УДК 575.852

Горбунов К.Ю., Любецкий В.А.

Институт проблем передачи информации им. А.А. Харкевича РАН, г. Москва, Россия

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ПРЕОБРАЗОВАНИЯ ХРОМОСОМНЫХ СТРУКТУР: УСЛОВИЯ АБСОЛЮТНОЙ ТОЧНОСТИ

АННОТАЦИЯ

В статье излагается модификация ранее разработанного авторами алгоритма преобразования одной хромосомной структуры в другую. Доказываются достаточные условия его абсолютной точности.

КЛЮЧЕВЫЕ СЛОВА

Хромосомная структура; хромосомная перестройка; линейный алгоритм; точный алгоритм; принцип парсимонии; цена операции; комбинаторная оптимизация.

Gorbunov K.Yu., Lyubetsky V.A.

Institute for Information Transmission Problems of the Russian Academy of Sciences, Moscow, Russia

A MODIFIED ALGORITHM FOR TRANSFORMATION OF CHROMOSOMAL STRUCTURES: A CONDITION OF ABSOLUTE EXACTNESS

ABSTRACT

In the article the modification of an algorithm for transformation of one chromosomal structure into another one is presented. The algorithm was developed by the authors earlier; for its modification a sufficient condition of absolute exactness has been proved.

KEYWORDS

Chromosomal structure; chromosomal rearrangement; linear algorithm; exact algorithm; parsimony principle; operation cost; combinatorial optimization.

Определения и постановка задачи

Гены располагаются в геноме в линейных и кольцевых хромосомах, каждый ген имеет направление. В модели хромосомной структуры нуклеотидный состав не учитывается, и считается, что все гены на хромосоме расположены друг за другом, без пересечения. Итак, в модели *хромосомная структура* представляется ориентированным графом, состоящим из цепей и циклов, включая петли (кольцевые хромосомы из одного гена); каждое ребро графа представляет ген и ему приписывается *имя* этого гена, которое в модели обозначается обычно числом. В этом смысле ребро с приписанным ему именем называется здесь *геном*, а цепи и циклы – *хромосомами*.

На рис. 1 в [1] слева приведены две хромосомные структуры *a* и *b* с равным генным составом, без паралогов, в обоих представлены гены с именами от 1 до 11.

Рассматриваются следующие операции, они называются *стандартными*, над любой структурой; рисунки, иллюстрирующие операции, приведены в первом дополнительном материале к [2]. Операции применяются к одной хромосоме или к паре хромосом.

(1) *Двойная переклейка.* Разрезать две вершины, имеющиеся в графе и по-новому отождествить (говорим: склеить) четыре образовавшихся края.

(2) *Полуторная переклейка.* Разрезать вершину и по-новому отождествить (склеить) один образовавшийся край с каким-то свободным краем в графе.

(3–4) *Одинарные переклейки*: *разрез* и *склейка*. Разрезать вершину или, наоборот, отождествить (склеить) два свободных края.

Все операции обратимы: двойная и полуторная переклейки обратны каждая к себе, разрез и склейка взаимно обратны.

Каждой операции приписывается её цена – положительное число.

Рассматривается **задача**: преобразовать одну данную структуру *a* в другую данную структуру *b* последовательностью операций минимальной суммарной цены. Искомую последовательность называем *кратчайшей*.

Для двух структур *a* и *b* полезно введённое нами понятие *общего графа a+b*; для удобства

читателя определим его сначала для равных составов, а затем для общего случая. Это – неориентированный граф без петель, вершины которого – имена краёв всех генов; например, начало гена 3 обозначается 3_1, конец – 3_2. Ребро общего графа соединяет две вершины, если соответствующие им края отождествлены (вместо этого говорят: склеены) в *a* или в *b*; оно помечается соответственно как *a*- или *b*-ребро. Если они склеены в обеих структурах, то соединяются двумя рёбрами, одно помечено *a* и другое *b.* Например, общий граф структур, показанных слева на рис. 1 в [1], показан справа на том же рисунке. Таким образом, общий граф *a+b* несёт информацию о склейках вершин как в *a*, так и в *b*. Легко видеть, что *a+b* всегда состоит из цепей и циклов (в том числе, изолированных вершин и циклов длины 2), в которых *a*-рёбра и *b*-рёбра чередуются. Эти цепи и циклы будем называть *компонентами*.

Граф *c+c* состоит из циклов длины 2 и изолированных вершин. Граф такого вида назовём *финальным* (или: *финального вида*). Легко видеть, что наша задача эквивалентна задаче приведения графа *a+b* к финальному виду следующими *операциями над общим графом*, которые иллюстрируются в первом дополнительном материале к [2].

Двойная переклейка: удаление двух одинаково помеченных рёбер и соединение четырёх образовавшихся концов двумя новыми неинцидентными рёбрами с той же пометкой.

Полуторная переклейка: удаление ребра и соединение одного из его концов ребром с той же пометкой с вершиной, не инцидентной ребру с этой пометкой.

Разрез: удаление любого ребра.

Склейка: добавление ребра (скажем, с пометкой *a*) между вершинами, каждая из которых не инцидентна ребру с пометкой *a*.

Эти четыре операции назовём *стандартными.*

Рассмотрим случай *неравного генного состава*, т.е. множества имён генов в структурах *a* и *b* могут не совпадать (но имена в структуре не повторяются). Ген, который представлен в *a* и в *b* назовём *общим*; ген, представленный лишь в одной из структур – *особым*: соответственно, имеются *a*- и *b*-особые гены.

