Лаборатория математических методов и моделей
в биоинформатике
Институт проблем передачи информации
Российской академии наук

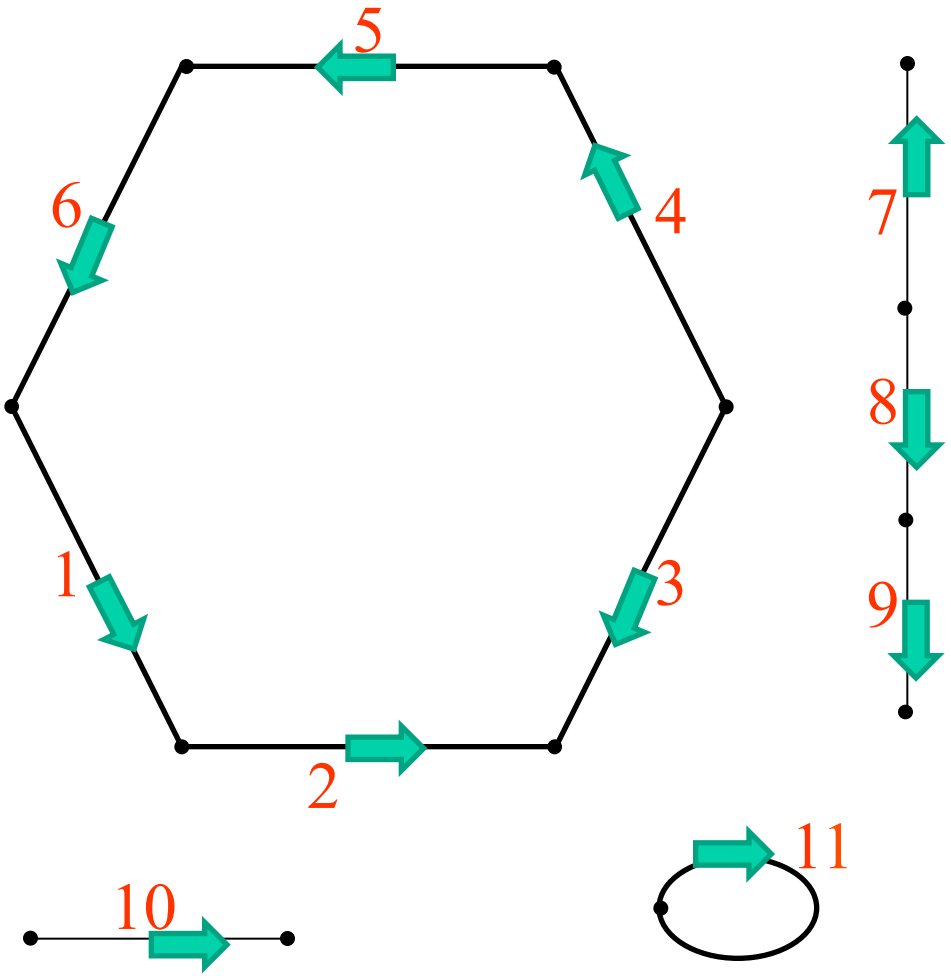
В.А. Любецкий, К.Ю. Горбунов

АЛГОРИТМ РАССТАНОВКИ ПО ДЕРЕВУ ПРЕДКОВЫХ ХРОМОСОМНЫХ СТРУКТУР: КОРРЕКТНОСТЬ И РЕЗУЛЬТАТЫ

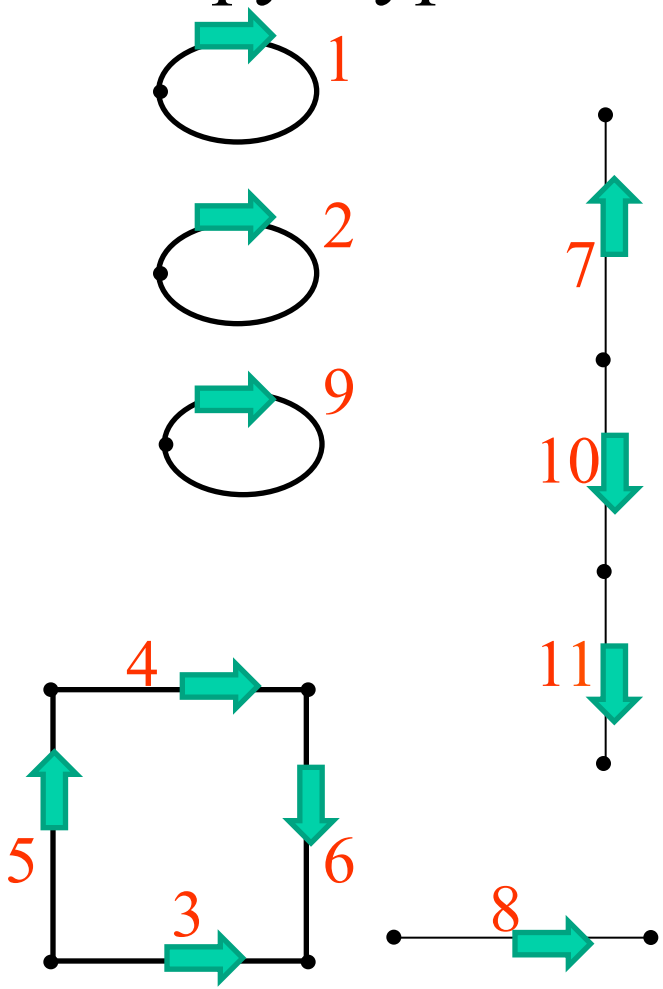
Накоплен значительный объем данных о хромосомных структурах геномов. Это позволило предложить следующую модель хромосомной структуры: пример двух хромосомных структур *a* и *b*.

Гены показаны направленными отрезками и занумерованы числами; структуры могут содержать **одинаковые или разные** наборы генов. Хромосомы линейные или циклические. Пример:

