

Article

Linear Time Additively Exact Algorithm for Transformation of Chain-Cycle Graphs for Arbitrary Costs of Deletions and Insertions [†]

Konstantin Gorbunov and Vassily Lyubetsky *

Institute for Information Transmission Problems of the Russian Academy of Sciences, Bolshoi Karetnyi, 19, 127994 Moscow, Russia; gorbunov@iitp.ru

* Correspondence: lyubetsk@iitp.ru; Tel.: +7-910-464-6917

[†] Dedicated to the 70-th anniversary of A. L. Semenov.

Received: 18 October 2020; Accepted: 5 November 2020; Published: 10 November 2020

Abstract: We propose a novel linear time algorithm which, given any directed weighted graphs a and b with vertex degrees 1 or 2, constructs a sequence of operations transforming a into b . The total cost of operations in this sequence is minimal among all possible ones or differs from the minimum by an additive constant that depends only on operation costs but not on the graphs themselves; this difference is small as compared to the operation costs and is explicitly computed. We assume that the double cut and join operations have identical costs, and costs of the deletion and insertion operations are arbitrary strictly positive rational numbers.

Keywords: discrete optimization; exact algorithm; additively exact algorithm; graph transformation; graph of degree 2; chain-cycle graph; operation cost; minimization of total cost

1. Introduction and Basic Definitions

We consider the problem of constructing an algorithm for efficient solution of the below problem. We are given directed graphs a and b in which each vertex has degree either 1 or 2 and each edge is assigned with its *unique* name, a natural number (in this sense, a graph is referred to as a weighted graph with unique names). We consider a vertex in a graph as two *joined* (identified) endpoints of the adjacent edges. The following operations over such graphs are well known: cut any vertex (*Cut*) or join two currently free (i.e., of degree 1) ends (*OM*); cut a vertex and join one of the thus formed free ends with any currently free end (*SM*); cut two vertices and join the four thus formed free ends (*DM*). The latter two operations are compositions of the two former ones, but they are considered as independent operations. These four operations were defined in [1] and are traditionally referred to as *DCJ* (i.e., double cut and join) *operations*; they were depicted in [2] (Figure 1).

There are two more operations: *remove* (*Rem*) a connected fragment of edges with names in a but not in b or, vice versa, *insert* (*Ins*) such a fragment with names not in a but in b . When removing a fragment, the arising free ends are joined; when inserting a fragment, first, some vertex is cut (if it is not extreme) and, then, after the insertion, two pairs of arising free ends are joined. These operations are analogous to standard deletion of a subword in a word and to insertion of a word as a subword. Each operation is assigned with its cost, a strictly positive rational number.

It is required to find a sequence (composition) of these operations with the minimum total cost which transforms the given graphs a and b , the first into the second. Such a sequence is said to be the shortest. In [2], previously obtained results were stated in detail and references on this problem were presented. Among these references, we point out the two latest papers [3,4], which, in turn, contain further historical references. The previous results concerning this problem and including its applied

aspects, were overviewed in [5] (Chapter 10) and [6]. A principal distinction between our Theorem 1 and preceding results is the fact that we do not assume the condition of equal costs of deletion and insertion operations, which essentially simplified the problem. In our paper, we assume no restrictions on costs of these operations. As before, we still keep the condition of equal costs of DCJ operations. Thus, in this paper we prove the following:

Theorem 1. *If DCJ operations have the same cost w and the deletion and insertion operations have arbitrary costs w_a and w_i , then the **Algorithm** described below in Sections 2, 4, and 5 outputs an additively shortest sequence of operations and has linear time complexity.*

We denote the additive exactness constant by k . The proof of the exactness splits into **three cases**: (I) both costs w_a and w_i are not less than w (then, $k \leq 2w$), (II) both are not greater than w (then $k = 0$), or (III) one of them is less and the other greater than w (an expression for k is given at the beginning of Section 4). From the description of a problem equivalent to this (Section 2, Stage 0), it follows that the two possibilities in the last case are symmetric and, therefore, do not differ from each other. In each of the cases, the description of the Algorithm, given in Section 2, slightly changes. The linear time complexity of each of these versions of the Algorithm directly follows from its description. The arXiv publication [2] is preparatory to the present paper and contains figures illustrating the algorithm operation.

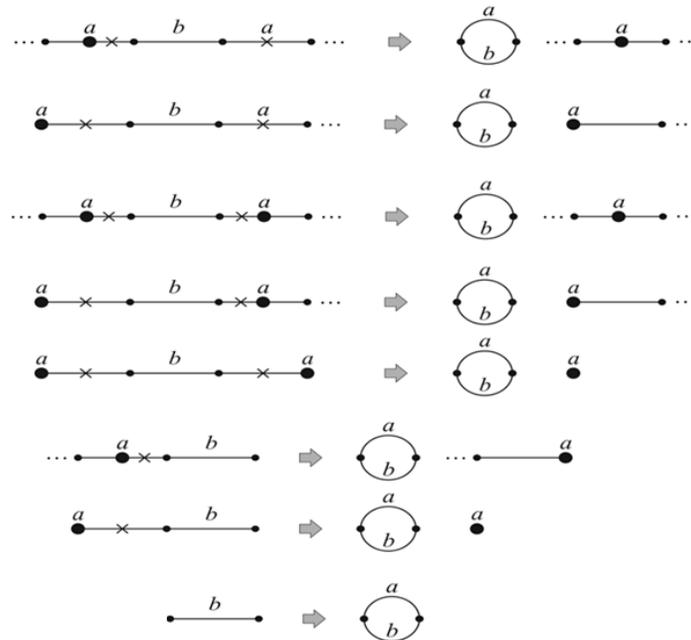


Figure 1. Cutting out a conventional b -edge. Singular vertices are marked by large circles. The cases differ by the form of edges adjacent to the edge to be cut out. From top to bottom: singular nonhanging and conventional; hanging and conventional; both singular nonhanging; hanging and singular nonhanging; both hanging; only one singular nonhanging; only one hanging; no adjacent edges.

2. Description of the Algorithm for Case II

We list the stages of this algorithm, which are then explained in sequence and in detail below, in this section (see also Table 1).

Stage 0: Transform initial given graphs a and b into the new graph, denoted by $a + b$ below.

Stage 1.: Cut out conventional edges in $a + b$.

Stage 2.0: Solve the integer linear programming problem which outputs the set of disjoint elements (each of them consisting of pairs, triples, or quadruples of chains of the current graph) with the largest aggregate quality.

Stage 2.1: For each element in this set, perform the interaction between its chains; the interaction is uniquely determined by the element and combines the chains of the element into a single chain.

Stage 3: Circularize chains of sizes strictly larger than 0 into cycles, and then break all cycles into many cycles of size 2; then, remove all singular vertices and loops.

Now, we pass to a detailed description of each stage.

Stage 0: An initial pair of graphs a and b is transformed in linear time into a new («breakpoint») graph $a + b$ such that the original problem is equivalent to the problem of reducing $a + b$ to the simplest form, referred to as final. A *final form* is a graph consisting of cycles of length 2 (one edge of each cycle is labeled by a , and the other, by b) and isolated conventional vertices; the definition of «conventional» is given below. The proof of the equivalence of these two problems literally repeats the proof of Corollary 5 in [7]; that proof used the equality of costs of DCJ operations only, which is assumed in Theorem 1. In [7], there was also given a linear time algorithm that transforms a solution of the second problem into a solution of the first (and vice versa).

The definition of the graph $a + b$ is given in [7]; in other variants, it was known from earlier works, for instance, from [8,9]. We recall the definition from [7], which describes vertices and edges of $a + b$. Vertices in $a + b$ are all endpoints of edges that occur in both a and b (they are denoted by n_i , where n is the edge name, $i = 1$ for a tail of an edge, and $i = 2$ for its head) and also vertices uniquely corresponding to every maximal connected segment of edges («block») in either $a \setminus b$ or $b \setminus a$, which we label by a or b , respectively. Vertices of the first type are referred to as *conventional*, and those of the second type are referred to as *singular*. Edges in $a + b$ connect vertices whenever the latter are joined in a or in b or if an extremity of a block is joined with a vertex in a or in b ; edges are labeled by a or b , respectively. Edges of the first type are said to be *conventional*, and those of the second type are said to be *singular*. If a singular endpoint in $a + b$ is of degree 1, then both the edge and the endpoint itself are said to be *hanging*.

An operation o' over $a + b$ is defined through an operation o over a by commutativity: $o'(a + b) = o(a) + b$. Thus, over a breakpoint graph, the five operations listed below are allowed, which are shown in Figure 5 in [2]. In fact, the number of operations is six, since the fifth one, removal of a singular vertex, divides into removal of an *a-singular* vertex, which corresponds to deleting a block in a , and removal of a *b-singular* vertex, which is inverse to inserting a block in b .

Double intermerging (DM): deletion of two edges with the same label and joining the four thus formed endpoints by two new edges with this label. If the operation involves a loop, then its vertex is regarded as having two endpoints. If two adjacent singular vertices are formed, then the edge connecting them is deleted and they are *merged* into a single vertex.

Sesquialteral intermerging (SM): deletion of an edge and adding an edge with the same label connecting one of the thus formed free ends with a free end of an edge with an alternative label, or with a hanging or singular isolated vertex with the same label (with possible *merging* of adjacent singular vertices).

Single merging (OM): adding an a -edge between free vertices such that each of them is either a conventional vertex incident to a b -edge or an isolated vertex, a -hanging or a -isolated, with possible *merging* of adjacent singular vertices. A similar definition is given with b instead of a .

Cut (Cut): deletion of any edge.

The above-listed operations are referred to as DCJ (or sometimes *standard*) operations. A supplementary operation is *removal (Rem)* of a singular vertex. Namely, if it is of degree 2, it is removed and the edges incident to it are joined into an edge with the same label; if it is hanging, it is removed together with the edge incident to it; if it is isolated or has a loop, the vertex and the loop are removed.

Each standard operation is assigned with its original *cost* w ; removal of an a -singular vertex has original *cost* $w_a = w_a$, and removal of a b -singular vertex has original *cost* $w_b = w_b$.

An inclusion-maximal connected fragment of conventional edges is called a *segment*; depending on its length, it can be either even or odd.

The size of a component in $a + b$ is the number of conventional edges plus half the number of singular nonhanging edges in it. For conventional isolated vertices and loops, the size is defined to be 0, and, for singular isolated vertices, the size is defined to be -1.

Our algorithm successively generates graphs G starting from $a + b$; all these G are of the form $c + d$ for their initial graphs c and d ; all these G together form a sequence that begins with $a + b$ and ends with a graph of a final form. Now, we pass to the description of the Algorithm, which consists of three consecutive stages.

Stage 1: From all components other than cycles of size 2, cut out conventional edges, i.e., apply a DM to a pair of edges adjacent to the edge to be cut out, or similarly apply an SM or OM if one or two of the adjacent edges do not exist. Such a derived operation is called *cutting out* (conventional edges) (Figure 1).

A chain of an odd (even) size is called *odd (even)*; 0 is an even number, and -1 is odd. The definition of a *type of a chain* plays a crucial role. First, assume that a chain *does not contain conventional edges*. Then, $1a$ is an odd chain with one hanging b -edge; $2a^*$ is an odd chain with two hanging b -edges; $2a'$ is a b -singular isolated vertex; $2a$ denotes type $2a^*$ or $2a'$. Type $3a^*$ is an odd chain without hanging edges, with two extremal a -edges, and having a b -singular vertex; $3a'$ is a chain aa ; $3a$ denotes type $3a^*$ or $3a'$. Type $1a^*$ is an even chain with one hanging a -edge that has a b -singular vertex; $1a'$ is a hanging a -edge; $1a$ denotes type $1a^*$ or $1a'$. Similar definitions are given with b instead of a . Type 2^* is an even chain with two hanging edges nonincident to each other; $2'$ is two hanging edges incident to a common conventional vertex; 2 denotes type 2^* or $2'$. Type 3 is an even chain without hanging edges but having singular vertices; 0 is a chain without singular vertices. The chain types were presented in Figure 6 in [2]. The type of a chain *with conventional edges* is defined as the type of a chain obtained by cutting them out; it does not depend on the order of cuttings (see Lemma 6 in [7]).

A *hanging extremity* of a chain not of type 0 is an extremity with an adjacent hanging edge or odd segment. After cutting out this segment, a hanging edge appears. A chain $2a$ (or $2b$, $3a$, $3b$) is a chain of type $2a'$ (respectively, $2b'$, $3a'$, $3b'$) if and only if it contains b -singular vertices only (a -singular only, a -singular only, b -singular only). A chain of type 2 is a chain of type $2'$ if and only if it contains a conventional vertex with only a -singular vertices on one side and only b -singular vertices on the other.

An *interaction* in G is a chain of operations successively applied to G . In Section 3, we introduce a key notion of $P(s)$, the *quality* of an interaction s . We also give there a convenient formula for computing it; see Equation (1b), which is consistent with Equation (1a). A term equality $1a + 1b = 1^*_b$ means that an interaction is applied to two chains of types $1a$ and $1b$ and outputs a chain of type 1^*_b . The same applies for other term equalities that are defined with the help of type designations, the + sign, and parentheses. Each interaction below corresponds to its term equality, which can be regarded as a designation (name) of this interaction. In square brackets, we give the interaction quality.

Stage 2: *2-interactions* are SM applied to two different chains with the following term equalities (a cut chain is always given the first; on the right-hand side, we do not present conventional isolated vertices): $1a + 1b = 1^*_b [w_a + w_b]$, $3a^* + 2b^* = 1_a [w_b]$, $3a^* + 2b' = 1_a [w_a]$, $3a' + 2b^* = 1_a [w_a]$, $3a' + 2b' = 1'_a [w_a]$, $3b + 2a = 1_b [w_b]$, $3 + 2 = 1^*_b [w_a + w_b - 1]$, $(1a + 2b^*) + 3 = 1^*_b [w_a + 2w_b - 1]$, $(1a + 2b') + 3 = 1^*_b [2w_a + w_b - 1]$, $(1b + 2a) + 3 = 1^*_b [w_a + 2w_b - 1]$, $(3a^* + 1b) + 2 = 1^*_b [w_a + 2w_b - 1]$, $(3a' + 1b) + 2 = 1^*_b [2w_a + w_b - 1]$, $(3b + 1a) + 2 = 1^*_b [w_a + 2w_b - 1]$, $1a + 2 = 2a^* [w_a + w_b - 1]$, $1b + 2 = 2b^* [w_a + w_b - 1]$, $3 + 1a = 3a^* [w_a + w_b - 1]$, $3 + 1b = 3b^* [w_a + w_b - 1]$, $(3b + 1a) + (1a + 2b^*) = 1^*_b [w_a + 3w_b - 1]$, $(3b + 1a) + (1a + 2b') = 1^*_b [2w_a + 2w_b - 1]$, $(3a^* + 1b) + (1b + 2a) = 1^*_b [w_a + 3w_b - 1]$, $(3a' + 1b) + (1b + 2a) = 1^*_b [2w_a + 2w_b - 1]$, $1a + (1a + 2b^*) = 2a^* [w_a + 2w_b - 1]$, $1a + (1a + 2b') = 2a^* [2w_a + w_b - 1]$, $1b + (1b + 2a) = 2b^* [w_a + 2w_b - 1]$, $(3b + 1a) + 1a = 3a^* [w_a + 2w_b - 1]$, $(3a^* + 1b) + 1b = 3b^* [w_a + 2w_b - 1]$, $(3a' + 1b) + 1b = 3b^* [2w_a + w_b - 1]$, $1a + 2b^* = 2 [w_b]$, $1a + 2b' = 2 [w_a]$, $1b + 2a = 2 [w_b]$, $3a^* + 1b = 3 [w_b]$, $3a' + 1b = 3 [w_a]$, $3b + 1a = 3 [w_b]$, $3 + ((3 + 2b^*) + 2a) = 1^*_b [w_a + 3w_b - 2]$, $3 + ((3 + 2b') + 2a) = 1^*_b [2w_a + 2w_b - 2]$, $(3a^* + (3b + 2)) + 2 = 1^*_b [w_a + 3w_b - 2]$, $(3a' + (3b + 2)) + 2 = 1^*_b [2w_a + 2w_b - 2]$, $(3a^* + 2) + 2 = 2a^* [w_a + 2w_b - 2]$, $(3a' + 2) + 2 = 2a^* [2w_a + w_b - 2]$, $(3b + 2) + 2 = 2b^* [w_a + 2w_b - 2]$, $3 + (3 + 2a) = 3a^* [w_a + 2w_b - 2]$, $3 + (3 + 2b^*) = 3b^* [w_a + 2w_b - 2]$, $3 + (3 + 2b') = 3b^* [2w_a + w_b - 2]$, $(3 + 2b^*) + 2a = 2^* [2w_b - 1]$, $(3 + 2b') + 2a = 2^* [w_a + w_b - 1]$, $3a^* + (3b + 2) = 3 [2w_b - 1]$, $3a' + (3b + 2) = 3 [w_a + w_b - 1]$, and OM with the equalities $1a + 1a = 3a^* [w_a + w_b - 1]$, $1b + 1b = 3b^* [w_a + w_b - 1]$.

