ISSN 0032-9460, Problems of Information Transmission, 2011, Vol. 47, No. 3, pp. 274–288. © Pleiades Publishing, Inc., 2011.
Original Russian Text © K.Yu. Gorbunov, V.A. Lyubetsky, 2011, published in Problemy Peredachi Informatsii, 2011, Vol. 47, No. 3, pp. 64–79.

LARGE SYSTEMS

The Tree Nearest on Average to a Given Set of Trees

K. Yu. Gorbunov and V. A. Lyubetsky

Kharkevich Institute for Information Transmission Problems,
Russian Academy of Sciences, Moscow
gorbunov@iitp.ru lyubetsk@iitp.ru

Received October 13, 2010; in final form, May 26, 2011

Abstract—We formulate the problem of constructing a tree which is the nearest on average to a given set of trees. The notion of "nearest" is formulated based on a conception of events such that counting their number makes it possible to distinguish each of the given trees from the desired one. These events are called divergence, duplication, loss, and transfer; other lists of events can also be considered. We propose an algorithm that solves this problem in cubic time with respect to the input data size. We prove correctness of the algorithm and a cubic estimate for its complexity.

DOI: 10.1134/S0032946011030069

The following problem is well known and has been studied for a long time in connection with various applications (for instance, in species evolution theory [1–5]). There is given a collection of trees G_i, where i ranges from 1 to some n, and it is required to find a tree S^* which is the *nearest on average* to each G_i. Usually it is assumed that the trees G_i, and then also S^*, are binary and rooted. Below (see Remark 2), we discuss how one can pass from nonbinary and nonroot trees to this simpler case. After specifying the italicized notions, the problem consists in finding the global minimum of a functional on a tree space; below we specify this functional and the space. Discrete optimization problems are known to rarely have an efficient and mathematically strict solution. For the rather general problem in question, we propose a worst-case (with respect to input data) solution algorithm of cubic complexity and prove that it strictly solves the described problem in this time. On typical inputs, it works even faster. A computer program realizing the algorithm is freely available at http://lab6.iitp.ru/ru/super3gl/, together with execution examples and a user manual. As one of possible interpretations of the problem, we propose an evolution model, which is formally described below and in [4] and, at a biological level, in [3, 5].

Thus, leaves of each tree G_i ("gene tree") are labeled by pairs $\langle k, l\rangle$ of positive integers; the first integer is referred to as a "gene," and the second, as a "species." In essence, this is a relation "gene k occurs in species l," which will be referred to as the "gene-species" relation. In a gene tree, a species l can be accompanied by several genes $\langle k_1, l\rangle$, $\langle k_2, l\rangle, \ldots$. In distinct gene trees G_i and G_j, species can be the same. We denote by V_0 the set of all species occurring in leaves of all the trees G_i.

Let us agree that the root of any tree is "at the top." Denote by e^- and e^+ the upper and lower endpoints of an edge e. An edge is understood as a pair of vertices: starting point e^- and ending point e^+. We denote an incoming edge of a vertex g by b_g. Each tree is considered together with its "root edge," which is a specially added edge which goes up from the root and corresponds to the time when the common ancestor of all genes or species that occur in the tree lived; the upper endpoint of the root edge is referred to as a "superroot." Edges of a species tree S are called *tubes* (in particular, a root edge is referred to as a root tube); this term is introduced only to distinguish between edges in S and edges in G_i. On vertices of any tree, we define the relation

"below": $g_1 < g_2$ if $g_1 \neq g_2$ and there is a path from the superroot to g_1 passing through g_2; throughout what follows, a "path" is understood as a *shortest* path with respect to the number of edges. The relation "below" between edges of a tree is defined similarly. Distinct edges are said to be *incomparable* if neither of them is below the other. Otherwise, the edges are comparable and lie on a common path from a leaf to the superroot. Edges outgoing from one vertex and the farthest from the root are said to be *adjacent*, as well as subtrees that have these edges as their root edges; they form a pair of adjacent edges and, respectively, a pair of adjacent subtrees. On the set of all vertices and tubes in S, we define a unified ordering relation $y < x$ as follows: a vertex or tube y is "below" a vertex or tube x in S if $y \neq x$ and there is a path from the superroot to y passing through x; accordingly, "x is above y." We denote $y \leq x$ if either $y < x$ or $y = x$. Each subtree (all that is below some vertex g) contains its root tree b_g but does not contain its upper endpoint. A clade M_s in a species tree is the set of species assigned to leaves that are below a vertex s in S. A clade M_g (assumed: in on of the gene trees G_i) is the set of species assigned to leaves that are below a vertex g in G_i. We call the vertices s and g the roots of the corresponding clades. A clade M_e is the set of species assigned to all leaves below an edge/tube e in a tree G or S.

Let P be a fixed collection of sets of species including V_0 and all of its one-element subsets but not including the empty set. A *tree space* $\boldsymbol{P}$ consists of all species trees S such that their set of leaves is in a one-to-one correspondence with the species in V_0 and all clades belong to P. In this sense, we call P a *collection of clades*. A *standard collection* P is the set of all clades in all initial gene trees G_i extended by the set V_0.

An *embedding* (with no transfers) of a tree G in a tree S is a mapping f of all vertices $V(G)$ of the gene tree G to vertices $V(S)$ and tubes $E(S)$ of the species tree S satisfying the following conditions:

1. The superroot in G is mapped to the root tube in S; each leaf g in G is mapped to a leaf s in S according to the gene-species relation;
2. Let g_1 be a son of g: if $f(g)$ is a vertex, then $f(g_1) < f(g)$; if $f(g)$ is a tube, then $f(g_1) \leq f(g)$;
3. Let g_1 and g_2 be sons of g: if $f(g)$ is a vertex, then a path from $f(g_1)$ to $f(g_2)$ in S passes through $f(g)$.

Note that an embedding is everywhere defined but is not necessarily injective or surjective.

For a given embedding f, a *duplication* is a nonsuperroot vertex g in G for which $f(g)$ is a tube. A *divergence* is a vertex g for which $f(g)$ is a nonleaf vertex. A *loss* is a pair $\langle e, s\rangle$ such that e is an edge in G, s is a vertex in S, and $f(e^+) < s < f(e^-)$.

Remark 1. These definitions are based on an intuitive conception of the process of "evolution of a protogene in a protospecies" located in a root tube. This process can briefly be described as follows. Figures 1 and 2 illustrate the notions of gene duplication and loss, and also the divergence. Duplication of a gene is appearance of two its copies irrelative to a furcation in S; since duplication corresponds to no vertex in S, it is drawn inside a tube. Divergence of a gene is appearance of two its copies in a fork in S, where one copy ("life line") goes to one—and the other, to the other—of two tubes outgoing from this fork (to adjacent tubes), and both copies are not lost in them. A loss of a gene happens when two copies appear in a fork in S and the copy is lost in one of the adjacent tubes and is not lost in the other; in this case the life line is drawn only in the adjacent tube where the copy is not lost. If both copies of a gene are lost in adjacent tubes, then the gene was already lost before the furcation. A copy of a gene is also a gene. In the course of evolution, a gene as a sequence changes; the more the time, the more.

