ISSN 0005-1179, Automation and Remote Control, 2018, Vol. 79, No. 12, pp. 2203–2216. © Pleiades Publishing, Ltd., 2018.
Original Russian Text © K.Yu. Gorbunov, V.A. Lyubetsky, 2018, published in Avtomatika i Telemekhanika, 2018, No. 12, pp. 124–141.

OPTIMIZATION, SYSTEM ANALYSIS, AND OPERATIONS RESEARCH

A Linear Algorithm for Restructuring a Graph

K. Yu. Gorbunov[*,a] **and V. A. Lyubetsky**[*,**,b]

[*] *Institute for Information Transmission Problems (Kharkevich Institute), Russian Academy of Sciences, Moscow, Russia*
[**] *Lomonosov State University, Moscow, Russia*
e-mail: [a]*gorbunov@iitp.ru,* [b]*lyubetsk@iitp.ru*

Received January 31, 2017

Abstract—We propose an algorithm, linear in both running time and memory, that constructs a sequence of operations that transform any given directed graph with degree of any vertex at most two to any other given graph of the same type with minimal total cost. This sequence is called the shortest one. We allow four standard operations of re-gluing graphs with equal cost and two more additional operations of inserting and deleting a connected section of edges that also have equal cost. We prove that the algorithm finds a minimum with this restriction on the costs.

Keywords: graph, cycle, chain, graph transformation, operation cost, combinatorial problem, optimization on graphs, linear algorithm

DOI: 10.1134/S0005117918120093

1. INTRODUCTION AND PROBLEM SETTING

We consider the *graph transformation problem*: we are given directed graphs a and b, for which the degree of each vertex is $\leqslant 2$, and each edge is assigned a label in the form of a natural number, well-known operations on graphs that preserve this condition on vertex degrees, and the cost of each operation. Such graphs are called *structures*. The problem is to find the *shortest* sequence of operations that converts a to b. The cost is a strictly positive rational number, and "shortest" means a sequence that has minimal total cost; we call it the *minimum cost*. The above-mentioned operations on such graphs include: re-gluing of chains and cycles at vertices (these operations are called *standard*), *inserting* and *removing* connected sections of edges (they are called *additional*). These operations are described in detail, for example, in [1, Fig. 1]; in essence, *double intermerging* consists of cutting two vertices and identifying (gluing) the four edges together in a new way. *Sesquialteral intermerging* consists in the same cutting, followed by gluing together one of the new endpoints with some free endpoint. A *splitting* consists of cutting the vertex, *merging* is the inverse operation. *Removal* means removing a connected section of edges whose labels belong to the original but do not belong to the final structure; *insertion* means a substitution instead of the vertex a connected part with edges whose labels belong to the final but not to the original structure, or declaring this part to become a new component.

The problem has a long history and numerous applications, links to which can be found in [2–5]; it is NP-complete and therefore cannot be solved in general. The authors of the present work have obtained linear algorithms, both in running time and in memory, for solving this problem for the following cases: (1) when the costs of all operations are the same [6]; (2) when the costs of all operations except insertion are equal to one, and the cost of the latter operation is from one to two [1]. In this work we consider the case (3) when the cost w of insertion and removal is the same and does not exceed the cost of other operations. For the sake of brevity, the statement is proved under the following assumption: w does not exceed half of the costs of other operations which are the same (the second half of the statement will be verified in the next paper). Details of some essentially obvious and purely technical calculations of the proof are given on the web site http://lab6.iitp.ru/-/lagr17.

2. DEFINITIONS

For structures a and b, their *combined graph*, denoted $a + b$, is an undirected graph consisting of connected *components*, i.e., chains (including isolated vertices) and cycles, including loops. The definition of $a + b$ has been given in detail in [1], but essentially $a + b$ is described below. We call edges in structures a and b and in intermediate structures from a to b *arcs*, and for edges in $a + b$ we keep the usual term. An arc is called *ordinary* if it is represented in both a and b, otherwise it is called *singular*.

Vertices and edges in $a + b$ can be either *ordinary* or *singular*: an ordinary vertex is an endpoint of an arc represented in both a and b, an ordinary edge connects two endpoints of arcs represented in both a and b and glued in a or in b; a singular vertex is a connected maximal segment of singular arcs in a or in b, a singular edge connects such a segment with the adjacent endpoint of an ordinary arc, i.e., connects a singular vertex with an ordinary one. A *loop* in $a + b$ corresponds to a cycle of singular arcs, that is, it connects a singular vertex with itself.

A *pendant* edge is an edge incident to a singular vertex of degree one; such an edge is located at the end of a chain. This singular vertex is also called *pendant*. Non-pendant singular edges occur in $a + b$ in pairs: these are edges incident to a common singular vertex. Any edge or singular vertex has a source in a or b and is called respectively an a- or b-edge, a- or b-vertex. The *size* of a component in $a + b$ is the sum of the number of its ordinary edges and half of the number of its singular non-pendant edges. For isolated ordinary vertices and loops the size is assumed to be zero, for isolated singular vertices (not loops), equal to negative one. A combined graph is called final (of *final type*) if each of its components is an isolated ordinary vertex or cycle without singular vertices of size (or length, which is the same in this case) two, one of them an a- and another a b-edge; we call such cycles *final*. Cycles of size two are called 2-cycles.

The work [1] defines five operations on a combined graph: counterparts of standard operations, which are also called *standard*, and one *additional* operation. The latter is called *removal*, and it consists of removing a singular vertex: removing it, we merge two singular edges incident to it into one ordinary edge. Counterparts of standard operations on the structure are also called *splitting*, *merging*, *sesquialteral*, and *double* intermerging. Double intermerging consists of the following: remove two edges of a combined graph with the same label and connect the four resulting endpoints with two new non-incident edges with the same label. A splitting is a removal of an edge from the combined graph; merging means adding an edge to the graph, sesquialteral intermerging is a splitting followed by merging one of the endpoints of the removed edge to a vertex of degree at most one. The latter two operations are admissible if the resulting graph does not contain two identically labeled edges incident to an ordinary vertex. If as a result of one of the last three operations there appears an edge with singular endpoints (both a- or both b-vertices), then it is replaced by a single singular vertex. Both splitting and removal can lead to an isolated vertex appearing if they are applied to an end edge or vertex. On the contrary, in the case of a structure removal occurs along with the endpoints of an edge.

Cutting an ordinary edge (say, with label a) from a cycle of size strictly larger than two consists of the following: replace two adjacent b-edges with two other b-edges; one of them forms, together with the cut edge, the final 2-cycle, and the other connects the two "distant" ends of the replaced edges. Obviously, this is a counterpart of double intermerging. Similarly, we define cutting from a chain of a non-end ordinary edge, and for an end edge we use a counterpart of sesquialteral intermerging. Inversion corresponds to double intermerging that keeps the segment in the same place, but with reversed order of the edges.