Two additional 2-interactions are SM with the equalities $3a^* + 3b = 3 [w_b - w_a]$ and $2a + 2b^* = 2^* + 1'_a [w_b - w_a]$ (see Figure 2). Lemma 1 below demonstrates that the quality of a 2-interaction depends only on types of chains on the left-hand side of a term equality.

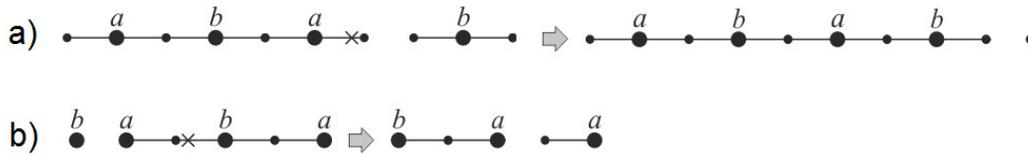


Figure 2. Interactions $3a^* + 3b = 3$ and $2a' + 2b^* = 2 + 1'_a$. In both cases, *sesquialteral intermerging* (SM) is applied.

Unlike the algorithms in [2,7], where interactions are performed in the same order as they are listed, now, the order of interactions is described in the following nontrivial way:

Notice 1. Denote the graph obtained after Stage 1 by G' . For each of the above-listed 2-interactions s , we call its *element* an unordered pair, triple, or quadruple (depending on the number of arguments in the composition of s , which we also denote by s) of chains in G' that have the same types as the types of arguments in s . The pairs, triples, and quadruples of such *types* themselves will be referred to as a *polytypes*; a type of a chain in a polytype may occur several times. For instance, for the interaction $f = (1a + 1b = 1'_b)$, its element is any pair of chains of types $1a$ and $1b$, and its polytype is the pair $\{1a, 1b\}$ of these types. To each element, precisely one 2-interaction corresponds, which is specified by its polytype. The *quality of an element* is defined to be the quality of this 2-interaction. By a *domain*, we call any set of elements (usually, from different interactions) where the elements are disjoint. The *quality of a domain* M is the aggregate quality of its elements. A domain with the maximum quality is called *maximal*.

Thus, Stage 2 consists of applying, simultaneously and independently of each other, 2-interactions corresponding to some maximal domain M for G' to elements of M .

Let us find a maximal domain M for G' . To this end, we use integer linear programming (ILP) with at most 51 variables and at most 10 nontrivial (i.e., not of the form $x \geq 0$) constraints. Namely, to each 2-interaction s we assign a nonnegative integer-valued variable x_s whose value must be equal to the number of elements of this 2-interaction in the sought-for domain M . This condition on the vector $\{x_s\}$ is expressed by the following linear relations: for every type t of a chain occurring in G' and corresponding to an argument of one of the 2-interactions, we impose the constraint $\sum_s c_{ts} \cdot x_s \leq l_t$,

where c_{ts} is the number of occurrences t in the arguments of any 2-interaction s (c_{ts} can be 0, 1, or 2), and l_t is the number of chains of type t in G' . Maximize the target function $F(\{x_s\}) = \sum_s P(s) \cdot x_s$, where

$P(s)$ is the quality of a 2-interaction s and the summation is over all 2-interactions. A solution to this ILP problem gives a maximal domain M .

Define *autonomous reduction* as the following sequence of operations: cut out all conventional edges. Circularize chains of sizes strictly larger than 0 into cycles using an OM or SM operation (after SM, there remains one extremal vertex or one extremal edge) (see Figure 3a). When circularizing a chain of type 2^* , choose a variant with joining two b -singular vertices; after circularizing a chain of type $3a^*$ or $3b^*$, cut out the arising conventional edge. Then, break all cycles into cycles of size 2 using a DM operation, which cuts out a cycle of size 2 with an a -singular vertex from a cycle (see Figure 3b). Remove all singular vertices and loops. The *autonomous cost* $A(G)$ of a graph G is defined as the total cost of the sequence of operations in the autonomous reduction of G (see Lemma 1 below).

Stage 3: Perform autonomous reduction of the remaining components to a final form.

End of the Algorithm description.

It follows from the proof presented in Section 3 that, for the graph G'' obtained after Stage 2, there exists no interaction with a strictly positive quality, which is by no means evident a priori.

Below, we consider the case $w_a \leq w_b$, since the description of the Algorithm (up to interchanging a and b) and the subsequent proof do not depend on which of the removal costs is smaller, w_a for an a -singular vertex or w_b for a b -singular vertex.

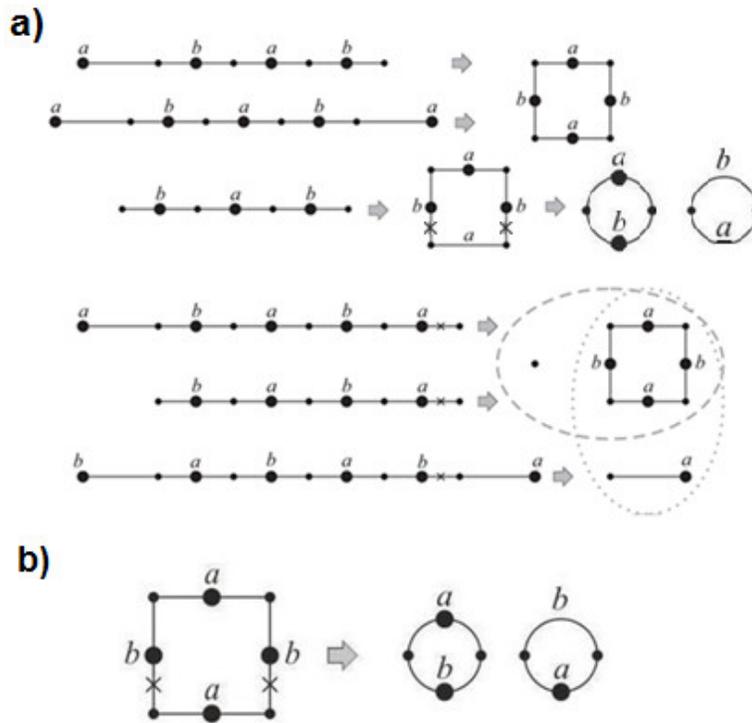


Figure 3. (a) Circularizing a chain. Chain types, from top to bottom: $1b$, $2b$, $3b$, $1a$, 3 , 2 . (b) Cutting out a cycle of size 2 from a cycle of size 4.

Table 1. Basic parameters predefined in or computed by the Algorithm for Case II. *OM*, single merging; *SM*, sesquialteral intermerging; *DM*, double intermerging.

Notation	Meaning
w	Equal costs of the <i>Cut</i> , <i>OM</i> , <i>SM</i> , and <i>DM</i> operations
$w_d = w_a$	Cost of the <i>Rem</i> operation
$w_i = w_b$	Cost of the <i>Ins</i> operation
x_s	Number of elements corresponding to the 2-interaction s in the sought-for domain M
t	Type of an argument in any 2-interaction; this t is one of the following types: $1a, 1b, 2a, 2b^*, 2b', 3a^*, 3a', 3b, 2, 3$.
c_{ts}	Number of occurrences of the type t in arguments of the 2-interaction s
l_t	Number of chains of the type t in the breakpoint graph G'
$P(s)$	Quality of the 2-interaction s

3. Proof of Theorem 1 (Case II)

First, we make an obvious remark concerning the **linear runtime** of the Algorithm. For Stage 0, this follows from the fact that $a + b$ is constructed by one-time examination of all components in a and b . Stage 1 requires one-time examination of all components in $a + b$. The number of interactions executed at Stage 2 is linear, since each of them reduces the number of chains in the corresponding graph, and each interaction is executed in constant time. Similarly, the number of operations executed at Stage 3 is linear, since each of them reduces the number of singular vertices or the number of chains. Linear runtime of solving the ILP problem at Stage 2 follows from [10], where it was shown that the time required for solving an ILP problem with a fixed number of variables and constraints is

polynomial in the logarithm of the maximum absolute value of a coefficient of the problem. In our problem, this coefficient is not greater than the problem size.

The rest of this section is devoted to the proof of additive exactness of the Algorithm. If, when executing a standard operation, singular vertices are joined, the operation is said to be *special*; otherwise, it is *nonspecial*; a removal operation is special by definition.

Clearly, we may assume that $w = 1$. For a graph G , we use the following *notation*: d is the total size of all components in it (we call it the *size* of G), f is the number of odd chains, c is the number of cycles (excluding loops), B is the number of singular vertices, S is the sum of integral parts of halved segment lengths plus the number of extremal (on a chain) odd segments minus the number of cyclic segments, D is the number of chains of types $1a$, $1b$, $3a$, $3b$, and 3 , and K_b is the number of components containing a b -singular vertex.

Lemma 1. *Let w_a and w_b be the removal costs for singular a - and b -vertices, $w_a \leq w_b$, and let all other operations have cost 1. Then, the autonomous cost $A(G)$ of a graph G is*

$$A(G) = (1 - w_a) \cdot (0.5d + 0.5f - c) + w_a \cdot (B + S + D) + (w_b - w_a) \cdot K_b. \tag{1a}$$

Proof. Denote the right-hand side of this equality by $A'(G)$. Let us check the equality for each component of the graph separately and then sum up the obtained equalities. Denote by I_b the indicator function which is 1 if G contains a b -singular vertex and 0 otherwise. \square

(1) For a conventional isolated vertex, the equality is trivial, since $A'(G)$ is equal to w_a for an a -loop and w_b for a b -loop. For a singular isolated vertex (odd chain), we have $d = -1$, $f = 1$, and $B = 1$; for a b -isolated vertex, $K_b = 1$ and the other quantities are zero. $A'(G)$ is equal to w_a for an a -isolated vertex and w_b for a b -isolated vertex, as well as $A(G)$.

(2) For a cycle without singular edges (and hence without singular vertices), we have $d > 0$ and d even, $c = 1$, $S = 0.5d - 1$; the other quantities are zero, and $A'(G) = 0.5d - 1$. Autonomous reduction includes only cutting out conventional edges by a DM operation. Each cutting reduces a large cycle by 2 edges; the last cycle requires no operation. Therefore, $0.5d - 1$ DM operations are executed.

(3) For a cycle with singular edges, we have $d > 0$ and d even, $B > 0$, $S \geq 0$, $c = 1$, $K_b = I_b$, and the other quantities are zero; $A'(G) = (1 - w_a)(0.5d - 1) + w_a(B + S) + (w_b - w_a)I_b$. When cutting out conventional edges, the size d of a graph does not change. Each cutting out a conventional edge or a cycle of size 2 reduces the size of a large cycle by 2; the last cycle requires no operation. Thus, $0.5d - 1$ DM operations are executed. The number of nonspecial operations among them is S , since every segment of length l for an even l requires $0.5l$ nonspecial cuttings, and, for an odd l , $l - 0.5(l - 1)$ nonspecial cuttings and one special cutting. Therefore, the number of Rem operations is $B - (0.5d - 1 - S)$ (the number of b -removals among them is I_b), and $A(G) = 0.5d - 1 + w_a(B - (0.5d - 1 - S) - I_b) + w_b I_b = A'(G)$.

(4) For an odd chain without singular edges, we have $d > 0$ and d odd, $f = 1$, $S = 0.5(d + 1)$ (such a chain is an extremal odd segment), and the other quantities are zero. Its autonomous reduction requires only DM operations, i.e., cutting out conventional edges, and, at the end, when a conventional edge without adjacent conventional edges remains, one OM. Each DM separates a final ab cycle from a current chain, and the chain length (equal to its size) reduces by 2. Therefore, $0.5(d + 1)$ operations are required in total, which equals $A(G)$, since the cost of a standard operation is 1. For $A'(G)$, we obtain the same quantity.

(5) For an odd chain with singular edges, we have $d > 0$ and d odd, $f = 1$, $B > 0$, D is either 0 or 1, $S \geq 0$, $K_b = I_b$, and the other quantities are zero. Cutting out conventional edges requires S nonspecial DM or SM operations. After the cuttings, a chain either has an odd size strictly greater than 0 or turns into a singular isolated vertex. In the first case, the chain is circularized into a cycle by a special OM (if $D = 0$) or by a nonspecial OM (if $D = 1$), after which the obtained cycle is finalized by special operations (this cycle may contain only one conventional edge, namely, the one by which the chain was circularized into a cycle). Every standard operation in autonomous reduction increases the number of cycles in the graph by 1, an OM increases the graph size by 1, and a DM or SM does not change it. Thus, finally, the graph size becomes $d + 1$, and it contains $0.5(d + 1)$ cycles of size 2. In the

second case, the graph size remains to be d , and it also contains $0.5(d + 1)$ cycles of size 2 formed when making cuttings (a singular isolated vertex is of size -1 and $D = 0$). Since the initial graph had no cycles, $0.5(d + 1)$ standard operations are made in total, $S + D$ among them being nonspecial. Therefore, the number of Rem operations is $B - (0.5d + 0.5 - S - D)$ (the number of b -removals among them is I_b), and $A(G) = 0.5(d + 1) + w_a(B - (0.5d + 0.5 - S - D) - I_b) + w_b I_b = A'(G)$.

(6) For an even chain without singular edges, we have $d > 0$ and d even, $S = 0.5d$, and the other quantities are zero. Its autonomous reduction requires only DM operations, cuttings of conventional edges. Each of them separates an ab cycle from the current chain; therefore, the length of the chain, equal to its size, reduces by 2. Therefore, in total, $A(G) = 0.5d$ operations are executed. For $A'(G)$, we obtain the same quantity.

(7) For an even chain with singular edges, we have $d > 0$ and d even, $B > 0$, D is either 0 or 1, $S \geq 0$, $K_b = I_b$, and the other quantities are zero. Cutting out conventional edges requires S nonspecial DM or SM operations. After a cutting, the chain has an even size either strictly greater than 0 or equal to 0. In the first case, the chain is circularized by a special SM (if $D = 0$) or a nonspecial SM (if $D = 1$), after which the obtained cycle is finalized by special operations. Every standard operation in the autonomous reduction increases the number of cycles in the graph by 1, and a DM or SM does not change the size of the graph. Therefore, finally, the graph size remains to be d , and it contains $0.5d$ cycles of size 2. The same happens in the second case. Since the initial graph had no cycles, in total, $0.5d$ standard operations are applied, $S + D$ among them being nonspecial. Thus, the number of Rem operations is $B - (0.5d - S - D)$ (the number of b -removals among them is I_b), and $A(G) = 0.5d + w_a(B - (0.5d - S - D) - I_b) + w_b I_b = A'(G)$.

Recall that an interaction in G is a sequence s of operations successively applied to G . The *quality* $P(s)$ of an interaction s is defined as

$$P(G,s) = A(G) - A(s(G)) - c(s), \tag{1b}$$

where $s(G)$ is the graph obtained after applying the composition s to G , and $c(s)$ is the *total cost of operations* in s . The quality shows what total cost can be saved by using the interaction s as against the autonomous reduction of G , i.e., without using s . By examining all 2-interactions (see the example below), we check that $P(G,s)$ is determined by the *polytype* of the 2-interaction, consisting of the types of its arguments or, equivalently, by the corresponding term equality. Accordingly, the quality of an element α (for the substitution in s) is determined by its polytype, which coincides with the polytype of s itself.