В случае неравного состава, кроме четырёх указанных операций разрешаются ещё две *дополнительные* операции над хромосомой – удаление и вставка, их рисунки приведены в первом дополнительном материале к [2]. *Удаление*: удалить из хромосомы связный отрезок из *a*-особых генов. Отрезок может удаляться из кольцевой или линейной хромосомы, а также, если он сам – хромосома. Если у удаляемого отрезка имеются два соседних гена, их края склеиваются между собой. *Вставка*: вставить в хромосому связный отрезок из *b*-особых генов. Отрезок может вставляться в любую хромосому, а также – в виде новой линейной или кольцевой хромосомы.

В случае неравного состава определение *общего графа a+b* изменяется следующим образом. Он содержит *обычные* вершины – края общих генов вида k_1 и k_2, и *особые* вершины – максимальные по включению связные участки из *a*-особых или из *b*-особых генов. Последние будем называть *блоками*. Блок принадлежит одной из структур, и соответствующая ему особая вершина помечается как *a*- или *b*-вершина, ей также *приписывается* множество (точнее, последовательность) генов, составляющих блок. Общий граф содержит следующие рёбра. *Обычное* ребро соединяет две обычные вершины, если соответствующие им края отождествлены (склеены) в *a* или в *b*; *особое* ребро соединяет обычную вершину с особой, если в *a* или в *b* край, соответствующий обычной вершине, отождествлён (склеен) с краем блока, соответствующего особой вершине. Такое ребро помечается как *a*- или *b*-ребро. Здесь также возможны двойные обычные рёбра. *Петля* в *a+b* соответствует циклу, который является блоком; иными словами, особая вершина этого блока соединяется с собой. *Висячим* называется особое ребро, инцидентное особой вершине степени 1.

Как и прежде, общий граф неориентированный и состоит из связных компонент – цепей и циклов. Невисячие особые рёбра присутствуют в нём парами – рёбра, инцидентные одной особой вершине; такую пару удобно считать за одно двойное ребро; с этой оговоркой сохраняется чередование *a*- и *b*-рёбер. Поэтому *размером компоненты* назовём сумму в ней числа обычных рёбер с половиной числа особых невисячих рёбер. Для изолированных обычных вершин и петель считаем размер равным 0, для изолированных особых вершин (не петель) – равным "минус 1". Общий граф называется *финальным* (*финального вида*), если каждая его компонента – изолированная обычная вершина или цикл без особых рёбер размера (в данном случае, то же самое – длины) 2, одно ребро из *a* и другое из *b*.

В [2] на рис. 1 приведён пример двух структур с неравным генным составом, а на рис. 2 показан их общий граф.

Над общим графом разрешаются четыре *стандартные* операции, которые уточняются следующим образом [2–4]. *Двойная переклейка*: удаление двух одинаково помеченных рёбер общего графа и соединение четырёх образовавшихся концов двумя новыми неинцидентными рёбрами с

той же пометкой. Если при этом образуется ребро с особыми концами (оба относятся к *a* или оба к *b*), то оно заменяется одной особой вершиной, которой приписана конкатенация последовательностей двух исходных особых вершин. *Полуторная переклейка*: удаление ребра общего графа и соединение ребром с той же пометкой одного из его концов с обычной вершиной, не инцидентной ребру с этой пометкой, или с особой вершиной степени не больше 1 с той же пометкой (с возможным последующим отождествлением двух особых вершин). *Склейка*: добавление ребра (скажем, с пометкой *a*) между вершинами, каждая из которых является или обычной, не инцидентной ребру с пометкой *a* или особой степени не больше 1 с той же пометкой (с возможным последующим отождествлением двух особых вершин). *Разрез*: удаление любого ребра.

Кроме того, вводится только одна *дополнительная* операция: *удаление* особой вершины (блока). А именно, если эта вершина степени 2, то она удаляется и инцидентные ей рёбра сливаются в одно ребро, на которое переносится пометка вершины, показано на рисунке из первого дополнительного материала к [2]; если вершина степени 1, то она удаляется вместе с инцидентным ей ребром; если вершина степени 0 или с петлёй, то вершина и петля удаляются.

В [2–3] мы свели задачу о преобразовании указанными шестью операциями одной хромосомной структуры в другую уже при неравных составах к задаче приведения их общего графа к финальному виду этими пятью операциями (называем такое приведение *финализацией* графа). Сведение произведено для случая, когда цены всех стандартных операций одинаковы, а цены операций удаления и вставки любые. Отметим: на общем графе удалению участка хромосомы соответствует удаление (особой) *a*-вершины, вставке участка хромосомы – удаление (особой) *b*-вершины. Таким образом, *цена* финализации – сумма цен операций с оговоркой, что удаление *b*-вершины имеет цену вставки, разрез *b*-ребра имеет цену склейки, и склейка *b*-ребром имеет цену разреза.

Здесь мы рассмотрим случай, когда все цены, кроме операции вставки, одинаковы (скажем, равны 1), а цена вставки больше 1, но не превышает 2, т.е. равна $1+\varepsilon$, где $0\le\varepsilon\le1$. Это соотношение цен можно назвать *нестационарным*, предполагая, что при нём идёт уменьшение числа генов в геноме. В [4] мы описали алгоритм, выдающий решение, цена которого отличается от цены оптимального решения не более, чем на ε. Здесь мы опишем уточнение этого алгоритма, позволяющее доказать два достаточных условия его абсолютной точности.