An *embedding* f of a collection of trees $\{G_i\}$ in a tree S is a collection of embeddings $f = \{f_i\}$ where each f_i is an embedding of G_i in S.

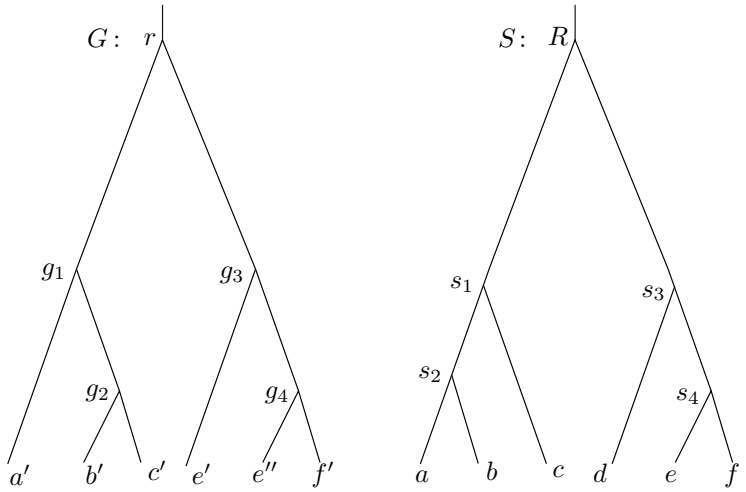

Fig. 1. Illustration of the notions of duplication, gene loss, and divergence. Gene tree G and species tree S in leaves of which: gene a' is taken from species a, etc.; two genes e' and e'' are taken from the same species e. Species d is not represented in G.

Problem 1 is to find a global minimum point of the functional

$$c(\{G_i\}, f, S) = \sum_i \big(c_l l(f_i, G_i, S) + c_d d(f_i, G_i, S)\big) \tag{1}$$

in the above-specified space $\boldsymbol{P}$, where c_l and c_d are fixed nonnegative numbers and S and all the f_i are variables over which the global minimization is performed. Recall that $\boldsymbol{P}$ consists of trees with the set V_0 of leaves. In (1) we use the following notation: $l(f, G_i, S)$ is the number of losses in G_i under the embedding f_i of G_i in S, and c_l is the cost of a single loss; thus, $c_l \sum\limits_i l(f_i, G_i, S)$ is the total cost of all losses in all the G_i. Similarly, $d(f_i, G_i, S)$ is the number of duplications in G_i under the embedding f_i of G_i in S, and c_d is the cost of a single duplication, so that $c_d \sum\limits_i d(f_i, G_i, S)$ is the total cost of all duplications in all the G_i. Thus, the "proximity" of each G_i and S is defined through the embedding f_i. We call an embedding $f^* = \{f_i^*\}$ a *scenario* for a collection of trees $\{G_i\}$ if it is a solution to problem (1). Then the value c^* of the functional (1) at the global minimum point $\langle f^*, S^*\rangle$ is referred to as the *minimal cost* (of the scenario), and S^* itself is referred to as a *supertree* for the collection $\{G_i\}$.

An intensional interpretation of a solution to problem (1) depends on the parameter P. Our computer experiments have shown that in evolution problems it is reasonable to first take a standard collection P and then extend it with differences of sets contained in P; the role of such differences is explained in [6].

A *pair scenario* h for one gene tree G and a given species tree S is an embedding of G in S which minimizes the functional (1) where i takes precisely one value, i.e., $n = 1$ and $G = G_1$. Here S is fixed, and the only variable is h. If f^* is a scenario for some collection $\{G_i\}$ and S^* is a supertree corresponding to this collection, then all the f_i^*, clearly, are pair scenarios for each pair G_i and S^*.

For any G and S, a pair scenario $h(G, S)$ is unique and even independent of the choice of fixed nonnegative values of the costs c_d and c_l of a single duplication and single loss. This scenario h is explicitly described in Lemma 1 below. Therefore, when finding a supertree S^*, into each term on the right-hand side of (1) instead of f_i we may substitute the corresponding unique scenario $h(G_i, S) = h_i$ as a function of S, and then the functional (1) becomes independent of the variables f_i. In what follows, we assume that this substitution is made.

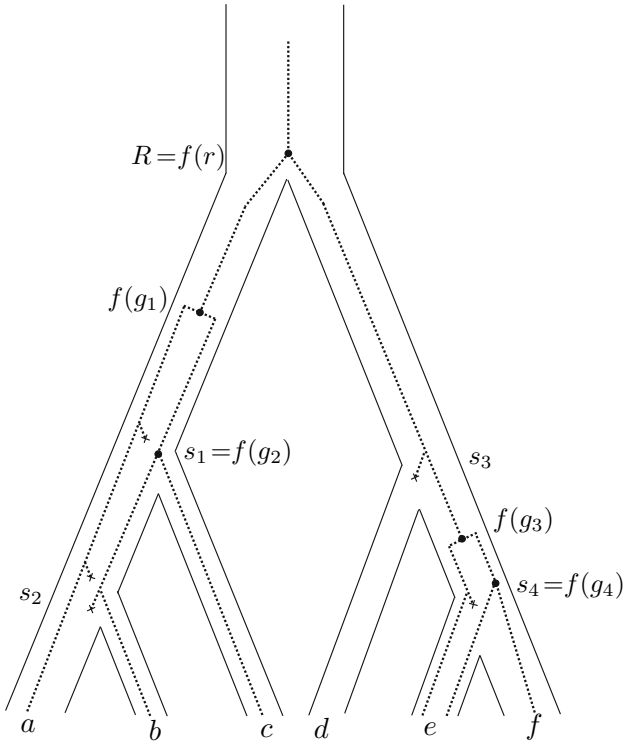

Fig. 2. Illustration of the notions of duplication, gene loss, and divergence. Values of the embedding f of G in S are shown by bold dots inside tubes of S, except for values on leaves in G that coincide with the corresponding leaves in S. The value $f(g_1)$ is shown inside the tube (though formally it equals this tube), and the vertex g_1 corresponds to a duplication event by definition. The same for g_3. Values of f for all other interior vertices of G coincide with the corresponding interior vertices of S and correspond to divergence events by definition. For the edge $l = (g_1, a')$, vertices s_1 and s_2 lie between the values of f at the endpoints of l, and the pairs $\langle l, s_1\rangle$ and $\langle l, s_2\rangle$ by definition correspond to loss events; losses are shown as legs with crossed ends. Similarly, the pairs $\langle (g_2, b'), s_2\rangle$, $\langle (g_3, e'), s_4\rangle$, and $\langle (r, g_3), s_3\rangle$ are losses.