Структура *a*



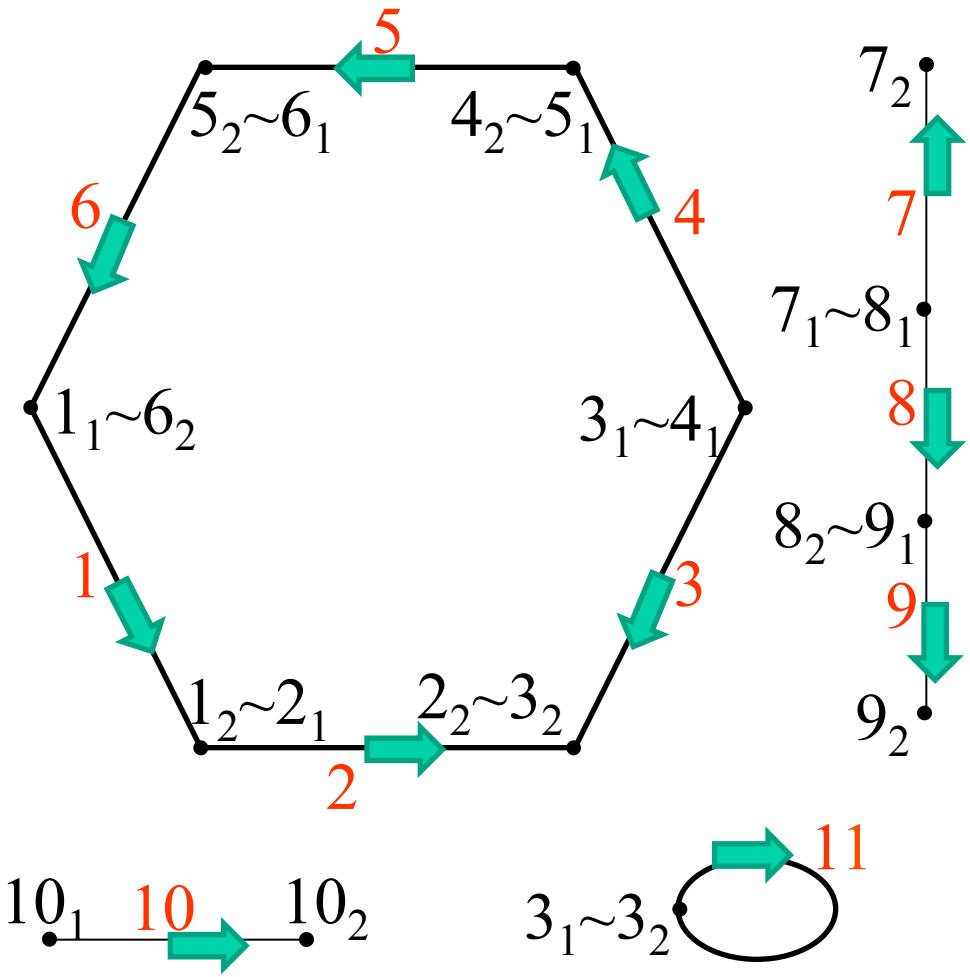
Структура *b*



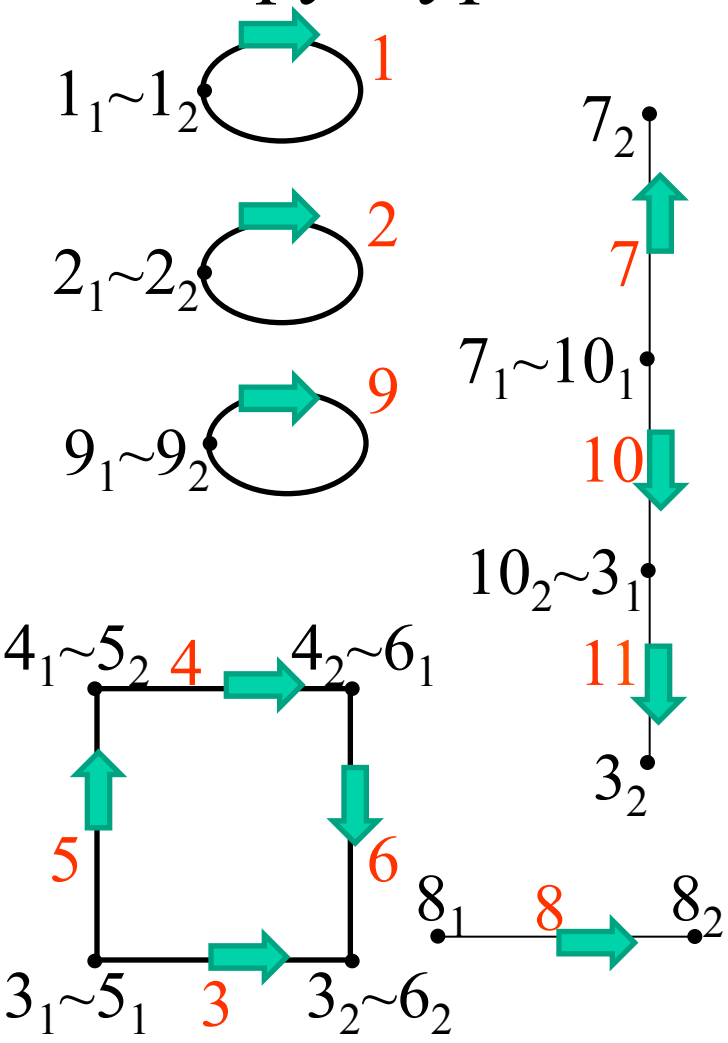
Показаны те же две структуры, но теперь края генов склеены.

Для гена *i* его начало обозначено i_1 , его конец обозначен i_2.

Структура *a*



Структура *b*



Задача: **восстановить хромосомные структуры в нелистовых вершинах данного дерева (возможно, небинарного), если они заданы в его листьях. При этом минимизировать сумму (по всем рёбрам дерева) расстояний между структурами на концах ребра**. Структуры в листьях могут иметь разные множества генов.

Расстояние между структурами, приписанными концам ребра, определяется как **сумма**: (число пар краёв генов, которые в одной структуре склеены, а в другой – не склеены) **+** (число генов, присутствующих в одной структуре и отсутствующих в другой).

Нами предложен **точный алгоритм решения этой задачи, вычислительная сложность которого –** произведение числа листьев дерева на квадрат числа неортологичных генов во всех листьях.

Ценовой вариант задачи:

Даны 4 цены: цена склейки двух краёв генов, цена их расклейки, цена потери гена, цена возникновения гена.

При восстановлении хромосомных структур нужно минимизировать сумму (по всем рёбрам дерева) цен событий между структурами на концах ребра.

Прежний вариант соответствует одинаковым ценам.

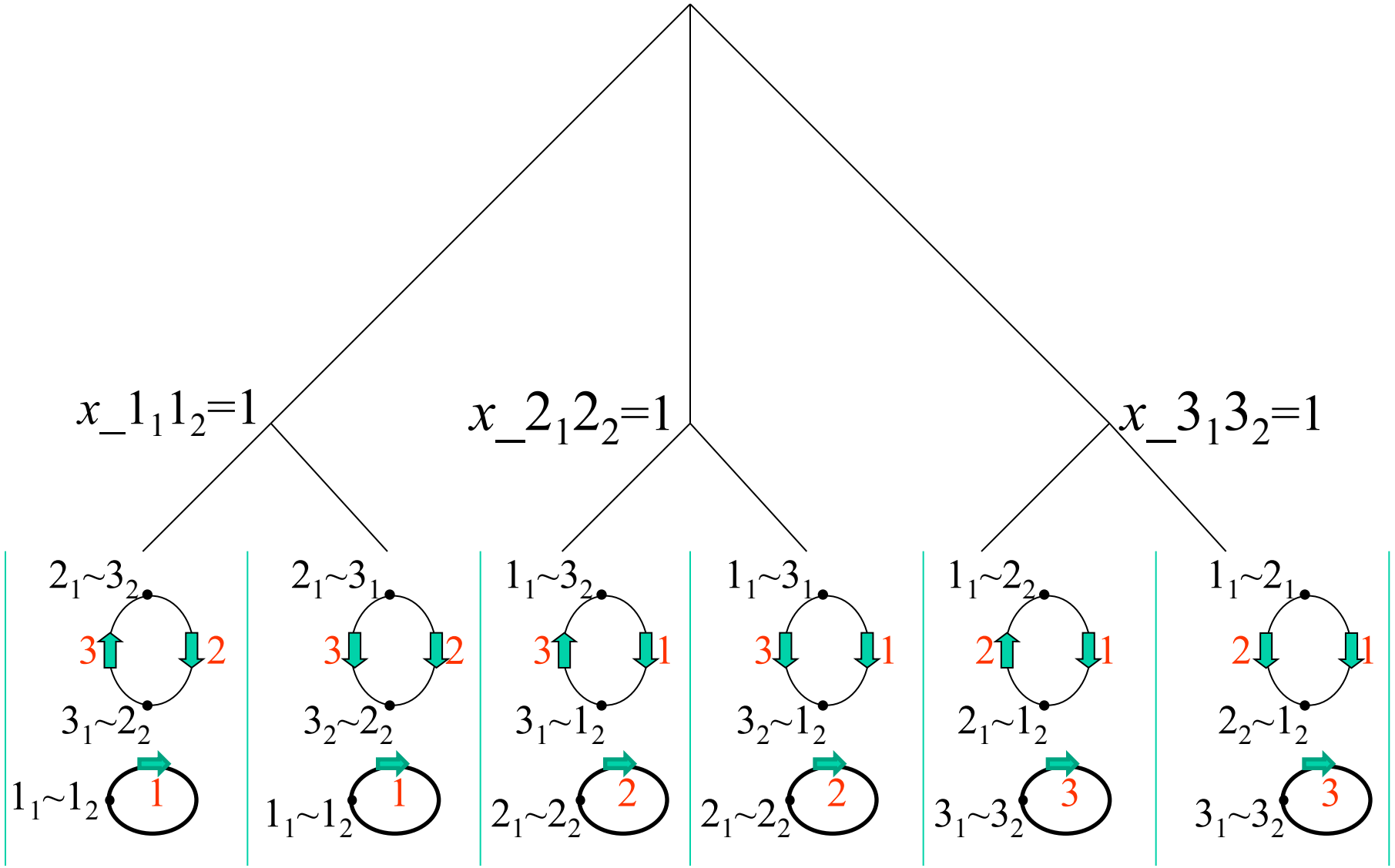
Наш алгоритм выдаёт точное решение, когда цена склейки двух краев не больше цены их расклейки, и цена потери гена не больше цены его возникновения.