In [6, Corollary to Theorem 1], it has been proved that if the costs of all operations are equal, then the transformation of a to b by the six operations on structures is equivalent with the same total cost to a transformation of $a + b$ by the five specified operations to a final form where the costs of operations are the same as over the structures, while the cost of removing an a-vertex is equal to the removal cost of the arc segment, and the removal cost of a b-vertex equals to the cost of inserting such a segment. This proof remains literally the same if the costs of only all standard operations are equal, and the costs of removal and insertion are arbitrary. Therefore, we will consider reducing $a + b$ to the final form instead of reducing a to b.

A chain of odd (even) size is called *odd* (*even*); zero is considered an even number, and (-1) is odd. In [6], the notion of a *type* of a chain of a combined graph was defined; namely, if there are no ordinary edges in a chain then $1a$ is an odd chain with one pendant b-edge, $2a$—an odd chain with two pendant b-edges or an isolated singular b-vertex (in the second case we are talking about type $2a'$), $3a$—an odd chain without pendant edges with outermost a-edges, 1_a—an even chain with one pendant a-edge, 2—an even chain with two pendant edges, 3—an even chain without pendant edges. We define the types with a replaced by b in a similar way. Type 1 is the union of types 1_a and 1_b, type 0 corresponds to a chain without singular vertices. The *alphabetical type* of a chain is the letter in its type's notation. If there are ordinary edges in the chain, then the *chain type* is equal to the type of a chain obtained by cutting these edges [6, Lemma 6].

3. ALGORITHM OF AUTONOMOUS REDUCTION, AUTONOMOUS COST AND QUALITY OF A SET OF GRAPH COMPONENTS

Let M be any set of chains, cycles and loops from the combined graph $a + b$. An *autonomous reduction* of the set M to a final form is the following sequence of actions applied to each element of M independently of each other: remove the loop; cut all ordinary edges out from the chain and the cycle, close a chain of strictly positive size into a cycle. The latter is done by merging the endpoints for an odd chain and sesquialteral intermerging with cutting the ordinary end vertex for an even chain; if such a vertex does not exist, i.e., both edges of the chain are singular, then with the same intermerging we remove the penultimate non-pendant edge and keep the pendant edge. Then the cycle (already consisting only of singular edges) is divided into 2-cycles, namely: suppose that we are talking about a segment $abba$ in some large cycle; double intermerging removes the left and right a-neighbors of a special vertex and replaces them with two edges, which leads to two smaller cycles, and one of them is the cycle abb. In the cycle abb we delete the singular vertex and get the final 2-cycle; we divide the larger cycle in the same way; for details, see [6, § 2.2].

The *autonomous cost* $C(M)$ of a set M is the total cost of the sequence of actions specified above. For the set M we use the following *notation*: B is the total number of singular vertices; S is the sum of whole parts of halves of the lengths of the maximal connected sections made of ordinary edges (which we call *segments*) added to the number of extreme (on a chain) odd segments (i.e., consisting of odd number of edges) less the number of cyclic segments; D is the sum of component defects, where the *defect* of a component is equal to one if it is a chain of types $1a$, $1b$, $3a$, $3b$ or 3, and zero in all other cases. The *quality* M is the number of cycles in M (but not loops) plus half the number of even chains in M, see [6, § 3].

Lemma 1. *Let M be the set of components (chains, cycles, loops) of a combined graph G; d, the total size of the components; f, the number of odd chains in M; c, the number of cycles in M. Then the total cost of autonomous reduction of all components from M to the final form is equal to*

$$C(M) = (1 - w)(0.5d + 0.5f - c) + w(B + S + D).$$

Proof. For each component K, we calculate the cost of its autonomous reduction to the final form. It is easy to see (for more details see [6]) that the number of operations required is $B+S+D$, where B, S, D are defined before the statement of Lemma 1 and are computed for K. Denoting by U and Q the number of deletions of a singular vertex and standard operations, we get: $B+S+D=U+Q$, $U=B+S+D-Q$. This means that the total cost of all operations is $Q+w(B+S+D-Q)=(1-w)Q+w(B+S+D)$.

When autonomously reducing a component to the final form, a standard operation increases the quality of the combined graph by one, and removal a special vertex does not change it. Let d be the size of the component. The final form consists of a cycle for $0.5d$ final cycles, for an odd chain of $0.5(d+1)$ final cycles, and for an even chain of $0.5d$ final cycles and one isolated ordinary vertex. Subtracting the quality of the initial graph from the quality of the final graph, we obtain that for a cycle $Q=0.5d-1$, for an odd chain $Q=0.5(d+1)$, and for an even chain $Q=0.5d$. Recall that an isolated singular vertex is considered an odd chain of size minus one. This completes the proof of Lemma 1.

4. MAXIMAL REGION IN A SET OF CHAINS AND A DESCRIPTION OF THE MAIN ALGORITHM

Let L be the set of chains consisting of singular edges. We call a *pair* in L two chains from L that have the following types: $\{2a,3b\}$, $\{2b,3a\}$, $\{1a,1b\}$, $\{1a,2b\}$, $\{1b,2a\}$, $\{1a,3b\}$, $\{1b,3a\}$; each pair of types is tied to one specific action from the list 1–7 (see below); the entry in braces is called the *type* of a pair. The set of pairs in L is called a *region* if the pairs do not intersect. A *region* is divided into *subregions* with the same type of pairs. One specific action out of the seven below is applied to all pairs from the subregion independently of each other; thus, the choice of action is uniquely determined by the type of subregion. So, each of the seven actions is associated with its type: in their descriptions below, equality joins the types of the two arguments in the left-hand side with the type of their value in the right-hand side. The + sign in the list of actions 1–7 is non-commutative.

1. $2a+3b=1_b$: two chains of types $2a$ and $3b$ are replaced with a chain of type 1_b; for this, in $3b$ we remove the extreme edge and glue the singular b-vertex incident to it (from the inside) with the extreme b-vertex in the $2a$-chain; in all items 1–7, an isolated ordinary vertex is formed and a sesquialteral intermerging is applied.

2. $2b+3a=1_a$: replace chains of types $2b$ and $3a$ with a chain of type 1_a; for this, in $3a$ we remove the extreme edge and glue the incident a-vertex with the extreme a-vertex in the $2b$-chain.

3. $1a+1b=1_b$: replace chains of types $1a$ and $1b$ with a chain of type 1_b; for this, we delete the outermost edge in the $1a$-chain and glue the incident a-vertex with the extreme a-vertex in the $1b$-chain.

4. $1a+2b=2$: replace chains of types $1a$ and $2b$ with a chain of type 2; for this, in the $1a$-chain we remove the extreme non-pendant edge and glue the incident a-vertex with the extreme a-vertex in the $2b$-chain.