In this way, we computed the qualities of 2-interactions that were presented above, after each of them, in square brackets. For instance, let us find the quality of the interaction $1a + 1b = 1b^*$ (Figure 4), $c(s) = 1$, and the operation is special (generally, all operations of Stage 2 are special). This 2-interaction does not change the values of d , c , and S , reduces B and K_b by 1, and reduces D and f by 2. Hence, $P(s) = w_a + w_b$.



Figure 4. Interaction $1a + 1b = 1b^*$, an SM operation.

Let E be any reducing sequence for a graph G , which is arbitrarily divided into connected fragments of operations; we call them interactions here and denote by s . Denote by $T(R,E)$ the *total cost of operations* in a reducing subsequence which starts in E with some intermediate graph R and continues to a final form. Denote by $P(R,E)$ the aggregate quality of all interactions in E starting from R , where $P(s)$ is formally defined in Equation (1b).

Lemma 2. For any graph G , we have $T(G,E) = A(G) - P(G,E)$.

Proof. It suffices to check $A(R) - T(R,E) = P(R,E)$. This is proven by induction on the number of interactions from the end of the sequence E . If R is of a final form, all the three quantities are zero. Let $A(R') - T(R',E) = P(R',E)$ for a subsequence starting with a graph R' next to R . Let s be an interaction taking R into R' , $c(s)$ be the total cost of its operations, and $P(s)$ be its quality, defined formally by (1').

Then $P(R,E) = P(R',E) + P(s)$, $T(R,E) = T(R',E) + c(s)$, and $P(s) = A(R) - A(R') - c(s)$. Hence, $A(R) - T(R,E) = A(R') + c(s) + P(s) - T(R',E) - c(s) = A(R') - T(R',E) + P(s) = P(R',E) + P(s) = P(R,E)$. \square

If E is a sequence of operations constructed by the Algorithm, then it contains 2-interactions, and all other operations (i.e., those from Stages 1 and 3) by Equation (1b) have formal quality 0 (for uniformity, these operations can be viewed as formal interactions).

Notice 2. Thus, $P(G)$ is the *aggregate quality of 2-interactions* (equivalently, interactions in all stages of the algorithm), equal (!) to the quality of the maximal domain on chains in the initial $a + b$ (equivalently, on chains after all cuttings of Stage 1).

By Lemmas 1 and 2,

$$T = T(G) = (1 - w_a) \cdot (0.5d + 0.5f - c) + w_a \cdot (B + S + D) - P(G) + (w_b - w_a) \cdot K_b \tag{2}$$

is the *total cost of operations of the algorithm* on a graph G . Denote the first and second term in this expression by T' and T'' respectively.

Now, we use the following important equivalence:

Exactness of the algorithm is equivalent to the *triangle inequality*: for any operation o and any graph G , we have

$$c(o) \geq T(G) - T(o(G)),$$

where $o(G)$ is the result of applying o to G .

The proof is performed by induction on $C(G)$, the *minimum total cost* over all reducing sequences. If G is of a final form, then $T(G) \leq C(G)$ (let a reducing sequence be empty). Consider some nonempty shortest reducing sequence for G , and denote the first operation in it by o . By the induction hypothesis, $T(o(G)) \leq C(o(G))$. Then, by this inequality, $T(G) = c(o) + T(o(G)) \leq c(o) + C(o(G)) = C(G)$, i.e., $T(G) = C(G)$. The converse is even simpler, but we use only this implication. Note that this argument works for any algorithm, not only for the one described above.

Thus, to complete the proof of Theorem 1, it remains to check the triangle inequality for each of the above-mentioned operations o .

1. o is the Rem operation. When passing from G to $o(G)$, B reduces by 1. Consider several cases.

1.1. An isolated singular vertex is removed (chain of type $2a'$ or $2b'$). Then S , D , and c do not change; d increases by 1; f reduces by 1. P does not increase when removing any chain. Therefore, $T' + T'' - P$ reduces by at most w_a . If a b -vertex is removed, then K_b reduces by 1 and T reduces by at most w_b .

1.2. A singular vertex is removed from a cycle, or a loop is removed. Then, S does not reduce.

1.3. An interior singular vertex (i.e., on both sides of it there are other singular vertices) is removed from a chain. The type of the chain does not change, changes from $3b^*$ to $3b'$, or changes from $2a^*$ to $2a'$. Then, P does not increase, since no element increases its quality when making this change.

1.4. A hanging vertex which is the only singular vertex in a chain is removed from the chain. Then, S and D do not reduce, and K_b reduces by 1 if a b -vertex is removed.

1.5. A hanging vertex that is not unique in a chain is removed from the chain. If, when passing from G to $o(G)$, S does not change (the segment adjacent to the hanging edge is even), then the hanging extremity becomes nonhanging, and the following changes in the type of the chain are possible: $1a$ changes into $3a$, $1b$ changes into $3b$, 2 changes into 1 , $2a$ changes into $1a$, $2b$ changes into $1b$, or 1 changes into 3 . In the first three cases, D does not change and P does not increase. Indeed, all elements containing type $1a$ do not increase their quality when making the change from $1a$ to $3a$; the same applies to the two other changes. In the last three cases, D increases by 1, P either does not change or increases by at most w_a (consider the inverse change), and all other quantities do not change. Then, $T' + T'' - P$ reduces by at most w_a . If S increases by one (the segment adjacent to the hanging edge is odd), then the hanging extremity remains to be hanging, and the type of the chain does not change. Therefore, D and P do not change.

1.6. A nonhanging exterior singular vertex (on the right or on the left of it there are no other singular vertices) is removed from a chain, and it is the only singular vertex in the chain. Then, the chain has one of the types $3a$, $3b$, $2a$, $2b$, or 1 , since, after cutting out conventional edges, we obtain a

chain with one singular vertex, and P does not increase. If the chain is of type $3a$ or $3b$, then S increases by 1, since both segments adjacent to the removed vertex are even and the resulting segment is odd, and D reduces by 1. If the chain is of type $2a$, $2b$, or 1, then D and S do not change (in the cases of $2a$ or $2b$, two odd segments are replaced with one odd; in the case of 1, one even and one odd segments are replaced with one even). Therefore, $T' + T'' - P$ reduces by at most w_a .

1.7. A nonhanging exterior singular vertex is removed from a chain, and it is not a unique singular vertex in the chain. If, when passing from G to $o(G)$, S does not change (the extremal segment is odd, and the next one is even), then the hanging extremity becomes nonhanging, and we repeat the arguments from case 1.5. If S increases by 1, then either the type of the chain does not change (then, $T' + T'' - P$ does not change) or (if both segments adjacent to the removed vertex are even) the nonhanging extremity becomes hanging. Then, one of the following changes in the type of the chain is possible: $1a$ changes into $2a$, $1b$ changes into $2b$, 3 changes into 1, $3a$ changes into $1a$, $3b$ changes into $1b$, or 1 changes into 2. In the first three cases, D reduces by 1 and P does not increase. In the last three cases, D does not change and P either does not change or increases by at most w_a . Thus, in all cases, $T' + T'' - P$ reduces by at most w_a .

2. o is an OM.

2.1. Extremities of one chain are joined. This is possible for odd chains only. Then, d increases by 1, f reduces by 1, and c either increases by 1 or does not change, such that T' either reduces by $1 - w_a$ or does not change. Consider possible chain types. For a chain of type 0, S reduces by 1, since an odd segment turns into a cyclic one of length greater by 1. Then, T reduces by 1 (now, $c(o) = 1$). For a chain of type $1a$ or $1b$, B and S do not change, D reduces by 1, and P either does not change or reduces by at most $w_a + w_b$. Therefore, T changes by at most 1. For types $2a$ or $2b$, D does not change and P either does not change or reduces by at most w_b . If at both ends of the chain there are hanging edges, then after joining their endpoints, B reduces by 1 and S does not change. If at least on one side there is an odd extremal segment, then B does not change and S reduces by 1. For types $3a$ or $3b$, B and S do not change, D reduces by 1, and P either does not change or reduces by at most w_b . Therefore, T changes by at most 1.

2.2. Extremities of different chains are joined. In Table 2, we present results of joining the endpoints of nonhanging edges labeled with a or of hanging edges labeled with b (the added edge is labeled by b). Such a join is referred to as a b -join (similarly for an a -join). The result of an a -join coincides with the result of a b -join obtained by interchanging the labels a and b in chains.

Define *types of extremities*: $0a$ is an extremity of an odd chain of type 0, 0 is an extremity of an even chain of type 0, $1a$ is a hanging extremity of a chain of type $1a$, $1a'$ is a nonhanging extremity of a chain of type $1a$, $2a^*$ is an extremity of a chain of type $2a^*$ (respectively, $2a'$), $3a^*$ is an extremity of a chain of type $3a^*$ (respectively, $3a'$), 1 is a hanging extremity of a chain of type 1 with singular b -vertices, $1'$ is a nonhanging extremity of a chain of type 1 with singular b -vertices (respectively, $1 \setminus b$ and $1' \setminus b$ are extremities of chains of type 1 without singular b -vertices), 2 is an extremity of a chain of type 2, and 3 is an extremity of a chain of type 3.

Denote by ΔX the *increment* of X , i.e., the difference of values of X when passing from G to $o(G)$, $\Delta X = X(o(G)) - X(G)$. In Table 2, we present *number segments* which contain all possible increments of T (in parentheses) and P (in square brackets) from Equations (1a)–(2) when joining an extremity of a chain with the row name and an extremity of a chain with the column name, i.e., when passing from G to $o(G)$. These segments are merely lower and upper estimates for T and P , which is sufficient for the following step. The increments are given as functions of the costs w_a and w_b . At the bottom of the same cell, we give the type of the chain obtained as a result of this passing, i.e., joining the initial chains. For instance, joining the extremity $2a^*$ or $2a'$ with any extremity has the same increment of all arguments in Equation (1a) except for ΔK_b , and the corresponding ΔP is given in the cells of row $\{2a^*, 2a'\}$ in Table 2. Here, $w_{b,a}$ is equal to w_b for $2a^*$ and w_a for $2a'$. Similarly, this applies to the pairs $3a^*$ and $3a'$, 1 and $1 \setminus b$, and $1'$ and $1' \setminus b$. Next, (0) means that the corresponding quantity “does not change”, $(-w_a)$ means “reduces by at most w_a ”, $(+w_a)$ means “increases by at most w_a ”, $(1 - w_a)$ means “increases by at most $1 - w_a$ ”, etc.; an ellipsis denotes numbers between the ends of the given number segments.

Bounds for ΔT are easily computed through bounds for ΔP and increments of other arguments in Equation (2), with the latter being easy to find.

Thus, the problem consists precisely of estimating ΔP . For that, we use a functional $P^-(M, M_1)$, where M is a set consisting of two chains in G whose types are the one from the row name and the one from the column name, while M_1 consists of a single chain in G with the name given in the corresponding cell. The functional itself is the *minimum* increment of P when replacing M with M_1 over all graphs G containing M . Given the functional $P^-(M, M_1)$, one can compute a functional $P^+(M, M_1)$ as a function of w_a and w_b which is the *maximum* increment of P when replacing M with M_1 over all graphs G containing M . Specifically, we have the equality $P^+(M, M_1) = -P^-(M_1, M)$. Note that Table 2 is symmetric around the main diagonal.

Table 2. Results of joining extremities of chains with types specified in a row and column; the type of resulting chain is given at the bottom of a cell.

	$0a$	0	$1a$	$1a'$	$2a^*, 2a'$	$3a^*, 3a'$	$1, 1 \setminus b$	$1', 1 \setminus b$	2	3
$0a$	(0) [0] $0a$	(0) [0] 0	(0) [0] $1a$	(0) [0] $1a'$	(0) [0] $2a^*, 2a'$	(0) [0] $3a^*, 3a'$	(0) [0] 1	(0) [0] 1	(0) [0] 2	(0) [0] 3
0	(0) [0] 0	(+1) [0] $0b$	(0...+1) [0...-1] 3	(0...+1) [0...-1] 2	(0...+ w_b) [0...- w_b] 1	(0...+ w_b) [0...- w_b] 1	(+1...1- w_b) [0...+ w_b] $3b^*, 3b'$	(+1...1- w_b) [0...+ w_b] $2b^*, 2b'$	(+1...0) [0...-1] 1b	(+1...0) [0...+1] 1b
$1a$	(0) [0] $1a$	(0...+1) [0...-1] 3	(-1...+ w_b) [1- w_b - w_b ...- w_b -1] $3a^*$	(- w_b ...+ w_b) [0...- w_b - w_b] $1a$	(- w_b ...0) [0...- w_b] $1a$	(- w_b ...+ w_b) [0...- w_b] $3a^*$	(- w_b ...1- w_b) [0...-1] 3	(- w_b ...+ w_b) [0...- w_b - w_b] 1	(-1...0) [1- w_b - w_b ...- w_b - w_b] 1	(- w_b ...+ w_b) [0...- w_b - w_b] 3
$1a'$	(0) [0] $1a'$	(0...+1) [0...-1] 2	(- w_b ...+ w_b) [0...- w_b - w_b] $1a$	(-1...+ w_b) [1- w_b - w_b ...- w_b -1] $2a^*$	(- w_b ...+ w_b) [0...- w_b - w_b] $2a^*$	(- w_b ...0) [0...- w_b] $1a$	(- w_b ...+ w_b) [0...- w_b - w_b] 1	(- w_b ...1- w_b) [0...-1] 2	(- w_b ...+ w_b) [0...- w_b - w_b] 2	(-1...0) [1- w_b - w_b ...- w_b - w_b] 1
$2a^*, 2a'$	(0) [0] $2a^*, 2a'$	(0...+ w_b) [0...- w_b] 1	(- w_b ...0) [0...- w_b] $1a$	(- w_b ...+ w_b) [0...- w_b - w_b] $2a^*$	(- w_b ...0) [0...- w_b] $2a^*$	(- w_b ...+ w_b) [w_b ...- w_b] $1a$	(- w_b ...0) [0...- w_b] 1	(w_b ...- w_b) [- w_b ...- w_b] 2	(- w_b ...0) [0...- w_b] 2	(- w_b ...+ w_b) [0...- w_b - w_b] 1
$3a^*, 3a'$	(0) [0] $3a^*, 3a'$	(0...+ w_b) [0...- w_b] 1	(- w_b ...+ w_b) [0...- w_b - w_b] $3a^*$	(- w_b ...0) [0...- w_b] $1a$	(- w_b ...+ w_b) [w_b ...- w_b] $1a$	(- w_b ...0) [0...- w_b] $3a^*, 3a'$	(w_b ...- w_b) [- w_b ...- w_b] 3	(- w_b ...0) [0...- w_b] 1	(- w_b ...+ w_b) [0...- w_b - w_b] 1	(- w_b ...0) [0...- w_b] 3
$1, 1 \setminus b$	(0) [0] 1	(+1...1- w_b) [0...+ w_b] $3b^*, 3b'$	(- w_b ...1- w_b) [0...-1] 3	(- w_b ...+ w_b) [0...- w_b - w_b] 1	(- w_b ...0) [0...- w_b] 1	(w_b ...- w_b) [- w_b ...- w_b] 3	(1- w_b ...1-2 w_b) [0...+ w_b] $3b^*, 3b'$	(1- w_b + w_b ... [0...+ w_b] 1b	(1- w_b ...- w_b) [0...+1] 1b	(1- w_b ... [0...+ w_b] 1b
$1', 1 \setminus b$	(0) [0] 1	(+1...1- w_b) [0...+ w_b] $2b^*, 2b'$	(- w_b ...+ w_b) [0...- w_b - w_b] 1	(- w_b ...1- w_b) [0...-1] 2	(w_b ...- w_b) [- w_b ...- w_b] 2	(- w_b ...0) [0...- w_b] 1	(1- w_b + w_b ... [0...+ w_b] 1b	(1- w_b ...1-2 w_b) [0...+ w_b] $2b^*, 2b'$	(1- w_b ... [0...+1] 2b*	(1- w_b ...- w_b) [0...+1] 1b
2	(0) [0] 2	(+1...0) [0...+1] 1b	(-1...0) [1- w_b - w_b ...- w_b - w_b] 1	(- w_b ...+ w_b) [0...- w_b - w_b] 2	(- w_b ...0) [0...- w_b] 2	(- w_b ...+ w_b) [0...- w_b - w_b] 1	(1- w_b ...- w_b) [0...+1] 1b	(1- w_b ... [0...+1] 2b*	(1- w_b ...-1) [- w_b ...2- w_b - w_b] 2b*	(- w_b ...+ w_b) [1...1- w_b - w_b] 1b
3	(0) [0] 3	(+1...0) [0...+1] 1b	(- w_b ...+ w_b) [0...- w_b - w_b] 3	(-1...0) [1- w_b - w_b ...- w_b - w_b] 1	(- w_b ...+ w_b) [0...- w_b - w_b] 1	(- w_b ...0) [0...- w_b] 3	(1- w_b ... [0...+1] 1b	(1- w_b ...- w_b) [0...+1] 1b	(- w_b ...+ w_b) [1...1- w_b - w_b] 1b	(1- w_b ...-1) [- w_b ...2- w_b - w_b] 3b

Thus, to compute $P^-(M, M_1)$, we need Lemma 3 below, where M consists of different chains, denoted by 1 and 2, and M_1 consists of one corresponding chain, denoted by 3. If D is a maximal domain in a graph G containing M , we denote by G' the graph that contains all chains from M and all chains from elements in D which intersect with M . These elements themselves form a maximal domain D' in G' . Indeed, if D' is not maximal, strictly extend it to a maximal domain in G' , which together with elements in $D \setminus G'$ forms a domain in G with quality strictly greater than that of D , a contradiction.