We say that a set V from P is a *basis* set if it can be partitioned into two parts from P, then each part, in turn, can be partitioned into two parts from P, etc., until one-elements sets representing species are obtained. It is not known beforehand which sets from P are basis sets; in particular, we do not know whether V_0 itself is a basis set. However, if a solution S^* to problem (1) exists, then V_0 is a basis set, since the required partitioning of V_0 is defined by clades in S^*, which in this case are contained in P by definition.

To describe our algorithm, we also need the following notions. Let f be an embedding; then an edge e in G *enters* a tube b in S if $f(e^+) \le b < f(e^-)$. It happens that the set of all edges entering any tube b can be found efficiently for a pair scenario $h(G, S)$. Namely, for any set of species M we define $\mathrm{Ed}(M, G)$ as the set of edges e in G such that $M_e \subseteq M$ and there is no edge $e' > e$ with this property. There can be several such edges e, and all of them are incomparable in G. Then, for any tube b, the set of all edges entering b coincides with $\mathrm{Ed}(M_b, G)$; this is precisely the result of Lemma 2 (a).

Recall that a subtree in G or S defined by a vertex g has root g, and b_g is its root edge/tube. In what follows, a collection P of sets and a collection $\{G_i\}$ of trees are always fixed, and so usually they are not mentioned explicitly.

Description of the algorithm. For each gene tree G_i and all sets V from P, the sets $\mathrm{Ed}(V, G_i)$ can be constructed by direct search according to their definition.

Then, using joint induction on the growing number of elements in V, we construct some specific trees $S(V)$. More precisely, we compute some number, the "price" $c(V)$ of V, and then, using it, trivially construct $S(V)$ for which all clades belong to P and precisely the species from V are assigned to leaves. We refer to any tree of the form $S(V)$ as a *basis* tree.

Thus, let V be a basis set from P. Initial step: for V we take one-element sets from P, each of them consisting of one species; the cost $c(V)$ is set to zero by definition, and the corresponding tree $S(V)$ consists by definition of a single leaf to which this species is assigned.

Induction step: we consider all possible partitions of the set V from P into two *basis* sets V_1 and V_2. Intuitively, this means checking whether a fork at the root of the further tree $S(V)$ will be defined by a partition of V into V_1 and V_2. If there are no such partitions, the set V is marked as "nonbasis"; all sets with fewer elements than in V are already marked as either "basis" or "nonbasis." If V_0 is marked as "nonbasis," the algorithm outputs the message "problem (1) has no solution." Below we consider the case where V is a basis set.

For any partition of a basis set V into basis sets V_1 and V_2, by the induction hypothesis we have already computed the costs $c(V_1)$ and $c(V_2)$ and constructed the basis trees $S(V_1)$ and $S(V_2)$. For each G_i, we set $l(i) = |\mathrm{Ed}(V_1, G_i)| + |\mathrm{Ed}(V_2, G_i)|$ and $d(i) = l(i) - |\mathrm{Ed}(V, G_i)|$; here, $|\cdot|$ denotes the cardinality of a set. Now we look through all trees in the collection $\{G_i\}$ in an arbitrary order, and for each of them look through all vertices in G_i. For every such vertex, if an edge of one of its sons belongs to $\mathrm{Ed}(V_1, G_i)$ and an edge of another one belongs to $\mathrm{Ed}(V_2, G_i)$, we reduce the numbers $l(i)$ by 2 and the numbers $d(i)$ by 1. We denote the resulting numbers by $l(V, V_1, V_2, G_i)$ and $d(V, V_1, V_2, G_i)$.

Now we find a partition of V into V_1^* and V_2^* for which the functional

$$c(V, V_1, V_2) = \sum_i [c_l l(V, V_1, V_2, G_i) + c_d d(V, V_1, V_2, G_i)] + c(V_1) + c(V_2) \tag{2}$$

attains its minimum over all partitions of a fixed V into basis sets V_1 and V_2. By definition, let $c(V)$ be the value of the functional (2) at this *minimal partition* $\langle V_1^*, V_2^* \rangle$. We call the obtained $c(V)$ the *cost of the set* V. After that, the basis tree $S(V)$ is by definition obtained by adding a root to the basis trees $S(V_1^*)$ and $S(V_2^*)$; the root corresponds to V, and its sons, to V_1^* and V_2^*. The costs $c(V_1)$ and $c(V_2)$ of V_1 and V_2 are obtained inductively as values of the functional (2) at some of its minimal partitions. The algorithm description is completed.

There can be several minimal partitions; it would be interesting to obtain a nontrivial bound on their number.

To characterize all basis trees composing a collection $\{S(V) : V \text{ is a basis set}\}$, we have to slightly extend Problem 1: find a global minimum of the functional (1) on the same tree space $\boldsymbol{P}$ where summation over the trees G_i is extended by summation over all their subtrees G' for which edges from $\mathrm{Ed}(V, G_i)$ are root edges and V_0 is replaced by V. We obtain a new functional

$$c(\{G_i\}, f, S) = \sum_i \sum_{G'} \big(c_l l(f_{G'}, G', S) + c_d d(f_{G'}, G', S)\big). \tag{3}$$

This extension of Problem 1 will be called *Problem 2*. If $V = V_0$, then all G' in G_i coincide with G_i, and therefore the functional (3) coincides with (1) and Problem 2 coincides with Problem 1. For any trees G' and S, a unique (again, as will be seen from Lemma 1) pair scenario $h(G', S)$ can be substituted for $f_{G'}$, and then minimization over variables $f_{G'}$ in (3) is not needed, as well as in (1). A solution S^* to Problem 2 will also be referred to as a *supertree* (for a species set V). In what follows, we assume that this substitution of $h(G', S)$ instead of $f_{G'}$ in (3) is made.

Theorem. *Let P be a collection of clades.*

(a) *The set V_0 is a basis set if and only if the tree $S(V_0)$ found by the algorithm is a solution to Problem* 1.

(b) *For any basis set V from P, the tree $S(V)$ found by the algorithm is one of solutions to Problem* 2. *Conversely, any solution to Problem* 2 *is of the form $S(V)$ under an appropriate choice of a sequence of minimal partitions.*

(c) *If P is a standard collection and the average number of leaves in the collection of gene trees $\{G_i\}$ is of order $|V_0|$, then the algorithm finds the set $\{S(V) : V$ is a basis set$\}$ in a number of steps of order $|P|^3 + |P|^2|V_0|n \le Cn^3|V_0|^3$. In this time, the algorithm either outputs a solution to Problem* 1 *or reports that it does not exist.*

The proof of the theorem uses Lemmas 1–3 given below and will be presented after proving them.