Описание алгоритма

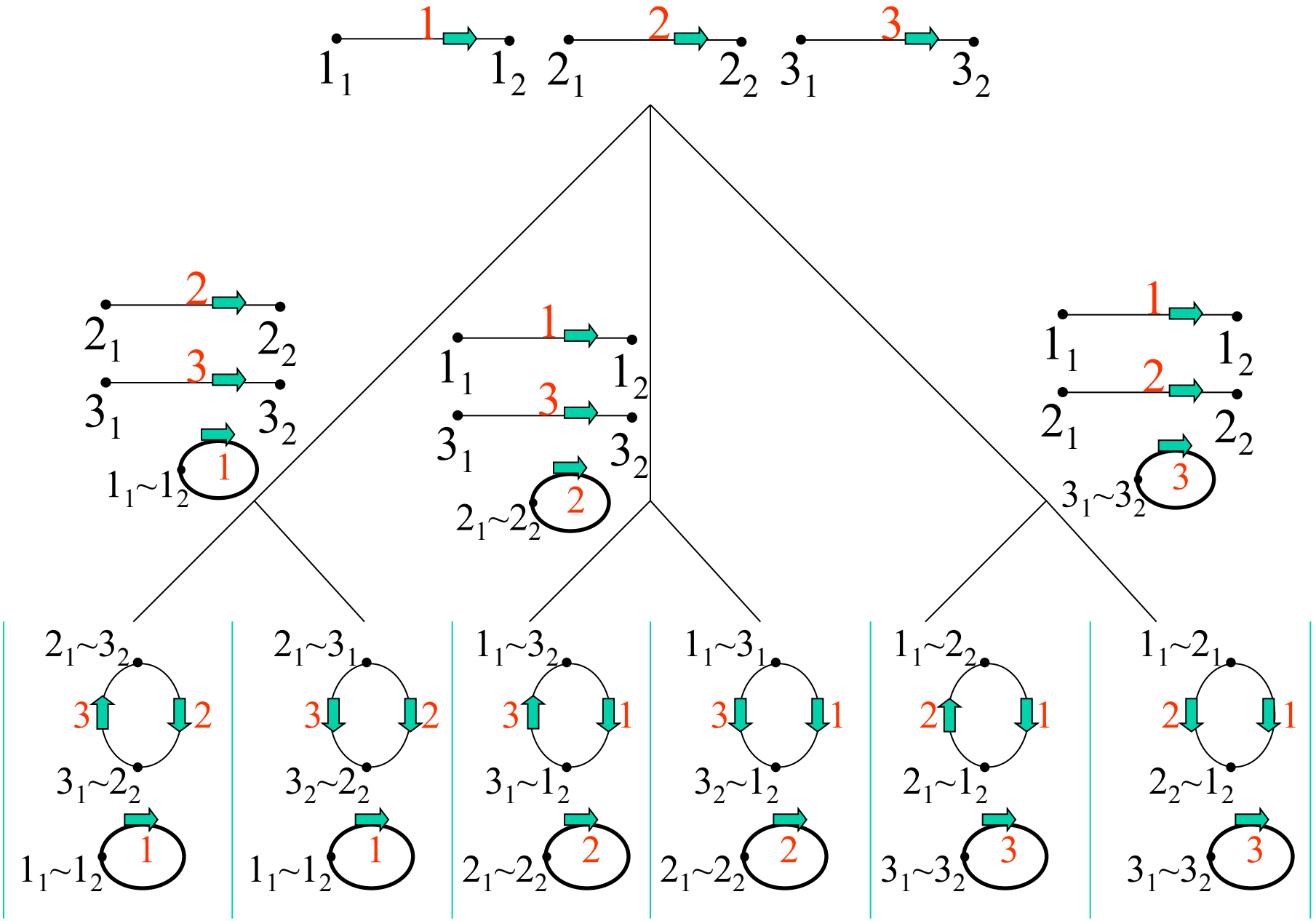
Для каждой внутренней вершины дерева введем свою переменную x_{ij} для каждой пары краёв $i \neq j$ генов. Переменная равна 1, если соответствующие края склеены и 0 в противном случае. Также, для каждого гена i вводится переменная y_i, равная 0, если ген присутствует в структуре и 1, если отсутствует. Таким образом, в листьях значения переменных даны. Для каждой переменной расставим ее значения во внутренних вершинах так, чтобы минимизировать цену разметки – сумму цен событий по этой переменной (алгоритм динамического программирования).

В полученном «решении» возможны противоречия, т.е. пары равных 1 переменных, противоречащих друг другу (один край гена склеивается с двумя краями или край отсутствующего гена с чем-то склеен).

Пример реконструкции эволюции структуры из трех генов
Пусть все цены равны



Решение (найденные предковые структуры):

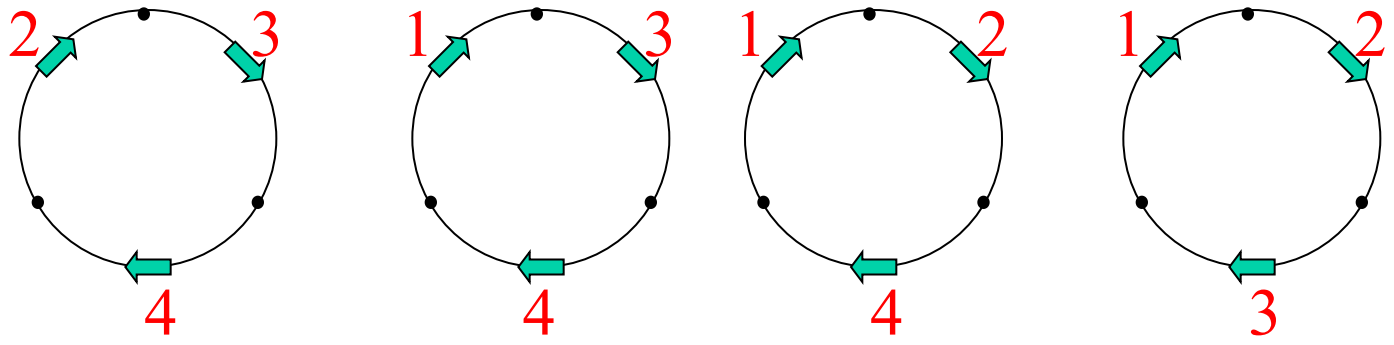


Еще один пример: пусть цена расклейки меньше цены склейки.

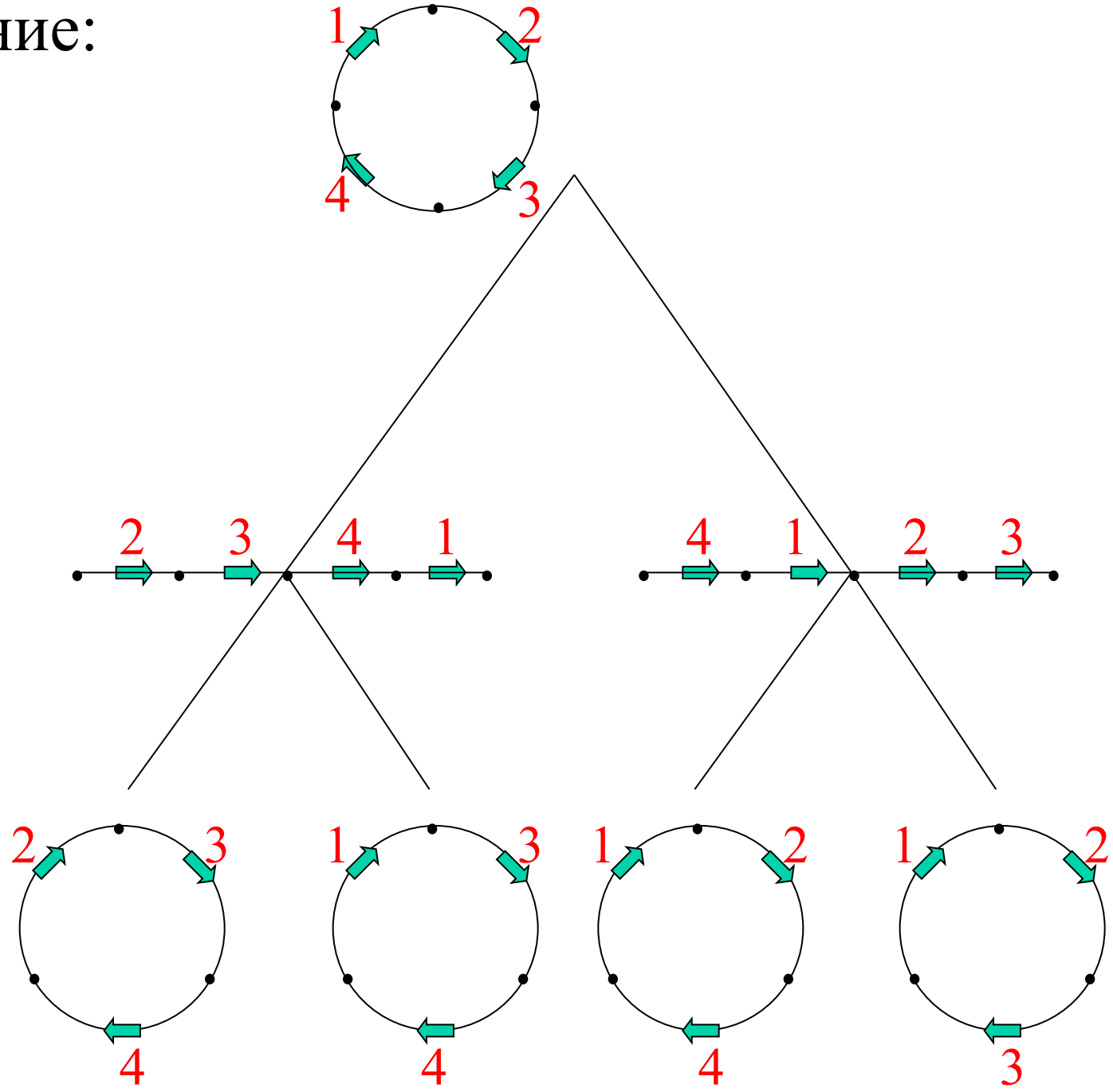
$x\_1_2 2_1 = 1$, $x\_2_2 3_1 = 1$,
$x\_3_2 4_1 = 1$, $x\_4_2 1_1 = 1$

