общего графа* $a + b$ двух орграфов a и b. Его вершины имеют вид n_1 или n_2, для каждого ребра с именем n, представленного в a и b; индекс указывает на начало или конец такого ребра. Эти вершины в $a + b$ называются *обычными.* Кроме того, вершинами в $a + b$ являются *блоки* — максимальные по включению связные участки из рёбер, принадлежащих лишь одному из графов a и b. Эти вершины в $a+b$ называются *особыми* и помечаются множеством имён блока. У общего графа $a + b$ имеются следующие рёбра. *Обычное* ребро соединяет две обычных вершины, если соответствующие им края склеены (отождествлены) в a или в b. *Особое* ребро соединяет обычную вершину с особой, если в a или в b край, соответствующий обычной вершине, склеен с краем блока, соответствующего особой вершине. Такое ребро помечается как a- или b-ребро соответственно. *Петля* в $a+b$ соответствует циклу, который является блоком; иными словами, особая вершина этого блока соединяется с собой. *Висячим* называется ребро общего графа, у которого один конец особый и ни с кем не соединён.

Общий граф состоит из цепей и циклов, которые также называются *компонентами. Размером* компоненты назовём сумму в ней числа обычных рёбер и половины числа особых невисячих рёбер. Дополним определение: размер петли равен 0, размер изолированной особой вершины (не петли), равен -1. В [1] доказано, что исходная задача эквивалентна задаче преобразования (приведения) общего графа к *финальному* виду, который определяется как состоящий из циклов длины 2 без особых вершин и изолированных обычных вершин. Приведение общего графа выполняется операциями, аналогичными перечисленным выше с одним исключением: вставка заменяется удалением особой b-вершины. Легко проверить, что доказательство в [1] эквивалентности этих двух формулировок задачи сохраняется без изменения, если цены всех стандартных операций равны, а цены удаления и вставки любые.

3. Приведение общего графа в случае постоянного состава и разных цен операций. Обозначим цену разреза c_1, склейки c_1', полуторной переклейки $c_{1.5}$, двойной переклейки c_2. Рассмотрим два варианта соотношения цен: $c_2 \leqslant c_1 \leqslant c_1' \leqslant c_{1.5}$ (*циклический* вариант) и $c_1 \leqslant c_1' \leqslant c_{1.5} \leqslant c_2$ (*линейный* вариант).

Будем искать кратчайшую последовательность среди всех минимальных последовательностей, т.е. решать задачу условной оптимизации. Такая последовательность называется *условно кратчайшей.* В рассматриваемом случае общий граф состоит из циклов и цепей, в которых чередуются a- и b-рёбра. В случае постоянного состава качеством $H(a+b)$ общего графа $a+b$ назовём число циклов сложенное с половиной числа чётных цепей в нем. Чётной называется цепь с чётным числом рёбер, а также цепь размера ноль; цепи нечётного размера не учитываются; понятия размера и длины в этом пункте совпадают. Пусть графы a и b имеют по n рёбер.

Лемма 1. *1) Каждая стандартная операция изменяет качество общего графа на* 0 *или* ± 1.

2) Для нефинального графа существует операция, увеличивающая его качество на 1.

3) Граф $a+b$ *финальный, если и только если* $a = b$; *для финального графа* $a+b$ *выполняется* $H(a+b) = n$.

4) Как безусловная, так и условная задачи для a *и* b *эквивалентны соответствующим задачам о приведении общего графа* $a+b$ *к финальному виду.*

5) Существует последовательность операций, преобразующая $a+b$ *к финальному виду, на каждом шаге которой качество увеличивается ровно на* 1; *её длина равна* $k = n - H(a+b)$.

6) Длина минимальной последовательности операций равна k.

7) Минимальными последовательностями для $a+b$ *являются в точности те, у которых каждая операция увеличивает качество общего графа на* 1. *Их длины равны* k.

Доказательство. 1) Последовательно рассмотрим стандартные операции.

Двойная переклейка. Если она применяется к одному циклу, то получается один или два цикла, качество не меняется или увеличивается на 1. Применяется к одной цепи: получается одна цепь той же длины или цикл вместе с цепью той же чётности, что и исходная. Качество не меняется или увеличивается на 1. Применяется к двум циклам: получается один цикл, качество уменьшается на 1. Применяется к циклу и цепи: получается одна цепь той же чётности, что и исходная. Качество уменьшается на 1. Применяется к двум цепям: получаются две цепи той же суммарной длины, что и исходные. Здесь возможны следующие преобразования: две чётные цепи в две чётные, ъ две нечётные в две нечётные (в обоих случаях качество не меняется), две чётные в две нечётные (качество уменьшается на 1), две нечётные в две четные (качество увеличивается на 1), чётная и нечётная в чётная и нечётная (качество не меняется).

Полуторная переклейка. Свободная вершина берется из цепи и является крайней. Применяется к цепи и вершина из неё: получается цепь той же длины (качество не меняется) или цикл вместе с цепью той же чётности (качество увеличивается на 1). Применяется к цепи и вершина из другой цепи: получаются две цепи той же суммарной длины, что исходные (возможны все три варианта изменения качества: те же, что и в случае двойной переклейки). Применяется к циклу и вершина из цепи: получается цепь той же чётности, что исходная (качество уменьшается на 1).

Разрез. Применяется к циклу: получается цепь нечётной длины (качество уменьшается на 1). Применяется к цепи: получаются две цепи суммарной длины на 1 меньшей, чем длина исходной цепи. Возможны переходы: чётная цепь в чётную и нечётную цепи (качество не меняется), нечётная цепь в две нечётные цепи (качество не меняется), нечётная цепь в две чётные цепи (качество увеличивается на 1).

Склейка. Применяется к одной цепи (обязательно нечётной): получается цикл (качество увеличивается на 1). Применяется к двум цепям: получается цепь длины на 1 большей суммы длин исходных цепей. Возможны следующие переходы: чётная и нечётная цепи в чётную цепь (качество не меняется), две нечётные цепи в нечётную цепь (качество не меняется), две чётные цепи в нечётную цепь (качество уменьшается на 1).