Lemma 3. We have $P(o(G)) - P(G) \geq P(o(G')) - P(G')$.

Proof. Recall that such a difference was called the *increment* of the corresponding quantity. Denote the left-hand side of the inequality by d , and the right-hand side, by d' . Consider $D \setminus D'$ (elements of D that do not contain chains 1 and 2), D_1 , a maximal domain in $o(G')$, and D_2 , the result of replacing D' with D_1 in D . Then, $d \geq P(D_2) - P(D) = P(D_1) - P(D')$, since $P(D_2) = P(D_1) + P(D_2 \setminus D_1)$ and $P(D) = P(D') + P(D_2 \setminus D_1)$. \square

Thus, we want to find $\min_{G'} P(o(G')) - P(G')$. Let X be a set containing at most two polytypes α^* and β^* (one for each element of D'), and let Y be a set containing at most four of the types of which these polytypes are composed (types of the chains contained in the corresponding elements of D') and also containing the types of the initial chains 1 and 2. There exists a G'' for which these polytypes correspond to disjoint elements, and for such a G'' the pair $D'' = \{\alpha^*, \beta^*\}$ is a polytype of a domain in G'' , not necessarily maximal. In the last case, $\sum_{\alpha^* \in X} P(\alpha^*)$ is smaller than the quality $P(G'')$, and such

an X is inessential when searching for the minimum. Given G'' , we form the corresponding $o(G'')$ and find in it an actual maximal domain and the quality $P(o(G''))$, i.e., the minuend is definitely correct, and the subtrahend may be smaller than required. Note that $P(D) = P(D')$, as well as for other domains, since the quality of any domain does not change when changing chains but preserving their types. Through an exhaustive search over all such X , we find the desired lower bound; in this way, Table 2 is filled. Take note that the number of possible sets X is not greater than the squared number of interactions, and X contains at most eight types. Therefore, due to an exhaustive search over all X and finding a maximal domain in $o(G'')$, computing $P(M, M_1)$, takes constant time.

Note that if the initial chains are even, then T' increases by $1 - w_a$; otherwise, it does not change; simple observations of this kind are omitted in what follows. Let us show how seven cells in Table 2 are filled, with the other being filled similarly.

(1) Cell $1a, 1a$. After the join, we obtain a chain of type $3a^*$. If the initial chains have hanging edges, then B reduces by 1 and S does not change. If at least one of the initial chains has an odd extremal segment, then, vice versa, B does not change and S reduces by 1, D reduces by 1, and K_b reduces by 1. Therefore, $T + P$ reduces by $w_a + w_b$. Our functionals are as follows: $P(1a, 1a; 3a^*) = -w_a - 1$ and $P^+(1a, 1a; 3a^*) = \min(1 - w_a - w_b, 0)$. In other words, either P reduces by at least $w_a + w_b - 1$ but at most $w_a + 1$ (if $w_a + w_b - 1 > 0$), or P does not increase and reduces by at most $w_a + 1$ (otherwise). Therefore, we have $|\Delta T| \leq 1$.

(2) Cell $1a, 2$. After the join, we obtain a chain of type 1 (it is not involved in interactions). Similarly to case (1), we obtain that $T + P$ reduces by $w_a + w_b$. We have $P(1a, 2; 1) = -w_a - w_b$, $P^+(1a, 2; 1) = \min(1 - w_a - w_b, 0)$. In other words, either P reduces by at least $w_a + w_b - 1$ (if $w_a + w_b - 1 > 0$) but at most $w_a + w_b$, or P does not increase and reduces by at most $w_a + w_b$ (otherwise). Therefore, $|\Delta T| \leq 1$.

(3) Cell $2a, 3a$. After the join, we obtain a chain of type $1a$. Then, $T + P$ reduces by $w_b - w_a$ if the type of the $3a$ -chain is $3a^*$ and does not change if it is $3a'$. We have $P(2a, 3a^*; 1a) = -w_b$, $P(2a, 3a'; 1a) = -w_a$, $P^+(2a, 3a^*; 1a) = w_a$, and $P^+(2a, 3a'; 1a) = w_a$. In other words, P reduces by at most w_b in the first case and by at most w_a in the second and increases by at most w_a in both cases.

(4) Cell $2a, 1'$. After the join, we obtain a chain of type 2. Then, B , S , and D do not change. We have $P(2a, 1; 2) = -w_b$, $P^+(2a, 1; 2) = \max(w_a + w_b - 1, 0)$.

(5) Cell $1, 3$. After the join, we obtain a chain of type $3b$. The increment of $T + P$ is $1 - w_b$ if the chain of type 1 contains a b -vertex, and $1 - w_a$ otherwise. We have $P(1, 3; 3b) = \min(1 - w_a - w_b, 0)$ and $P^+(1, 3; 3b) = w_b$. Subtracting from the increment of $T + P$ three possible increments of P , equal to $1 - w_a - w_b$, 0, or w_b , we obtain a number of absolute value no greater than 1.

(6) Cell $2, 2$. After the join, we obtain a chain of type $2b^*$. The increment of $T + P$ is $1 - w_a - w_b$. We have $P(2, 2; 2b^*) = -w_a$ and $P^+(2, 2; 2b^*) = 2 - w_a - w_b$. Subtracting from the increment of $T + P$ all possible increments of P , equal to $-w_a$ or $2 - w_a - w_b$, we obtain a number of absolute value no greater than 1.

(7) Cell $2, 3$. After the join, we obtain a chain of type $1b$. The increment of $T + P$ is $1 - w_b$. We have $P(2, 3; 1b) = \min(1 - w_a - w_b, 0)$ and $P^+(2, 3; 1b) = 1$. Subtracting from the increment of $T + P$ all possible increments of P , equal to $1 - w_a - w_b$, 0 or 1, we obtain a number of absolute value no greater than 1.

Thus, the result for OM follows from the fact that, according to Table 2, T changes by at most 1. The next simple lemma is used in what follows.

Lemma 4. Let $X(G)$ be a function of a graph G , and let $\Delta X(G, G_1)$ be its increment when passing from G to G_1 . Then,

1. $\Delta X(G, G_1) = -\Delta X(G_1, G)$. In particular, if o is an operation, then $\Delta X(G, o(G)) = -\Delta X(o(G), G)$. If o is a composition of operations o_1 and o_2 (first o_1 and then o_2), then $\Delta X(G, o(G)) = \Delta X(G, o_1(G)) + \Delta X(o_1(G), o(G))$.
2. For any graphs G and G_1 and any operations o and o_1 , if $\Delta X(G, G_1) = \Delta X(o(G), o_1(G_1))$, then $\Delta X(G, o(G)) = \Delta X(G_1, o_1(G_1))$.

Proof. The first claim is obvious. The second follows from the equalities $\Delta X(G, o(G)) = \Delta X(G, G_1) + \Delta X(G_1, o_1(G_1)) - \Delta X(o(G), o_1(G_1)) = \Delta X(G_1, o_1(G_1))$. \square

3. o is a Cut. This operation is the inverse to OM; thus, the result follows from case 2 and Lemma 4.1.

4. o is an SM. Represent o as a composition of a cut and a join. By Lemma 4.1, the increment obtained in a composition of operations is the sum of increments obtained in each of them.

4.1. Reversal (i.e., rearrangement of a connected fragment of edges in reverse order at the same place) of an extremal fragment of the chain. Denote by 1 and 2 the extremities of the reversed fragment and, by $2'$, the arising free endpoint of the remaining chain. Consider two joins: the first is the inverse to the cut (the first operation of the reversal), and the second is the join from the reversal. They are either both a - or both b -joins. We go over all rows of Table 2 in the role of the endpoint $2'$ and over pairs of columns as the extremities 1 and 2; the latter are either of types $1a$ or $1a'$ or they are a pair of identical extremities of types $0a$, $2a^*$, $2a'$, $3a^*$, or $3a'$. The increment of T when making a cut is opposite to the increment of T when joining 2 and $2'$, as for any mutually inverse operations. By Lemma 4.1 it suffices to check the following: when joining the extremities 2 and $2'$ and extremities 1 and $2'$, we obtain increments with the absolute value of their difference no greater than 1.

Let us give examples; other pairs of joins are considered similarly. Cases of a -joins are considered analogously to b -joins.

(1) The pair of cells $1a, 1a$ and $1a, 1a'$ defines two joins: $2' = 1a$, $2 = 1a$, $1 = 1a'$. For the first of them, ΔT and ΔP are, respectively, $-1 + x$ and $1 - w_a - w_b - x$ with $0 \leq x \leq w_a + 1$; for the second, these increments are $-w_b + y$ and $0 - y$ with $0 \leq y \leq w_a + w_b$. The difference of increments of T is strictly greater than 1 in absolute value if $x - y > 2 - w_b$ or $x - y < -w_b$ (this region in the x, y plane is called *forbidden*). This difference is equal to the increment of T upon the reversal, and we want to show that the forbidden region is empty. The difference R of increments of P (the second minus the first) is $x - y + w_a + w_b - 1$. In the forbidden region, $R > w_a + 1$ or $R < w_a - 1$. However, R is equal to the increment of P resulting when making the reversal. The reversal result is obtained by replacing a chain of type $3a^*$ (the result of joining the extremities 2 and $2'$) with a chain of type $1a$ (the result of joining the extremities 1 and $2'$). This increment of P lies in the segment $[0, w_a]$, since $P(3a^*; 1a) = 0$ and $P(3a^*; 1a) = w_a$. Hence, the forbidden region is empty. Computation of $P(M; M_1)$ in the cases where $|M| = 1$ or $|M_1| = 2$ is performed as in Lemma 3.

(2) The pair of identical cells $1a, 3a$ defines two joins: $2' = 1a$, $2 = 3a$, $1 = 3a$. For these joins, ΔT is either $-w_b + x$ and $-w_b + y$ (if the $3a$ chain is of type $3a^*$) or $-w_a + x$ and $-w_a + y$ (if this chain is of type $3a'$). Although both joins are made according to the same cell, ΔT takes different values within the same segment because of other parameters of the graph. Respectively, ΔP is $0 - x$ and $0 - y$. The forbidden region, which is shown to be empty, is as follows: $x - y > 1$ or $x - y < -1$. We have $R = x - y$. However, R is equal to the increment of P upon the reversal (replacing a chain $3a^*$ (the result of joining the extremities 2 and $2'$) with a chain $3a^*$ (the result of joining the extremities 1 and $2'$)), i.e., it is zero.

4.2. Circularization of an extremal fragment of a chain, i.e., the composition of a cut of the chain with closing one of the parts into a cycle or loop by a join. Denote by 1 and 2 the extremities of the circularized fragment and, by $2'$, the arising free endpoint of the remaining part of the chain. Consider two joins: the first is the inverse to the cut (the first operation of the circularization), and the second is the join from the circularization. They are either both a - or both b -joins.

Let us make Table 3 for closing a chain into a cycle, analogous to Table 2, following the description given in case 2.1. In Table 3, for each odd type of a chain, we give an interval of possible

increments of T and the corresponding interval of increments of P when closing this chain into a cycle by a join.

Table 3. Increments of T and P when circularizing a chain by joining extremities.

0	1a, 1b	2a, 2b	3a, 3b
$(w_a - 1)(-1 \dots w_a + w_b - 1)$	$(-1 \dots w_a + w_b - 1)$	$(-1 \dots w_b - 1)$	$(-1 \dots w_b - 1)$
[0]	$[0 \dots -w_a - w_b]$	$[0 \dots -w_b]$	$[0 \dots -w_b]$

In Table 2 we go over all rows in the role of extremity $2'$ and all columns in the role of extremity 2, the latter for all types of extremities of odd chains. By Lemma 4.1, it suffices to check the following condition:

the absolute value of the difference of ΔT for joins of 2 and $2'$ and joins of 1 and 2 is at most 1. (3)

Let us give examples; other pairs of joins are considered similarly. Cases of a -joins are considered analogously to b -joins.

(1) Cell 0,1a (i.e., $2' = 0, 2 = 1a, 1 = 1a'$). Here, ΔT and ΔP are, respectively, $0 + x$ and $0 - x$, and, in Table 3, for chain 1a, these increments are $-1 + y$ and $0 - y$. The forbidden region for condition (3) is as follows: $x - y > 0$ or $x - y < -2$. The difference R of increments of P is $x - y$. In the forbidden region, we have, respectively, $R > 0$ or $R < -2$. However, R is equal to the increment of P when replacing a chain of type 3 (the result of joining the extremities 2 and $2'$) with a chain of type 0 (the result of joining the extremities 1 and 2) and cannot take these values. Therefore, the forbidden region is empty.

(2) Cell 1a,1a (i.e., $2' = 1a, 2 = 1a, 1 = 1a'$). Here, ΔT and ΔP are, respectively, $-1 + x$ and $1 - w_a - w_b - x$. The forbidden region for condition (3) is as follows: $x - y > 1$ or $x - y < -1$; $R = x - y + w_a + w_b - 1$. In the forbidden region, $R > w_a + w_b$ or $R < w_a + w_b - 2$. However, R is equal to the increment of P when replacing a chain of type 3a with a chain of type 1a and cannot take these values. Therefore, the forbidden region is empty.

(3) Cell 2,3a (i.e., $2' = 2, 2 = 3a, 1 = 3a$). Here, ΔT and ΔP are, respectively, $-w_b + x$ (or $-w_a + x$) and $0 - x$, and, in Table 3, for chain 3a, these increments are $-1 + y$ and $0 - y$. The forbidden region for condition (3) is as follows: $x - y > w_b$ or $x - y < w_b - 2$ (or: $x - y > w_a$ or $x - y < w_a - 2$); $R = x - y$. In the forbidden region, $R > w_a$ or $R < w_b - 2$. However, R is equal to the increment of P when replacing a chain of type 1 with a chain of type 2 and (since $P^+(1;2) = \max(w_a + w_b - 1, 0)$) cannot take these values. Therefore, the forbidden region is empty.

4.3. SM is applied to a cycle (or a loop) and a chain, i.e., a cycle or a loop is cut and the obtained chain is lengthened. This operation is the inverse to the preceding one (case 4.2); therefore, by Lemma 4.1, we have $|\Delta T| \leq 1$.

4.4. SM is applied to two chains: the first is cut, and the second is joined. Denote by 1 the arising free extremity in the cut chain, which is joined with an extremity of the second chain, denoted by $2'$. The other arising free extremity of the cut chain is denoted by 2. Consider two joins: the first of them is the inverse to the cut (the first operation in our SM), and the second is the final OM. They are either both a - or both b -joins. We go over all rows of Table 2 in the role of extremity 1 and over all pairs of columns in the role of extremities 2 and $2'$. By Lemma 4.1, it suffices to check the following: for joins of extremities 1 and 2 and extremities 1 and $2'$ we obtain increments with the absolute value of their difference no greater than 1.