5. $1b+2a=2$: replace chains of types $1b$ and $2a$ with a chain of type 2; for this, in the $1b$-chain we remove the extreme non-pendant edge and glue the incident b-vertex with the extreme b-vertex in the $2a$-chain.

6. $1a+3b=3$: replace chains of types $1a$ and $3b$ with a chain of type 3; for this, in the $3b$-chain we remove the extreme non-pendant edge and glue the incident b-vertex with the extreme b-vertex in the $1a$-chain.

7. $1b+3a=3$: replace chains of types $1b$ and $3a$ with a chain of type 3; for this, in the $3a$-chain we remove the extreme non-pendant edge and glue the incident a-vertex with the extreme a-vertex in the $1b$-chain.

The quality of action $P(s)$ is equal to the difference between autonomous costs $C(s)$ and $C(\{t\})$, where t is the result of the action on its argument s, and from this difference we also deduct the cost of the action.

The following are some obvious results of calculating the quality $P(j)$ for each of the actions j, $j = \overline{1,7}$. The value d, the total size of the components in the set, is increased by one after a merging and does not change with sesquialteral or double intermerging.

1, 2. $2a + 3b = 1_b$, $2b + 3a = 1_a$. The values of B and D decrease by one, f is reduced by two, $P(1) = P(2) = (1 - w) + 2w - 1 = w$.

3. $1a + 1b = 1_b$. The value of B is decreased by one, f and D decrease by two, $P(3) = (1 - w) + 3w - 1 = 2w$.

4, 5. $1a + 2b = 2$, $1b + 2a = 2$. The values B and D decrease by one, f decreases by two, $P(4) = P(5) = (1 - w) + 2w - 1 = w$.

6, 7. $1a + 3b = 3$, $1b + 3a = 3$. The values of B and D decrease by one, f is reduced by two, $P(6) = P(7) = (1 - w) + 2w - 1 = w$.

We use that, according to the calculations above, P is a multiple of w.

Consider a set L of chains along with its region of *maximum total quality* of actions compared to any other region for L. Such a region is called *maximal*. Since the pairs do not intersect in a region, actions can be performed simultaneously for all pairs of the region in an arbitrary order.

We now describe the *main algorithm*, which brings the graph $a + b$ to its final form.

The algorithm starts with the initial combined graph $a + b$ and includes 4 steps: 1) delete all loops; 2) cut all ordinary edges; we denote the resulting set of chains made of singular edges by L and denote the set of chains and cycles by M; 3) for L, find the maximal region N and apply actions 1–7 to $\langle L, N\rangle$; to the resulting set of chains L_1 add all the cycles that were present in M; and, finally, 4) perform an autonomous reduction of the remaining components to the final form.

The *algorithm for constructing a maximal region* N is trivial: we define a linear order on the types of pairs defined by items 1–7, and then for each type we sequentially add to N pairs from L of this type that do not intersect already added pairs, as long as possible (it does not matter which chains of a given type make up the next pair).

This completes the description of the main algorithm.

The justification for stage 3 is non-trivial: to find the region N we use Lemma 2, and the statement about the algorithm's accuracy is contained in the final theorem of this section, which uses Lemma 1.

Denote by $T(G)$ the total cost of operations in the *final* sequence that the proposed algorithm builds, applied to the combined graph G. Let us prove that

$$T(G) = C(G) - P(G), \tag{1}$$

where $P(G)$ is the total quality of actions for any maximum region $N(G)$ and the specified $L(G)$; the notation $C(G)$ was introduced before the statement of Lemma 1, where the set of connected components of the graph G was taken as M. *Denote* by $T'(R)$ the total cost of operations in the final sequence, starting from some intermediate R.

The equality $T(R) = T'(R)$ follows from the fact that $T(R)$ is the minimum cost for the graph R, which will be proved in the theorem.

Thus, for (1) to hold it suffices that all graphs R in the final sequence, which by the proposed algorithm leads $a + b$ to the final form, satisfy that

$$C(R) - T'(R) = K'(R), \tag{2}$$

where $K'(R)$ is the total quality of all actions in this sequence starting from R. Equality (2) can be proved by induction from the end of the sequence. The induction base is obvious. For operations that are not actions, the inductive step is obvious; for action operations, it follows from the definition of the quality of action (the order of pairs within one type is arbitrary). Note that $K'(G) = P(G)$, and we get that

$$T(G) = C(G) - P(G),$$

and now using Lemma 1 we find that

$$T(G) = (1 - w)(0.5d + 0.5f - c) + w(B + S + D) - P(G). \tag{3}$$

Lemma 2. *For any set of chains L, the maximum region N can be constructed in linear time.*

Proof. Suppose that we have constructed a nonmaximal region N of pairs which is included in some maximal region N'. Let us verify that any pair l of the smallest type that does not intersect with any pair from N is such that $N \cup \{l\}$ is also contained in some maximum region N''. To prove this, let us consider as l pairs of all possible types.

1, 2. Let l be of type $\{2a, 3b\}$ or $\{3a, 2b\}$. Consider the first case. If l is contained in N', we assume $N'' = N'$. Otherwise, let S be the set of partners of chains from pairs in $N' \setminus N$ such that the pair intersects with l. A *partner* chain in a pair is the other chain in this pair. A partner is marked with $*$ in the pair type. Looking ahead, we note that N'' specified below have the same action quality as N', and therefore represent maximum regions (the first and last cases where the quality of N'' is strictly greater than the quality of N' are impossible, i.e., such pair l does not exist).

If S is empty, then $N'' = N' \cup \{l\}$; if $S = \{3b^*\}$, then $N'' = \{N' \setminus \{2a, 3b^*\}\} \cup \{l\}$; if $S = \{1b^*\}$, then $N'' = \{N' \setminus \{1b^*, 2a\}\} \cup \{l\}$; if $S = \{2a^*\}$, then $N'' = \{N' \setminus \{2a^*, 3b\}\} \cup \{l\}$; if $S = \{1a^*\}$, then $N'' = \{N' \setminus \{1a^*, 3b\}\} \cup \{l\}$; if a $S = \{3b^*, 2a^*\}$, then $N'' = \{N' \setminus \{2a, 3b^*\} \setminus \{2a^*, 3b\}\} \cup \{l\} \cup \{2a^*, 3b^*\}$; if $S = \{1b^*, 2a^*\}$, then $N'' = \{N' \setminus \{1b^*, 2a\} \setminus \{2a^*, 3b\}\} \cup \{l\} \cup \{1b^*, 2a^*\}$; if $S = \{3b^*, 1a^*\}$, then $N'' = \{N' \setminus \{2a, 3b^*\} \setminus \{1a^*, 3b\}\} \cup \{l\} \cup \{1a^*, 3b^*\}$; if $S = \{1b^*, 1a^*\}$, then $N'' = \{N' \setminus \{1b^*, 2a\} \setminus \{1a^*, 3b\}\} \cup \{l\} \cup \{1a^*, 1b^*\}$.