$x\_3_2 4_1 = 1$
$x\_2_2 3_1 = 1$
$x\_4_2 1_1 = 1$

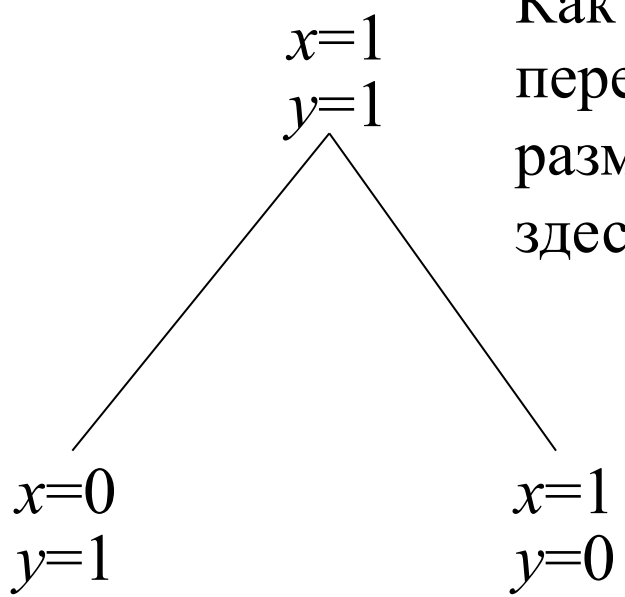
$x\_1_2 2_1 = 1$
$x\_2_2 3_1 = 1$
$x\_4_2 1_1 = 1$



Решение:



Противоречие – пара переменных, равных 1, если эти значения противоречат друг другу.



Как устранить противоречия? Менять значения переменных, но желательно так, чтобы цена разметки не увеличилась. Например, заменить здесь две единицы в корне на нули.

Если цена $c(0\to1)$ перехода $0\to1$ равна цене $c(1\to0)$ перехода $1\to0$, то цена разметки не изменится. Если $c(0\to1)<c(1\to0)$, то цена разметки уменьшится на $2d$, где $d=c(1\to0)-c(0\to1)$, так что противоречия вообще не могло быть. Если $c(0\to1)>c(1\to0)$, то цена разметки увеличится на $2d$, в данном случае разметка не будет оптимальной.

Устранение противоречий

Противоречие – пара переменных, равных 1. Перебираем вершины дерева. В каждой вершине, если противоречие есть (т.е. пара значений 1,1) заменяем ее на пару 0,0.

Сначала рассмотрим случай $c(0\to1)=c(1\to0)$.

Покажем, что при устранении противоречий цена разметки дерева не увеличивается. Дефектом разметки назовем величину отличия её цены от первоначальной. Назовем ребро противоречивым, если один его конец размечен парой 1,1, а другой – парой 0,0.

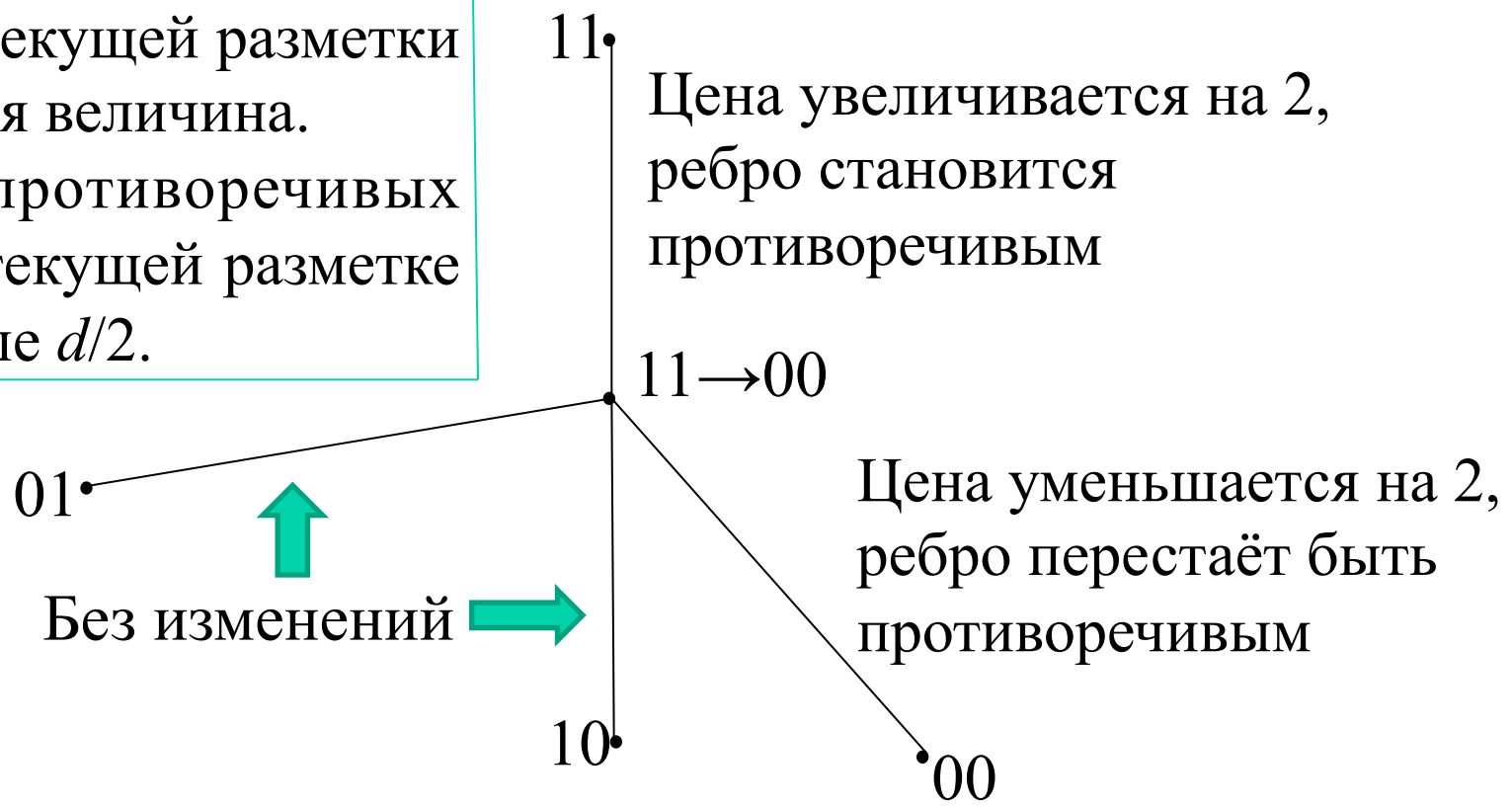
Покажем:

1) Дефект текущей разметки d – четная величина.

2) Число противоречивых рёбер в текущей разметке не меньше $d/2$.

Доказываем:

1) Дефект текущей разметки d – четная величина.
2) Число противоречивых рёбер в текущей разметке не меньше $d/2$.



В конце противоречивых ребер не останется, по свойству 2 дефект разметки станет нулевым. Новых противоречий (по другим парам переменных) не возникнет.

Видим, что устранение противоречий не увеличивает цену разметки не только для дерева, но и для произвольного графа.