Алгоритм финализации общего графа

Итак, дан общий граф $a+b$ и число ε, $0\le\varepsilon\le1$. Пусть цены стандартных операций и удаления *a*-вершины равны 1, а цена удаления *b*-вершины равна $1+\varepsilon$. В описываемый далее алгоритм мы включили некоторые *эвристические усовершенствования*, рассчитанные на *эвристическое* его использование при неравных ценах стандартных операций.

Шаги 1 и 2 те же, что и в [4], т.е. удаление *a*-петель и вырезание обычных рёбер.

Для описания дальнейших шагов определим типы компонент общего графа, построенного после выполнения шагов 1–2. Их обобщения на компоненты исходного общего графа однозначно определяются по правилу: компонента имеет тип *T*, если после выполнения шага 2 (т.е. вырезания из неё обычных рёбер) она превращается в компоненту типа *T* (см. лемму 6 из [3]). Исключением является случай, когда компонента в исходном графе не содержит особых рёбер, в этом случае припишем ей тип 0.

Нечетной (*четной*) *цепью* назовём цепь нечетного (четного) размера. *a-Цепью* называется нечётная цепь, у которой крайние невисячие ребра помечены *a*, или изолированная *b*-вершина. Аналогично определяется *b-цепь*. Цепям (кроме изолированных обычных вершин) припишем следующие *типы*. *a*-Цепи приписываем типы: 1*a*, если в ней одно висячее ребро; 2*a*, если в ней два таких ребра или если это изолированная *b*-вершина; 3*a*, если у неё нет висячих рёбер. *b*-Цепям тип приписывается аналогично. Чётной цепи приписывается тип: 1, если в ней одно висячее ребро и имеется *b*-вершина и *a*-вершина; 2, если в ней два висячих ребра; 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).

Циклу, содержащему *a*-вершину, но не *b*-вершину, припишем тип «*a*-цикл»; симметрично – «*b*-цикл». Циклу, в котором имеются как *a*-вершины, так и *b*-вершины, приписываем тип «(*a*,*b*)-цикл». Петле с *b*-вершиной припишем тип «*b*-петля». Среди цепей: в типе 2*a* выделяем подтип 2*a′* – если это изолированная *b*-вершина (обобщение на исходный граф: нет *a*-вершин) и 2*a** для остальных цепей, аналогично для типа 2*b*. В типе 3*a* выделяем подтип 3*a′* – если это цепь размера 1 (обобщение: нет *b*-вершин) и 3*a** для остальных цепей, аналогично для типа 3*b*. В типе 1_a выделяем подтип $1'_a$ – если это цепь размера 0, т.е. если она состоит из одной обычной вершины и инцидентной ей *a*-вершины (обобщение: нет *b*-вершин) и 1^*_a для остальных цепей, аналогично для

типа 1_b. В типе 2 выделяем подтип 2′ – если это цепь размера 0, т.е. в ней два висячих ребра и нет других рёбер (обобщение: с одной стороны, все a-вершины, с другой – все b-вершины) и 2* для остальных цепей. Цепь, содержащую a- и b-вершины назовём (a,b)-цепью; если цепь содержит только вершины одного типа, назовём её, соответственно, $(a\backslash b)$-цепью или $(b\backslash a)$-цепью. Компоненту, содержащую b-вершину, но не являющуюся b-циклом, назовём *ценной*.

Шаг 3. Последовательно выполняем следующие операции между компонентами указанных типов: каждая из них выполняется, пока возможно. Описание даём для первой операции, для других оно аналогично (кроме пункта 3.0). На рисунках маленькие кружочки означают обычные вершины, большие – особые. Перечислим усовершенствования, внесённые здесь по сравнению с описанием в [4]. Они связаны с тем, что мы желаем максимально избавиться от цепей типов $2a'$, $3b'$ и $1'_b$, (назовём эти цепи *проблемными*) поскольку эти цепи не взаимодействуют с (a,b)-циклами, что затрудняет слияние b-вершин.

Вводится предварительный шаг 3.0, на котором проводятся взаимодействия $1'_b+2a'=2a'$ и $1'_b+3b'=3b'$ (это частный вид взаимодействий 4.17 и 4.18 ниже). Это позволяет уменьшить число цепей типа $1'_b$. На шагах 3.1, 3.3, 3.4, 3.5, 3.8, 3.14, 3.15 выбирается $c=b$. Это связано с тем, что цепь типа 1^*_b взаимодействует с проблемными цепями, а цепь типа 1^*_a – не взаимодействует. На шаге 3.2 взаимодействие $2a+3b=1_b$ разбивается: сначала $2a'+3b^*=1^*_b$ и $2a^*+3b'=1^*_b$, затем $2a^*+3b^*=1^*_b$ и $2a'+3b'=1_b$. Это связано с желанием максимально избавиться от проблемных цепей. Аналогичное разбиение проводится также на шагах 3.4, 3.5, 3.8, 3.9, 3.10, 3.12–3.19.

3.0. $1'_b+2a'=2a'$, $1'_b+3b'=3b'$. В $1'_b$ -цепи расклеить ребро и особую вершину склеить с особой вершиной $2a'$-цепи, рис. 1a. В $3b'$-цепи расклеить любое ребро и особую вершину склеить с особой вершиной $1'_b$-цепи, рис. 1b.