3. Let l be of type $\{1a, 1b\}$. The reasoning is similar.

4, 5. Let l be of type $\{1a, 2b\}$. A number of cases is impossible: for example, the intersection of pairs in $1a$ with a partner $1b^*$ and an intersection in $2b$ with a partner $3a^*$ would correspond to a smaller type. The remaining cases are treated similarly.

6, 7. Types $\{1b, 2a\}$, $\{1a, 3b\}$ and $\{1b, 3a\}$ are treated similarly.

Linear running time of this algorithm is obvious. This completes the proof of Lemma 2.

Theorem. *Let w be the cost of insertion and removal, $w \leqslant 0.5$, and suppose that the costs of other operations are equal to one. The algorithm proposed above constructs a sequence of operations with a minimum total cost. Its running time and memory are linear in the total size of the original structures.*

Proof. Let $c(G)$ denote the minimal total cost of the sequence of operations leading the combined graph G to the final form.

Let us show that $T(G) = c(G)$ for any combined graph G. It suffices to check that

$$T(G) \leqslant c(G). \tag{4}$$

Consider the shortest sequence that reduces G to the final form. Suppose that in this sequence, o is the first operation with cost $c(\mathrm{o})$, which applied to G, and $\mathrm{o}(G)$ is the result of applying it. Then $c(\mathrm{o}(G)) < c(G)$, and we can prove by induction on $c(G)$.

In what follows we will *verify* the key inequality for any operation o and any G:

$$c(\mathrm{o}) \geqslant T(G) - T(\mathrm{o}(G)). \tag{5}$$

Then (5) will imply (4). Indeed, by the induction hypothesis we have: $T(\mathrm{o}(G)) \leqslant c(\mathrm{o}(G))$. Using (5), we get that $T(G) \leqslant T(\mathrm{o}(G)) + c(\mathrm{o}) \leqslant c(\mathrm{o}(G)) + c(\mathrm{o}) = c(G)$, i.e., (4).

Thus, we can proceed to checking inequality (5), which essentially uses equality (3); let us consider all possible operations o. As already mentioned, P is always a multiple of w. *Denote* by $T(G)_1$ and $T(G)_2$ the first and second terms in $T(G)$ from (3).

1. o is a *removal* of a singular vertex. When passing from G to $\mathrm{o}(G)$, the value of B decreases by one. Let us consider the possible cases.

1.0. An isolated singular vertex is removed, which is considered to be a chain of type $2a$ or $2b$. In the transition from G to $\mathrm{o}(G)$, the values S, D, c do not change, d is increased by one, f decreases by one. The value of P does not change or decreases by w; indeed, by removing the chain P, by definition, it does not increase, and cannot decrease by more than w since all pairs containing type $2a$ or $2b$ have quality w. Thus, T either does not change or decreases by w.

1.1. A singular vertex is removed from the cycle, or a loop is deleted. When passing from G to $\mathrm{o}(G)$, the value of S does not change (if both segments are adjacent to the deleted vertex, are even, or if the deleted vertex is the only one in the cycle) or increases by one, and the other quantities do not change. Thus, T does not change or decreases by w.

1.2. An internal singular vertex is removed from the chain, i.e., there are other singular vertices on both sides of it. The type of the chain does not change: it is determined by the size of the chain, the types of endpoints (pendant or non-pendant) and (in case of a non-pendant endpoint) the length of the outermost segment (if it is odd, after cutting it a pendant edge appears; otherwise it does not appear). Now we can repeat the argument from 1.1.

We call a *pendant edge* not only the edge of a chain that ends with a pendant edge, but also an edge incident to an odd segment.

1.3. A pendant vertex is removed from the chain. Let us consider the possible cases.

1.3.1. The removed singular vertex is the only singular vertex in the chain. A chain can have the type $2a$, $2b$ or 1, because after cutting ordinary edges we obtain a chain with one singular vertex, pendant or isolated. Therefore P does not change or decreases by w. Other values do not change. Thus, T does not change or decreases by w.

1.3.2. The removed singular vertex is not the only singular vertex in the chain. If during the transition from G to $\mathrm{o}(G)$ the value of S does not change (i.e., the segment adjacent to the pendant edge is even), then the pendant edge becomes non-pendant, and one of the following chain type changes may occur: $1a \to 3a$, $1b \to 3b$, $2 \to 1$, $2a \to 1a$, $2b \to 1b$, $1 \to 3$. In the first three cases D does not change, P does not change or decreases by w. Indeed, P cannot decrease by more than w, since the only pair $\{1a, 1b\}$, having quality $2w$, after, say, $1a \to 3a$ turns into a pair with quality w. In the last three cases, D is increased by one, P does not change or increases by w (this can be proved by considering the replacement). Other values do not change. Thus, T does not change or decreases by w. If S is incremented by one (that is, the segment adjacent to the pendant edge is odd), then the pendant edge remains pendant and the type of the chain does not change. Therefore, D and P do not change. Thus, T does not change.

1.4. A non-pendant extreme singular vertex is removed from the chain, i.e., there are no other singular vertices on the right or left of it. Let us consider possible cases.

1.4.1. The deleted vertex is the only singular vertex in the chain. A chain can have the type $3a$, $3b$, $2a$, $2b$ or 1, because after cutting ordinary edges we obtain a chain with one singular vertex. The value P does not change or decreases by w. If the chain is of type $3a$ or $3b$, then S is increased by one (since both segments adjacent to the vertex to be deleted are even, and the resulting segment is odd), D decreases by one. Thus, T either does not change or decreases by w. If the chain has type $2a$, $2b$ or 1, then D and S do not change (in case of types $2a$ or $2b$ two odd segments are replaced by one odd segment, in case of type 1, even and odd segments are replaced by one even segment). Thus, T does not change or decreases by w.

1.4.2. The deleted vertex is not the only singular vertex in the chain. If during the transition from G to $\mathrm{o}(G)$ the value of S does not change (i.e., the extreme segment is odd, and the next is even), then the pendant edge becomes non-pendant, and one of the following chain type changes may occur: $1a \to 3a$, $1b \to 3b$, $2 \to 1$, $2a \to 1a$, $2b \to 1b$, $1 \to 3$. The argument here is the same as in 1.3.2. If S is increased by one, then the type of the chain does not change (then T does not change) or (if both segments adjacent to the removed vertex are even), the non-pendant edge changes into a pendant edge. Then one of the following chain type changes may occur: $1a \to 2a$, $1b \to 2b$, $3 \to 1$, $3a \to 1a$, $3b \to 1b$, $1 \to 2$. In the latter three cases D decreases by one, P does not change or decreases by w. In the last three cases D does not change, P does not change or increases by w. Thus, in all cases the value of T either does not change or decreases by w.

2. o—*merging* (adding an edge).