Ценовой вариант $c(0\to1)<c(1\to0)$

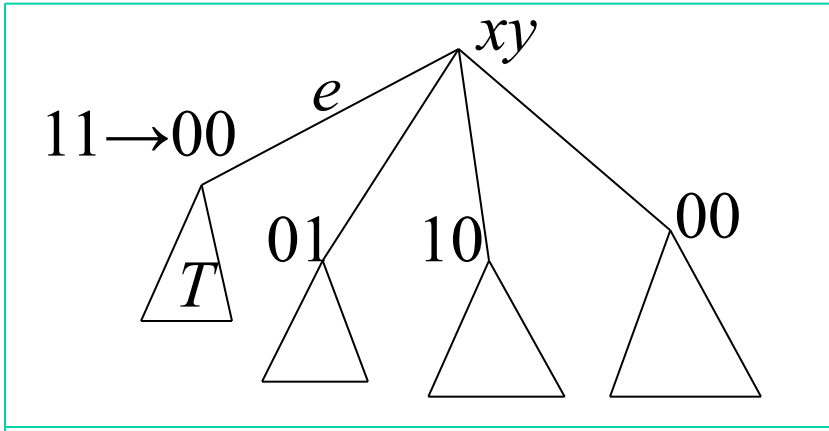
Покажем: если $c(0\to1)<c(1\to0)$, то противоречий не возникнет. Пусть $d=c(1\to0)-c(0\to1)$. Фиксируем пару переменных x,y. Индукцией по высоте дерева докажем: для любой (не обязательно минимальной разметки) если в корне дерева имеется противоречие, то при устранении противоречий цена разметки уменьшится не менее, чем на $2d$, иначе цена не увеличится.

Индукционный шаг:

1) $xy=00$. Цена ни по какому направлению не увеличивается.

2) $xy=01$ или 10. На ребре e увеличение d, но в дереве T уменьшение $2d$.

3) $xy=11\to00$. На каждом ребре, отличном от e, уменьшение $\geq d$, и в дереве T уменьшение $2d$ (используем, что сыновей > 1).



Пусть T – максимальное по включению поддерево с противоречием в корне.

Устранение противоречий в T уменьшает цену разметки, а она уже минимальна.

Быстрый алгоритм устранения противоречий

Упорядочим все края генов в каком-либо линейном порядке L, например, лексикографически. В произвольном порядке перебираем вершины дерева и для каждой вершины перебираем края **всех** генов в порядке L. Для каждого края удаляем все его склейки в данной структуре, если их четное число, или, если нечетное, оставляя одну склейку с краем, наибольшем в этом порядке. Затем перебираем гены в порядке возрастания их номеров и для каждого края отсутствующего гена, если он с чем-то склеен (а он теперь может быть склеен не более чем с одним краем) удаляем эту склейку и объявляем этот ген присутствующим в структуре. Конец алгоритма.

Отметим также: в биологических примерах применения алгоритма мы брали цену склейки равной 3, а цену расклейки равной 2, и тогда после устранения противоречий цена разметки часто незначительно увеличивается.

Использование «биологического» расстояния

На биологических примерах алгоритм часто выдавал предковые структуры с большим числом хромосом (к тому же линейных), даже если во всех листьях было по одной циклической хромосоме. Варьирование цен событий не помогало.

Изменение определения расстояния между структурами помогло решить эту проблему. Биологическое расстояние – это минимальное количество операций, требуемых для преобразования одной структуры в другую. Мы использовали известный набор операций, называемый «Double Cut and Join».

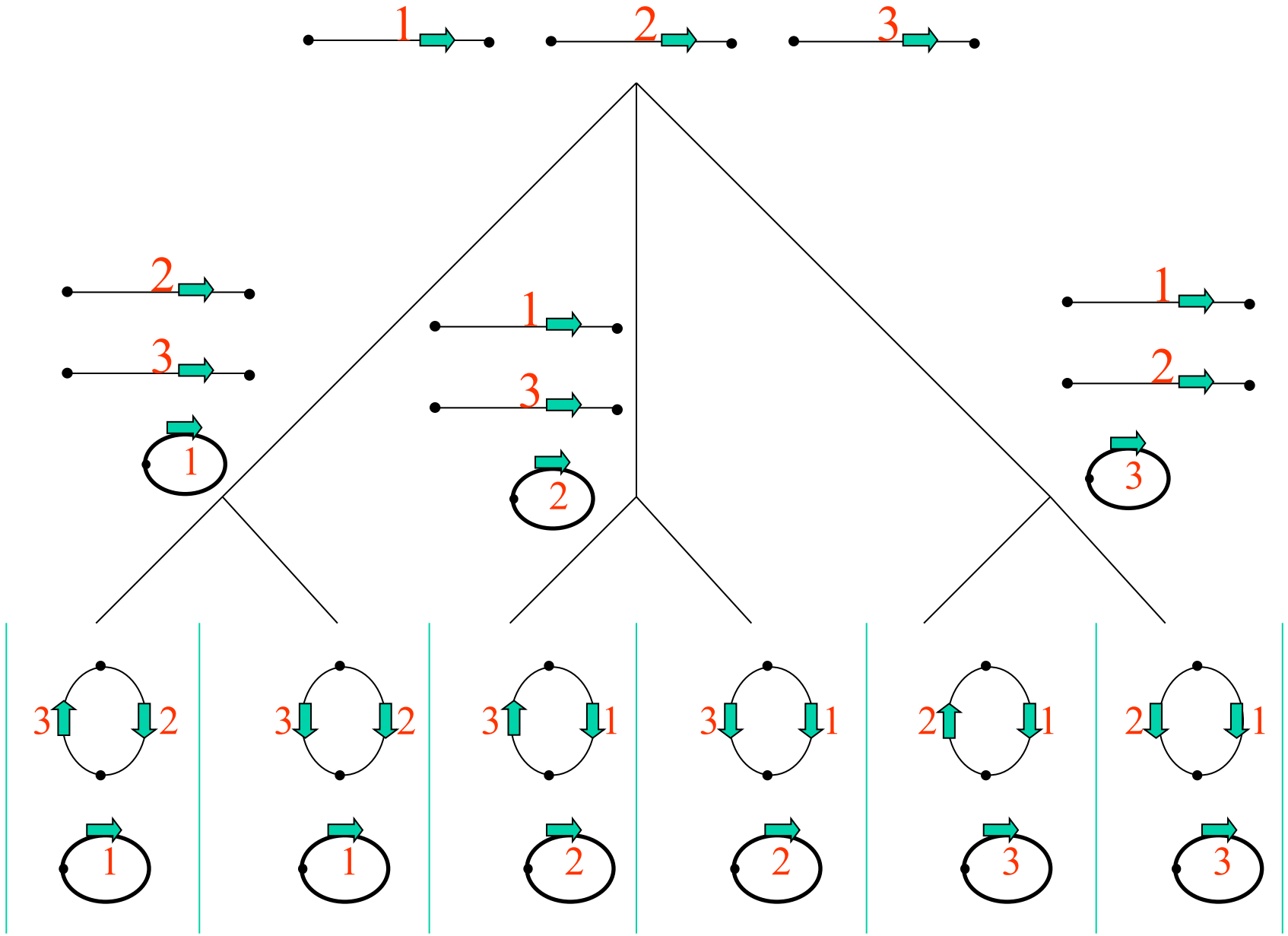
Для минимизации функционала применялся метод «спуска» из различных начальных расстановок структур по нелистовым вершинам дерева.