Vertices g and s are said to be *matching* if they are not superroots and also either (1) g and s are leaves obeying the gene-species relation, or (2) two partitions of the set M_g coincide: the one defined by a fork in g and the one defined by a fork in s. The latter means: ($M_{g1} \subseteq M_{s1}$ and $M_{g2} \subseteq M_{s2}$) or ($M_{g1} \subseteq M_{s2}$ and $M_{g2} \subseteq M_{s1}$), where g_1 and g_2 are sons of g, and s_1 and s_2 are sons of s. We denote by $\sup M$ the vertex in S which is the least upper bound of the set M of leaves (species) in S. Recall that b_s denotes a tube in S with an endpoint at s. Denote by $h(g) = h(G, S)$ the following mapping of vertices g of G to vertices and tubes of S:

$$\begin{cases} \text{If } g \text{ is a superroot in } G, \text{ then } h(g) \text{ is a root tube in } S; \text{ otherwise:} \\ \text{if the vertices } g \text{ and } \sup M_g \text{ are matching, then } h(g) = \sup M_g; \\ \text{otherwise: } h(g) = b_s, \text{ where } s = \sup M_g. \end{cases} \tag{4}$$

Lemma 1. *The map $h(g)$ is an embedding with the following property: for any embedding f of G in S, $f \ne h$, the numbers of duplications and losses for f are not less than these numbers for h, and at least one of these numbers for f is strictly greater than that for h.*

This immediately implies that for any nonnegative costs c_l and c_d the embedding h is a unique pair scenario for G and S.

Proof. Let us check that h is an embedding. Property 1 holds trivially. The nonstrict inequality in property 2 follows from the fact that $\sup M_{g1} \le \sup M_g$. The strict inequality in property 2: the vertices g and $\sup M_g$ are matching; therefore, $h(g_1) \le b_{s1}$ or $h(g_1) \le b_{s2}$, i.e., $h(g_1) < s$. Property 3: similarly to the aforesaid, $h(g_1) \le b_{s1}$ and $h(g_2) \le b_{s2}$ (or symmetrically); i.e., $h(g_1)$ and $h(g_2)$ belong to different adjacent subtrees.

Up to the end of the proof, let f be different from h, i.e., $f \ne h$.

For any vertex g in G we have

$$f(g) \ge h(g). \tag{5}$$

Indeed, by property 2 we have $f(g) \ge \sup M_g$. If $f(g) = \sup M_g$, then property 3 implies that g and $\sup M_g$ are matching vertices, whence we get $f(g) = h(g)$. If $f(g) > \sup M_g$, then $f(g) \ge h(g)$.

$$\text{If } f(g) > \sup M_g, \text{ then } f(g) \text{ is a tube.} \tag{6}$$

Assume that $f(g)$ is some vertex s. There exists a son s_1 of s for which $s_1 \ge \sup M_g$ does not hold. By property 3, for sons g_1 and g_2 of g we have $M_{g1} \subseteq M_{s1}$ or $M_{g2} \subseteq M_{s1}$, which contradicts the condition $f(g) > \sup M_g$.

From (5) and (6) we have the following:

$$\text{For any vertex } g, \text{ if } h(g) \text{ is a tube, then } f(g) \text{ is a tube.} \tag{7}$$

Hence, the number of duplications for f is not less than the number of duplications for h. Let us prove the same for losses.

Let $\langle e, s\rangle$ be a loss for h. Let $f(e^+) < s$. Then $\langle e, s\rangle$ is a loss for f, taking (5) into account. Otherwise, since $f(e^+) \geq h(e^+) < s$, we have $s = f(e^+)$ or $s < f(e^+)$. The former is impossible, since $f(e^+)$ is a tube according to (6). Let $s < f(e^+)$. Let us show that a path from e^+ to any leaf l contains an edge e' such that $\langle e', s\rangle$ is a loss for f; thus, to an initial loss $\langle e, s\rangle$ for h there corresponds a set consisting of at least two distinct losses of the form $\langle e', s\rangle$ for f. Indeed, $h(e^+) < s$ means $\sup M_{e^+} < s$. Consider any vertex g on the path from e^+ to l; let us show that $f(g) \neq s$. If this is not the case, then $\sup M_g \leq \sup M_{e^+} < f(g) = s$ and $f(g)$ is a tube according to (6), a contradiction. By property 2 we have $f(l) \leq f(g) \leq f(e^+)$; furthermore, we have $f(l) < s < f(e^+)$ (the first inequality holds by the condition, and the second, due to $f(l) = h(l) \leq h(e^+) < s$). Hence, in this path there are neighboring vertices k^+ and k^- for which $f(k^+) < s < f(k^-)$, i.e., the edge $\langle k^+, k^-\rangle$ and the vertex s form a loss for f. Thus, to each loss for h there corresponds either the same loss for f ("first case"; consider it as a one-element set) or a set of losses for f of cardinality strictly greater than one ("second case"). Let us prove that these sets are disjoint. Let $\langle e_1, s_1\rangle$ and $\langle e_2, s_2\rangle$ be two losses for h. The corresponding losses for f are of the form $\langle e_1', s_1\rangle$ and $\langle e_2', s_2\rangle$. If $s \neq s_1$, these pairs are distinct. Otherwise, we have $s_1 = s_2$ and $e_1 \neq e_2$. By the definition of a loss, the edges e_1 and e_2 are incomparable in G, and therefore e_1' and e_2' are distinct. Hence, the number of losses for f is not less than the number of losses for h.

Now assume that the number of duplications for f and h is the same. Let us show that then the second case occurs at least once (i.e., $s < f(e^+)$), and therefore the number of losses for f is strictly greater than the number of losses for h. Since $f \neq h$, there exists a vertex g in G such that $f(g) > h(g)$, i.e., $f(g) > \sup M_g$, and $f(g)$ is a tube according to (6). By (7) and by the assumption, we have

$$\{k \mid h(k) \text{ is a tube}\} = \{k \mid f(k) \text{ is a tube}\}; \tag{8}$$

then $h(g)$ is a tube. Consider a vertex s for which $h(g) < s < f(g)$. Consider a path in G from g to the superroot. For any vertex g' in it, we have $h(g') \neq s$. If this is not the case, then $h(g') = s$, and by property 2 we have $f(g') \geq f(g) > s = h(g') = \sup M_{g'}$, $f(g')$ is a tube by (6), and we obtain a contradiction to (8). Hence, in this path there are two neighboring vertices g' (maybe, equal to g) and g'' (maybe, equal to the superroot) such that $h(g') < s < h(g'')$, i.e., the edge $e = (g'', g')$ and the vertex s form a loss for h. By property 2 we have $f(e^+) \geq f(g)$, and by the choice of g and s we have $f(g) > s$; hence, $f(e^+) > s$, i.e., the second case takes place. △

For any tube b, the set of all edges entering b and the set $\mathrm{Ed}(M_b, G)$ are related as follows.

Lemma 2. *If h is a pair scenario for G and S, then*:

(a) *The tube b in S is entered by precisely the edges from* $\mathrm{Ed}(M_b, G)$;

(b) *If a tube b_1 is a son of a tube b, then for any edge e in G entering b_1 there exists precisely one edge $e' \geq e$ entering b.*

Proof. (a) Let e be an edge from $\mathrm{Ed}(M_b, G)$. Then $\sup M_{e^+} \leq b^+$ and $\sup M_{e^-} \geq b^-$. By the definition (4) of a pair scenario h, we conclude that e enters b. Conversely, if e enters b, then $\sup M_{e^+} \leq b^+$ and $\sup M_{e^-} \geq b^-$, i.e., e belongs to $\mathrm{Ed}(M_b, G)$.