Let us give examples; other pairs of joins are considered similarly. Cases of a -joins are considered analogously to b -joins.

(1) The pair of cells 1a,0 and 1a,1a (i.e., $1 = 1a, 2 = 0, 2' = 1a$). When making the join according to the first cell, we have $\Delta T = 0 + x, 0 \leq x \leq 1$. When making the join according to the second cell, $\Delta T = -1 + y, 0 \leq y \leq w_a$. The difference of these expressions (the second minus the first) is $y - x - 1$. Its absolute value is greater than 1 if $y - x > 2$ or $y - x < 0$ (the forbidden region). The corresponding difference R of increments of P is $x - y + 1 - w_a - w_b$, which, for $y - x > 2$, yields $R < -1 - w_a - w_b$, and for $y - x < 0, R > 1 - w_a - w_b$. However, R is equal to the increment of P when making the original SM operation. The

result is obtained by replacing chains of types 3 (joining the extremities 1 and 2) and 1a (the chain with extremity 2') with chains of types 3a* (joining the extremities 1 and 2') and 0 (the chain with extremity 2). The equalities $P^-(1a,3; 3a^*,0) = -w_a$ and $P^+(1a,3; 3a^*,0) = \min\{0, 1 - w_a - w_b\}$ imply that such values of R are impossible and that the forbidden region is empty.

(2) The pair $2a^*,1$ and $2a^*,1'$ (i.e., $1 = 2a^*, 2 = 1, 2' = 1'$). When making the join according to the first cell, we have $\Delta T = -w_b + x$. When making the join according to the second cell, $\Delta T = -w_b + y$. The difference of these expressions (the second minus the first) is $y - x$. The forbidden region is $x - y < -1$ or $x - y > 1$. The corresponding difference R of increments of P is $x - y + w_a$. In the forbidden region, $R < -1 + w_a$ or $R > 1 + w_a$. However, R is equal to the increment of P when replacing chains of types 1 (joining the extremities 1 and 2) and 1 (the chain with extremity 2') with chains of types 2 (joining the extremities 1 and 2') and 1 (the chain with extremity 2). The equalities $P^-(1,1; 2,1) = 0$ and $P^+(1,1; 2,1) = \max\{0, w_a + w_b - 1\}$ imply that the forbidden region is empty.

(3) The pair $1,1a$ and $1,1$ (i.e., $1 = 1, 2 = 1a, 2' = 1$). When making the join according to the first cell, we have $\Delta T = -w_b + x$. When making the join according to the second cell, $\Delta T = 1 - 2w_b + y$. The difference of these expressions is $y - x + 1 - w_b$. The forbidden region is $x - y < -w_b$ or $x - y > 2 - w_b$. The corresponding difference R of increments of P is $x - y + w_b$. In the forbidden region, $R < 0$ or $R > 2$. Under the SM, chains of types 3 and 1 are replaced with chains of types 1a and 3b. The equalities $P^-(3,1; 1a,3b) = w_b$ and $P^+(3,1; 1a,3b) = 1$ imply that the forbidden region is empty.

(4) The pair $(1,1 \setminus b), 1a$ and $(1,1 \setminus b), 3$ (i.e., $1 = 1$ or $1 = 1 \setminus b, 2 = 1a, 2' = 3$). When making the join according to the first cell, we have $\Delta T = -w_{b,a} + x$, where the index b occurs in the case $1 = 1$, and index a , in the case $1 = 1 \setminus b$. When making the join according to the second cell, $\Delta T = -w_{b,a} + y$. The forbidden region is $x - y < -1$ or $x - y > 1$. The corresponding difference R of increments of P is $x - y + 1$. In the forbidden region, $R < 0$ or $R > 2$. Under the SM, chains of types 1 and 3 are replaced with chains of types 1a and 1b. The equalities $P^-(1,3; 1a,1b) = \min(1, w_a + w_b)$ and $P^+(1,3; 1a,1b) = w_a + w_b$ imply that the forbidden region is empty.

(5) The pair $1,3a^*$ and $1,1'$ (i.e., $1 = 1, 2 = 3a^*, 2' = 1'$). When making the join according to the first cell, we have $\Delta T = -w_b + x$. When making the join according to the second cell, $\Delta T = 1 - w_b + w_a - y$. The forbidden region is $x + y < w_a$ or $x + y > 2 + w_a$. The corresponding difference R of increments of P is $x + y - w_a$. In the forbidden region, $R < 0$ or $R > 2$. Under the SM, chains of types 1 and 3 are replaced with chains of types 3a* and 1b. The equalities $P^-(1,3; 3a^*,1b) = w_b$ and $P^+(1,3; 3a^*,1b) = \min(1, w_a + w_b)$ imply that the forbidden region is empty.

(6) The pair $2,1a$ and $2,1$ (i.e., $1 = 2, 2 = 1a, 2' = 1$). When making the join according to the first cell, we have $\Delta T = -1 + x$. When making the join according to the second cell, $\Delta T = 1 - w_b - y$. The forbidden region is $x + y < 1 - w_b$ or $x + y > 3 - w_b$. The corresponding difference R of increments of P is $x + y - 1 + w_a + w_b$. In the forbidden region, $R < w_a$ or $R > 2 + w_a$. Under the SM, chains of types 1 and 1 are replaced with chains of types 1a and 1b. The equalities $P^-(1,1; 1a,1b) = w_a + w_b$ and $P^+(1,1; 1a,1b) = w_a + w_b$ imply that the forbidden region is empty.

(7) The pair $2,2$ and $2,3$ (i.e., $1 = 2, 2 = 2, 2' = 3$). When making the join according to the first cell, we have $\Delta T = 1 - w_b - x$. When making the join according to the second cell, $\Delta T = -w_b + y$. The forbidden region is $-x - y > 0$ or $-x - y < -2$. The corresponding difference R of increments of P is $-x - y + w_a + 1$. In the forbidden region, $R > w_a + 1$ or $R < w_a - 1$. Under the SM, chains of types 2b* and 3 are replaced with chains of types 1b and 2. The equalities $P^-(2b^*,3; 1b,2) = 0$ and $P^+(2b^*,3; 1b,2) = w_a$ imply that the forbidden region is empty.

5. o is a DM. Represent o as a composition of two cuts and two joins. One can easily check that, when making any operation, $B + S$ changes by at most 1. This property of an operation is called the $B + S$ property.

5.1. Reversal in a cycle or in a chain. Let both cuts be interior (i.e., on both sides of each cut there are singular vertices); then, P and D do not change, since the chain type does not change. The result follows from the $B + S$ property. Let a reversal be made in a chain where at least one cut is exterior. The extremity of the chain adjacent to an exterior cut is called *exterior*; let it be the left-hand extremity. The other cut is called *interior*, as well as the edge that it cuts. To reduce the number of cases to consider, note that, according to the upper row of Table 2, joining any chain with an odd chain of

type 0, as well as the inverse cut operation, does not change T . We call this the *odd cutting property*. Therefore, if at one end (or at both ends) outside the chain there is a fragment (fragments) of oddly many conventional edges, then we can cut it (them) out, make the reversal, and then again join it (them) to the resulting chain. Thus, it remains to consider the case where exterior cuts are made at extreme edges. If the reverted segment has no singular vertices, the chain does not change. Otherwise, consider two possibilities.

(1) Two singular vertices are joined into one, i.e., B reduces by 1. Then, one cut is interior and the other exterior. Consider variants of changing the parity of the extremal segment. If it does not change, the result follows from the $B + S$ property. If an even extremal segment is replaced with an odd one, S increases by 1, since an extremal odd segment appears. The following variants of changing the chain type are possible: $3a$ changes into $1a$, 1 changes into 2 , $1a$ changes into $2a$, and 3 changes into 1 . In the first two variants, D does not change, and P does not reduce and increases by at most w_a . In the last two variants, D reduces by 1 and P does not increase and reduces by at most w_a . If an odd extremal segment is replaced with an even one, S does not change. Now, the following variants of changing the chain type are possible: $1a$ changes into $3a$, 2 changes into 1 , $2a$ changes into $1a$, and 1 changes into 3 . In the first two variants, D does not change, and P does not increase and reduces by at most w_a . In the last two variants, D increases by 1, and P does not reduce and increases by at most w_a . In all the cases, T changes by at most 1.

(2) Assume that no merging of two singular vertices into one occurs, i.e., B does not change, and let one cut be interior and the other exterior. Consider variants of changing the parity of the extremal segment. If it does not change, the result follows from the $B + S$ property. If an even extremal segment is replaced with an odd one, S either does not change or increases by 1. The following variants of changing the chain type are possible: $3a$ changes into $1a$, 1 changes into 2 , $1a$ changes into $2a$, and 3 changes into 1 . In the first two variants, D does not change, and P does not reduce and increases by at most w_a . In the last two variants, D reduces by 1, and P does not increase and reduces by at most w_a . If an odd extremal segment is replaced with an even one, S either does not change or reduces by 1. Now, the following variants of changing the chain type are possible: $1a$ changes into $3a$, 2 changes into 1 , $2a$ changes into $1a$, and 1 changes into 3 . In the first two variants, D does not change, and P does not increase and reduces by at most w_a . In the last two variants, D increases by 1, and P does not reduce and increases by at most w_a . In all the cases, $|\Delta T| \leq 1$. If both cuts are exterior, then, taking into account the odd cutting property, the chain does not change.

5.2. Breaking a cycle into two cycles or a cycle and a loop (cutting out a cycle or a loop from a cycle). In this case, T' either reduces by $1 - w_a$ or does not change; T'' either does not change, or increases by w_a , or reduces by w_a ; P does not change; and $(w_b - w_a)K_b$ either does not change or increases by $w_b - w_a$. Therefore, $|\Delta T| \leq 1$.

5.3. Cutting out a fragment from a chain and circularizing it (we consider the case where a cycle is formed; the case of a loop reduces to removal of a vertex). Here, c increases by 1. If this fragment contains no singular vertices, consider two cases. Let no merging of two singular vertices into one occur. Then, one fragment splits into two, one of the new segments being cyclic and, hence, even. Then S reduces by 1, since a cyclic segment appears, and the chain type does not change; T reduces by 1. Now, let two singular vertices merge into one. Then, one odd segment is circularized with adding an additional conventional edge, S does not change, the chain type does not change, and T reduces by 1.

If the fragment contains singular vertices, we consider the same two possibilities as in case 5.1 and repeat all arguments with replacing the reversal operation by circularization. The only case that must be considered separately is the case where both cuts are exterior. One can easily see that the odd cutting property holds for the circularization operation, as well as for the reversal. Therefore, it suffices to consider the case where both cuts are made at extremal edges. This results in a 0-chain of length 1. If merging of vertices occurs, then no segment appears in the cycle, and S increases by 1 due to the appearance of the above-mentioned 0-chain. In this case, the initial chain is of type $3a$ or $3b$, such that, when it turns into a chain of type 0, D reduces by 1, and P does not increase and reduces by at most w_a . In all the cases, $|\Delta T| \leq 1$.

Let there be no merging of vertices; then, two extremal segments of the chain are transformed into other two segments with the same total length. One of them is the above-mentioned 0-chain of length 1, and the other is located inside a cycle. If both extremal segments are odd, then S reduces by 1, the initial chain is of type either $2a$ or $2b$, D does not change, and P does not increase and reduces by at most w_b . If at least one of the extremal segments is even, then S does not change, the initial chain is of type $1a$, $1b$, $3a$, or $3b$, D reduces by 1, and P does not increase and reduces by at most $w_a + w_b$. In all the cases, $|\Delta T| \leq 1$.

5.4. Joining two cycles into one, joining a cycle and a loop, or joining two loops. In this case, T' either increases by $1 - w_a$ or does not change, T'' does not change, increases by w_a , or reduces by w_a , P does not change, and $(w_b - w_a)K_b$ either does not change or reduces by $w_b - w_a$. Therefore, $|\Delta T| \leq 1$.

5.5. One cut is made in a chain, and the other in a cycle or loop. The case of a loop is trivial; consider the case of a cycle. The cycle is cut and inserted into a chain. If no merging of vertices occurs, then this operation is the inverse to that considered in case 5.3. Otherwise, B reduces by 1. Furthermore, c reduces by 1. If the cut in the chain is interior or the parity of the extremal segment does not change, the result follows from the $B + S$ property. Next, we repeat the arguments from case 5.1, subcase (1).

5.6. Each cut is made in a separate chain.

5.6.1. Both cuts are interior. In what follows, we implicitly use the $B + S$ property without referring to it. To trace the changes of D and P , consider types of both chains. If the set of types of the two chains does not change or the change consists in replacing the «asterisk» with the «prime» in the type of one of the chains or vice versa, then the result follows from the fact that, when making such a replacement, P changes by at most $w_b - w_a$, since $P(2b^*; 2b') = P(3a^*; 3a') = w_a - w_b$ and $P(2b'; 2b^*) = P(3a'; 3a^*) = 0$. In what follows, we call this case the identical transformation and omit it from consideration. Consider other cases.

(1) Both chains are odd, and either both are a - or both are b -chains. T' does not change. If both chains are of type $1a$, the only nonidentical change of types is transformation of this pair into a pair of chains of types $2a$ and $3a$ or $2a$ and $3a'$: D reduces by 1, and $(w_b - w_a)K_b$ either reduces by $w_b - w_a$ or does not change. Since $P(1a, 1a; 2a, 3a) = -2w_a$, $P(1a, 1a; 2a, 3a') = -1$, and $P^+(1a, 1a; 2a, 3a) = P^+(1a, 1a; 2a, 3a') = 1 - w_a - w_b$, we have $|\Delta T| \leq 1$. Type $1b$ is considered similarly. Any other pair of types of the initial chains (except for $2a$ and $3a$, which give the inverse transformation) results in the identical change.

(2) Both chains are odd, one of them being an a -chain and the other a b -chain. If their pair of types is $1a, 1b$, then, when making a DM, it transforms into either a 1,1 or a 2,3 pair: T' reduces by $1 - w_a$; $(w_b - w_a)K_b$ in the first case either reduces by $w_b - w_a$ or does not change and, in the second case, does not change. In the first case, D reduces by 2, and P reduces by $w_a + w_b$. In the second case, D reduces by 1, and P reduces by at most 1 and at least $\min(1, w_a + w_b)$ (since $P^+(1a, 1b; 2, 3) = \max(-1, -w_a - w_b)$). If the pair of initial types is $1a, 2b$ or $1a, 2b'$, then, when making a DM, it transforms into a pair 1,2: T' reduces by $1 - w_a$; $(w_b - w_a)K_b$ does not change, reduces by $w_b - w_a$ (in the case of $2b$), or increases by $w_b - w_a$ (in the case of $2b'$); D reduces by 1; P reduces by at most 1 and at least w_b (in the case of $2b$) or w_a (in the case of $2b'$). The pair $1a, 3b$ is considered similarly. If the pair of initial types is $2a, 2b$ or $2a, 2b'$, then, when making a DM, it transforms into a pair 2,2: T' reduces by $1 - w_a$; $(w_b - w_a)K_b$ either does not change or increases by $w_b - w_a$; D does not change; P does not increase and reduces by at most $2 - w_a - w_b$, since $P(2a, 2b; 2, 2) = P(2a, 2b'; 2, 2) = w_a + w_b - 2$ and $P^+(2a, 2b; 2, 2) = P^+(2a, 2b'; 2, 2) = 0$. The pairs $3a, 3b$ or $3a', 3b$ are considered similarly. If the initial pair is $2b, 3a$, $2b', 3a$, $2b, 3a'$, or $2b', 3a'$, then, when making a DM, it transforms into a pair 1,1: $(w_b - w_a)K_b$ does not change, increases by $w_b - w_a$ (for the last three pairs), or reduces by $w_b - w_a$ (for the first pair); T' reduces by $1 - w_a$; D reduces by 1; P reduces by w_b (in the first three cases) or w_a (in the last case). Similarly, this applies to the pair $2a, 3b$. In all the cases, $|\Delta T| \leq 1$.

(3) Both chains are even. The only nontrivial transformation relates to pairs 2,3 and 1,1; other transformations are either identical or inverse to those considered in case (2). T' does not change. When a pair 2,3 transforms into 1,1, D reduces by 1, $(w_b - w_a)K_b$ either does not change or reduces by $w_b - w_a$, and $\Delta P = \min(0, 1 - w_a - w_b)$.