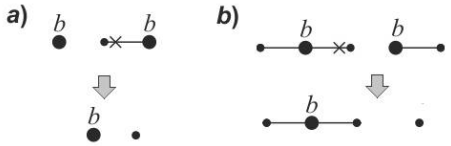

Рис.1. Шаг 3.0 алгоритма

3.1. $1a+1b=1_b$. Расклеим крайнее невисячее ребро (назовём его *внешним*) в цепи типа $1a$ и соответствующую особую вершину склеим с крайней особой вершиной другой цепи (полуторная переклейка), рис. 2.

Рис.2. Шаг 3.1 алгоритма

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

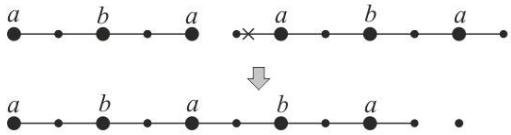

Рис.3. Шаг 3.2 алгоритма

3.3. $2+3=1_b$. В 3-цепи расклеим внешнее a-ребро и особую вершину склеим с крайней особой вершиной 2-цепи, рис. 4.

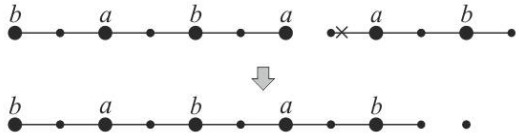

Рис.4. Шаг 3.3 алгоритма

3.4. $1a+2b+3=2+3=1_b$, $1b+2a'+3=2+3=1_b$, $1b+2a^*+3=2+3=1_b$. Сначала выполняем $1a+2b=2$ (описание ниже, шаг 3.12), затем $2+3=1_b$.

3.5. $1b+3a+2=3+2=1_b$, $1a+3b'+2=3+2=1_b$, $1a+3b^*+2=3+2=1_b$. Сначала $1b+3a=3$ (описание ниже, шаг 3.13), затем $2+3=1_b$.

3.6. $1a+2=2a$, $1b+2=2b$. В $1a$-цепи расклеим внешнее ребро и особую вершину склеим с крайней a-вершиной 2-цепи, рис. 5

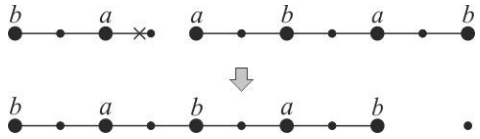

Рис.5. Шаг 3.6 алгоритма

3.7. 1a+3=3a, 1b+3=3b. В 3-цепи расклеим крайнее b-ребро и особую вершину склеим с крайней b-вершиной 1a-цепи, рис. 6.

Рис.6. Шаг 3.7 алгоритма

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

3.9. 1a+1a+2b=3a+2b=1$_a$, 1b+1b+2a'=3b+2a=1$_b$, 1b+1b+2a*=3b+2a=1$_b$. Сначала 1a+1a=3a (описание ниже, пункт 3.11), затем 2b+3a=1$_a$.

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

3.11. 1a+1a=3a, 1b+1b=3b. Склеим крайние b-вершины двух 1a-цепей, рис. 7.

Рис.7. Шаг 3.11 алгоритма

3.12. 1a+2b=2, 1b+2a'=2, 1b+2a*=2. В 1a-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой a-вершиной 2b-цепи, рис. 8.

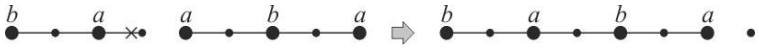

Рис.8. Шаг 3.12 алгоритма

3.13. 1b+3a=3, 1a+3b'=3, 1a+3b*=3. В 3a-цепи расклеим внешнее ребро и особую вершину склеим с крайней a-вершиной 1b-цепи, рис. 9.

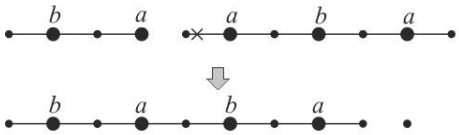

Рис.9. Шаг 3.13 алгоритма

3.14. 2a'+2b+3+3=2+3=1$_b$, 2a*+2b+3+3=2+3=1$_b$. Сначала 2a'+2b+3=2 (описание ниже, шаг 3.18), затем 2+3=1$_b$.

3.15. 3a+3b'+2+2=3+2=1$_b$, 3a+3b*+2+2=3+2=1$_b$. Сначала 3a+3b'+2=3 (описание ниже, шаг 3.19), затем 2+3=1$_b$.

В описании шагов 3.16–3.18 используются вспомогательные взаимодействия 2a+3=1a и 3b+2=1b, которые сами по себе не присутствуют в алгоритме. Их описания ниже.

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

3.17. 2a'+3+3=1a+3=3a, 2a*+3+3=1a+3=3a, 2b+3+3=1b+3=3b. Сначала 2a'+3=1a, затем 1a+3=3a.

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

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

Вспомогательное взаимодействие 2a+3=1a. В 3-цепи расклеить внешнее b-ребро и особую вершину склеить с крайней b-вершиной 2a-цепи, рис. 10.

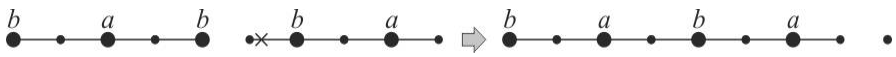

Рис.10. Взаимодействие 2a+3=1a.

Вспомогательное взаимодействие 3b+2=1b. В 3b-цепи расклеить внешнее ребро и особую

вершину склеить с крайней b-вершиной 2-цепи, рис. 11.

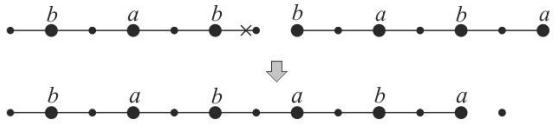

Рис.11. Взаимодействие 3b+2=1b.