2.1. Consider the case when endpoints of the same chain are merged together, which is possible only for odd chains. In this case d increases by one, f decreases by one, c increases by one or does not change (the latter case occurs if the chain has type $2a'$ or $2b'$, then the operation results in a loop), so $T(G)_1$ decreases by $1 - w$ or does not change. Consider different chain types. For a chain of type 0, the value of S is reduced by one, because the odd segment is converted to a cyclic one with length increased by one. Thus, T is reduced by one (recall that now $c(\mathrm{o}) = 1$). For the chain type $1a$ or $1b$, the values B and S do not change, D decreases by one, P does not change or decreases by w or $2w$. Thus, T changes by at most one. For chain type $2a$ or $2b$, the value D does not change, P does not change or decreases by w. If at both endpoints of the chain there are pendant edges, then when merging their ends together, B decreases by one, S does not change. If at least on one side the pendant edge is "replaced" with an odd segment, then B does not change, S decreases by one. If the chain has type $2a'$ or $2b'$, then B and S do not change. In all cases T will change by at most one. For chain type $3a$ or $3b$, the values B and S do not change, D decreases by one, P does not change or decreases by w. Thus, T will change by at most one.

2.2. Consider the case when endpoints of different chains are merged together. The table shows the result of merging the ends of non-pendant edges marked with a or pendant edges marked with b (thus, the added edge is marked with b). We will call such merging *a-merging* (symmetrically, *b-merging*). The result of a b-merging is equal to the result of a *symmetric* a-merging, which is obtained by swapping the labels a and b in merged chains.

Notation for endpoint types: $0a$—the endpoint of an odd chain of type 0, 0—the endpoint of an even chain of type 0, $1a$—the pendant endpoint of a chain of type $1a$, $1a'$—non-pendant end of a chain of type $1a$, $2a$—the end of a chain of type $2a$, $3a$—the end of a chain of type $3a$, 1—the pendant end of a chain of the type 1, $1'$—a non-pendant end of a chain of type 1, 2—the endpoint of a chain of type 2, 3—the endpoint of a chain of type 3.

We enumerate possible pairs of chain endpoint types. Each cell in the table shows in parentheses the changes T when moving from G to $\mathrm{o}(G)$, where 0 means "does not change," $(-w)$—"decreases by w," w—"increases by w," $(1 - w)$—"increases by $1 - w$" and so on; no other changes are possible. Corresponding changes of P with the same notation are given in square brackets. At the end of a cell we indicate the type of a chain resulting from merging original chains. The table is symmetric with respect to the main diagonal.

Let us check the cell $(1a, 1a)$; the rest are filled in the same way. If both source chains are even, then $T(G)_1$ increases by $1 - w$, otherwise it does not change. After merging, we get a chain of

Table. Results of merging the endpoints of two chains

	$0a$	0	$1a$	$1a'$	$2a$	$3a$	1	$1'$	2	3
$0a$	(0) $[0]$ $0a$	(0) $[0]$ 0	(0) $[0]$ $1a$	(0) $[0]$ $1a$	(0) $[0]$ $2a$	(0) $[0]$ $3a$	(0) $[0]$ 1	(0) $[0]$ 1	(0) $[0]$ 2	(0) $[0]$ 3
0	0 $[0]$ 0	(1) $[0]$ $0b$	$(0,w,2w)$ $[0,-w,-2w]$ 3	$(0,w,2w)$ $[0,-w,-2w]$ 2	$(0,w)$ $[0,-w]$ 1	$(0,w)$ $[0,-w]$ 1	$(1,1-w)$ $[0,w]$ $3b$	$(1,1-w)$ $[0,w]$ $2b$	$(1,1-w,1-2w)$ $[0,w,2w]$ $1b$	$(1,1-w,1-2w)$ $[0,w,2w]$ $1b$
$1a$	0 $[0]$ $1a$	$(0,w,2w)$ $[0,-w,-2w]$ 3	$(-2w,-w,0,w)$ $[0,-w,-2w,-3w]$ $3a$	$(-w,0,w)$ $[0,-w,-2w]$ $1a$	$(-w,0)$ $[0,-w]$ $1a$	$(-w,0,w)$ $[0,-w,-2w]$ $3a$	$(-w,0,w)$ $[0,-w,-2w]$ 3	$(-w,0,w)$ $[0,-w,-2w]$ 1	$(-2w,-w,0)$ $[0,-w,-2w]$ 1	$(-w,0,w)$ $[0,-w,-2w]$ 3
$1a'$	0 $[0]$ $1a$	$(0,w,2w)$ $[0,-w,-2w]$ 2	$(-w,0,w)$ $[0,-w,-2w]$ $1a$	$(-2w,-w,0,w)$ $[0,-w,-2w,-3w]$ $2a$	$(-w,0,w)$ $[0,-w,-2w]$ $2a$	$(-w,0)$ $[0,-w]$ $1a$	$(-w,0,w)$ $[0,-w,-2w]$ 1	$(-w,0,w)$ $[0,-w,-2w]$ 2	$(-w,0,w)$ $[0,-w,-2w]$ 2	$(-2w,-w,0)$ $[0,-w,-2w]$ 1
$2a$	0 $[0]$ $2a$	$(0,w)$ $[0,-w]$ 1	$(-w,0)$ $[0,-w]$ $1a$	$(-w,0,w)$ $[0,-w,-2w]$ $2a$	$(-w,0)$ $[0,-w]$ $2a$	$(w,0,-w)$ $[-w,0,w]$ $1a$	$(-w,0)$ $[0,-w]$ 1	$(0,w)$ $[0,-w]$ 2	$(-w,0)$ $[0,-w]$ 2	$(-w,0)$ $[0,-w]$ 1
$3a$	0 $[0]$ $3a$	$(0,w)$ $[0,-w]$ 1	$(-w,0,w)$ $[0,-w,-2w]$ $3a$	$(-w,0)$ $[0,-w]$ $1a$	$(w,0,-w)$ $[-w,0,w]$ $1a$	$(-w,0)$ $[0,-w]$ $3a$	$(0,w)$ $[0,-w]$ 3	$(-w,0)$ $[0,-w]$ 1	$(-w,0)$ $[0,-w]$ 1	$(-w,0)$ $[0,-w]$ 3
1	0 $[0]$ 1	$(1,1-w)$ $[0,w]$ $3b$	$(-w,0,w)$ $[0,-w,-2w]$ 3	$(-w,0,w)$ $[0,-w,-2w]$ 1	$(-w,0)$ $[0,-w]$ 1	$(0,w)$ $[0,-w]$ 3	$(1-w,1-2w)$ $[0,w]$ $3b$	$(1,1-w,1-2w)$ $[0,w,2w]$ $1b$	$(1-w,1-2w,1-3w)$ $[0,w,2w]$ $1b$	$(1-w,1-2w)$ $[0,w]$ $3b$
$1'$	0 $[0]$ 1	$(1,1-w)$ $[0,w]$ $2b$	$(-w,0,w)$ $[0,-w,-2w]$ 1	$(-w,0,w)$ $[0,-w,-2w]$ 2	$(0,w)$ $[0,-w]$ 2	$(-w,0)$ $[0,-w]$ 1	$(1,1-w,1-2w)$ $[0,w,2w]$ $1b$	$(1-w,1-2w)$ $[0,w]$ $2b$	$(1-w,1-2w)$ $[0,w]$ $2b$	$(1-w,1-2w,1-3w)$ $[0,w,2w]$ $1b$
2	0 $[0]$ 2	$(1,1-w,1-2w)$ $[0,w,2w]$ $1b$	$(-2w,-w,0)$ $[0,-w,-2w]$ 1	$(-w,0,w)$ $[0,-w,-2w]$ 2	$(-w,0)$ $[0,-w]$ 2	$(-w,0)$ $[0,-w]$ 1	$(1-w,1-2w,1-3w)$ $[0,w,2w]$ $1b$	$(1-w,1-2w)$ $[0,w]$ $2b$	$(1-2w,1-3w)$ $[0,w]$ $2b$	$(1-w,1-2w,1-3w)$ $[0,w,2w]$ $1b$
3	0 $[0]$ 3	$(1,1-w,1-2w)$ $[0,w,2w]$ $1b$	$(-w,0,w)$ $[0,-w,-2w]$ 3	$(-2w,-w,0)$ $[0,-w,-2w]$ 1	$(-w,0)$ $[0,-w]$ 1	$(-w,0)$ $[0,-w]$ 3	$(1-w,1-2w)$ $[0,w]$ $3b$	$(1-w,1-2w,1-3w)$ $[0,w,2w]$ $1b$	$(1-w,1-2w,1-3w)$ $[0,w,2w]$ $1b$	$(1-2w,1-3w)$ $[0,w]$ $3b$