(4) One chain is odd, and the other even: T' does not change. If a pair $1b,1$ transforms into $2b,3$ or $2b',3$, then D does not change; $(w_b - w_a)K_b$ does not change, increases by $w_b - w_a$ (for the first case), or reduces by $w_b - w_a$ (for the second case); ΔP is not less than $w_b - 1$ and $w_a - 1$, respectively, and not greater than 0, whence the desired follows. Transformation into a pair $3a,2$ is similar. If a pair $1b,2$ transforms into $2b,1$ or $2b',1$, then D reduces by 1, $(w_b - w_a)K_b$ either does not change or reduces by $w_b - w_a$, and ΔP is not less than $-w_b$ and not greater than $\min(0, 1 - w_a - w_b)$, whence the desired follows. Transformation of a pair $1a,3$ into $3a,1$ is similar. Transformation of a pair $2a,1$ into $1a,2$ is inverse to the one considered above. If a pair $2a,3$ transforms into $3a,2$ or $3a',2$, then D does not change, ΔP is not less than $w_a - 1$ and not greater than $1 - w_b$, and K_b does not change. Transformations $2a,3$ into $3a',2$ and $2b',3$ into $3b,2$ are considered similarly. Transformations $2a,3$ into $1a,1$, $3a,1$ into $1a,3$, and $3a,2$ into $1a,1$ are inverse to those considered above. For a pair $3a,3$, the identical transformation is only possible.

5.6.2. One of the cuts is exterior. We reduce this case to an SM or OM operation as follows: the extremity of the chain adjacent to an exterior cut is called exterior. The other cut is called interior, as well as the edge that it cuts. If an exterior cut is at neither an extremal edge nor the edge next to it, we remove two conventional edges from the exterior end. Then, T reduces by 1 both in the initial graph and in the resulting one, which allows us to apply Lemma 4.2. Therefore, it suffices to consider cases where the exterior cut is at an extremal edge of a chain or at the next to it.

(1) The exterior cut is at the next to extremal edge. If it is conventional, remove two extremal edges together with the exterior cut, thus replacing the DM with an SM (see Figure 5a). Then, T reduces by 1 both in the initial graph and in the resulting one, which by Lemma 4.2 reduces the problem to the SM case. If the edge with the exterior cut is singular, remove the extremal edge together with its endpoints, replacing an odd extremal segment of length 1 with a hanging edge (the singular vertex becomes an endpoint of a hanging edge). Again, the DM is changed to an SM with reducing T by 1 in both graphs (see Figure 5b).

(2) The exterior cut is at the extremal edge, and this edge is conventional. If the interior edge is conventional or its singular vertex is on the side that is joined with the exterior extremity, move the exterior cut outside the end of the chain (i.e., make it "fictitious") and move the interior cut to the next edge in the opposite direction from the potential singular vertex (and, if this edge is missing, then outside the end of the chain) (see Figure 5c). Thus, the DM is changed to an SM (or, respectively, to the identity operation, which changes nothing). The result of the operation does not change. If the singular vertex is on the other side, move the exterior cut from the end to the next edge and, accordingly, move the interior cut to the neighboring edge (see Figure 5d). The result of the operation does not change, and the problem reduces to that considered in the preceding case.

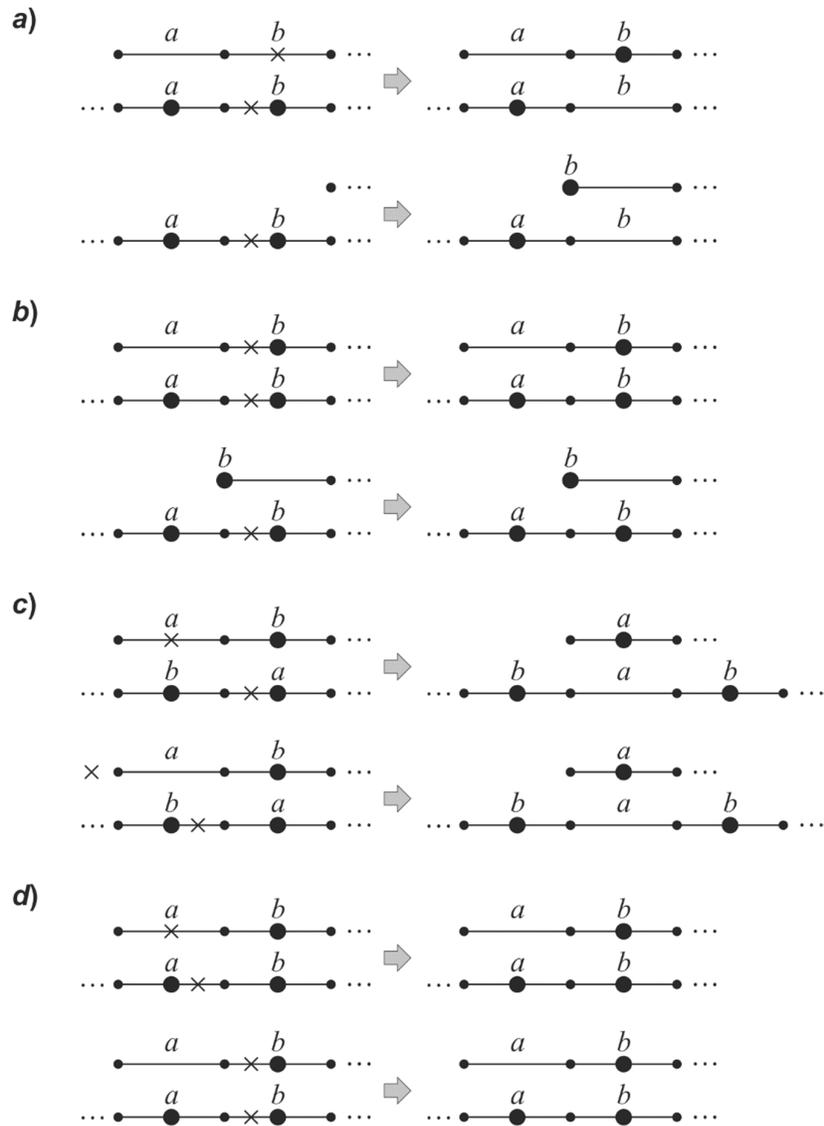


Figure 5. DM with an exterior cut at a conventional edge. **(a)** Reduction to an SM for an exterior cut at the next to extremal conventional edge. Top: initial DM. Bottom: SM obtained from the initial DM by removing two conventional edges. **(b)** Reduction to an SM for an exterior cut at the next to extremal singular edge. **(c)** Reduction to an SM for an exterior cut at an extremal conventional edge. The fictitious cut outside the end of the chain corresponds to replacement of the DM with an SM. **(d)** Reduction of a cut at an extremal conventional edge to the case considered in Figure 5b.

(3) The exterior cut is at the extremal edge, and this edge is singular. If the interior edge is conventional or its singular vertex is on the side that is joined with the exterior extremity, we proceed as in the preceding case (see Figure 6a). Otherwise, move the exterior cut outside the end of the chain and move the interior cut to the neighboring edge in the opposite direction from the singular vertex (and, if this edge is missing, then outside the end of the chain) (see Figure 6b–d). Thus, the DM is replaced with an SM or OM. To obtain the result of the first operation (DM) from the result of the second (SM or OM), merge the two singular vertices by deleting the conventional edge that separates them, and add a chain from one conventional edge (in the case of OM) or add two extremal conventional edges to the chain (if the interior cut was moved to a conventional edge), or replace a hanging edge with an extremal segment of length 1 (if the interior cut was moved to a singular edge). In all the cases, the transformation does not change T , which completes the analysis of the cases.

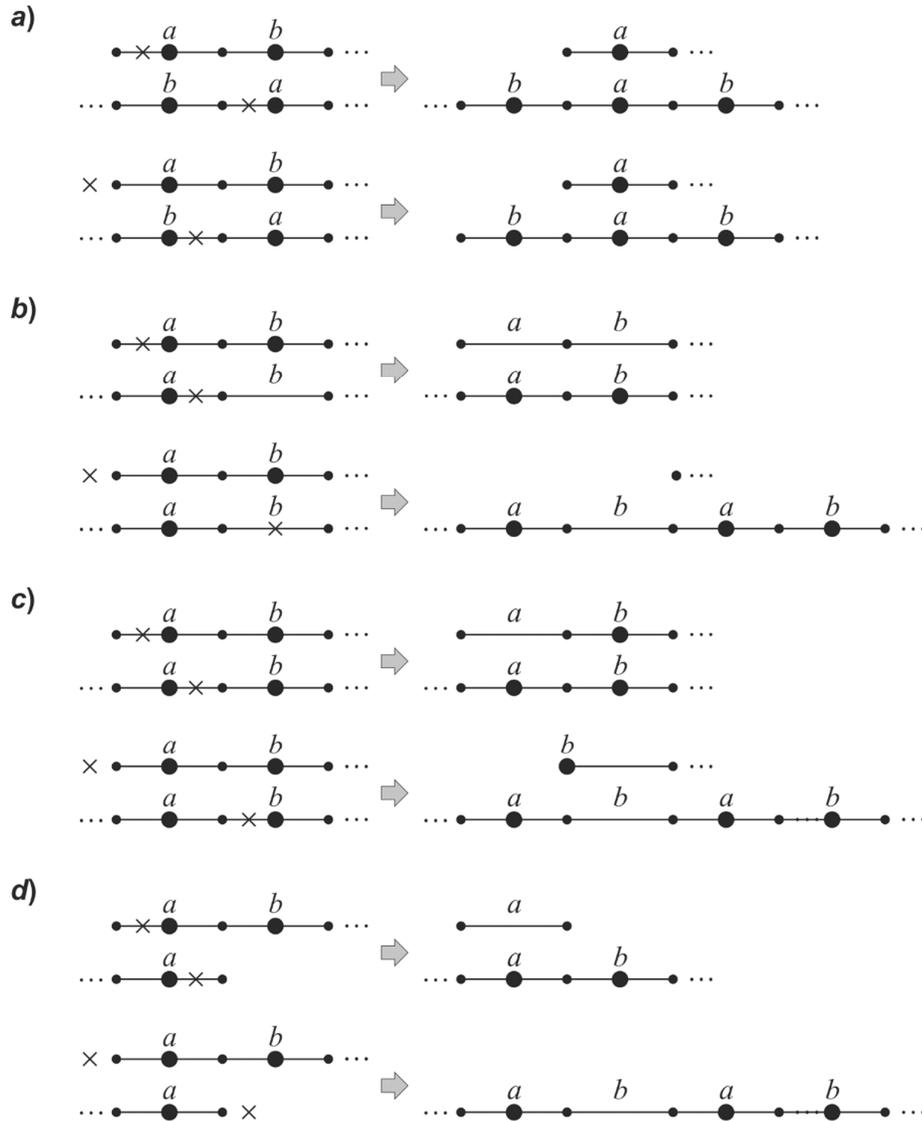


Figure 6. DM with an exterior cut at an extremal singular edge. Reduction to an SM: first case (a), second case (b), and third case (c); reduction to an OM (d).

Theorem 1 is proven.

4. Algorithm and Proof of Theorem 2 (Case III)

Theorem 2 considers Case III; it is a particular case of Theorem 1.

Theorem 2. *If DCJ operations are assigned with equal costs w and the deletion and insertion operations are assigned with any costs w_d and w_i such that one of them is strictly less and the other strictly greater than w , then the algorithm described below is of linear time complexity and outputs a sequence of operations with total cost superior to the cost of the shortest sequence by at most an additive constant that depends only on the operation costs.*

Without loss of generality, we may assume that $w = 1$. Recall that $w_d = w_a$ and $w_i = w_b$. In Theorem 2, we distinguish three subcases: (1) $w_d + w_i \leq 2$, (2) $w_d + w_i > 2$ and $\max(w_d, w_i) \leq 2$, and (3) $\max(w_d, w_i) > 2$, and we enumerate them, respectively, by 1, 2, and 3. The additive exactness constant k mentioned in the theorem is in the first subcase not greater than $w_b - 1$; in the second subcase, constant k is not greater than $4w_b + 2w_a - 6$; in the third subcase, constant k is not greater than $6w_b + 2w_a - 9$. In Section 4a, we describe the algorithm for the first subcase. Algorithms for the second and third subcases, as

well as proofs of their exactness, will be given elsewhere due to natural restrictions on the size of the paper. These algorithms are akin to the algorithm for the first subcase, and the proofs of their exactness are akin to the proof for this subcase, which is given in Section 4.2. As above, we may confine ourselves with the case of $w_a \leq w_b$.

4.1. Description of the Algorithm for the First Subcase

In what follows, all comparisons are made with the Algorithm from Section 2 (Case II).

Stage 0: The same, transformation of initial graphs a and b into $a + b$.

Stage 1: The same, cutting out all conventional edges.

Let us recall what *types* of chains may occur in $a + b$. These are odd chains: $1a$, $1b$, $2a$ ($2a^*$ and $2a'$), $2b$ ($2b^*$ and $2b'$), $3a$ ($3a^*$ and $3a'$), $3b$ ($3b^*$ and $3b'$), and 0, and even chains: 1_a (1_a^* and $1_a'$), 1_b (1_b^* and $1_b'$), 2 (2^* and $2'$), 3, and 0. Chains of types $2a'$, $3b'$, $1'_b$ are called *problem chains*. Components of $a + b$ are divided into *types* (a,b) -, a -, b -, and 0- accordingly as they contain both a - and b -vertices, a - but no b -vertices, b - but no a -vertices, or no singular vertices at all.

Stage 2: In essence, the same: Consider former 2-interactions between different chains that were performed in Case II. Their *a-quality* (instead of the quality that was used in Case II) is shown in square brackets (its definition is given right after the algorithm description, in Section 4b): $1a + 1b = 1'_b [w_a + 1]$, $3a^* + 2b^* = 1_a [1]$, $3a^* + 2b' = 1_a [w_a]$, $3a' + 2b^* = 1_a [w_a]$, $3a' + 2b' = 1'_a [w_a]$, $3b + 2a = 1_b [1]$, $3 + 2 = 1'_b [w_a]$, $(1a + 2b^*) + 3 = 1'_b [w_a + 1]$, $(1a + 2b') + 3 = 1'_b [2w_a]$, $(1b + 2a) + 3 = 1'_b [w_a + 1]$, $(3a^* + 1b) + 2 = 1'_b [w_a + 1]$, $(3a' + 1b) + 2 = 1'_b [2w_a]$, $(3b + 1a) + 2 = 1'_b [w_a + 1]$, $1a + 2 = 2a^* [w_a]$, $1b + 2 = 2b^* [w_a]$, $3 + 1a = 3a^* [w_a]$, $3 + 1b = 3b^* [w_a]$, $(3b + 1a) + (1a + 2b^*) = 1'_b [w_a + 2]$, $(3b + 1a) + (1a + 2b') = 1'_b [2w_a + 1]$, $(3a^* + 1b) + (1b + 2a) = 1'_b [w_a + 2]$, $(3a' + 1b) + (1b + 2a) = 1'_b [2w_a + 1]$, $1a + (1a + 2b^*) = 2a^* [w_a + 1]$, $1a + (1a + 2b') = 2a^* [2w_a]$, $1b + (1b + 2a) = 2b^* [w_a + 1]$, $(3b + 1a) + 1a = 3a^* [w_a + 1]$, $(3a^* + 1b) + 1b = 3b^* [w_a + 1]$, $(3a' + 1b) + 1b = 3b^* [2w_a]$, $1a + 1a = 3a^* [w_a]$, $1b + 1b = 3b^* [w_a]$, $1a + 2b^* = 2 [1]$, $1a + 2b' = 2 [w_a]$, $1b + 2a = 2 [1]$, $3a^* + 1b = 3 [1]$, $3a' + 1b = 3 [w_a]$, $3b + 1a = 3 [1]$, $3 + ((3 + 2b^*) + 2a) = 1'_b [w_a + 1]$, $3 + ((3 + 2b') + 2a) = 1'_b [2w_a]$, $(3a^* + (3b + 2)) + 2 = 1'_b [w_a + 1]$, $(3a' + (3b + 2)) + 2 = 1'_b [2w_a]$, $(3a^* + 2) + 2 = 2a^* [w_a]$, $(3a' + 2) + 2 = 2a^* [2w_a - 1]$, $(3b + 2) + 2 = 2b^* [w_a]$, $3 + (3 + 2a) = 3a^* [w_a]$, $3 + (3 + 2b^*) = 3b^* [w_a]$, $3 + (3 + 2b') = 3b^* [2w_a - 1]$, $(3 + 2b^*) + 2a = 2^* [1]$, $(3 + 2b') + 2a = 2^* [w_a]$, $3a^* + (3b + 2) = 3 [1]$, $3a' + (3b + 2) = 3 [w_a]$, $3a^* + 3b = 3 [1 - w_a]$, and $2a + 2b^* = 2^* + 1'_a [1 - w_a]$.