Шаг 4. Здесь производятся взаимодействия, которые, в отличие от взаимодействий шага 3, не уменьшают общее число операций (точнее, сохраняют его), но позволяют заменить "дорогую" операцию удаления b-вершины на другую, более дешёвую операцию. Отметим: неценные компоненты во взаимодействиях на шаге 4 не участвуют, за исключением двух последних пунктов 4.23–4.24.

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 вариантов). Смысл этого разделения поясним на примере пунктов 4.2 и 4.2′. Если цепь типа $2a$ в пункте 4.2 имеет тип $2a^*$, применять взаимодействие пункта 4.2 необязательно (разве что при дешёвой полуторной переклейке), поскольку далее (на шаге 4.21) все (a,b)-цепи размера больше нуля замыкаются в (a,b)-циклы, (a,b)-циклы сливаются друг с другом и получившийся (a,b)-цикл разбивается на циклы размера 2. Если же эта цепь имеет тип $2a'$, это взаимодействие (пункт 4.2′) следует применить, чтобы «уничтожить» эту проблемную цепь.

По сравнению с описанием в [4] здесь внесены усовершенствования, направленные на максимально возможное уничтожение проблемных цепей.

4.1. «b-петля»+любой тип t с b-вершиной = тип t. Объединить b-вершину петли с b-вершиной компоненты типа t двойной переклейкой (если эта цепь не изолированная b-вершина, рис. 12a) или полуторной переклейкой (иначе, рис. 12b).

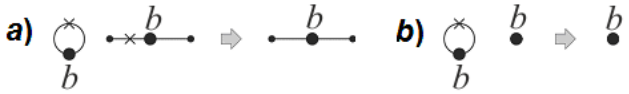

Рис.12. Шаг 4.1 алгоритма

4.1′. То же, что и 4.1.

4.2. $2a+2b^*=2+1'_a$. 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. 13.

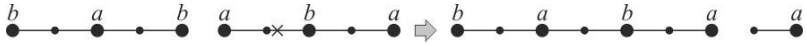

Рис.13. Шаг 4.2 алгоритма

4.2′. $2a'+2b^*=2+1'_a$.

4.3. $3a^*+3b=3$. В $3a^*$-цепи расклеить внешнее ребро и особую вершину склеить с крайней обычной вершиной $3b$-цепи, рис. 14.

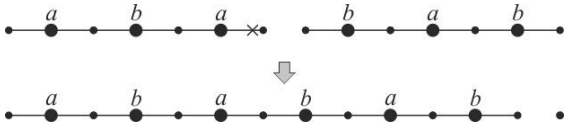

Рис.14. Шаг 4.3 алгоритма

4.3′. $3a^*+3b'=3$.

4.4. $2a+3=1a$, $2b^*+3=1b$. В 3-цепи расклеить внешнее b-ребро и особую вершину склеить с крайней особой вершиной $2a$-цепи, рис. 15.

Рис.15. Шаг 4.4 алгоритма

4.4′. $2a'+3=1a$

4.5. $3a^*+2=1a$, $3b+2=1b$. В $3a$-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 2-цепи, рис. 16.

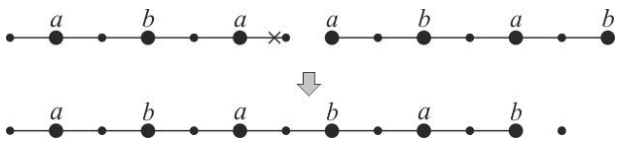

Рис.16. Шаг 4.5 алгоритма

4.5′. $3b'+2=1b$.

4.6. $2a+2a=2a$, $2b^*+2b^*=2b^*$. Склеить крайние особые вершины двух цепей, рис. 17.

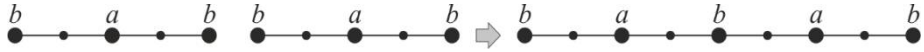

Рис.17. Шаг 4.6 алгоритма

4.6′. $2a'+2a=2a$.

4.7. $3a^*+3a^*=3a^*$, $3b+3b=3b$. Две крайние обычные вершины цепей соединить обычным ребром с последующим его вырезанием, рис. 18.

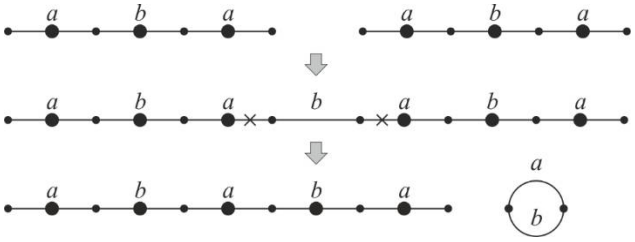

Рис.18. Шаг 4.7 алгоритма

4.7′. $3b'+3b=3b$.

4.8. $1a+2a=1a$, $1b+2b^*=1b$. Склеить крайние особые вершины двух цепей, рис. 19.

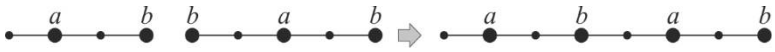

Рис.19. Шаг 4.8 алгоритма

4.8′. $1a+2a'=1a$.

4.9. $1a+3a^*=1a$, $1b+3b=1b$. Две крайние обычные вершины цепей соединить обычным ребром с последующим его вырезанием, рис. 20.