5) Последовательность строится последовательным применением п. 2.

Опустим простые доказательства остальных пунктов. □

Для наглядности на странице http://lab6.iitp.ru/-/graph_transformation_algorithms (п. 1) приведена иллюстрация работы наших алгоритмов, которая не существенна для формального понимания доказательств. Далее будем упоминать эти рисунки без упоминания ссылки.

Опишем точный линейный алгоритм приведения к финальному виду в случаях циклического и линейного соотношений цен. Для циклического варианта он состоит из трёх шагов.

Шаг 1. Если имеется цикл длины строго большей двух, двойной переклейкой разбиваем его на два цикла, один из которых имеет длину 2, как показано на рис. 2a.

Шаг 2. Склейкой каждую нечётную цепь замыкаем в цикл, после чего применяем шаг 1, рис. 2b.

Шаг 3. Полуторной переклейкой каждую ненулевую чётную цепь замыкаем в цикл, один край цепи становится нулевой цепью, рис. 2c. Затем применяем шаг 1.

Алгоритм решения условной задачи для линейного варианта состоит также из трёх шагов.

Шаг 1. Тот же, что и в предыдущем алгоритме.

Шаг 2. Разрезом от каждой нечётной цепи отделяем крайнюю вершину, получаем чётную цепь на 1 меньшей длины и нулевую цепь, рис. 3a.

Шаг 3. Полуторной переклейкой каждую ненулевую чётную цепь укорачиваем на 2 ребра и замыкаем два её крайних ребра в цикл, пока в общем графе не останется ненулевых цепей, рис. 3b.

Если ни один шаг не применим, то общий граф уже имеет финальный вид, и к нему применяется пустая последовательность операций. □

Теорема 1. *Указанные линейные алгоритмы точно решают задачу условной оптимизации для циклического и линейного варинатов цен.*

Доказательство. Минимальность полученной последовательности следует из леммы 1. Из неё же следует линейность алгоритма по времени.

Докажем, что полученная последовательность кратчайшая. В общем графе *a-цепью* назовём нечётную цепь, у которой крайние ребра помечены a. Аналогично определяется *b-цепь*. Пусть n_1 — число циклов в общем графе, ℓ_1 — сумма их длин; определим аналогичные величины n_a, ℓ_a для a-цепей, n_b, ℓ_b для b-цепей, и n_2, ℓ_2 для чётных ненулевых цепей. Обозначим $n_3 = n_a + n_b$, $\ell_3 = \ell_a + \ell_b$. Суммарная цена в полученной алгоритмом последовательности для циклического варианта равна

$$c_2[0.5(\ell_1 + \ell_2 + \ell_3) - n_1 - 0.5n_3 - n_2] + c_{1.5}n_2 + c_1 n_a + c_1' n_b,$$

а для линейного варианта равна

$$c_2(0.5\ell_1 - n_1) + 0.5c_{1.5}(\ell_2 + \ell_3 - n_3) + c_1 n_a + c_1' n_b.$$

Эту суммарную цену для общего графа G *обозначим* $c(G)$. Кратчайшую цену для приведения графа G к финальному виду *обозначим* $C(G)$. Индукцией по величине $C(G)$ покажем, что для всех графов G выполняется неравенство $c(G) \leqslant C(G)$. Отсюда $c(G) = C(G)$, что и требуется.

Число вершин в графе G фиксировано, поэтому множество минимальных цен конечно. Индукция идет по естественному порядку в этом множестве цен. Если $C(G) = 0$, то граф G финального вида и $c(G) = 0$.

Индуктивный шаг. Пусть для всех графов G', у которых $C(G') < C(G)$, выполняется неравенство $c(G) \leqslant C(G)$. Докажем его для G. Рассмотрим приводящую последовательность для G. Обозначим o — ее первую операцию, $c(o)$ — ее цену, $o(G)$ — результат ее применения к G. Достаточно проверить неравенство

$$c(o) \geqslant c(G) - c(o(G)).$$

Поскольку операция o увеличивает качество G на 1, после её применения число циклов увеличиться на 1 или число чётных цепей — на 2. Переберём случаи.

1) o — двойная переклейка, применяется к циклу и разбивает его на два цикла, рис. 2a. Тогда n_1 увеличивается на 1, другие величины в формуле для $c(G)$ не меняются. Выполняется равенство $c(o) = c_2 = c(G) - c(o(G))$.

2) o — двойная переклейка, применяется к цепи, вырезает из нее цикл, рис. 4a.

Получается цикл и ненулевая цепь той же чётности, что и первоначальная. Величина n_1 увеличивается на 1, величина ℓ_1 увеличивается настолько, насколько уменьшается сумма $\ell_2+\ell_3$ (пусть на некоторое p). Другие величины в формуле для $c(G)$ не меняются. Следовательно, для циклического варианта выполняется $c_2 = c(G) - c(o(G))$, а для линейного варианта $c(G) - c(o(G)) = c_2 - 0.5c_2 p + 0.5c_{1.5}p \leqslant c_2$.

3) o — двойная переклейка, применяется к двум нечётным цепям: a-цепи и b-цепи, рис. 4b. Получаются две ненулевые чётные цепи той же суммарной длины, что и исходные. Величины n_a и n_b уменьшаются на 1, величина n_3 уменьшается на 2, n_2 увеличивается на 2, ℓ_3 уменьшается настолько, насколько увеличивается ℓ_2. Для циклического варианта имеем $c(G) - c(o(G)) = -c_2 + 2c_2 - 2c_{1.5} + c_1 + c_1' = c_2 + c_1 + c_1' - 2c_{1.5} \leqslant c_2$. Для линейного варианта имеем $c(G) - c(o(G)) = -c_{1.5} + c_1 + c_1' \leqslant c_2$.