type $3a$. If both chains contain pendant edges, B decreases by one and S does not change. If at least one of the source chains has an odd extreme segment instead of a pendant edge, then, on the contrary, B does not change and S decreases by one. The value of D decreases by one, P does not change or decreases by w, by $2w$, or $3w$. Let us show that P cannot decrease by $4w$ and more. Consider the partners of two chains of type $1a$ in the maximum region. The only case when the sum of the qualities of two pairs is $4w$ corresponds to partners of type $1b$. Then one of them can become the partner of a "new" chain of type $3a$, which gives an increase in quality by w. The value of P will not increase, otherwise it would have decreased during the reverse replacement, which is impossible since any partner of a chain of type $3a$ is a partner of $1a$.

So, the initial statement regarding merging follows from the fact that, according to the table, T changes by at most one.

3. o is a *splitting*. The operation is inverse to merging, so the result follows from item 2.

4. o is a *sesquialteral intermerging*. We distinguish three cases.

4.1. The operation is applied to one chain, i.e., gluing is carried out with an endpoint of the same chain where degluing is done. There are two cases.

4.1.1. There is an inversion of the extreme part of the chain. Consider an intermediate state when the cut is made, and the gluing has not yet been done. From this state, one can go to the initial state by merging, and one can pass to the final state by merging together with another endpoint of the same chain. Both mergings are a- or b-mergings. In the latter case, consider the a-mergings that are symmetric to b-mergings. We iterate over pairs of cells from the table that are adjacent horizontally and correspond to the pair of above-mentioned mergings (these are pairs of columns where one endpoint of the chain differs by the "stroke"), or pairs of matching cells (where endpoints of the chain are the same). We have to check that it is impossible for the absolute value of the difference between changes in T during the transition from the intermediate to the initial and final state to exceed one.

Consider, for example, a pair of cells $(1a, 1a)$ and $(1a, 1a')$; other pairs are treated similarly. When switching to the initial state corresponding to the first cell, the change in T is equal to $(-2w + iw)$, where $i = 0, 1, 2, 3$. When switching to the final state corresponding to the second cell, the change T is equal to $(-w + jw)$, where $j = 0, 1, 2$. The difference of these expressions (final minus initial) is $w - (i - j)w$. The corresponding difference between changes in P is $0 + (i - j)w$. Under the considered replacement $3a \to 1a$, the value of P either does not change or increases by w. Therefore, $i - j = 0$ or $i - j = 1$. Then the first expression is w or zero, which does not exceed one in absolute value.

4.1.2. There is a looping in the extreme part of the chain. It is a composition of two operations: splitting the original chain into two chains and closing one of them into a cycle or loop by merging it together. Consider an intermediate state when the splitting is made, and the merging has not yet been done. From this state, it is possible to pass to the initial state by merging, or to the final state by merging that loops the chain. Both are a- or b-mergings, and further consideration is similar to the previous case.

4.2. The operation is applied to the cycle (or loop) and a chain, i.e., a cycle (or loop) is cut and the resulting chain is lengthened. This operation is inverse to the previous one, so here again the change in T does not exceed one in absolute value.

4.3. The operation is applied to two chains: the first one being cut and the second one being glued together. We denote by C the chain which is that part of the cut chain that is glued to the second chain, which we denote C_2. The other part of the cut chain is denoted by C_1. When we have three chains C, C_1, C_2, the initial state (before the operation) is achieved by merging C together with C_1, and the final state (after the operation)—by merging together C with C_2. In order to show that T in the initial and final states differs by at most one, we have to check that the difference of its changes during the transitions does not exceed one in absolute value. We enumerate triples of types of endpoints to be merged. Since both mergings are a- or b-mergings, it is enough to consider the case of a-merging.

If in the initial and final states the value of P is the same, then the assertion follows from the fact that for the considered sesquialteral intermerging $T(G)_1$ changes by at most $1 - w$, and $T(G)_2$ changes by at most w. In particular, this condition is satisfied if the types of merged endpoints of C_1 and C_2 coincide. It remains to enumerate such triples of endpoints of chains C, C_1, C_2 that the latter two types are different. This means that we need to enumerate pairs of different cells that are in the same row of the table. Here the row corresponds to the endpoint of the chain C, and the two columns to the endpoints of chains C_1 and C_2. For each *such pair* we consider the differences between the changes in T and P (as in item 4.1.2). The replacement type $\langle$left cell type, right column type$\rangle \to \langle$right cell type, left column type$\rangle$ limits the latter difference. Let us check that under this constraint, the first difference does not exceed one in absolute value.

Consider a pair of cells $(1a, 0)$ and $(1a, 1a)$; all other pairs are treated in the same way. When switching to the initial state (merging with respect to the first cell), the change in T is equal to $0 + iw$, where $i = 0, 1, 2$. In the transition to the final state (merging with respect to the second cell), it is equal to $(-2w + jw)$, where $j = 0, 1, 2, 3$. The difference between these expressions (the final less the initial) is $(-2w - (i - j)w)$. The corresponding difference in the change of P is $0 + (i - j)w$. The transition from the initial state to the final state corresponds to the replacement $\{3, 1a\} \to \{3a, 0\}$. Since types 3 and 0 do not participate in actions, P either does not change or decreases by w. Therefore, $i - j$ is zero or negative one. Then the first difference is equal to $(-2w)$ or $(-w)$, which does not exceed one in absolute value.