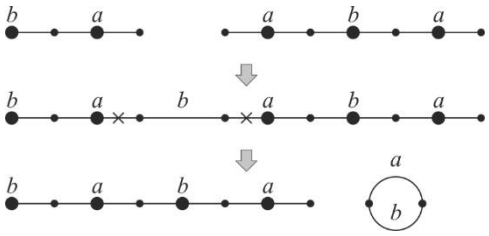

Рис.20. Шаг 4.9 алгоритма

4.9′. $1b+3b'=1b$.

4.10. $2a+2=2$, $2b^*+2=2$. Склеить крайние особые вершины двух цепей, рис. 21.

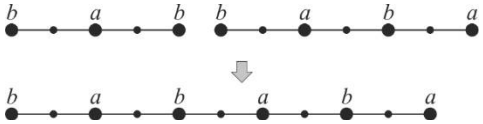

Рис.21. Шаг 4.10 алгоритма

4.10′. $2a'+2=2$.

4.11. $3a^*+3=3$, $3b+3=3$. Две крайние обычные вершины цепей соединить обычным ребром с последующим его вырезанием, рис. 22.

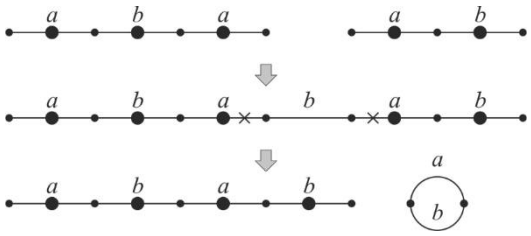

Рис.22. Шаг 4.11 алгоритма

4.11′. $3b'+3=3$.

4.12. $2+2=2+1'_a$. 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. 23.

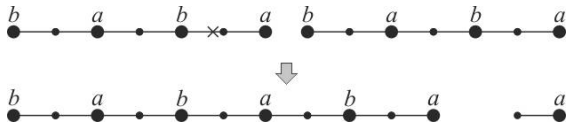

Рис.23. Шаг 4.12 алгоритма

4.12′. Пустое действие.

4.13. $3+3=3$. В 3-цепи расклеить внешнее a-ребро и образовавшийся край этой цепи склеить с b-краем другой 3-цепи, рис. 24.

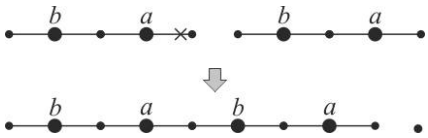

Рис.24. Шаг 4.13 алгоритма

4.13′. Пустое действие.

4.14. $1^*_a+1^*_a=1^*_a$, $1_b+1_b=1_b$. В 1_a-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной другой 1_a-цепи, рис. 25.

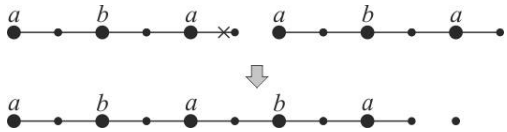

Рис.25. Шаг 4.14 алгоритма

4.14′. $1'_b+1_b=1_b$.

4.15. $1a+1_b=1a$, $1b+1^*_a=1b$. В 1_b-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной $1a$-цепи, рис. 26.

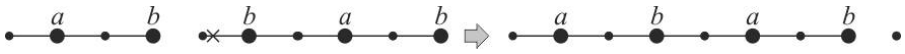

Рис.26. Шаг 4.15 алгоритма

4.15′. $1a+1'_b=1a$.

4.16. $1a+1^*_a=1a$, $1b+1_b=1b$. В $1a$-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 1^*_a-цепи, рис. 27.

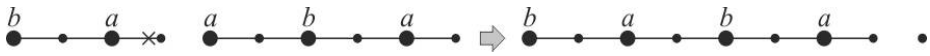

Рис.27. Шаг 4.16 алгоритма

4.16′. $1b+1'_b=1b$.

4.17. $2a+1_b=2a$, $2b^*+1^*_a=2b^*$. В 1_b-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной $2a$-цепи, рис. 28.

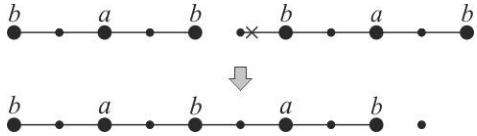

Рис.28. Шаг 4.17 алгоритма

4.17′. $2a'+1_b=2a$, $2a+1'_b=2a$.

4.18. $3a^*+1^*_a=3a^*$, $3b+1_b=3b$. В $3a^*$-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 1^*_a-цепи, рис. 29.

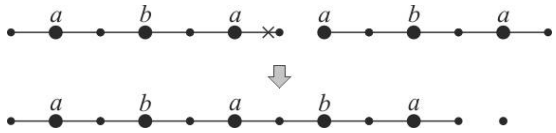

Рис.29. Шаг 4.18 алгоритма

4.18′. $3b'+1_b=3b$, $3b+1'_b=3b$.

4.19. $2+1^*_a=2$, $2+1_b=2$. В 1^*_a-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 2-цепи, рис. 30.

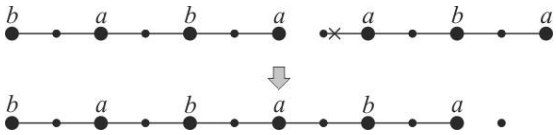

Рис.30. Шаг 4.19 алгоритма

4.19′. $2+1'_b=2$.

4.20. $3+1^*_a=3$, $3+1_b=3$. В 3-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 1^*_a-цепи, рис. 31.