4) o — полуторная переклейка, применяется к цепи, отрезает от неё цикл, рис. 2c. Получается цикл и (возможно, нулевая) цепь той же чётности, что и исходная цепь. Величина n_1 увеличивается на 1, величина ℓ_1 увеличивается настолько, насколько уменьшается сумма $\ell_2+\ell_3$ (пусть на некоторое p). Если полученная цепь нулевая, то ещё n_2 уменьшается на 1. Для циклического варианта имеем $c(G) - c(o(G)) = c_2 \leqslant c_{1.5}$ (ненулевой случай) и $c(G) - c(o(G)) = c_{1.5}$

(нулевой случай). Для линейного варианта, учитывая очевидное неравенство $p \geqslant 2$, имеем $c(G) - c(o(G)) = c_2 - 0.5c_2p + 0.5c_{1.5}p \leqslant c_{1.5}$.

5) o — полуторная переклейка, применяется к двум нечётным цепям: a-цепи и b-цепи, рис. 4c. Получаются две чётные цепи той же суммарной длины, что и исходные; одна из них может быть нулевой. Если обе цепи ненулевые, рассуждение аналогично случаю 3. Иначе (теперь величина n_2 увеличивается на 1) для циклического варианта имеем $c(G) - c(o(G)) = -c_2 + c_2 - c_{1.5} + c_1 + c'_1 = c_1 + c'_1 - c_{1.5} \leqslant c_{1.5}$ и аналогично для линейного варианта.

6) o — склейка, применяется к нечётной цепи, замыкает ее в цикл, рис. 2b. Здесь $c(o) = c_1$, если цепь является a-цепью и $c(o) = c'_1$, если цепь является b-цепью. Величина n_1 увеличивается на 1, величина n_3 и одна из величин n_a или n_b уменьшается на 1, ℓ_1 увеличивается на некоторое $p \geqslant 2$, ℓ_3 уменьшается на $p - 1$. Для циклического варианта имеем $c(G)-c(o(G)) = -0.5c_2+c_2-0.5c_2+c(o) = c(o)$. Для линейного варианта имеем $c(G)-c(o(G)) = -0.5c_2p + c_2 + 0.5c_{1.5}(p - 2) + c(o) \leqslant c(o)$.

7) o — разрез, применяется к нечётной цепи, разрезая ее на две чётные, рис. 4a и 4d, при этом в a-цепи разрезается ребро с пометкой a или в b-цепи разрезается ребро с пометкой b. Здесь $c(o) = c_1$, если разрезаемое ребро помечено a, и $c(o) = c'_1$, если оно помечено b. Получаются две чётные цепи на 1 меньшей суммарной длины, чем исходная; одна из них или обе могут быть нулевыми. Если обе цепи ненулевые, величина n_2 увеличивается на 2, величина n_3 и одна из величин n_a или n_b уменьшается на 1, ℓ_2 увеличивается на некоторое p, ℓ_3 уменьшается на $p + 1$. Для циклического варианта имеем: $c(G) - c(o(G)) = 0.5c_2 - 0.5c_2 + 2c_2 - 2c_{1.5} + c(o) \leqslant c(o)$. Если одна или обе полученные цепи нулевые, n_2 увеличивается на 1 или не меняется. Соответственно, в неравенстве двойка заменяется на 1 или на 0, и оно остаётся верным. Для линейного варианта имеем $c(G) - c(o(G)) = -0.5c_{1.5}p + 0.5c_{1.5}p + c(o) = c(o)$. □

4. Приведение общего графа в случае переменного состава и специальных цен операций. Дан общий граф $a+b$ и число ε, $0 \leqslant \varepsilon \leqslant 1$. Разрешены все операции, т.е. стандартные операции, удаления a- и b-вершин (особых вершин с пометкой a или b). Пусть цены стандартных операций и a-удаления равны 1, а цена b-удаления вершины равна $1 + \varepsilon$. Термин *конец*, естественно, относится к концу ребра или к изолированной вершине в общем графе.

Предлагаемый алгоритм компьютерно тестировался в общем случае, если цена b-удаления больше цены всех других операций. Как правило, алгоритм находил ответ, близкий к кратчайшей последовательности. Эта более общая ситуация здесь не рассматривается, но с учётом возможного эвристического использования в описание алгоритма, которое мы сейчас приведём, включены соответствующие пояснения; они не используются в доказательстве, которое приводится ниже.

Описание алгоритма.

Шаг 1. Удалить особые a-петли.

Шаг 2. Вырезать все обычные ребра, не входящие в 2-циклы (т.е. циклы размера 2), замыкая их в финальные 2-циклы двойной (если ребро не крайнее, рис. 5a) или полуторной (если оно крайнее, рис. 5b) переклейками или склейкой (если оно изолированное, рис. 5c). Если цена двойной переклейки не больше цены полуторной, то сначала выполняются всевозможные двойные переклейки, иначе — всевозможные полуторные.

Шаг 3. Напомним и обобщим некоторые определения из [1]. a-Вершиной назовем особую вершину, помеченную a (аналогично для b-вершины). *Нечетной* (*четной*) *цепью* назовём цепь нечётного (чётного) размера. a-*Цепью* называется нечётная цепь, у которой крайние невисячие ребра помечены a, или изолированная b-вершина. Аналогично определяется b-цепь. Оставшимся после шагов 1–2 цепям и циклам (кроме финальных 2-циклов и изолированных обычных вершин) *приписываем тип*: циклу, содержащему a-вершину, но не b-вершину — “a-цикл”; симметрично — “b-цикл”. Циклу, в котором имеются как a-вершины, так и b-вершины, приписываем тип “цикл”. Особой b-петле приписываем тип “петля”. a-Цепи приписываем типы: $1a$, если в ней одно висячее ребро; $2a$, если в ней два таких ребра; $2a'$ — если это изолированная b-вершина; $3a$, если у неё нет висячих рёбер, но имеются как a-, так и b-вершины (тогда размер цепи строго больше 1); $3a'$, если нет ни висячих ребер, ни b-вершин (тогда размер цепи равен 1). b-Цепям тип приписывается аналогично. Отметим, что выделение типов “без штриха” и “со штрихом” связано с тем, что важно выделить цепи, в которых нет b-вершин или a-вершин (в [1] не было такой необходимости, и типы $2a'$, $3a'$ входили, соответственно, в типы $2a$ и $3a$). Чётной цепи приписывается тип: 1, если в ней одно висячее ребро и имеется b-вершина и a-вершина; $1'$, если она состоит из одной обычной вершины и инцидентной ей a-вершины; $1''$, если она состоит из одной обычной вершины и инцидентной ей b-вершины; 2, если в ней два висячих ребра и есть невисячие ребра; $2'$, если в ней два висячих ребра и нет других ребер; 3, если в ней имеется хотя бы одно ребро и нет висячих рёбер. Среди цепей типа 1 выделим цепи типа 1_a (если висячая вершина — a-вершина) и 1_b (если она — b-вершина).