(b) Consider a path from e to the root edge. Let e_1 be the first edge in this path for which $b^- \leq h(e_1^-)$. Then either e_1^+ coincides with the upper endpoint of the edge preceding e_1 in this path or $e_1 = e$. In both cases $h(e_1^+) \leq b$, and hence e_1 enters b. Since only one of two comparable edges may enter a tube, no other edge in this path enters b. △

For any gene tree G and species tree S and the corresponding pair scenario $h(G, S)$, define the "locus" of each evolution event: the *locus* of a duplication g is the tube $h(g)$, and the *locus* of a loss $\langle e, s\rangle$ is the tube b_s, which is technically more convenient than what was said in Remark 1. Recall

that a supertree for V is a tree minimizing the functional $C(V,S)$ given by (3), and a minimal partition is a partition minimizing another functional $c(V,V_1,V_2)$ given by (2).

We say that a nonleaf vertex g in a gene tree is *paralogical* if the clade M_g consists of a single species. For any pair scenario, a paralogical vertex is a duplication in a leaf tube, and vice versa: any duplication in a leaf tube is a paralogical vertex. Terms in (1) that correspond to paralogical vertices can be discarded, since their sum is a constant, having no effect on minimization.

Lemma 3. (a) *Let S_0 be a species tree with a leaf set V_0, and let S be its subtree with a leaf set V. The total cost $Z(S)$ of events (duplications and losses) occurring in S in pair scenarios for all G_i and a given S_0 equals $C(V,S)$.*

(b) *If V_1,V_2 is a partition at the root of any tree S with leaf set V and if subtrees at the root are of the form $S(V_1)$ and $S(V_2)$, then $c(V,V_1,V_2)=C(V,S)$.*

If, moreover, S is a minimal tree, then the partition V_1,V_2 is minimal.

(c) *Let S be an arbitrary tree with leaf set V, and let S_1 be any its proper subtree with leaf set V_1. If $[S,S_2/S_1]$ is the result of replacing the subtree S_1 in S with a subtree S_2 having the same leaves as S_1, and $C(S_2)\le C(S_1)$, then $C([S,S_2/S_1])\le C(S)$.*

Proof. (a) Let us show the following: the total cost Z of these events for a single G_i equals one term $C(G_i,V,S)$ in the sum from $C(\{G_i\},V,S)$ corresponding to this G_i. Let G' be a subtree in G_i whose root edge belongs to $\mathrm{Ed}(V,G_i)$. In what follows, G stands for G_i.

The formulated statement follows from a more general one: any event occurring under a pair scenario for G' and S is an event in S under the pair scenario for G and S_0, and conversely, any event occurring in S under the pair scenario for G and S_0 is an event under the pair scenario for G' and S, for a unique G'.

By the definition (4) of a pair scenario h, we have the following: if a vertex g belongs to a subtree G', then $h_{G'}(g)$ under the pair scenario for G' and S coincides with $h_1(g)$ under the pair scenario for G and S_0, and conversely: if $h_1(g)$ belongs to S, then there exists a unique G' containing g and $h_{G'}(g)=h_1(g)$. Indeed, the clade of $h_1(g)$ in S is contained in V, and then, by the above-mentioned definition, the clade of g is contained in V; thus, moving upwards from g, we find a root edge of the desired unique G'.

Check: if there is a duplication or loss in S under the pair scenario h_1, then it remains the same event for precisely one pair scenario $h_{G'}$ for G' and S, and vice versa. For a duplication, this immediately follows from the preceding paragraph.

Let $\langle e,s\rangle$ be a loss under the pair scenario h_1 for G and S_0, and let b_s belong to S. Then e^+ belongs to some G'. If e is not a root edge in G', then $\langle e,s\rangle$ is a loss under the pair scenario $h_{G'}$, since the images of e^+ and e^- do not change. If e is a root edge in G', then $\langle e,s\rangle$ is also a loss under the pair scenario $h_{G'}$, since the image of e^+ remains the same and $h_{G'}(e^-)$ equals the root tube of S; i.e., $h_{G'}(e^+)<s<h_{G'}(e^-)$.

Let $\langle e,s\rangle$ be a loss for $h_{G'}$. If e is not a root edge in G', then $\langle e,s\rangle$ is a loss under the pair scenario h_1 too, since the images of e^+ and e^- do not change. If e is a root edge in G', then $\langle e,s\rangle$ is also a loss for h_1, since $h_{G'}(e^+)=h_1(e^+)$ and $h_{G'}(e^-)<h_1(e^-)$.

Summing the terms $C(G_i,V,S)$ over i, we obtain assertion (a).

(b) The second claim of this item immediately follows from the first: if this partition is not minimal, we pass to trees over a minimal partition and obtain a tree over V with a strictly smaller cost C, which is impossible.

We check the first claim by induction. If S consists of a single leaf, then V consists of a single species, and $C(V,S)=0$ by (a) (we omit terms corresponding to paralogical vertices), so $c(V)=0$ by definition.

Induction step: by the induction assumption, for the trees $S(V_1)$ and $S(V_2)$ we have the following: $c(V_1) = C(V_1, S(V_1))$, by (a) we have $c(V_1, S(V_1)) = Z(S(V_1))$, and similarly $c(V_2) = C(V_2, S(V_2)) = Z(S(V_2))$. Then

$$c(V, V_1, V_2) = \sum_i [c_l l(V, V_1, V_2, G_i) + c_d d(V, V_1, V_2, G_i)] + Z(S(V_1)) + Z(S(V_2));$$

below we show that the first term for each G_i is the cost of events at the root tube of S under a pair scenario for G_i and for an arbitrary S_0 with leaf set V_0 that contains S as a subtree. Therefore, the right-hand side is the cost of events in all tubes of this tree, i.e., $Z(S(V))$. According to (a), this equals $C(V, S)$. Denote by b the root tube of S, and by b_1 and b_2, the tubes outgoing from b.