Набор операций (Double Cut and Join):

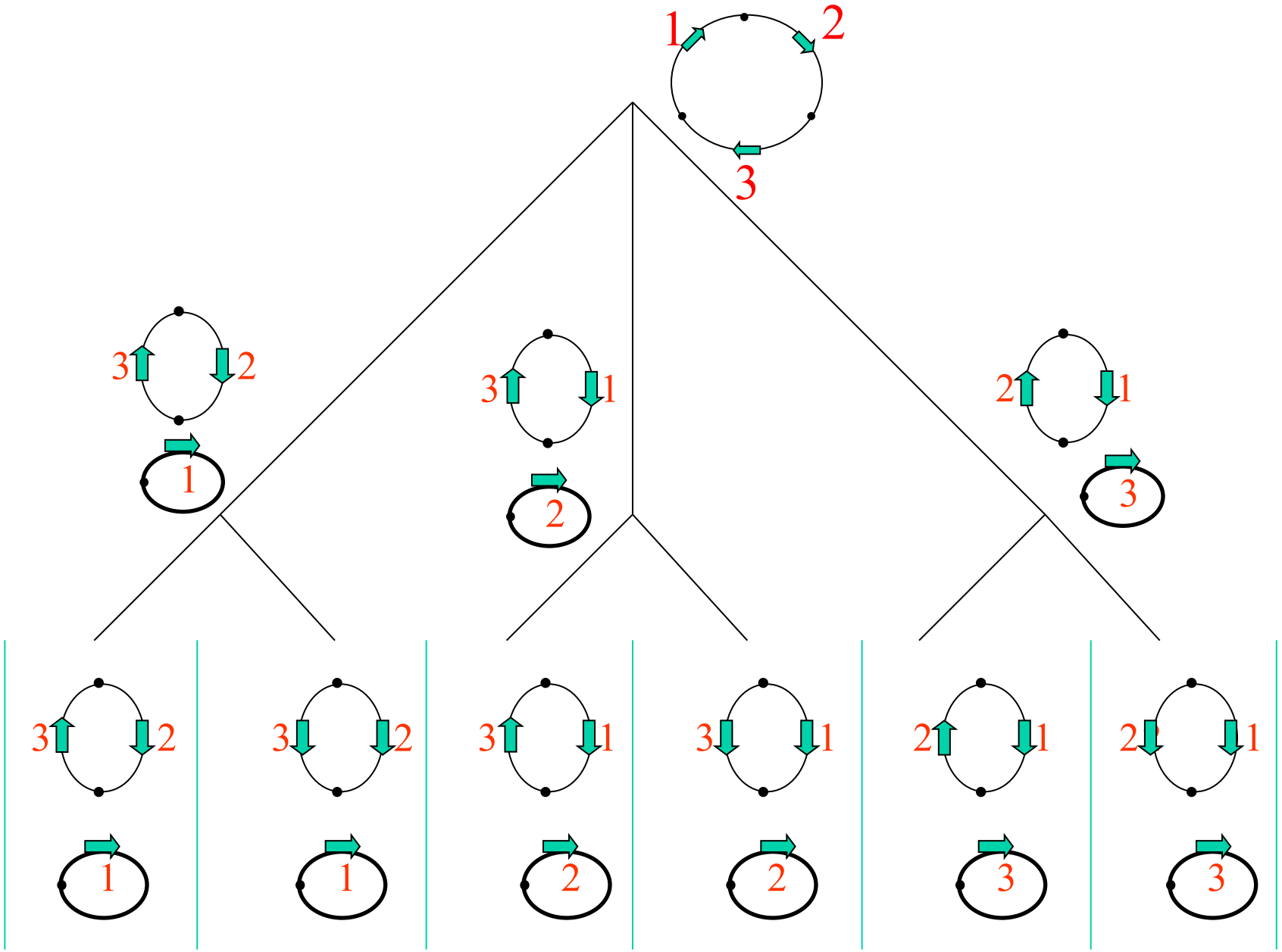
1) расклейка двух склеек структуры и склейка четырёх освободившихся краёв генов по-другому;
2) расклейка одной склейки и склеивание одного из освободившихся краёв гена с каким-то свободным краем;
3) разрез одной склейки и обратная операция склейки двух свободных краёв.
4) удаление/вставка участка хромосомы, в случае разного состава генов в листьях. В этом случае также налагалось условие: отсутствовать в вершине могут лишь гены, отсутствующие хоть в одном ее сыне.

- Эти операции позволяют выполнять все биологически содержательные перестройки хромосом: инверсия, транслокация, трансверсия, вырезание сегмента с его зацикливанием, вставка цикла в цепь и другие.

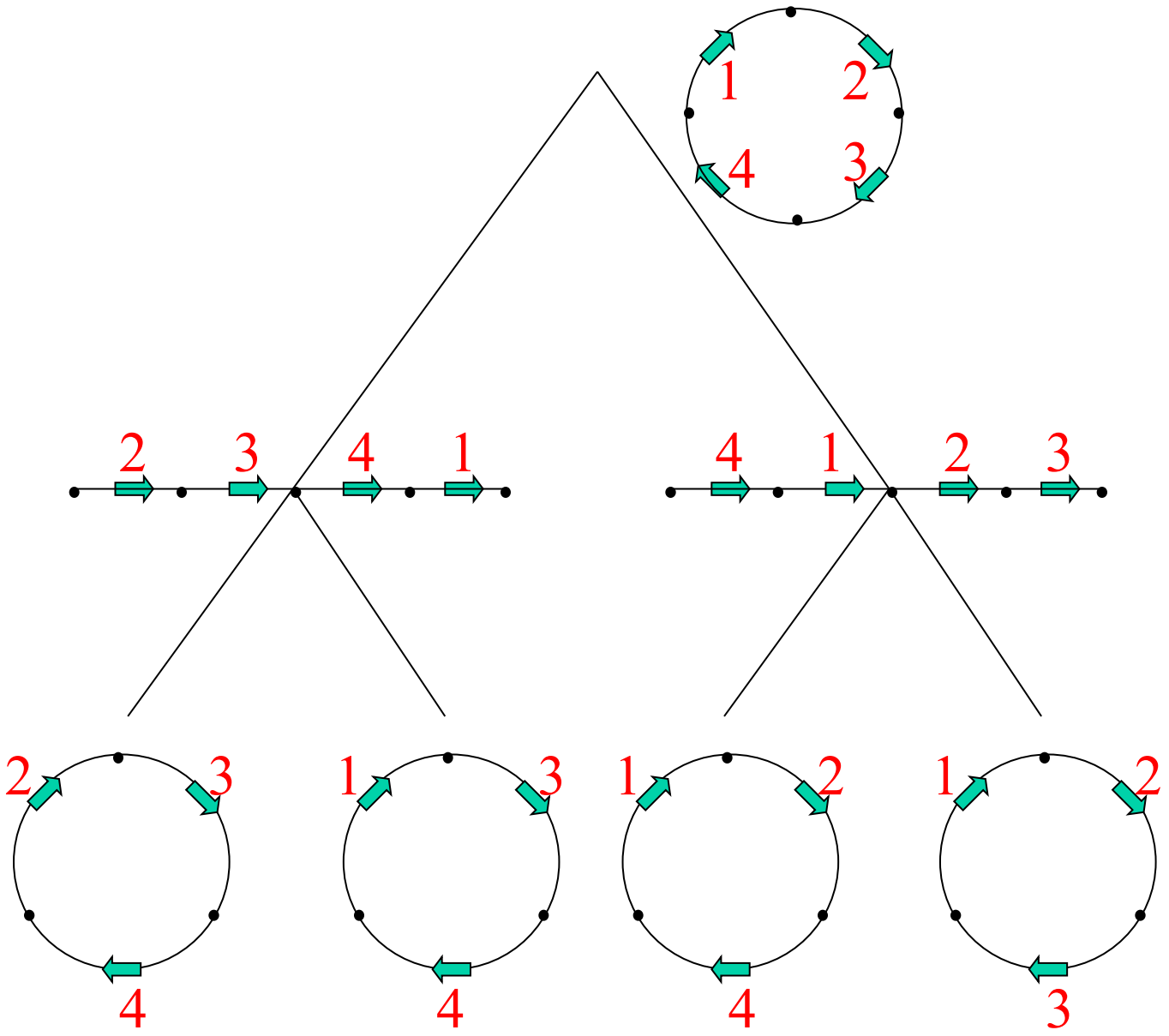
Прежнее решение первого примера: 15 склеек – по одной на верхних рёбрах и по две на нижних.