Один пункт на шагах 3–4 может включать несколько последовательных преобразований, которые отделяются друг от друга знаком равенства. В описании одного пункта перед каждым знаком равенства указываются (разделяемые знаком +) типы цепей до выполнения операции, а после равенства — тип цепи, полученной после её выполнения. Изолированные обычные вершины и финальные 2-циклы не указываются. Для краткости приводится описание только *первого равенства*; описания последующих равенств аналогичны. Типом $2a$ называем объединение типов $2a$ и $2a'$, типом $3b$ — типов $3b$ и $3b'$, типом 1_b — типов 1_b и $1''$ и типом 2 — типов 2 и $2'$ (это позволяет сократить число преобразований). Тип 1_c означает отложенный выбор между цепями типов 1_a и 1_b — двумя возможными результатами соответствующей операции. Оба результата храним до шагов 4.15–4.24, на которых в каждой паре результатов выбираем один.

Итак, продолжим описание алгоритма. На примере пункта 3.2 поясним содержания пунктов: последовательно применяем к парам цепей типов $2a$ и $3b$ указанную операцию, получаем цепь типа 1_b, затем аналогично последовательно преобразуем цепи типа $2b$ и $3a$, $2b'$ и $3a$, $2b$ и $3a'$, $2b'$ и $3a'$ в цепи соответственно типов 1_a, 1_a, 1_a и $1'$. Таким образом, последовательно выполняем все пункты.

3.1.$1a + 1b = 1_c$. Расклеим крайнее невисячее ребро (назовём его *внешним*) в одной из двух цепей типа $1a$ или $1b$ и соответствующую особую вершину склеим с крайней особой вершиной другой цепи (полуторная переклейка). Два варианта показаны на рис. 6.

3.2. $2a + 3b = 1_b$, $2b + 3a = 1_a$, $2b' + 3a = 1_a$, $2b + 3a' = 1_a$, $2b' + 3a' = 1'$. В $3b$-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной $2a$-цепи, рис. 7.

3.3. $2 + 3 = 1_c$. В 3-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной 2-цепи. В зависимости от того, какое из двух внешних рёбер расклеивается, получается цепь типа 1_a или 1_b, рис. 8a и 8b.

3.4. $1b + 2a + 3 = 2 + 3 = 1_c$, $1a + 2b + 3 = 2 + 3 = 1_c$, $1a + 2b' + 3 = 2 + 3 = 1_c$. Сначала выполняем $1b + 2a = 2$ (описание ниже); затем $2 + 3 = 1_c$.

3.5. $1a + 3b + 2 = 3 + 2 = 1_c$, $1b + 3a + 2 = 3 + 2 = 1_c$, $1b + 3a' + 2 = 3 + 2 = 1_c$. Сначала $1a + 3b = 3$ (описание ниже); затем $2 + 3 = 1_c$.

3.6. $1a + 2 = 2a, 1b + 2 = 2b$. В $1a$-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной 2-цепи, рис. 9.

3.7. $1a + 3 = 3a$, $1b + 3 = 3b$. В 3-цепи расклеим крайнее b-ребро и особую вершину склеим с крайней особой вершиной $1a$-цепи, рис. 10.

3.8. $1a+1a+2b+3b=2+3=1_c$, $1a+1a+2b'+3b=2+3=1_c$, $1b+1b+2a+3a=2+3=1_c$, $1b+1b+2a+3a'=2+3=1_c$. Сначала выполняем $1a+2b=2$ и $1a+3b=3$; затем $2+3=1_c$.

3.9. $1a+1a+2b=3a+2b=1_a$, $1a+1a+2b'=3a+2b'=1_a$, $1b+1b+2a=3b+2a=1_b$. Сначала $1a+1a=3a$ (описание ниже); затем $2b+3a=1_a$.

3.10. $1a+1a+3b=1a+3=3a$, $1b+1b+3a=1b+3=3b$, $1b+1b+3a'=1b+3=3b$. Сначала $1a+3b=3$; затем $1a+3=3a$.

3.11. $1a+1a=3a$, $1b+1b=3b$. Склеим крайние особые вершины двух $1a$-цепей, рис. 11.

3.12. $1a+2b=2$, $1a+2b'=2$, $1b+2a=2$. В $1a$-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной $2b$-цепи, рис. 12.

3.13. $1a+3b=3$, $1b+3a=3$, $1b+3a'=3$. В $3b$-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной $1a$-цепи, рис. 13.

3.14. $2a+2b+3+3=2+3=1_c$, $2a+2b'+3+3=2+3=1_c$. Сначала $2a+2b+3=2$, затем $2+3=1_c$.

3.15. $3a+3b+2+2=3+2=1_c$, $3a'+3b+2+2=3+2=1_c$. Сначала $3a+3b+2=3$, затем $2+3=1_c$. Упомянутые переходы описаны ниже.

3.16. $2a+3+3=1a+3=3a$, $2b+3+3=1b+3=3b$, $2b'+3+3=1b+3=3b$. Сначала $2a+3=1a$, затем $1a+3=3a$.