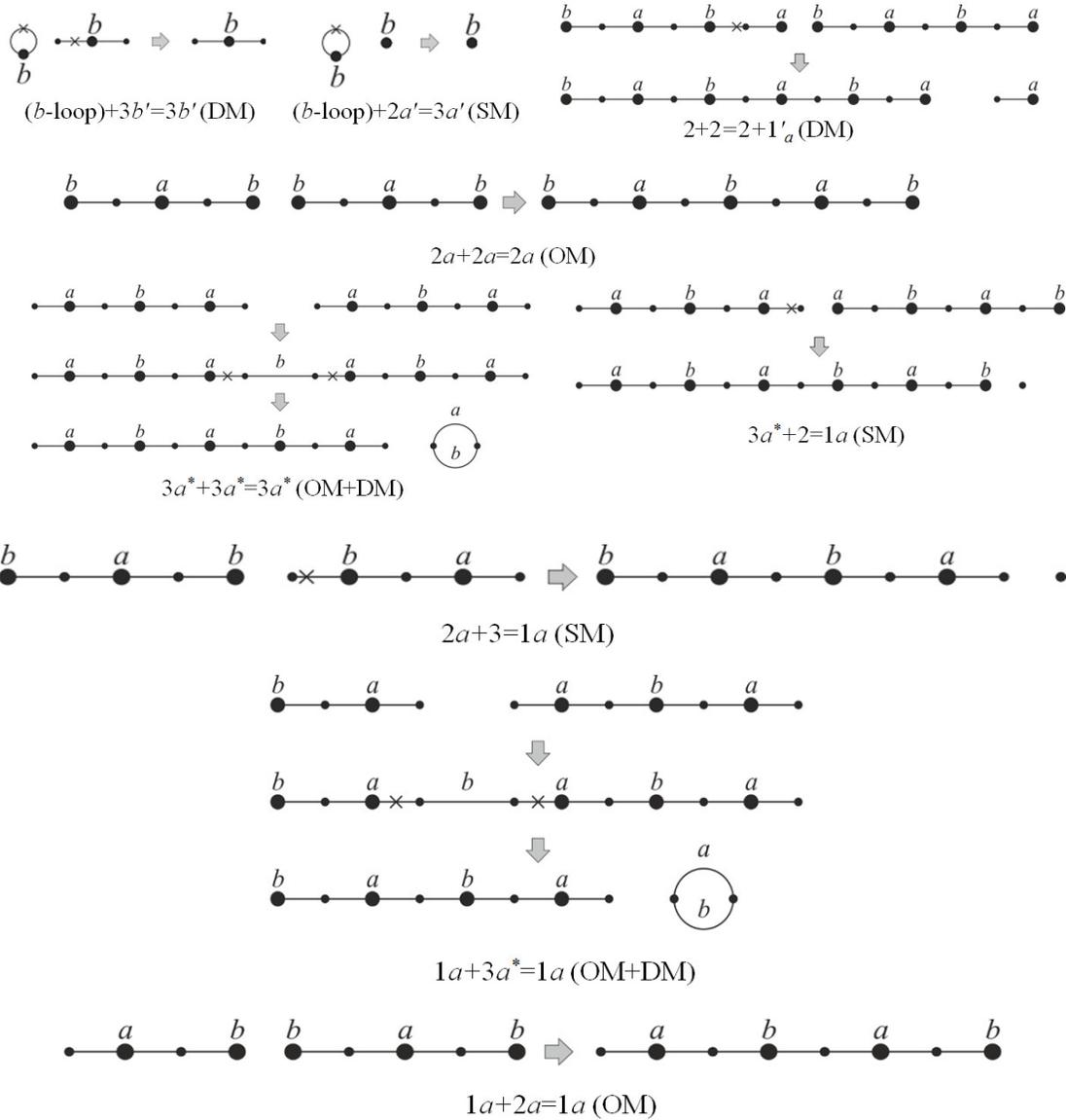
Construct a maximal domain on the set of chains of $a + b$ in the same way as in Case II but with the *a-quality* instead of the quality. Having found a maximal domain M , simultaneously and independently perform all 2-interactions that correspond to it.

Notice 3. To 2-interactions, we may add the following three interactions that are applied to two different chains, with the term equalities: $1'_b + 2a' = 2a'$, $1'_b + 3b' = 3b'$, and $1'_b + 1'_b = 1'_b$. They have zero *a-quality*s and reduce the number of problem chains, which produce an additive error of the algorithm. However, they are inessential for the proof of Theorem 2. These interactions were shown in Figure 7 in [2].

Stage 3: Execute the following *3-interactions*, which reduce the number of (a,b) -components and the number of b -components. These interactions are shown in Figure 7 below.

1. b -loop + «component with a b -singular vertex» = the same component (DM or SM operations);
2. $2 + 2 = 2 + 1'_a$. SM operation with cutting out two vertices of a 2-chain (an extremal a -singular vertex and the neighboring conventional) and joining the obtained extremity with an extremal b -singular vertex of the other 2-chain.
3. $2a + 2a = 2a$, $2b^* + 2b^* = 2b^*$. Joining hanging vertices of two chains.
4. $3a^* + 3a^* = 3a^*$, $3b + 3b = 3b$. OM and cutting out a conventional edge.
5. $3a^* + 2 = 1a$, $3b + 2 = 1b$. SM.
6. $2a + 3 = 1a$, $2b^* + 3 = 1b$. SM.
7. $1a + 3a^* = 1a$, $1b + 3b = 1b$. OM and cutting out a conventional edge.
8. $1a + 2a = 1a$, $1b + 2b^* = 1b$. OM.
9. $3a^* + 3 = 3$, $3b + 3 = 3$. OM and cutting out a conventional edge.
10. $2a + 2 = 2$, $2b^* + 2 = 2^*$. OM.
11. $3 + 3 = 3$. SM.
12. $1'_a + 1'_a = 1'_a$, $1_b + 1_b = 1_b$. SM.

13. $1a + 1b = 1a, 1b + 1^*_a = 1b$. SM.
14. $1a + 1^*_a = 1a, 1b + 1_b = 1b$. SM.
15. $2a + 1_b = 2a, 2b^* + 1^*_a = 2b^*$. SM.
16. $3a^* + 1^*_a = 3a^*, 3b + 1_b = 3b$. SM.
17. $3 + 1^*_a = 3, 3 + 1_b = 3$. SM.
18. $2 + 1^*_a = 2^*, 2 + 1_b = 2$. SM.



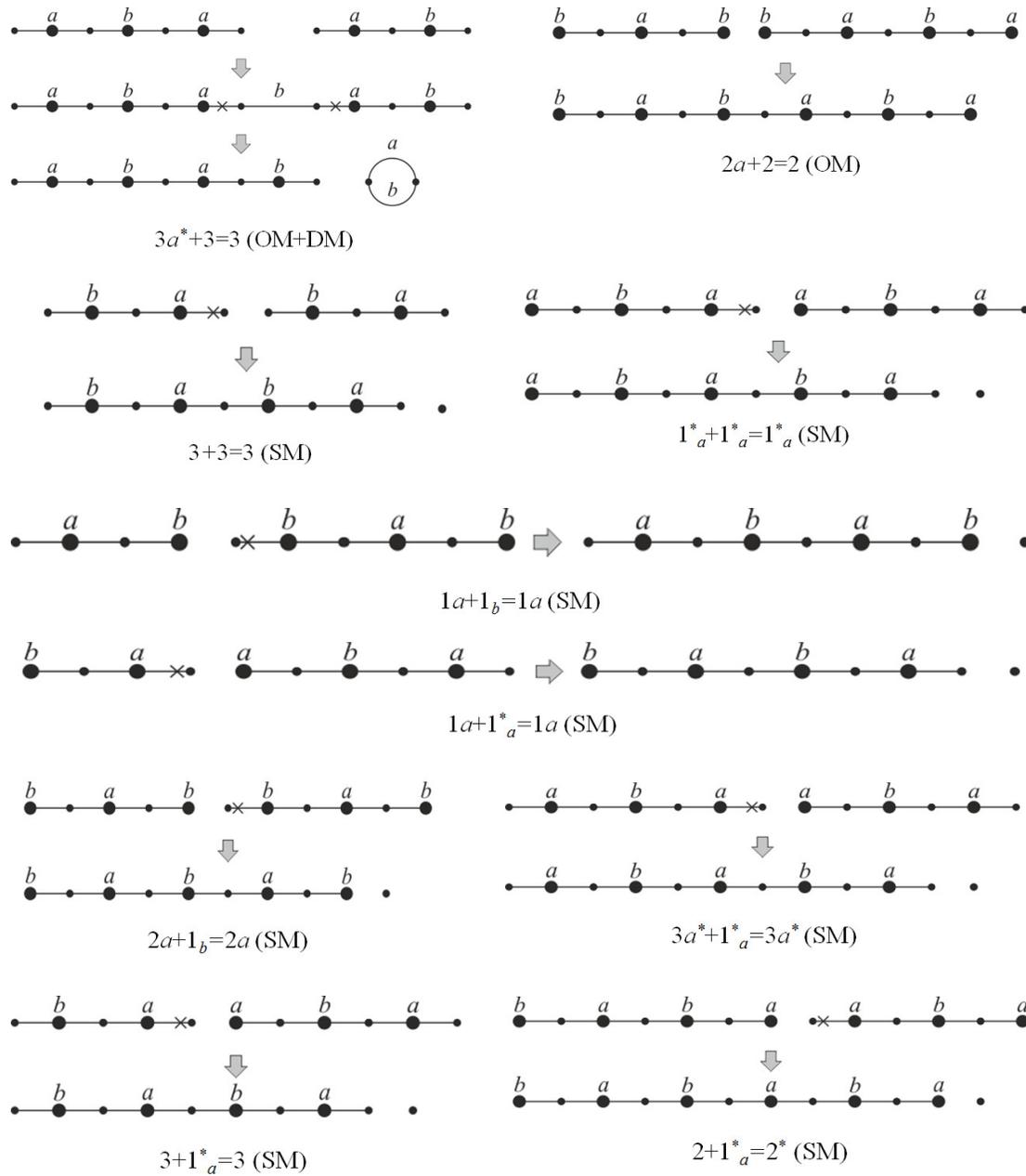


Figure 7. The 3-interactions.

Repeat Stage 3 until these interactions are possible.

Stage 4: Perform autonomous reduction of remaining components to a final form.

End of the algorithm description for the first subcase.

4.2. Proof of the Algorithm Exactness for the First Subcase

Define the *autonomous a-cost* $A_a(G)$ of a graph G to be the total cost of the sequence of operations given in the autonomous reduction for G after deduction of costs of all its b -special operations, i.e., those reducing the number of b -singular vertices. In other words, $A_a(G)$ accounts only for a -special (i.e., those reducing the number of a -singular vertices) and nonspecial (i.e., not reducing the number of singular vertices) operations. Together, they are called *a-operations*.

For a graph G , we use the following *notation*: B_a and B_b are the numbers of, respectively, a -singular and b -singular vertices in G ; S is the sum of integral parts of halved lengths of maximal

connected fragments consisting of conventional edges (referred to as *segments*) plus the number of extremal (on a chain) odd segments (i.e., those consisting of an odd number of edges) minus the number of cyclic segments; S_a is the number of segments enclosed between two singular a -edges; C_a is the number of a -cycles; D is the number of chains of types $1a$, $1b$, $3a$, $3b$, and 3 ; $N(t)$ is the number of chains of a certain type t . Denote by A the number of a -special DCJ operations in the autonomous reduction.

Lemma 5. *Let w_a and w_b be the removal costs for singular a - and b -vertices, and let all other operations have cost 1. Then,*

$$A_a(G) = (1 - w_a) \cdot (S_a - C_a + N(2b^*) + N(3a^*) + N(1a^*)) + w_a \cdot B_a + S + D.$$

Proof. For every component of G , let us check the equality $A = S_a - C_a + N(2b^*) + N(3a^*) + N(1a^*)$, and then sum up the obtained equalities. Each of them is implied by the following facts: when cutting out conventional edges from any component other than an a -cycle, the number of executed a -special operations is S_a , and, for an a -cycle, it is less by 1; the operation of circularization into a cycle is a -special if and only if this chain is of type $2b^*$, $3a^*$, or $1a^*$; breaking a cycle does not contain a -special DCJ operations. Next, the number of removals of a -singular vertices in the autonomous reduction is $B_a - A$, and, similarly to the proof of Lemma 1, we obtain that the number of nonspecial operations is $S + D$. □

Define the ***a-quality*** $P_a(s)$ of an interaction s to be

$$P_a(s) = A_a(G) - A_a(s(G)) - c_a(s),$$

where $s(G)$ is the graph obtained by applying s to G , and $c_a(s)$ is the total cost of a -operations in s . The number $P_a(s)$ shows what a -cost is saved when applying the interaction s as against the autonomous reduction of G , i.e., without using s . Take note that, in Case II, we used the *quality* of interaction, which is related to all operations instead of a -operations.

Using Lemma 4, we compute a -qualities of 2-interactions. For example, let us find the a -quality of the interaction $1a + 1b = 1b^*$ (Figure 4), where $c_a(s) = 1$. The interaction reduces B_a by 1, reduces D by 2, and does not change all other quantities in the expression for $A_a(G)$. Therefore, $P_a(s) = w_a + 1$.

Let E be any reducing sequence for a graph G , which is arbitrarily divided into connected fragments of operations; let us for a while call the latter interactions and denote them by s . Define $T_a(G, E)$, the *total cost of a -operations* in E , and $P_a(G, E)$, the *total a -quality* of all interactions in E , where $P_a(s)$ is formally defined above. The following lemma is proven similarly to Lemma 2 in Section 3:

Lemma 6. *For any graph G , we have $T_a(G, E) = A_a(G) - P_a(G, E)$.*

Lemma 7. *After Stage 3, the algorithm outputs a graph G''' with no polytype corresponding to any 2-interaction of two chains.*

Proof. It is important that the algorithms for Cases II (when $w_b = 1$) and III at Stages 0, 1, and 2 completely coincide. Maximal domains that are constructed in both cases coincide, since, through an exhaustive search over all 2-interactions, we check the following: for $w_b = 1$, the a -quality of any 2-interaction is equal to the quality if the same 2-interaction (computed in Section 2). Furthermore, through an exhaustive search, we check that all 3-interactions have zero quality. If, for G''' , there are two different chains corresponding to some 2-interaction (its a -quality is automatically strictly greater than zero for $0 < w_a < 1$), then, after Stage 2 of the *algorithm for Case II* applied to G with $w_b = 1$, we use 3-interactions, followed by this 2-interaction and, then, autonomous reduction. The aggregate quality of all interactions in such a reducing sequence is strictly greater than the aggregate quality in the sequence output by the algorithm for Case II. By Lemma 2, this means that we have obtained a sequence with total cost strictly less than the absolute minimum $C(G)$, which is impossible. □

Lemma 8. *After Stage 3, the algorithm outputs a graph G''' satisfying the following condition: it contains no more than two chains having b -singular vertices. These pairs of different chains are contained among the*

following pairs of types: $\{2a, 3a^*\}$, $\{1_a^*, 1_b\}$, $\{3b, 2b^*\}$, $\{2a, 1_a^*\}$, $\{3a^*, 1_b\}$, $\{1_a^*, 3b\}$, and $\{1_b, 2b^*\}$. If the initial G has no (a, b) -chains, then G''' contains at most one chain with a b -singular vertex.