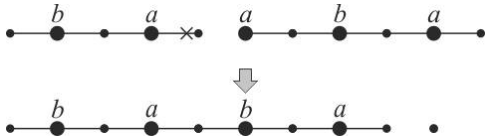

Рис.31. Шаг 4.20 алгоритма

4.20′. $3+1'_b=3$.

Шаги 4.21–4.24 одинаковы для обычного и «штрихованного» вариантов.

4.21. Замыкание цепей в циклы, кроме цепей типа $3b'$ (они ещё могут «пригодиться» на шаге 4.24), проблемных цепей и цепей типа 2′ (они в цикл не замыкаются). Цепи, имеющие невисячее ребро, замыкаем в циклы склейкой (цепи типа $2a^*$, $2b^*$, $3a$, $3b^*$), полуторной переклейкой с отождествлением двух особых вершин (цепи типа 1^*_a, 1^*_b, 2^*) или без отождествления (цепи типа $1a$, $1b$, 3). При замыкании цепи типа 2^* выбираем вариант с отождествлением двух b-вершин, рис. 32. Из циклов, получившихся при замыкании цепей типа $3a^*$ или $3b^*$, вырезаем обычные рёбра.

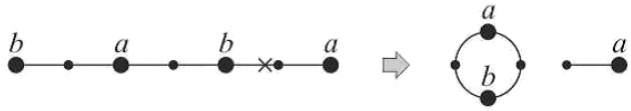

Рис.32. Замыкание в цикл цепи типа 2

4.22. Пока возможно, осуществлять взаимодействие (a,b)-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. 33. Впрочем, легко видеть, что фактически t может быть лишь (a,b)-циклом или цепью типа 2′.

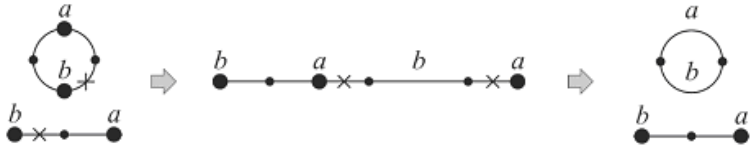

Рис.33. Шаг 4.22 алгоритма, t=2′

4.23. Если возможно, выполнить взаимодействие (a,b)-цикл+$2a'+2b'=2$. Сначала полуторной

переклейкой (a,b)-цикл+2a'=1a (рис. 34), затем 1a+2b'=2 (пункт 3.12).

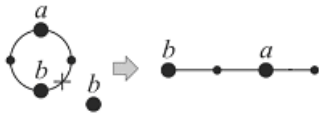

Рис.34. Первая часть шага 4.23 алгоритма

4.24. Если возможно, выполнить взаимодействие (a,b)-цикл+3a'+3b'=3. Сначала двойной и полуторной переклейками (a,b)-цикл+3b'=1b (рис. 35), затем 1b+3a'=3 (пункт 3.13).

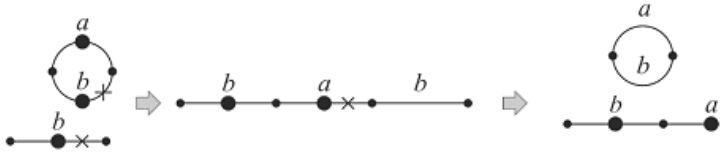

Рис.35. Первая часть шага 4.24 алгоритма

Шаг 5. Оставшиеся цепи (типов $1'_a$, $1'_b$, $2'$, $2a'$, $2b'$, $3b'$) приводим к финальному виду по отдельности. Из циклов размера больше 2 вырезаем циклы размера 2 так, чтобы происходило отождествление двух b-вершин (соответственно, в вырезанный цикл включается a-вершина), рис. 36. Из циклов размера 2 удаляем особые вершины.

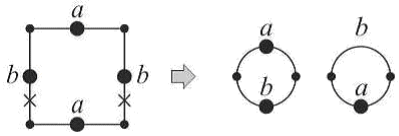

Рис.35. Вырезание из (a,b)-цикла a-цикла

Алгоритм описан. Пусть B' – число b-циклов в графе a+b. Напомним обозначения из [4]: B – число особых вершин в a+b; S – сумма целых частей половин длин максимальных отрезков в a+b, которые состоят из обычных рёбер (называем из *сегментами*), 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), D – сумма дефектов компонент графа a+b (дефект цепей типа 1a, 1b 3a, 3b и 3 равен 1, дефект цепей других типов или цикла нулевой), P – разность величин D, вычисленных до и после применения шага 3 алгоритма, т.е. число операций, сэкономленных на шаге 3. Величина ε определена выше. Пусть C=B+S+D–P+ε(B'+1).

Теорема 1. Алгоритм строит последовательность операций, суммарная цена которой равна одному из трёх значений C–ε, C, C+ε. Минимально возможная суммарная цена последовательности операций, приводящей граф a+b к финальному виду, также равна одному из этих значений. Время работы алгоритма линейное.

Ключевой момент в доказательстве первого утверждения теоремы 1: после выполнения шага 4 остаётся не более двух ценных компонент. Если их остаётся две, то одна из них – (a,b)-цикл, вторая – проблемная цепь. Подробное доказательство теоремы 1 приведено в [4].

Следствие 1. Если после шага 4 остаётся не более одной ценной компоненты, то алгоритм выдаёт абсолютно точное решение. Если остаётся две ценных компоненты, алгоритм выдаёт решение, цена которого может превышать цену оптимального решения не более, чем на ε.