3.17. $3b+2+2=1b+2=2b$, $3a+2+2=1a+2=2a$, $3a'+2+2=1a+2=2a$. Сначала $3b+2=1b$; затем $1b+2=2b$. Упомянутые описания приведены в шаге 4.

3.18. $2a+2b+3=2a+1b=2$, $2a+2b'+3=2a+1b=2$. Сначала $2b+3=1b$; затем $1b+2a=2$.

3.19. $3a+3b+2=3a+1b=3$, $3a'+3b+2=3a'+1b=3$. Сначала $3b+2=1b$; затем $1b+3a=3$.

Шаг 4. Предполагается, что цена b-удаления больше цен других операций (напомним, смысл шага 4 — замена дорогой операции b-удаления на более дешёвую). Алгоритм зависит от того, цена двойной переклейки больше цены полуторной или наоборот. В первом случае последовательно применяем пункты 4.1–4.24, во втором случае — пункты 4.1'–4.24' (в случае равенства цен можно выбрать любой из этих случаев). Каждый пункт применяем пока возможно, затем переходим к следующему. Однако доказательства, приводимые после изложения алгоритма, применяются только к случаю равенства цен двойной и полуторной переклеек.

4.1. "петля"+любой тип t с b-вершиной = тип t. Если t не равен $2a'$, двойной переклейкой "вставить" петлю в компоненту типа t с отождествлением b-вершин. Иначе, сделать то же полуторной переклейкой, рис. 14.

4.1'. То же, что и 4.1.

4.2. "цикл"+любой тип t с b-вершиной и a-вершиной = тип t. Вставить цикл (двойной переклейкой, отождествляющей две b-вершины) рядом с b-вершиной из компоненты типа t с той стороны, в которой находится a-вершина; образовавшееся обычное ребро вырезать, рис. 15.

4.2'. То же, что и 4.2.

4.3. $2a+2b=2+1'$. Полуторная переклейка с отрезанием двух вершин $2b$-цепи (крайней a-вершины и соседней обычной вершины) и склейкой образовавшегося края с крайней b-вершиной $2a$-цепи, рис. 16.

4.3'. $2a'+2b=2+1'$.

4.4. $3a+3b=3$. В $3a$-цепи расклеить внешнее ребро и особую вершину склеить с крайней обычной вершиной $3b$-цепи, рис. 17.

4.4'. $3a+3b'=3$.

Ниже рассматривается первый случай, остальные аналогичны.

4.5. $2a+3=1a$, $2b+3=1b$. В 3-цепи расклеить внешнее b-ребро и особую вершину склеить с крайней особой вершиной $2a$-цепи, рис. 18.

4.5'. $2a'+3=1a$.

4.6. $3a+2=1a$, $3b+2=1b$. В $3a$-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 2-цепи, рис. 19.

4.6'. $3a+2'=1a$, $3b'+2=1b$.

4.7. $2a+2a=2a$, $2b+2b=2b$. Склеить крайние особые вершины двух цепей, рис. 20.

4.7'. $2a'+2a=2a$.

4.8. $3a+3a=3a$, $3b+3b=3b$. Две крайние обычные вершины цепей соединить обычным ребром с последующим его вырезанием, рис. 21.

4.8'. $3b'+3b=3b$.

4.9. $1a+2a=1a$, $1b+2b=1b$. Склеить крайние особые вершины двух цепей, рис. 22.

4.9'. $1a+2a'=1a$.

4.10. $1a+3a=1a$, $1b+3b=1b$. Две крайние обычные вершины цепей соединить обычным ребром с последующим его вырезанием, рис. 23.

4.10'. $1b+3b'=1b$.

4.11. $2a+2=2$, $2b+2=2$. Склеить крайние особые вершины двух цепей, рис. 24.

4.11'. $2a'+2=2$, $2a+2'=2$, $2b+2'=2$.

4.12. $3a+3=3$, $3b+3=3$. Две крайние обычные вершины цепей соединить обычным ребром с последующим его вырезанием, рис. 25.

4.12'. $3b'+3=3$.

4.13. $2+2=2+1'$. Полуторная переклейка с отрезанием двух вершин 2-цепи (крайней a-вершины и соседней обычной вершины) и склейкой образовавшегося края с крайней b-вершиной другой 2-цепи, рис. 26.

4.13'. $2'+2=2+1'$.

4.14. $3+3=3$. В 3-цепи расклеить внешнее a-ребро и образовавшийся край этой цепи склеить с b-краем другой 3-цепи, рис. 27.

4.14'. Пустое действие.

4.15. $1_a+1_a=1_a$, $1_b+1_b=1_b$, $1_b+1_c=1_b$ (определяется $c=b$). В 1_a-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной другой 1_a-цепи, рис. 28.

4.15'. $1''+1_b=1_b$, $1''+1_c=1_b$ (определяется $c=b$).

4.16. $1a+1_b=1a$, $1b+1_a=1b$, $1a+1_c=1a$ (определяется $c=b$). В 1_b-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной $1a$-цепи, рис. 29.

4.16'. $1a+1''=1a$.

4.17. $1a+1_a=1a$, $1b+1_b=1b$, $1b+1_c=1b$ (определяется $c=b$). В $1a$-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 1_a-цепи, рис. 30.

4.17'. $1b+1''=1b$.

4.18. $2a+1_b=2a$, $2b+1_a=2b$, $2a+1_c=2a$ (определяется $c=b$). В 1_b-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной $2a$-цепи, рис. 31.

4.18'. $2a'+1_b=2a$, $2a+1''=2a$, $2a'+1_c=2a$ (определяется $c=b$).

4.19. $3a+1_a=3a$, $3b+1_b=3b$, $3b+1_c=3b$ (определяется $c=b$). В $3a$-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 1_a-цепи, рис. 32.

4.19'. $3b'+1_b=3b$, $3b+1''=3b$, $3b'+1_c=3b$ (определяется $c=b$).