By Lemma 2 (a), for each gene tree G the tubes b, b_1, and b_2 are entered by edges of G, respectively, from $\mathrm{Ed}(V, G)$, $\mathrm{Ed}(V_1, G)$, and $\mathrm{Ed}(V_2, G)$. By definition, the sets $\mathrm{Ed}(V_1, G)$ and $\mathrm{Ed}(V_2, G)$ are disjoint, and any two edges from their union M are incomparable in G. By Lemma 2 (b), each edge e in G entering b_1 or b_2 has a unique ancestor, namely, an edge $e' \geq e$ entering b. In vertices of G lying on the path from e to e' there are duplications in b, and in the first vertex of the path or on the edge e there is, respectively, a divergence or loss. The edge e' from $\mathrm{Ed}(V, G)$ generates a subtree in G with a root at the ending point of e' and leaves at starting points of edges from M. For edges from $\mathrm{Ed}(V, G)$, we obtain a *forest* of such trees, their number being $|\mathrm{Ed}(V, G)|$; edges from M bijectively correspond to leaves, and edges from $\mathrm{Ed}(V, G)$ are root edges. These trees contain $d(G) = |\mathrm{Ed}(V_1, G)| + |\mathrm{Ed}(V_2, G)| - |\mathrm{Ed}(V, G)|$ vertices of G. Under the pair scenario for G_i and S_0, each of them is mapped either to a tube b, and then is a duplication, or to a vertex r (fork at the root) and then is a divergence. Conversely: if the image of a vertex is b or r, then it is one of these $d(G)$ vertices, since, when moving from this vertex along any path in G, we necessarily come to an edge from M. Of these $d(G)$ vertices, divergences are those for which the edge of one son belong to $\mathrm{Ed}(V_1, G)$, and the edge of the other, to $\mathrm{Ed}(V_2, G)$. All other vertices are duplications. For any edge e from M, a pair $\langle e, r\rangle$ is a loss if and only if e is not a son of a divergence. Conversely: any loss of the form $\langle e, r\rangle$ corresponds to the edge e from M. Thus, we have shown that $l(V, V_1, V_2, G_i)$ is the number of losses at the fork, and $d(V, V_1, V_2, G_i)$ is the number of duplications in b.

(c) Arbitrarily extend S to some species tree S_0 with leaf set V_0. Let us show that we have the following:

$$\begin{aligned} &\text{If } C(S_2) \leq C(S_1), \text{ then } C([S_0, S_2/S_1]) \leq C(S_0), \text{ and} \\ &\text{if } C(S_2) < C(S_1), \text{ then } C([S_0, S_2/S_1]) < C(S_0). \end{aligned} \tag{9}$$

Consider a pair scenario for any particular G_i and S_0 and compare the related events in S_0 before replacing the subtree S_1 with S_2 in S_0 and after this replacement, when we obtain a tree S_3 instead of S_0. According to (a), the total cost of events occurring in S_2 is not greater than the total cost of events occurring in S_1. Now it suffices to show that events in the part of S_0 that was not changed remain the same.

Let us check that each event occurring in the complement of S_2 in S_3 occurs in the complement of S_1 in S_0, and vice versa. If $\langle g, h(g)\rangle$ is a duplication in the complement of S_2 in S_3 after the replacement, then, by the definition (4), $h(g)$ is the same tube before the replacement, and therefore $\langle g, h(g)\rangle$ is a duplication before the replacement too. If $\langle e, s\rangle$ is a loss in the complement of S_2 in S_3 after the replacement, then $h(e^-)$ does not lie in S_2 (now our subtree does not include the superroot), and by (4) it did not lie in S_1 before the replacement. If $h(e^+)$ either does not lie in S_2, it has not changed under the replacement, and therefore $\langle e, s\rangle$ was also a loss before the replacement. If $h(e^+)$ lies in S_2, then by (4) it lied in S_1 before the replacement and, since s does not lie in S_2, we had $h(e^+) < s < h(e^-)$ before the replacement too, and hence $\langle e, s\rangle$ was also a loss before the replacement. Thus, claim (9) is proved.

Now (9) immediately implies (c). Indeed, from the condition we have $C([S_0, S_2/S_1]) \le C(S_0)$. Assume that (c) does not hold; then $C(S) < C([S, S_2/S_1])$, and by the second part of (9) we obtain $C(S_0) < C([S_0, S_2/S_1])$, a contradiction. $\triangle$

Proof of the theorem. (a) One implication in this statement follows from (b), since the set $\mathrm{Ed}(V_0, G_i)$ consists of the root edge of the gene tree G_i. The other implication is obvious.

(b) We use induction on the cardinality of V. For any basis set V, consider a minimal tree S^* with leaf set V and with all clades from P. Let V_1 and V_2 at the root fork in S^* correspond to subtrees S_1 and S_2; by the induction hypothesis, $S(V_1)$ and $S(V_2)$ are minimal. Choose S^* so that the subtrees S_1 and S_2 coincide with $S(V_1)$ and $S(V_2)$. To this end, replace S_1 with $S(V_1)$, then by Lemma 3 (c) the value of the functional $C(V, S)$ does not change; make the same for S_2. By Lemma 3 (b), this partition of V into V_1 and V_2 is minimal, and conversely, to any minimal partition there corresponds a minimal tree. Therefore, $S^* = S(V)$, as is claimed in item (b) of the theorem. The last assertion in (b) can easily be proved by induction.

(c) For each element from P we look through at most $|P|$ variants of its partition, and for each variant we look through all vertices in all gene trees, which corresponds to time of order $|P|^2|V_0|n$. Preliminary construction of the sets $\mathrm{Ed}(M, G_i)$ for all sets M in P requires time of order $|P||V_0|n$. Preliminary constriction of inclusion and intersection relations for the sets from P requires time of order $|P|^2|V_0|$. Preliminary construction of all variants of partitioning sets from P into two sets from P requires time of order $|P|^3$ (for each triple P_1, P_2, P_3 we must check that P_2 and P_3 are disjoint and $|P_2| + |P_3| = |P_1|$). Hence follows the aggregate time estimate of order

$$|P|^3 + |P|^2|V_0|n \le Cn^3|V_0|^3. \quad \triangle$$

Remark 2. (1) If gene trees are not binary, then the described algorithm should be modified as follows. Instead of the number of edges in $\mathrm{Ed}(V, G_i)$, one should consider the number of vertices such that at least one of their filial edges belongs to $\mathrm{Ed}(V, G_i)$. The same for $\mathrm{Ed}(V_1, G_i)$ and $\mathrm{Ed}(V_2, G_i)$. Instead of looking through divergences, i.e., vertices such that one of their filial edges belongs to $\mathrm{Ed}(V_1, G_i)$ and the other to $\mathrm{Ed}(V_2, G_i)$, one should look through vertices such that among their filial edges there is at least one edge from $\mathrm{Ed}(V_1, G_i)$ and at least one edge from $\mathrm{Ed}(V_2, G_i)$.

(2) If gene trees are not rooted, the following procedure is used for rooting. Let each leaf be assigned a label, the name of a taxonomic group to which the species represented in this leaf belongs. We call this label a *taxon*. From a given collection of gene trees, trees with a single taxon are deleted. For the remaining trees, taxons are partially ordered with respect to the age (in ascending order); such information is usually available from biological data. Formally, one can take any arrangement of these labels and any ordering defined on them. For each tree G, find the number k of the oldest taxons. For example, let us describe the procedure for the cases of $k = 1$ or $k = 2$, which usually take place in biological data. The general case can be treated similarly. Let $M(G)$ consist of the oldest taxon if $k = 1$, and of two most old taxons if $k = 2$ and the total number of taxons is at least three; otherwise, $M(G)$ consists of one of the oldest taxons (no matter which particular one). We compute $p(G)$, the "density" of $M(G)$ in G, as follows. First, for each edge $\{u, v\}$ (unordered pair) of G we compute a parameter d. Let b_u be the number of leaves with a taxon from $M(G)$ in the part U of the partition of G by this edge that is adjacent to u, and b_v, in the part V that is adjacent to v. Let l_u be the total number of leaves in U, and l_v, in V. Then $d_u = b_u/l_u$ is the fraction of leaves with a taxon from $M(G)$ among all leaves in U, and $d_v = b_v/l_v$, in V. Put $d = (\sqrt{d_u} - \sqrt{d_v})^2$. Find an edge $e(G)$ with the largest value of d (denote this value by $d_{\max}$). If there are several such leaves, put $d_{\mathrm{pmax}} = d_{\max}$. Otherwise, let d_{pmax} be the second largest value of d. Put $p = \sqrt{d_{\max}} - \sqrt{d_{\mathrm{pmax}}} + (d_{\max})^2$. In the collection of gene trees, retain only trees with p greater than a predetermined threshold. For each of such trees G, define a root at the middle of the edge $e(G)$. Now we can apply our algorithm to this collection of trees.