5. o is a double intermerging. We distinguish three cases.

5.1. Both degluings occur in the same chain or in the same cycle. We consider the case of a chain, the case of a cycle is treated similarly.

5.1.1. Inversion of a segment in a chain. The value of $T(G)_1$ and the size of the chain do not change. The case when there are no singular vertices in the inverted segment is obvious. Otherwise, consider two possibilities.

1. Two singular vertices are identified, that is, B is decremented by one. If both degluings are internal (i.e., there are singular vertices on both sides of each degluing), then the type of the chain does not change, and D and P do not change as well. Instead of two segments adjacent to the degluings (the absence of a segment is assumed to be a segment of length zero), one segment appears of length one greater than the sum of the lengths of the initial segments. The segments are not extreme, therefore S either does not change or increases by one. Let one degluing internal and the other external. If the parity of the extreme segment does not change, then the previous argument holds. If an even extreme segment is replaced by an odd one, then S increases by one. Possibilities for changing the chain type are: $3a \to 1a$, $1 \to 2$, $1a \to 2a$, $3 \to 1$. In the first two cases, D does not change, P does not change or increases by w. In the latter two cases, D decreases by one, P does not change or decreases by w. If an odd extreme segment is replaced by an even one, S does not change. Possibilities for changing the chain type are: $1a \to 3a$, $2 \to 1$, $2a \to 1a$, $1 \to 3$. In the first two cases D does not change, P does not change or decreases by w. In the last two cases, D is incremented by one, P does not change or increases by w. In all cases, T changes by at most w. The case when both degluings are external is impossible when identifying singular vertices.

2. There is no identification of two singular vertices, that is, B does not change. If both degluings are internal, the type of the chain does not change, as well as D and P. Two segments adjacent to the degluings are replaced by two other segments of the same total length, S changes by at most one. Let one degluing be internal and the other external. If the parity of the outermost segment does not change, then the argument from the previous case holds (when both degluings are internal). If

an even extreme segment is replaced by an odd one, then S does not change or is increased by one. Possibilities for changing the chain type: $3a \to 1a$, $1 \to 2$, $1a \to 2a$, $3 \to 1$. In the first two cases D does not change, P does not change or increases by w. In the last two cases D decreases by one, P does not change or decreases by w. If an odd extreme segment is replaced by an even one, then S does not change or decreases by one. Possibilities for changing the chain type: $1a \to 3a$, $2 \to 1$, $2a \to 1a$, $1 \to 3$. In the first two cases D does not change, P does not change or decreases by w. In the last two cases, D is incremented by one, P either does not change or increases by w. Suppose now that both degluings are external. If both extreme segments had the same parity and preserved it, then only S can change and by at most one. If both segments were even and became odd, then S is increased by one. Possibilities for changing the chain type: $3a \to 2a$, $3 \to 2$. The value of D is decremented by one, P changes by at most w. If both segments were odd and became even, then S decreases by one. Possibilities for changing the chain type: $2a \to 3a$, $2 \to 3$. The value of D is incremented by one, and P changes by at most w. If one segment was even, and the other was odd, and these properties are preserved, then no values change. In all cases, T changes by at most w.

5.1.2. Splitting a cycle into two cycles or into a cycle and a loop (cutting a cycle or loop from a cycle): $T(G)_1$ decreases by $1 - w$ or does not change, $T(G)_2$ changes by at most w, P does not change. Therefore, T changes by at most one.

5.1.3. Cutting a section from a chain and looping it: $T(G)_1$ is reduced by $1 - w$ or does not change. Similarly to item 5.1.1, although there may be a case when two singular vertices are identified, and both degluings are external. Then the two extreme segments of the chain merge into one, the length of which is one more than the sum of the lengths of the original segments. If both segments are even, then S is increased by one, and the chain is of type $3a$ or $3b$. Therefore, D decreases by one, P does not change or decreases by w. If one or both of the segments are odd, then S does not change, and the chain is of type $2a$, $2b$ or 1. Therefore, D does not change, P either does not change or decreases by w. Thus, the difference $T(G)_2 - P$ changes by at most w, and T by at most one.

5.2. One degluing occurs in a chain, the other in a cycle or loop, i.e., a cycle or loop breaks and is inserted into the chain. The value of $T(G)_1$ either increases by $1 - w$ or does not change. If two singular vertices are not identified, then the operation is inverse to that considered in item 5.1.2. Otherwise, B decreases by one. If the degluing of the chain is internal, then the argument from item 5.1.1 holds, referring to the case when the two degluings considered there are internal. If the degluing of the chain is external, then the argument from item 5.1.1 holds, referring to the case when one of the degluings considered there is external.

5.3. Each degluing occurs in its own chain.

5.3.1. Both degluings are internal. As before, $B + S$ changes by at most one. To track changes in D and P, consider the types of both chains.

1. Both chains are odd and both are a- or b-chains. The value of $T(G)_1$ does not change. If both chains are of type $1a$, then the only non-identical type change is the transition of this pair into a pair of chains of types $2a$ and $3a$: the value of D decreases by one. Then P either does not change or decreases by w or $2w$. Any other pair of types of source chains (except $2a$ and $3a$, giving the reverse transition) leads to the same result. Type $1b$ is treated similarly.

2. Both chains are odd, and one is an a-chain and another is a b-chain. If their types pair is $(1a, 1b)$, then during the gluing it transitions into a pair $(1, 1)$ or a pair $(2, 3)$: $T(G)_1$ decreases by $1 - w$. In the first case, D decreases by 2, P decreases by $2w$. In the second case, D decreases by one, P decreases by $2w$. If the pair of source types was $(1a, 2b)$, then under gluing it transitions into a pair $(1, 2)$: $T(G)_1$ decreases by $1 - w$, D decreases by one, P decreases by w or $2w$. We can consider the pair $(1a, 3b)$ similarly. If a pair of source types $(2a, 2b)$, then when gluing it goes into a pair $(2, 2)$: $T(G)_1$ decreases by $1 - w$, D does not change, P either does not change or decreases

by w or by $2w$. We consider the pair $(3a, 3b)$ in a similar way. If this pair is $(2a, 3b)$, then during gluing it goes into a pair $(1, 1)$: $T(G)_1$ decreases by $1 - w$, D decreases by one, P decreases at w. The pair $(3a, 2b)$ is treated similarly.

3. Both chains are even: $T(G)_1$ does not change. The only non-trivial transition refers to pairs $(1, 1)$ and $(2, 3)$, the remaining transitions are identical or inverse to those considered in item 2. When the pair $(2, 3)$ transforms into the pair $(1, 1)$, the value of D decreases by one, and P does not change.