4.20. $2+1_a=2$, $2+1_b=2$, $2+1_c=2$ (определяется $c=b$). В 1_a-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 2-цепи, рис. 33.

4.20'. $2'+1_a=2$, $2'+1_b=2$, $2+1''=2$, $2'+1_c=2$ (определяется $c=b$).

4.21. $3+1_a=3$, $3+1_b=3$, $3+1_c=3$ (определяется $c=b$). В 3-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 1_a-цепи, рис. 34.

4.21'. $3+1''=3$.

4.22. $1_a+1_c=1_a$, $1b+1_c=1b$, $1a+1_c=1a$, $2b+1_c=2b$, $3a+1_c=3a$. Везде определяется $c=a$.

4.22'. Пустое действие.

4.23. Для оставшихся цепей типа 1_c определяется $c=b$, и выполним $1_b+1_b=1_b$.

4.23'. Пустое действие.

4.24. Цепи, имеющие невисячее ребро, замыкаем в циклы склейкой (цепи типа $2a$, $2b$, $3a$, $3b$), полуторной переклейкой с отождествлением особых вершин (цепи типа 1_a, 1_b, 1_c, 2) или без отождествления (цепи типа $1a$, $1b$, 3). При замыкании цепи типа 1_c определяем $c=b$. При замыкании цепи типа 2 выбираем вариант с отождествлением двух b-вершин (рис. 35a), из образовавшейся цепи типа $1'$ удаляем a-вершину. Из циклов, получившихся при замыкании цепей типа $3a$ или $3b$, вырезаем обычные рёбра. Затем снова применяем шаг 4.2.

4.24'. То же, что и 4.24.

Шаг 5. Удаляем изолированные особые вершины и петли. Из оставшихся цепей удаляем особые вершины. Из циклов размера большего 2 вырезаем 2-циклы так, чтобы происходило отождествление двух b-вершин (соответственно, в 2-цикл включается a-вершина), рис. 35b. Из 2-циклов удаляем особые вершины.

Конец описания алгоритма. □

Докажем теорему о минимальности суммарной цены последовательности операций, которая получается в алгоритме, т.е. о точности (корректности) алгоритма.

Пусть B' — число b-циклов в графе $a+b$ (напомним, это циклы, содержащие b-вершину, но не содержащие a-вершин). Напомним обозначения из [1]: B — число особых вершин в $a+b$; S — сумма целых частей половин числа рёбер (назовём это число *длиной*) максимальных отрезков (*сегментов*) в $a+b$, которые состоят из обычных рёбер, плюс число нечётных (т.е. нечётной длины) крайних сегментов минус число циклических сегментов. *Крайним* называется сегмент, расположенный с краю цепи, включая и случай целой цепи. *Обычной* называется пара, состоящая из одной из стандартных операций вместе с её аргументом, результат которой не меняет число особых вершин. Далее "обычная" относится к операции, а её аргумент подразумевается заданным. *Дефект* цепи (или цикла) равен минимальному числу обычных операций в последовательности, которая приводит ее (или его) к финальному виду, не считая вырезания обычных рёбер на шаге 2; в последовательности могут встречаться и операции с их аргументами, которые не являются обычными; назовём их *особыми*. В [1] приведена зависимость дефекта от типа компоненты. Обозначим D сумму дефектов компонент графа $a+b$. Обозначим P разность величин D, вычисленных до и после применения шага 3 алгоритма. Заметим, что в любой последовательности операций, финализирующих общий граф, число особых операций равно числу особых вершин в нём, так что экономия числа операций может относиться лишь к обычным операциям. Поскольку все операции на шаге 3 особые, величина P равна числу операций, сэкономленных на шаге 3. Величина ε определена выше. Пусть $C=B+S+D-P+\varepsilon(B'+1)$.

Теорема 2. *Алгоритм строит последовательность операций, у которой суммарная цена равна одному из трёх значений $C-\varepsilon$, C, $C+\varepsilon$. Минимально возможная суммарная цена по-*

следовательности операций, приводящей граф $a+b$ к финальному виду, также равна одному из этих значений. Время работы алгоритма линейное по порядку.

Сначала докажем леммы 2–4, а затем саму теорему.

Лемма 2. *После выполнения каждого действия шага 4 (пп. 4.1–4.24 или 4.1'–4.24') действие не может быть применено снова (за исключением повторного применения пункта 4.2 на шаге 4.24), т.е. в результате дальнейших действий не появляются компоненты, к которым можно применить это действие.*

Доказательство. Перебираем операции шага 4 и для каждой из них показываем, что результат каждой дальнейшей операции не содержит типа, который присутствует в аргументе первой операции, но отсутствует в аргументе второй. Поскольку после выполнения операции отсутствуют компоненты хотя бы одного типа из её аргументов, она не может быть применена снова. □

Лемма 3. *После выполнения шага 4 остаётся 0, 1 или 2 связных компоненты, имеющих b-вершину и не являющихся исходными b-циклами.*

Доказательство. В силу леммы 2 после выполнения шага 4 останутся циклы и цепи без невисячих рёбер, между которыми невозможно ни одной из указанных на шаге 4 операций. Новые b-циклы могут возникать из цепей типа $3b'$; в силу пунктов 4.8–4.8' и леммы 2 возникнет не более одного нового b-цикла. Таким образом, указанные в формулировке леммы компоненты могут иметь типы $2a'$, $1''$, $2'$, цикл, b-цикл. Таким же способом, как для b-цикла, легко показать, что и компонент каждого из первых трёх типов останется не более одной. Осталось показать, что не может остаться никаких трёх типов из пяти. Предположим, что это не так. Если оставшиеся три типа не содержат b-цикла, между двумя из них (не являющимися циклом) возможна соответствующая операция, что противоречит лемме 2. Иначе, такая операция возможна между компонентой типа $3b'$, из которой возник b-цикл, и той из трёх компонент, которая не цикл. Снова получили противоречие. □