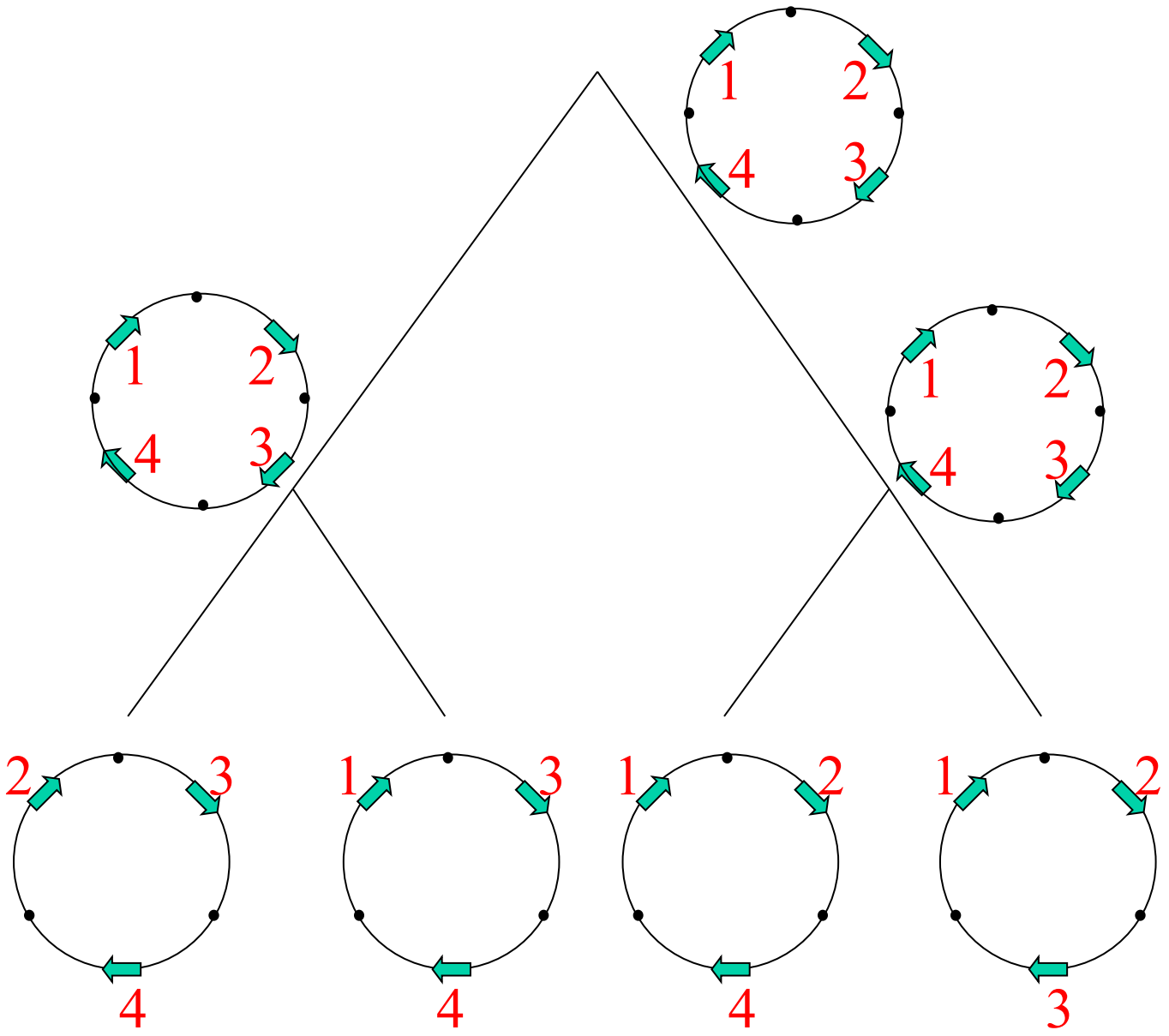
Новое решение первого примера: 6 переклеек – по одной на верхних рёбрах и по одной на трёх нижних.



Прежнее решение второго примера: 10 операций – по одной на верхних рёбрах и по 2 на нижних.



Новое решение второго примера: 4 операции – по одному удалению на нижних ребрах.



Биологический пример реконструкции хромосомных структур: рассмотрим пласто-мы девяти видов багрянок:

1. NC_021618 Grateloupia taiwanensis
2. NC_021075 Calliarthron tuberculosum
3. NC_020795 Chondrus crispus
4. NC_006137 Gracilaria tenuistipitata var. liui
5. NC_023133 Porphyridium purpureum
6. NC_004799 Cyanidioschyzon merolae strain 10D
7. NC_001840 Cyanidium caldarium
8. NC_007932 Pyropia yezoensis
9. NC_000925 Porphyra purpurea

В этих пластидах отберём **гены, связанные с 1й и 2й фотосистемами** и получим хромосомные структуры.

Порядок генов будет такой же, как на хромосоме; “–” указывает на комплементарную цепь, “С” в конце хромосомы указывают на цикличность.

Каждый геном состоит из одной циклической хромосомы.

Исходные хромосомные структуры

psaK -psaC psaI -psbJ -psbL -psbF -psbE -psaM psbA -psbY -psbV -psbX -psaJ -psaF psbD psbC psb28 psaE -psbH psbN -psbT -psbB -psbZ psbK -psaB -psaA -psaD psbI psaL |C

psbA -psbY -psbV -psbX -psaJ -psaF psbD psbC psb28 psaE -psbH psbN -psbT -psbB -psbZ psbK -psaB -psaA -psaD psbI psaL psaK -psaC psb30 psaI -psbJ -psbL -psbF -psbE -psaM |C

psaF psaJ psbX psbV psbY -psbA psbD psbC psb28 psaE -psbH psbN -psbT -psbB -psbZ psbK -psaB -psaA -psaD psbI psaL psaK -psaC psb30 psaI -psbJ -psbL -psbF -psbE -psaM |C

psaF psaJ psbX psbV -psbA psbD psbC -psbW psaE -psbH psbN -psbT -psbB psbK -psaB -psaA -psaD psbI psaL psaK -psaC psaI -psbJ -psbL -psbF -psbE -psaM |C

psbD psbC -psbV -psaB -psaA -psbK -psaE -psbZ psbA -psaJ -psaF psaD psbW -psbJ -psbL -psbF -psbE psbY -psaI -psaL psaM -psbX -psbI -psbH psbN -psbT -psbB -psaC -psaK |C

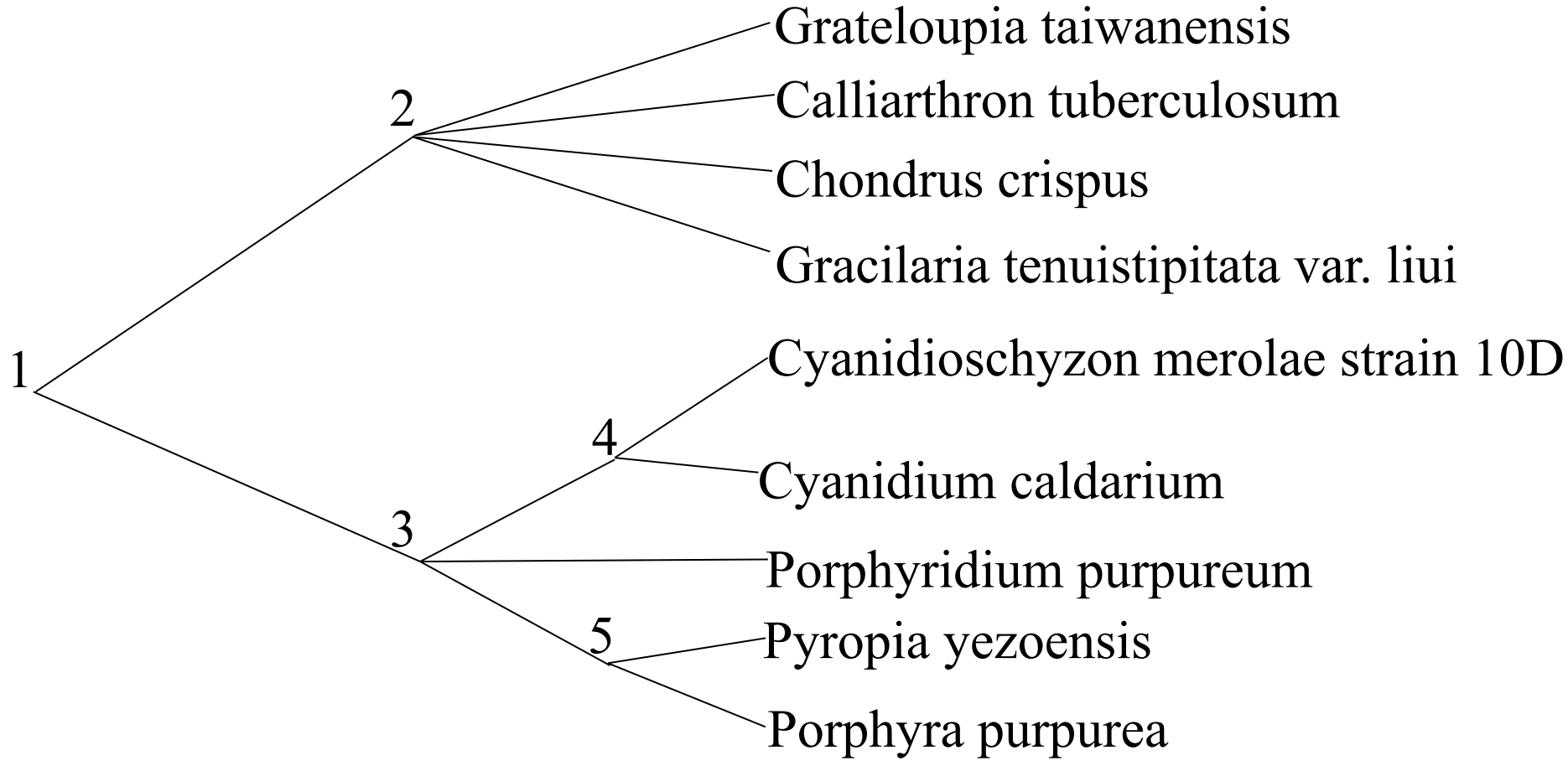
psaM psbY psbA -psaK -psaC psbD psbC psbW psbB psbT -psbN psbH -psaE psaA psaB -psbK psbZ -psaD psaF psaJ psbX psbV psaI -psbJ -psbL -psbF -psbE psaL -psbI |C