The obtained algorithm provided good results for collections of binary and nonbinary, rooted and nonrooted trees. A computer program for constructing a supertree, as well as execution examples and a user manual, is freely available at `http://lab6.iitp.ru/ru/super3gl/`.

In conclusion, let us state a mathematical problem, which in our opinion is one of key problems in mathematical description of evolution. We start with several definitions.

First, it is necessary to introduce a conception of lead time of evolution events. Maybe, to this end we should pass to a continuous description of the discrete picture described below. We proposed the following description of discrete time [3, 4]. We distinguish between an initial species tree S_0 and a new species tree S obtained from S_0 by dividing some tubes in S_0 into serial parts ("new tubes"). As a result, tubes with a single son appear in S. An algorithm for passing from S to S_0 is proposed in [3]. In the case of an embedding with no transfers, we have $S = S_0$. "Time slices" are a partition of the set of all tubes in S into disjoint sets enumerated from 1 to m; each set is one "time slice"; they must satisfy the following: For any tube b in the ith slice, its son b_1 belongs to the $(i+1)$st slice. The first slice contains the root tube of S, and the last (mth) slice consists of all tubes incoming to leaves of S. Then the ith slice consists of all tubes that have an incoming path of i tubes including the root tube. We write $b_1 \sim b_2$ if $b_1 \neq b_2$ and the tubes b_1 and b_2 belong to the same slice. Intuitively, tubes collected in one slice belong to the same time period, and simultaneous events among them are possible. By a partition of a tree G we call a tree G' obtained from G by dividing some its edges into serial parts, which results in appearance of "new edges" having a single son. In the case of an embedding with no transfers, we have $G' = G$.

Second, we have to describe the evolution event of a horizontal gene transfer [3, 4]. We have done this as follows. An *embedding* (with transfers) is a mapping f from all vertices $V(G')$ of some partition G' of G to vertices $V(S)$ and tubes $E(S)$ of S satisfying the following conditions:

1. The superroot of G' is mapped to the root tube of S; each leaf g in G' is mapped to a leaf s in S subject to the gene-species relation.

   In what follows, g, g_1, and g_2 are vertices in G';
2. Let g_1 be a son of g: if $f(g)$ is a vertex, then $f(g_1) < f(g)$, and if $f(g)$ is a tube, consider the following two cases. If g_2 is another son of g, then either $f(g_i) \leq f(g)$ for both sons or we have $f(g_i) \leq f(g)$ for one son and $f(g) \sim f(g_j)$ for the other; here $f(g_i)$ is a vertex or tube and $f(g_j)$ is a tube, $i, j = 1, 2$. If g with a parent g' has only one son g_1, then either $f(g_1) \leq f(g) \sim f(g')$ or $f(g) \sim f(g_1)$; here in the first expression $f(g_1)$ is a vertex or a tube and $f(g')$ is a tube, and in the second expression, $f(g_1)$ is a tube;
3. Let g_1 and g_2 be sons of g: if $f(g)$ is a vertex, then a path in S from $f(g_1)$ to $f(g_2)$ passes through $f(g)$; if g has only one son, then $f(g)$ is a tube.

Now a *duplication* of a gene is a vertex g in G' with two sons g_1 and g_2 for which $f(g)$ is a tube in S and for both sons we have $f(g_i) \leq f(g)$, $i = 1, 2$. A *divergence* is a vertex g in G' for which $f(g)$ is a vertex in S and each of the vertices g and $f(g)$ has two sons. A gene *loss* is a pair $\langle e, s\rangle$ for which e is an edge in G', s is a vertex in S having two sons, and $f(e^+) < s < f(e^-)$. A preserving *horizontal transfer* is a vertex g in G' with two sons g_1 and g_2 such that $f(g)$ is a tube in S and we have $f(g) \sim f(g_i)$ for precisely one of the sons g_i. A nonpreserving *horizontal transfer* is a vertex g in G' with a single son g_1 such that $f(g)$ is a tube and $f(g) \sim f(g_1)$. Usually, a nonpreserving transfer is considered as a series of two events: a preserving transfer and a gene loss at the source. Figures 3 and 4 illustrate an embedding with losses and transfers.

An analog of Problem 1, a scenario (with a transfer), pair scenario (with a transfer), etc., can be defined as above using a functional generalizing the functional (1):

$$c_{\text{trans}}(\{G_i\}, f, S) = \sum_i (c_l l(f_i, G_i, S) + c_d d(f_i, G_i, S) + c_t^+ t^+(f_i, G_i, S) + c_t^- t^-(f_i, G_i, S)). \quad (10)$$

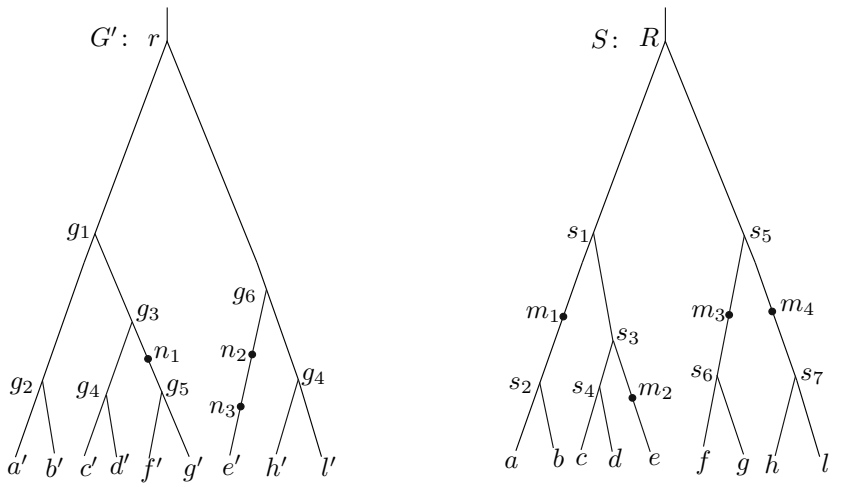

Fig. 3. Illustration of the notion of a horizontal transfer: gene tree G' and species tree S; notation at the leaves is the same as in Figs. 1 and 2. Vertices with a single son added to, respectively, G and S are marked with bold dots. In the ith slice of S there are tubes to which a path of i tubes leads from the superroot.