Лемма 4. *Число обычных операций в алгоритме равно $S + D - P$.*

Доказательство. Напомним [1], что минимальное число обычных операций, требуемых для приведения компоненты (после шага 2) к финальному виду без использования других компонент, равно её дефекту. Настоящий алгоритм отличается от описанного в [1] наличием шага 4. Любая операция шага 4 либо особая и не меняющая дефект результата по сравнению с суммарным дефектом аргументов, либо обычная и уменьшающая его на 1. Поэтому обычных операций в алгоритме столько же, сколько и раньше, т.е. $S + D - P$. □

Доказательство Теоремы 2. На шаге 5 для каждой компоненты, имеющей b-вершины, применяется ровно одна операция удаления b-вершины. По лемме 3 общее число таких операций равно $B'+n$, где n равно 0, 1 или 2. Всего особых операций B. В силу леммы 4 суммарная цена операций алгоритма равна $(1+\varepsilon)(B'+n)+(B-B'-n)+(S+D-P) = B+S+D-P+\varepsilon(B'+n)$, откуда следует первое утверждение теоремы. Линейность времени алгоритма и используемой памяти очевидна.

Второе утверждение теоремы докажем индукцией по минимальной суммарной цене M операций, приводящих общий граф к финальному виду; имеется лишь конечное число возможных значений M на любом ограниченном отрезке, которые рассматриваем по их возрастанию.

Базис индукции очевиден, опишем индуктивный шаг. Достаточно для любой операции o, применённой к произвольному общему графу G, проверить, что цена $c(o)$ операции o не меньше

$C(G) - C(o(G))$, где $C(G)$ — величина C, определённая в формулировке теоремы 2. Если o не меняет величину B', то в $C(o(G))$ по сравнению с $C(G)$ может измениться лишь величина $t = B + S + D - P$ и рассуждение такое же, как в [1], теорема 5, обосновывает неравенство. Если o увеличивает величину B', то учитывая, что t не может уменьшаться больше, чем на 1, получаем $C(G)-C(o(G)) \leqslant 1-\varepsilon \leqslant c(o)$. Таким образом, достаточно рассмотреть случай, когда o уменьшает величину B'. Это рассмотрение состоит из следующих пунктов 1–5.

1. o — операция удаления особой вершины. Если o уменьшает величину B' (очевидно, на 1), то o удаляет b-вершину и $C(G) - C(o(G)) \leqslant 1 + \varepsilon = c(o)$.

2. o — операция склейки. Она не может уменьшать величину B'.

3. o — операция разреза. Следует рассмотреть случай уменьшения B' на 1, когда разрезается b-цикл. Поскольку в b-цикле между b-вершинами нечётное количество обычных ребер, в результате разреза возникает либо цепь типа $3b$, либо цепь типа $2a$ (согласно классификации из [1, лемма 6] для компонент общего вида). В первом случае S не меняется, D увеличивается на 1, во втором наоборот. В обоих случаях B не меняется, P либо не меняется, либо увеличивается на 1. Таким образом, $C(G) - C(o(G)) \leqslant \varepsilon < c(o)$.

4. o — операция полуторной переклейки. Следует рассмотреть случай уменьшения B' на 1, когда разрезается b-цикл и получившаяся цепь склеивается с другой цепью. В предыдущем пункте показано, что при разрезе b-цикла величина t не уменьшается. Поэтому достаточно показать, что если при разрезе b-цикла t не меняется (т.е. P увеличивается на 1), то она не уменьшается и при склейке двух цепей.

Пусть в результате разреза b-цикла получается цепь типа $2a$. В таблице перечислены возможные типы склейки цепей, первая из которых — цепь типа $2a$, тип края второй цепи указан в заголовке столбца (обозначения соответствуют [1]).

Таблица. Результаты склейки краёв двух цепей.

$0a$	0	$1a$	$1a'$	$2a$	$3a$	1	$1'$	2	3
(0)	(0,+1)	(0,−1)	(0,+1,−1)	(0,−1)	(0,+1,−1)	(0,−1)	(0,+1,−1)	(0,−1)	(0,+1,−1)
[0]	[0,−1]	[−1,0]	[−1,−2,0]	[−1,0]	[0,−1,+1]	[−1,0]	[0,−1,+1]	[−1,0]	[−1,−2,0]
$2a$	1	$1a$	$2a$	$2a$	$1a$	1	2	2	1

Перебирая клетки таблицы, смотрим, какому изменению величины P соответствует уменьшение t. Например, в столбце $1a$ (цепь x имеет тип $1a$) оно соответствует сохранению P. Это невозможно, так как в результате склейки снова получается цепь типа $1a$ и, если при появлении цепи типа $2a$ величина P увеличилась, при ее исчезновении она должна уменьшиться. Аналогично, рассматривая столбец $1a'$, видим, что в результате склейки цепей типа $2a$ и $1a$ получается цепь типа $2a$, что эквивалентно исчезновению цепи типа $1a$. Но если величина P уменьшается при исчезновении цепи типа $2a$, то тем более она уменьшится при исчезновении цепи типа $1a$. Столбцы $2a$, 1, 2, 3 рассматриваются аналогично. Для столбца типа $3a$ имеем склейку $2a+3a = 1a$. Она не может увеличить величину P, поскольку тогда замена цепи типа $3a$ на цепь типа $1a$ (в результате операции o) увеличивала бы P на 2, что невозможно. Для столбца типа $1'$ имеем склейку $2a + 1 = 2$. Она не может увеличить величину P, поскольку тогда появление цепи типа 2 (в результате операции o) увеличивало бы P на 2, что невозможно. Случай, когда в результате разреза b-цикла получается цепь типа $3b$, рассматривается аналогично.

5. o — операция двойной переклейки. Рассмотрим возможные случаи, когда величина B' уменьшается.

5.1. Операция o применяется к двум b-циклам и отождествляет их в один b-цикл. Поскольку в b-циклах между b-вершинами нечётное количество обычных рёбер (достаточно, что в одном

b-цикле), в результате величина $B + S$ не уменьшится. Величины D и P не меняются, B' уменьшается на 1. Требуемый результат очевиден.