Доказательство. Из доказательства теоремы 1 [4] вытекает, что если после шага 4 остаётся 0, 1 или 2 ценных компоненты, то цена построенной алгоритмом последовательности равна, соответственно, $C(G)$–ε, $C(G)$ или $C(G)$+ε. Поэтому достаточно доказать, что если цена построенной алгоритмом последовательности равна $C(G)$–ε или $C(G)$, то такова же цена оптимальной последовательности, если же первая цена равна $C(G)$+ε, то цена оптимальной последовательности не меньше $C(G)$. Утверждение для $C(G)$–ε сразу следует из теоремы 1. Для двух других значений докажем его индукцией по минимальной суммарной цене M операций, приводящих общий граф G к финальному виду.

Базис индукции тривиален, опишем индуктивный шаг. Пусть o – первая операция в оптимальной последовательности операций, $c(o)$ – её цена, $o(G)$ – результат её применения к G. Из описания алгоритма следует, что если ценные компоненты присутствует в начальном общем графе, то хотя бы одна ценная компонента останется и после шага 4 алгоритма, а если в начальном графе нет ни одной ценной компоненты, то их не возникнет и после шага 4. Поэтому возможны лишь следующие случаи.

1) В графах G и $o(G)$ имеется ценная компонента. По предположению индукции цена оптимальной последовательности для G не меньше $c(o)+C(o(G))$. Учитывая установленное при доказательстве теоремы 1 неравенство $c(o)\geq C(G)-C(o(G))$ [4], получаем, что эта цена не меньше $C(G)$, откуда следует требуемое утверждение.

2) В графе G имеется ценная компонента, а в графе $o(G)$ её нет. По предположению индукции цена оптимальной последовательности для G равна $c(o)+C(o(G))-\varepsilon$. Легко видеть, что возможны лишь следующие два случая.

2.1. Операция o превращает ценную компоненту в b-цикл. В этом случае $c(o)=1$ и o увеличивает величину B' на 1. Тогда $C(G)-C(o(G))\leq 1-\varepsilon$. Отсюда $c(o)+C(o(G))-\varepsilon\geq C(G)$, что и требуется.

2.2. Операция o удаляет b-вершину из ценной компоненты. В этом случае $c(o)=1+\varepsilon$ и o не меняет величину B'. Тогда $C(G)-C(o(G))\leq 1$. Отсюда $c(o)+C(o(G))-\varepsilon\geq C(G)$, что и требуется.

Следствие 1 доказано. Следующее следствие формулирует достаточные условия абсолютной точности алгоритма в терминах исходных структур a и b и графа $a+b$.

Следствие 2. Алгоритм выдаёт абсолютно точное решение в любом из следующих случаев:

1) Структура a не содержит особых генов;

2) Структура b не содержит особых генов;

3) Среди компонент графа $a+b$ нет проблемных цепей (в частности, когда все хромосомы в a и b кольцевые).

Доказательство. В случаях 1 и 2 в общем графе не возникает (a,b)-циклов, поэтому после шага 4 остаётся не более одной ценной компоненты. По следствию 1 выдаваемое алгоритмом решение абсолютно точное. В случае 3 из описания алгоритма следует, что в ходе него не возникнет проблемных цепей и рассуждение аналогично. Следствие 2 доказано.

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

Литература

1. Горбунов К.Ю., Гершгорин Р.А., Любецкий В.А. Перестройка и реконструкция хромосомных структур // Молекулярная биология. – 2015. – Т. 49, № 3. – С. 372–383.
2. Lyubetsky V.A., Gershgorin R.A., Seliverstov A.V., Gorbunov K.Yu. Algorithms for reconstruction of chromosomal structures // BMC Bioinformatics. – 2016. – V. 17, no. 40, 23 pages. DOI: 10.1186/s12859-016-0878-z.
3. Горбунов К.Ю., Любецкий В.А. Линейный алгоритм минимальной перестройки структур // Проблемы передачи информации. – 2017. – Т. 53, вып. 1. В печати.
4. Горбунов К.Ю., Любецкий В.А. Линейный алгоритм кратчайшей перестройки графов при разных ценах операций // Информационные процессы. – 2016. – Т. 16, № 2. – С. 223–236.

References

1. Gorbunov K.Yu., Gershgorin R.A., Lyubetsky V.A. Rearrangement and Inference of Chromosome structures // Molecular Biology. – 2015. – V. 49, no. 3. – P. 327–338. DOI: 10.1134/S0026893315030073.
2. Lyubetsky V.A., Gershgorin R.A., Seliverstov A.V., Gorbunov K.Yu. Algorithms for reconstruction of chromosomal structures // BMC Bioinformatics. – 2016. – V. 17, no. 40, 23 pages. DOI: 10.1186/s12859-016-0878-z.
3. Gorbunov K.Yu., Lyubetsky V.A. Linear algorithm of the minimal reconstruction of structures // Problems of Information Transmission. – 2017. – V. 53, iss. 1. In press.
4. Gorbunov K.Yu., Lyubetsky V.A. A linear algorithm of the shortest transformation of graphs under different operation costs // Information Processes. – 2016. – Vol. 16, no. 2. – P. 223–236 (in Russian).

Поступила 21.10.2016

Об авторах:

Горбунов Константин Юрьевич, лаборатория № 6 Института проблем передачи информации им. А.А. Харкевича Российской академии наук, кандидат физико-математических наук, gorbunov@iitp.ru;

Любецкий Василий Александрович, заведующий лабораторией № 6 Института проблем передачи информации им. А.А. Харкевича Российской академии наук, доктор физико-математических наук, lyubetsk@iitp.ru.