Proof. First claim. Through an exhaustive search, we check that, for any three types of chains with b -singular vertices, there are two of them corresponding to a 2- or 3-interaction. The former is impossible by Lemma 7, and the latter is possible by the definition of Stage 3. \square

Second claim. Exhaustively searching over all types of chains with b -singular vertices, form all pairs of them that do not correspond to any 2- or 3-interaction. We obtained precisely the pairs listed in the lemma. Any two chains in this list do not correspond to these interactions.

Third claim. If an (a, b) -chain does not occur in the initial G , it does not appear while running the algorithm, i.e., in G''' , there are only a - and b -chains. If, in G''' , there are two different chains with b -singular vertices, then, by the second claim, their types form one of the listed pairs. However, every chain whose type contains an asterisk is an (a, b) -chain.

Denote by $T(G)$ the total cost of operations executed by the described algorithm on a graph G . Let C_b and C_{ab} be the numbers of b - and (a, b) -cycles in G , respectively; let I_{pb} be the indicator function of the property « G contains a b -loop but has no components with a singular b -vertex other than loops»; similarly, let I_{cb} be the indicator function of having a chain with a singular b -vertex; let $\varepsilon_b = w_b - 1$.

Denote by $P_a(G)$ the total a -quality of 2-interactions when running the algorithm on a graph G .

Lemma 9. Let w_a and w_b be removal costs for singular a - and b -vertices, and let all other operations have cost 1. Then,

$$T(G) = A_a(G) - P_a(G) + B_b + \varepsilon_b \cdot (C_b + C_{ab} + I_{cb} + I_{pb} + E), \text{ where } E = 0 \text{ or } E = 1.$$

Proof. We declare each operation outside Stages 2 and 3 as an interaction. It is easily checked that the a -quality of any such formal interaction, as well as that of any 3-interaction, is zero. By Lemma 6, the total cost of a -operations is $A_a(G) - P_a(G)$. In a reducing sequence, the number of b -special operations is B_b . Let us show that the number of b -removals among them is $C_b + C_{ab} + I_{cb} + I_{pb} + E$. If, in G , there are no components with a b -vertex other than b -loops, the claim is trivial. Otherwise, the number of b -removals is equal to the number of components with a b -vertex that occur in G''' , i.e., before Stage 4, since a removal operation was never applied in the algorithm before; at Stage 4, only autonomous reduction is used. This amounts to $C_b + C_{ab}$ original cycles, I_{cb} chains with a b -vertex, and E more chains from Lemma 8. \square

We prove the additive exactness of the algorithm by induction on the cost $C(G)$ of the shortest reducing sequence for a graph G . Denote by $T'(G)$ the value of $T(G)$ for $E = 0$. Assume that, for any operation o applied to an arbitrary breakpoint graph G , the «triangle inequality»,

$$c(o) \geq T'(G) - T'(o(G)),$$

is valid, where $c(o)$ is the cost of o and $o(G)$ is the result of applying o to G . If we take for o the first operation in the shortest sequence, by the induction hypothesis, we obtain $C(o(G)) \geq T'(o(G)) \geq T'(G) - c(o)$ and $C(G) = C(o(G)) + c(o) \geq T'(G) - c(o) + c(o) = T'(G)$. Therefore, the algorithm error is at most $\varepsilon_b E \leq w_b - 1$.

The proof of the triangle inequality follows the same lines as in Section 3 (Case II) by examining all operations o . In the proof, one should use P_a , the a -quality of the maximal domain on chains in G .

5. Algorithm and Proof of Theorem 1 for Case I

It might be interesting for the reader to compare the algorithms and proofs presented above with those related to Case I, where both costs w_a and w_b are not less than w , although this case was analyzed in a preparatory work [2]. As above, it suffices to consider the case of $w_a \leq w_b$.

Recall that, in Case I, the Algorithm consists of 8 stages.

Stage 0: The same as above: Transform initial graphs a and b into $a + b$.

Stage 1: The same as above: Cut out conventional edges.

Stage 2: Perform the same 2-interactions between chains as in Section 2 (except for the two additional ones) but in the same order as they are listed, independently of their quality (i.e., we need not choose a maximal domain M). The meaning of these interactions, as always, is that they save maximally many operations as against the number of operations in the autonomous reduction of a graph.

Stage 3: Perform 3'-interactions between chains, which are listed in [2]. The meaning of 3'-interactions is that they reduce the number of short chains in $a + b$ (to be precise, chains of types $2a'$, $2b'$, $3a'$, $3b'$, $1'a$, $1'b$, and $2'$), thereby reducing the additive error of the algorithm, which occurs precisely because of them.

Stage 4: Circularize chains (except for short ones) into cycles using OM or SM operations.

Stage 5: Join all cycles into one cycle and then detach final cycles from it.

Stage 6: Perform 6-interactions between a cycle and short chains, which reduce the number of the latter. These are interactions defined by the following term equalities: (a,b) -cycle + $2a' + 2b' = 2'$ (two DM operations) and (a,b) -cycle + $2' = 2'$ (DM and cutting out a conventional edge).

Stage 7 is applied if $w_b > 2$. Perform 7-interactions between two cycles or between a cycle and a short chain. Their meaning is to replace the removal operation for a b -singular vertex with two intermergings, which is advantageous if $w_b > 2$. These are the interactions defined by the following term equalities: (a,b) -cycle + b -cycle = (a,b) -cycle + cycle of length 2 (two DM operations), b -cycle + b -cycle = b -cycle + cycle of length 2 (two DM operations), b -cycle + $2'$ -chain = $2'$ -chain + cycle of length 2 (SM and cutting out an extremal conventional edge), (a,b) -cycle + $2a'$ -chain = (a,b) -cycle (SM and OM), (a,b) -cycle + $1b'$ -chain = (a,b) -cycle (two SM operations), b -cycle + $2a'$ -chain = $2a'$ -chain + cycle of length 2 (SM and cutting out an extremal conventional edge), b -cycle + $1b'$ -chain = $1b'$ -chain + cycle of length 2 (SM and cutting out an extremal conventional edge). If $w_a > 2$, then symmetric interactions with a replaced by b are also performed.

Stage 8: Remove singular vertices.

Now, we recall the proof of exactness of this algorithm, which has an additive error of at most $2w$. As all such proofs, it is based on the triangle inequality. However, for Case I, it has a distinction: $T(G)$ takes the form $T'(G) + E(G)$, where $T'(G)$ involves only easily computable characteristics of a graph G , and $E(G) \leq 2w$. The triangle inequality for $T'(G)$ implies that $C(G) \geq T'(G)$, where, as always, $C(G)$ is the total cost of the shortest reducing sequence for G . However, for $T'(G)$, the triangle inequality is not always valid. To overcome this obstacle, all graphs G are divided into ranks 1, 2, and 3. Graphs of ranks 1 and 2 have a simpler structure, although graphs of a final form fall into rank 3 (from the point of view of the current proof, they are more complicated). For graphs G of ranks 1 and 2, we prove a lower bound on $C(G)$, which is stronger than $C(G) \geq T'(G)$; for graphs of rank 1, it is even stronger than for those of rank 2. When passing, by operation o , from a graph of a larger rank to that of a smaller rank, the triangle inequality is weakened by a strictly positive number Δ (depending only on these ranks), i.e., $c(o) \geq T'(G) - T'(o(G)) - \Delta$. Under the inverse transformation, it is strengthened by Δ , i.e., $c(o) \geq T'(G) - T'(o(G)) + \Delta$. Then, the proof is made by induction on the shortest reducing sequence for G ; it also requires an exhaustive search over all pairs of ranks of G and $o(G)$. For example, if G is of rank 3 and $o(G)$ of rank 2, then, by the induction hypothesis, we have $C(o(G)) \geq T'(o(G)) + \Delta \geq T'(G) - c(o) - \Delta + \Delta$ and $C(G) = C(o(G)) + c(o) \geq T'(G) - c(o) + c(o) = T'(G)$. Here, we used the fact that the Δ by which the triangle inequality is weakened when passing from a graph of rank 3 to that of rank 2 is equal to the number by which the lower bound on $C(G)$ for graphs of rank 2 is strengthened. The analogous consistency of such inequalities holds for other pairs of ranks as well. All cases of the ranks are examined in [2].

6. Discussion

Usually, a small (slightly different from the original operation costs) explicitly specified additive error is not considered as a violation of the algorithm exactness. In this sense, we described an exact algorithm of linear complexity which constructs a shortest transformation of one weighted (in other words, labeled) directed graph into another under equal costs of DCJ operations and arbitrary costs of deletion and insertion operations. These graphs must be of degree 2 in the sense that each vertex

is of degree 1 or 2. The labeling cannot contain repetitions of names. Note that the above can be carried over to infinite (countable) recursively enumerated sequences. The idea of the current algorithm was based on the abstract theory developed in [11,12].

The theorem proven in the paper is rather unexpected from the mathematical point of view; a problem that seems to require exhaustive search or at least to be computationally very difficult is solved by an exact algorithm whose runtime is always linear in the size of input data. In this sense, the problem is not related to any application. However, it has arisen in the context of quite various applications. Of these, the most popular concerns the description of a biological genome by a set of chains and cycles that correspond to linear and circular chromosomes. In this case, the considered operations correspond to real genome transformations in the biological evolution of the genome. It is important that different operations can be assigned with different costs, which corresponds to different frequency of their occurrence in the evolution.

For instance, the possibility to take different costs of deletion w_d and insertion w_i is essential when mitochondria of deuterostomes are considered. Their mitochondria mainly have the same set of genes, and each gene has a unique function. However, the order of these genes can vary; it is substantially different in the purple sea urchin *Strongylocentrotus purpuratus* than in vertebrates (Jacobs et al. [13]), as well as among echinoderms themselves. Thus, gene losses and acquisitions are much rarer here than DCJ operations, and rare events should be assigned higher costs. A similar pattern is observed in plastids of plants, e.g., the order of plastid genes in the red alga *Porphyridium purpureum* notably differs from that in other red algae. In the next case where one genome is smaller than the other, the loss cost should exceed that of acquisition, which surely depends on the gene and the number of its copies. For example, the nuclear genome amplifies after hybridization of two species as in the case of polyploid strawberry or wheat (Bors and Sullivan [14]). Polyploidy is much more common in plants than in animals. Among animals, it was described in nematodes including ascarids and certain amphibians. However, other full-genome duplications have been reported in chordates (Putnam et al. [15]). The genome reduction can also occur, e.g., many genes (paralogs) are lost after a full-genome duplication. In this regard, the presented algorithm has been used, e.g., in [16].

Let us give an example from another area. In some robotics and image processing problems, it is assumed that the “terrain” is bound by barriers that can be described as chains and cycles. Chains and cycles are also used to represent parts that have long or compact forms and to describe the frame of an area or human pose, among other examples. Among many such applications, which are not related to genomics, let us mention, as an example, [17].

Further research will be aimed at reducing the additive error, up to zeroing it, constructing fast algorithms in the case of unequal costs of DCJ operations, allowing name repetitions in the graph labeling, and passing to graphs of degree 3. Moreover, it would be important to extend the list of operations by adding, for example, the operation of edge duplication in the initial graph, which will allow strictly considering the duplication of biological genes [18] from the mathematical point of view.

Author Contributions: Conceptualization, V.L. and K.G.; proof, K.G. and V.L.; writing—original draft preparation, K.G.; writing—review and editing, V.L.; supervision, V.L.; project administration, V.L.; funding acquisition, V.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Russian Foundation for Basic Research under research project No. 18-29-13037.

Acknowledgments: We are grateful to the anonymous reviewers for their thorough review, and we highly appreciate the comments and suggestions, which significantly contributed to improving the quality of the publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yancopoulos, S.; Attie, O.; Friedberg, R. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics* **2005**, *21*, 3340–3346, doi:10.1093/bioinformatics/bti535.

2. Gorbunov, K.Y.; Lyubetsky, V.A. An Almost Exact Linear Complexity Algorithm of the Shortest Transformation of Chain-Cycle Graphs. *arXiv* **2020**, arXiv:2004.14351.
3. Da Silva, P.H.; Machado, R.; Dantas, S.; Braga, M.D.V. Genomic Distance with High Indel Costs. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2016**, *14*, 728–732, doi:10.1109/tcbb.2016.2555301.
4. Compeau, P.E.C. A Generalized Cost Model for DCJ-Indel Sorting. *Lect. Notes Comput. Sci.* **2014**, *8701*, 38–51, doi:10.1007/978-3-662-44753-6_4.
5. Warnow, T. (Ed.) *Bioinformatics and Phylogenetics: Seminal Contributions of Bernard Moret*; Springer Nature: Aktiengesellschaft, Switzerland, 2019.
6. Yin, Z.; Tang, J.; Schaeffer, S.W.; Bader, D.A. Exemplar or matching: modeling DCJ problems with unequal content genome data. *J. Comb. Optim.* **2015**, *32*, 1165–1181, doi:10.1007/s10878-015-9940-4.
7. Gorbunov, K.Y.; Lyubetsky, V.A. Linear algorithm of the minimal reconstruction of structures. *Probl. Inform. Transm.* **2017**, *53*, 55–72, doi: 10.1134/S0032946017010057.
8. Alekseyev, M.A.; Pevzner, P.A. Multi-break rearrangements and chromosomal evolution. *Theor. Comput. Sci.* **2008**, *395*, 193–202, doi:10.1016/j.tcs.2008.01.013.
9. Alekseyev, M.A.; Pevzner, P.A. Breakpoint graphs and ancestral genome reconstructions. *Genome Res.* **2009**, *19*, 943–957, doi:10.1101/gr.082784.108.
10. Lenstra, H.W. Integer Programming with a Fixed Number of Variables. *Math. Oper. Res.* **1983**, *8*, 538–548, doi:10.1287/moor.8.4.538.
11. Kanovei, V.; Lyubetsky, V.A. Definable E0 classes at arbitrary projective levels. *Ann. Pure Appl. Log.* **2018**, *169*, 851–871, doi:10.1016/j.apal.2018.04.006.
12. Kanovei, V.; Lyubetsky, V.A. Non-uniformizable sets of second projective level with countable cross-sections in the form of Vitali classes. *Izv. Math.* **2018**, *82*, 61–90, doi:10.1070/im8521.
13. Jacobs, H.T.; Elliott, D.J.; Math, V.B.; Farquharson, A. Nucleotide sequence and gene organization of sea urchin mitochondrial DNA. *J. Mol. Biol.* **1988**, *202*, 185–217, doi:10.1016/0022-2836(88)90452-4.
14. Bors, R.H.; Sullivan, J.A. Interspecific Hybridization of *Fragaria vesca* subspecies with *F. nilgerrensis*, *F. nubicola*, *F. pentaphylla*, and *F. viridis*. *J. Am. Soc. Hortic. Sci.* **2005**, *130*, 418–423, doi: 10.1007/s10681-005-3496-9.
15. Putnam, N.H.; Butts, T.; Ferrier, D.E.K.; Furlong, R.F.; Hellsten, U.; Kawashima, T.; Robinson-Rechavi, M.; Shoguchi, E.; Terry, A.; Yu, J.-K.; et al. The amphioxus genome and the evolution of the chordate karyotype. *Nature* **2008**, *453*, 1064–1071, doi:10.1038/nature06967.
16. Gershgorin, R.A.; Gorbunov, K.Y.; Zverkov, O.A.; Rubanov, L.I.; Seliverstov, A.V.; Lyubetsky, V.A. Highly Conserved Elements and Chromosome Structure Evolution in Mitochondrial Genomes in Ciliates. *Life* **2017**, *7*, 9, doi:10.3390/life7010009.
17. Kerola, T.; Inoue, N.; Shinoda, K. Cross-view human action recognition from depth maps using spectral graph sequences. *Comput. Vis. Image Underst.* **2017**, *154*, 108–126, doi:10.1016/j.cviu.2016.10.004.
18. Mane, A.C.; Lafond, M.; Feijao, P.C.; Chauve, C. The distance and median problems in the single-cut-or-join model with single-gene duplications. *Algorithms Mol. Biol.* **2020**, *15*, 8–14, doi:10.1186/s13015-020-00169-y.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).