5.2. Операция o применяется к двум циклам, один из которых является b-циклом, а другой содержит оба типа особых вершин. Рассуждение такое же, как в предыдущем пункте.

5.3. Операция o применяется к b-циклу и цепи, вставляя цикл в цепь. Нужно показать, что величина $t = B + S + D - P$ при этом не уменьшается.

5.3.1. Если в цепи с обеих сторон от разреза имеются особые вершины, тип цепи не меняется, следовательно, не меняются величины D и P. Поскольку сегмент (из обычных рёбер), примыкающий к разрезу в b-цикле, нечётный, величина $B+S$ не уменьшается, что и требуется.

5.3.2. Пусть в цепи имеются особые вершины лишь с одной стороны от разреза. Если чётность крайнего сегмента цепи не меняется, то не меняется тип цепи и проходит рассуждение из предыдущего пункта. Если чётный крайний сегмент заменяется нечётным, то (учитываем, что в b-цикле сегмент нечётный) величина B не меняется, величина S увеличивается на 1. При этом возможны следующие варианты изменения типа цепи: $3a \longrightarrow 1a$, $1 \longrightarrow 2$, $1a \longrightarrow 2a$, $3 \longrightarrow 1$. При первых двух вариантах величина D не меняется, величина P либо не меняется, либо увеличивается на 1. При последних двух вариантах величина D уменьшается на 1, величина P либо не меняется, либо уменьшается на 1. Таким образом, величина t не уменьшается. Если нечётный крайний сегмент заменяется чётным, то величины B и S, очевидно, не меняются. При этом возможны следующие варианты изменения типа цепи: $1a \longrightarrow 3a$, $2 \longrightarrow 1$, $2a \longrightarrow 1a$, $1 \longrightarrow 3$. При первых двух вариантах величина D не меняется, величина P либо не меняется, либо уменьшается на 1. При последних двух вариантах величина D увеличивается на 1, величина P либо не меняется, либо увеличивается на 1. Таким образом, величина t не уменьшается.

5.3.3. В цепи нет особых вершин. Удобнее рассмотреть обратную к o операцию o' вырезания цикла из цепи в случае, когда все особые вершины расположены между разрезами. Достаточно доказать, что операция o' не может изменить величину t более, чем на 1, а если не происходит отождествления двух особых вершин и все сегменты в полученном цикле нечётные, то t не увеличивается. Рассмотрим два случая.

1) Две особые вершины отождествляются, т.е. величина B уменьшается на 1. Тогда два крайних сегмента цепи отождествляются с появлением дополнительного обычного ребра. Если оба сегмента чётные, величина S увеличивается на 1. В этом случае цепь имеет тип $3a$ или $3b$, так что в результате её исчезновения величина D уменьшается на 1, величина P либо не меняется, либо уменьшается на 1. Если один или оба сегмента нечётные, величина S не меняется. В этом случае цепь имеет тип $2a$, $2b$ или 1, так что в результате её исчезновения величина D не меняется, величина P либо не меняется, либо уменьшается на 1. Таким образом, во всех случаях величина t меняется не более чем на 1.

2) Не происходит отождествления двух особых вершин, т.е. величина B не меняется. Тогда два крайних сегмента цепи переходят в два сегмента той же суммарной длины — один сегмент в цепи (крайний), другой в цикле. Рассмотрим возможные случаи.

2.1. Пара чётных сегментов переходит в пару чётных или в пару нёчетных. Тогда величина S не меняется. Цепь имела тип $3a$, $3b$ или 3 и в результате её исчезновения величина D уменьшается на 1, величина P либо не меняется, либо уменьшается на 1. Таким образом, величина t либо не меняется, либо уменьшается на 1.

2.2. Пара нечётных сегментов переходит в пару чётных или в пару нечётных. Тогда величина S уменьшается на 1. Цепь имела тип $2a$, $2b$ или 2 и в результате её исчезновения величина D не меняется, величина P либо не меняется, либо уменьшается на 1. Таким образом, величина t либо не меняется, либо уменьшается на 1.

2.3. Пара сегментов разной чётности переходит в пару сегментов разной чётности, где чётный сегмент находится в цикле. Тогда величина S не меняется. Поскольку в цикле образовался чётный сегмент, крайние особые вершины цепи имели разные пометки. Тогда цепь имела тип $1a$ или $1b$ и в результате её исчезновения величина D уменьшается на 1, величина P либо не меняется, либо уменьшается на 1, либо на 2. Таким образом, величина t меняется не более чем на 1.

2.4. Пара сегментов разной чётности переходит в пару сегментов разной чётности, где нечётный сегмент находится в цикле. Тогда величина S уменьшается на 1. Поскольку в цикле образовался нечётный сегмент, крайние особые вершины цепи имели одинаковые пометки. Тогда цепь имела тип 1 и в результате её исчезновения величина D не изменилась, величина P не меняется. Таким образом, величина t уменьшается на 1. $\square$

СПИСОК ЛИТЕРАТУРЫ

1. Горбунов К. Ю., Любецкий В. А. Линейный алгоритм минимальной перестройки структур. *Проблемы передачи информации*, 2016, в печати. (*Problems of Information Transmission*, 2016, in press.)

2. Lyubetsky V. A., Gershgorin R. A., Seliverstov A. V., Gorbunov K. Yu. Algorithms for reconstruction of chromosomal structures. *BMC Bioinformatics*, 2016, 17:40.

A linear algorithm of the shortest transformation of graphs under different operation costs

Gorbunov K. Yu., Lyubetsky V. A.

A novel efficient algorithm is proposed for the task of transformation with the minimal cost of one marked oriented graph into another oriented graph. Both graphs are disjoint unions of chains and cycles.

KEYWORDS: linear algorithm, oriented graph, chain, cycle, graph transformation, operation cost, combinatorial optimization.