4. One chain is odd and the other is even: $T(G)_1$ does not change. If the pair $(1a, 1)$ goes to $(2a, 3)$, then D does not change, P does not change or decreases by w. Transition to the pair $(3a, 2)$ is treated similarly. If the pair $(1a, 2)$ goes to $(2a, 1)$, then D decreases by one, P does not change or decreases by w, and the same goes for the transition of pair $(1a, 3)$ to $(3a, 1)$. The transition of the pair $(2a, 1)$ to $(1a, 2)$ is inverse to the above. For the pair $(2a, 2)$, only the identical transition is possible. If the pair $(2a, 3)$ goes to $(3a, 2)$, then D does not change and P changes by at most w. Transitions $(2a, 3) \to (1a, 1)$, $(3a, 1) \to (1a, 3)$, $(3a, 2) \to (1a, 1)$ are inverse to those discussed above. For the pair $(3a, 3)$ only the identical transition is possible.

5.3.2. At least one of the deglings is external: the problem is now reduced to the case of a sesquialteral intermerging or merging. We will call the mentioned degluing and the edge of the chain to which it is adjacent *external*. We call the other degluing in o *internal*, like the edge that it deletes. If the external degluing is not on the extreme edge and not on the edge adjacent to it, then we remove two ordinary edges from the outer endpoint. At the same time, T will decrease by one in both the original graph and the resulting transformation: $T(G)_1$ will decrease by $1 - w$, $T(G)_2$—by w. Therefore, it suffices to consider the cases when the external degluing is on the extreme edge of a chain or on the penultimate edge.

1. External degluing is on the penultimate edge from the boundary. If it is ordinary, we will remove the two extreme edges along with the external degluing, replacing the double intermerging with sesquialteral. Then T decreases by one both in the initial graph and in the resulting one, which reduces the problem to that considered with sesquialteral intermerging. If the edge with the external degluing is singular, then we will remove the extreme edge along with its ends, replacing the odd extreme segment of length one by a pendant edge (the singular vertex becomes an endpoint of the pendant edge). We again replace the double intermerging with a sesquialteral with a decrease in the value of T in both graphs.

2. External degluing is on the extreme edge and it is ordinary. If the internal edge is ordinary or its singular vertex is from the side that is glued to the external edge, then we move the external degluing beyond the endpoint of the chain (i.e., make it "fictitious"), and the internal degluing, to the adjacent edge in the side opposite from the possible singular vertex (and if this edge is absent, then beyond the endpoint of the chain). Thus, double intermerging is replaced by a sesquialteral (or correspondingly identical operation). The result of the operation will not change. If a singular vertex is located on the other side, then we move the outer degluing from the edge to the adjacent edge and, accordingly, move the inner degluing to an adjacent edge. The result of the operation will not change, and the problem reduces to the degluing discussed in item 1.

3. External degluing is on the extreme edge, and it is singular. If the inner edge is ordinary or its singular vertex is from the side that is glued to the outer endpoint, then we argue the same way as in item 2. Otherwise, we will shift the external degluing beyond the endpoint of the chain, and the internal degluing—to the adjacent edge in the direction opposite from the singular vertex (and if this edge is absent, then beyond the endpoint of the chain). Thus, double intermerging is replaced by sesquialteral intermerging or merging. In order to obtain the result of the first operation (double intermerging) from the result of the second operation (sesquialteral intermerging or merging), we glue the two singular vertices, removing the common edge that separates them, and either add a

chain from one ordinary edge (in case of merging), add two extreme ordinary edges (if the internal degluing has shifted to an ordinary edge), or replace the pendant edge with an extreme segment of length one (if the internal degluing has shifted to a singular edge). In all cases, the conversion does not change T, which completes the enumeration of possible cases.

The linearity of running time and memory for this algorithm is obvious. This completes the proof of the theorem.

5. CONCLUSION

In this work, we have presented an algorithm for constructing a sequence of fixed operations with given costs that is minimal with respect to the total cost and transforms one graph to another. Such a sequence is called the shortest one. Regarding graphs we have assumed that the degree of each vertex is at most two; such graphs are called structures. Since the problem is NP-complete, it cannot be solved by any polynomial algorithm without any restrictions on the cost of operations. In this work, we have obtained a linear algorithm for solving the problem for the case when the cost c_2 of insertion and deletion operations does not exceed half the cost c_1 of standard operations (double and sesquialteral intermerging, splitting and merging). We have already prepared for publication a paper where we obtain a linear algorithm for solving this problem for the case $0.5c_1 \leqslant c_2 \leqslant c_1$, which together covers the case $c_2 \leqslant c_1$, and the case $c_1 \leqslant c_2$ is being investigated (for the case $c_1 = c_2$ the algorithm was previously published in [6]). All algorithms are accompanied by rigorous proofs of the fact that they find minima in the corresponding graph optimization problems.

ACKNOWLEDGMENTS

This work was carried out at the expense of the Russian Science Foundation, project no. 14-50-00150. The computations were performed on the supercomputer of the Joint Supercomputer Center of the Russian Academy of Sciences (JSCC RAS).

REFERENCES

1. Gorbunov, K.Yu. and Lyubetsky, V.A., A Linear Algorithm for the Shortest Transformation of Graphs with Different Operation Costs, *J. Commun. Technol. Electron.*, 2017, vol. 62, no. 6, pp. 653–662.

2. Lyubetsky, V.A., Gershgorin, R.A., Seliverstov, A.V., and Gorbunov, K.Yu., Algorithms for Reconstruction of Chromosomal Structures, *BMC Bioinformatics*, 2016, vol. 17, pp. 40.1–40.23.

3. Lyubetsky, V.A., Gershgorin, R.A., and Gorbunov, K.Yu., Chromosome Structures: Reduction of Certain Problems with Unequal Gene Content and Gene Paralogs to Integer Linear Programming, *BMC Bioinformatics*, 2017, vol. 18, pp. 537.1–537.18.

4. Braga, M.D.V. and Stoye, J., Sorting Linear Genomes with Rearrangements and Indels, *IEEE/ACM Trans. Computat. Biol. Bioinformatics*, 2015, vol. 12, no. 3, pp. 1–13.

5. Yin, Z., Tang, J., Schaeffer, S.W., and Bader, D.A., Exemplar or Matching: Modeling DCJ Problems with Unequal Content Genome Data, *J. Combinatorial Optimization*, 2016, vol. 32, no. 4, pp. 1165–1181.

6. Gorbunov, K.Yu. and Lyubetsky, V.A., Linear Algorithm for Minimal Rearrangement of Structures, *Probl. Inform. Transmiss.*, 2017, vol. 53, no 1, pp. 55–72.

This paper was recommended for publication by P.Yu. Chebotarev, a member of the Editorial Board