psbD psbC psaI -psbJ -psbL -psbF -psbE psaL -psbI psbA psaK -psaC psaE -psbH psbN -psbT -psbB -psbW -psaM psaF psaJ psbV psaD psbK -psaB -psaA |C

psaF psaJ psbX psbV -psbA psaL -psbI psaD psaA psaB -psbZ psbK psbB psbT -psbN psbH -psaE -psbW -psbC -psbD psaK -psaC psaI -psbJ -psbL -psbF -psbE -psaM |C

psaF psaJ psbX psbV psbA psaL -psbI psaD psaA psaB -psbZ psbK psbB psbT -psbN psbH -psaE -psbW -psbC -psbD psaK -psaC psaI -psbJ -psbL -psbF -psbE -psaM |C

Небинарное дерево видов таково:



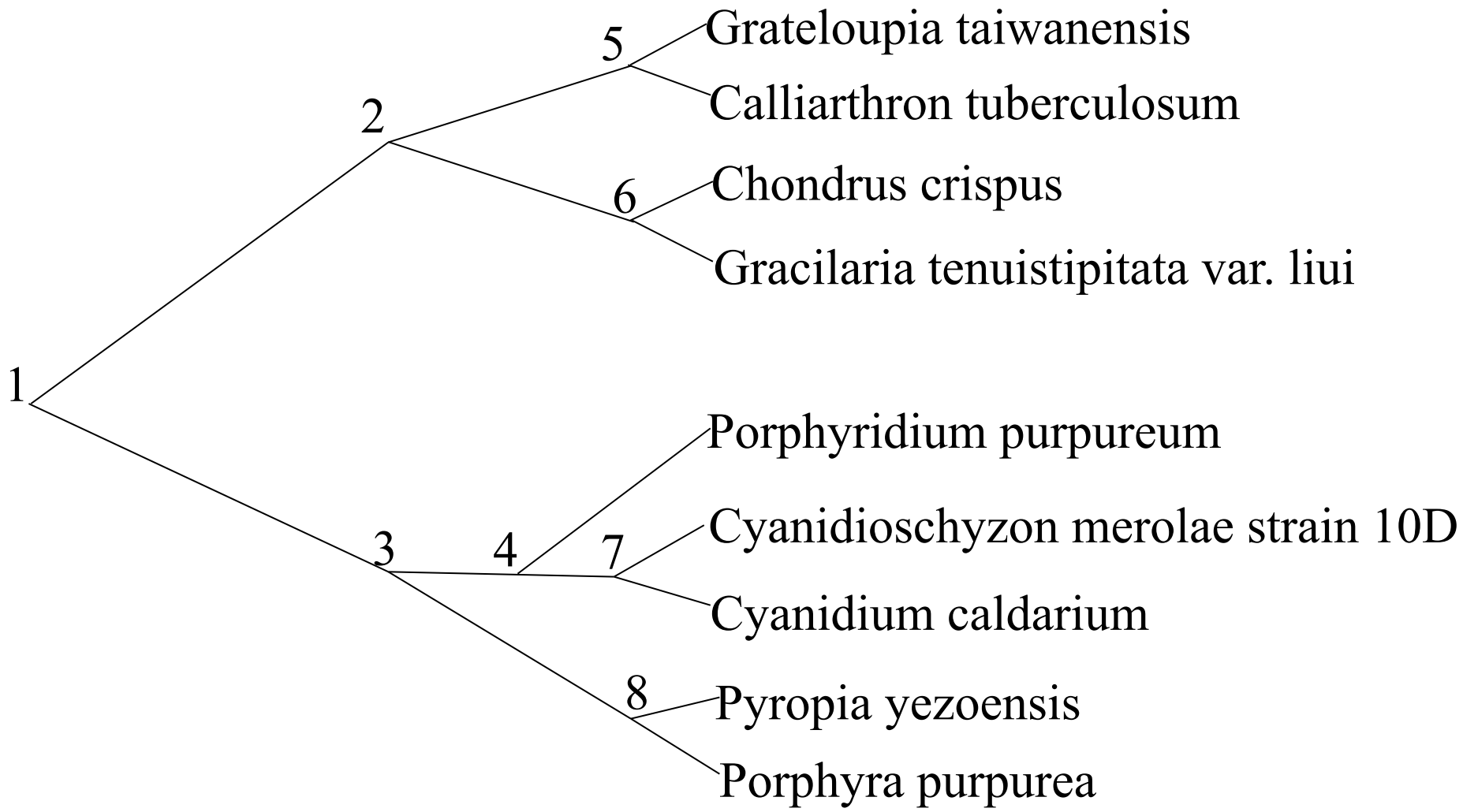
Получены предковые хромосомные структуры.

В вершине 4 две циклических хромосомы, а в остальных вершинах – по одной циклической хромосоме.

Полученные хромосомные структуры:

Вершина 1:
psaL –psbI psaD psaA psaB –psbK psbB psbT –psbN psbH –psaE –psbC –psbD
psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psaF psaJ psbV psbA |C
Вершина 2:
psaL psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psaF psaJ psbV –psbA
psbD psbC psaE –psbH psbN –psbT –psbB psbK –psaB –psaA –psaD psbI |C
Вершина 3:
psaL –psbI psaD psaA psaB –psbK psbB psbT –psbN psbH –psaE –psbC –psbD
psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psaF psaJ psbV psbA |C
Вершина 4:
psaL –psbI psaD psbK –psaB –psaA psbD psbC psaI –psbJ –psbL –psbF –psbE |C
psbB psbT –psbN psbH –psaE psaC –psaK –psbA –psbV –psaJ –psaF psaM |C
Вершина 5:
psaL –psbI psaD psaA psaB psbK psbB psbT –psbN psbH –psaE –psbC –psbD
psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psaF psaJ psbV psbA |C

Получено бинарное разрешение дерева видов:



Получены предковые хромосомные структуры.

В вершинах 4 и 7 по две циклических хромосомы, в остальных вершинах – по одной циклической хромосоме.

Полученные хромосомные структуры:

Вершина 1:
psaL –psbI psaD psaA psaB –psbK psbB psbT –psbN psbH –psaE –psbC –psbD psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psaF psaJ psbV –psbA |C

Вершина 2:
psaL psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psaF psaJ psbV –psbA psbD psbC psaE –psbH psbN –psbT –psbB psbK –psaB –psaA –psaD psbI |C

Вершина 3:
psaL –psbI psaD psaA psaB –psbK psbB psbT –psbN psbH –psaE –psbC –psbD psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psaF psaJ psbV –psbA |C

Вершина 4:
psaL –psbI psaM psaI –psbJ –psbL –psbF –psbE psbK –psaB –psaA –psbA –psbV –psaJ –psaF psaD |C; psbB psbT –psbN psbH –psaE –psbC –psbD psaK –psaC |C

Вершина 5:
psaL psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psbA –psbV –psaJ –psaF psbD psbC psaE –psbH psbN –psbT –psbB psbK –psaB –psaA –psaD psbI |C

Вершина 6:
psaL psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psaF psaJ psbV –psbA psbD psbC psaE –psbH psbN –psbT –psbB psbK –psaB –psaA –psaD psbI |C

Вершина 7:
psaL –psbI psaM psaI –psbJ –psbL –psbF –psbE |C; psaD psbK –psaB –psaA psbD psbC psbB psbT –psbN psbH –psaE psaC –psaK –psbA –psbV –psaJ –psaF |C

Вершина 8:
psaL –psbI psaD psaA psaB psbK psbB psbT –psbN psbH –psaE –psbC –psbD psaK –psaC psaI –psbJ –psbL –psbF –psbE –psaM psaF psaJ psbV –psbA |C

Из тех же девяти пластид были отобраны гены рибосомальных белков. Для соответствующих структур получены аналогичные результаты.

Спасибо за внимание!