Here $f = \{f_i\}$, $t^+(f, G, S)$ is the number of preserving transfers for f, c_t^+ is the cost of one preserving transfer, $t^-(f, G, S)$ is the number of nonpreserving transfers for f, and c_t^+ is the cost of one nonpreserving transfer.

Note that, unlike Lemma 1, there exist gene trees G and species trees S and values of costs of a single event for which a pair scenario (with transfers) is not unique.

Example. Let $G = ((a, c), b)$, $S = ((a, b), (c))$, and assume that to species a, b, and c in G there are assigned genes whose names are not given. The notation (c) indicates that a tube in S that joins the root with leaf c is divided into two serial parts. Here we have three time slices; at the ith slice there are tubes to which a path of i tubes leads from the superroot. Vertices in G and S are denoted in the same way as their clades, the root tube is denoted by r, and an edge/tube in G or S incoming to a leaf is denoted by the name of this leaf. Costs of event are as follows: $c_l = 1$, $c_d = 2$, $c_t^+ = 3$, and $c_t^- = 4$. Then there are two pair scenarios: (1) scenario f^* without transfers, where $f^*(\{a, c\}) = \{a, b, c\}$ and $f^*(\{a, b, c\}) = r$, which corresponds to one duplication $\{a, b, c\}$ and two losses $\langle b, \{a, b, c\}\rangle$ and $\langle b, \{a, b\}\rangle$; (2) scenario f^* with transfers, where G' is obtained from G by adding a new vertex g' to the edge a and where $f(\{a, c\}) = c$, $f(\{a, b, c\}) = \{a, b, c\}$, and $f(g') = a$, which corresponds to one preserving transfer $\{a, c\}$ and one loss $\langle b, \{a, b\}\rangle$. If we increase the cost of a preserving loss, then only one scenario remains, the first embedding; if we reduce this cost, then again only one scenario remains, the second embedding.

Problem. Prove a statement similar to the theorem for a more complicated functional (10).

Remark 3. Let us eliminate a misunderstanding concerning the algorithm from [3, 4], which is closely related to the algorithm of the present paper. The first phrase in [4, Section 3] reads: "The run time of the algorithms is proportional to the product of the number of edges in the gene tree and the number of tubes in the species tree already divided into time slices." In [7], the first of these numbers is denoted by $|G|$, and the second, by $|S'|$. Then the run time of the algorithm from [3, 4] is $O(|S'||G|)$. This estimate is proved in [3], and a little more formally, in [4]. Precisely the same estimate is claimed to be the main result of [7] (the end of the second paragraph on p. 94), though [7] contains references to both papers [3, 4].

Despite some evolutionary terminology, whose biological meaning is irrelevant for the present paper, the theorem and this problem have a pure mathematical content.

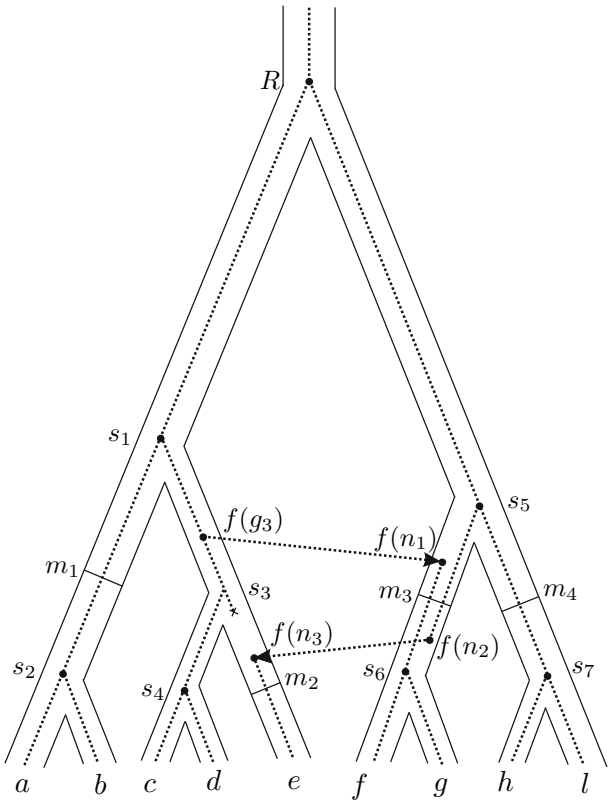

Fig. 4. Illustration of the notion of a horizontal transfer: embedding f of G' in S (for the trees shown in Fig. 3). Values of the embedding f of G' in S are shown by bold dots inside tubes of S, except for values on leaves in G' that coincide with the corresponding leaves in S. The value $f(g_3)$ is shown inside a tube (though formally it equals this tube), and the vertex g_3 corresponds by definition to a preserving transfer event. An arrow is drawn from $f(g_3)$ to $f(n_1)$, where n_1 is the corresponding son of g_3. The value $f(n_2)$ is shown inside a tube (though formally it equals this tube), and the vertex n_2 corresponds by definition to a nonpreserving transfer event. An arrow is drawn according to the same rule. A loss is shown as a leg with crossed end. Divergences: $R = f(r)$, $s_1 = f(g_1)$, $s_2 = f(g_2)$, $s_4 = f(g_4)$, $s_5 = f(g_6)$, $s_6 = f(g_5)$, and $s_7 = f(g_7)$. The vertex s_3 is not a value of f.

APPENDIX

Example of executing the algorithm. We illustrate the algorithm by an artificial example where we have ten gene trees G_i presented in Fig. 5. These trees are chosen so that it is easy to find a supertree S^* for them, which is given in the same figure. Let P be a standard collection, the loss cost be 2, and the duplication cost be 3. Here $V_0 = \{a, b, c, d, e\}$. Using the collection $\{G_i\}$, let us compute costs of all ten two-element sets V. Partitions $V = \{x\} \cup \{y\}$ and their costs computed according to formula (2) are presented in a table. The table also shows the number t of gene trees G_i where V is not a clade. Each of these trees generates two losses and 0 duplications, so their contribution to $c(V, V_1, V_2)$ equals the number of such trees multiplied by four. The other trees give zero contributions to $c(V, V_1, V_2)$. As a result, we find that the sets $\{a, b\}$ and $\{c, d\}$ have the minimal cost; i.e., for them we have $c(V) = 24$.

Now consider the three-element set $V = \{c, d, e\}$. Methodologically, we call its partition that coincides with the partition in S^* a *standard* partition, and say that all others are *nonstandard*. Of course, our algorithm does not use S^* but looks through all partitions. Here, a standard partition is only the partition of $V = \{c, d, e\}$ into $V_1 = \{c, d\}$ and $V_2 = \{e\}$. Let us compute the value $c(V, V_1, V_2)$ of the functional (2) on it. For that, we consider three cases: (1) $\{c, d\}$ is a

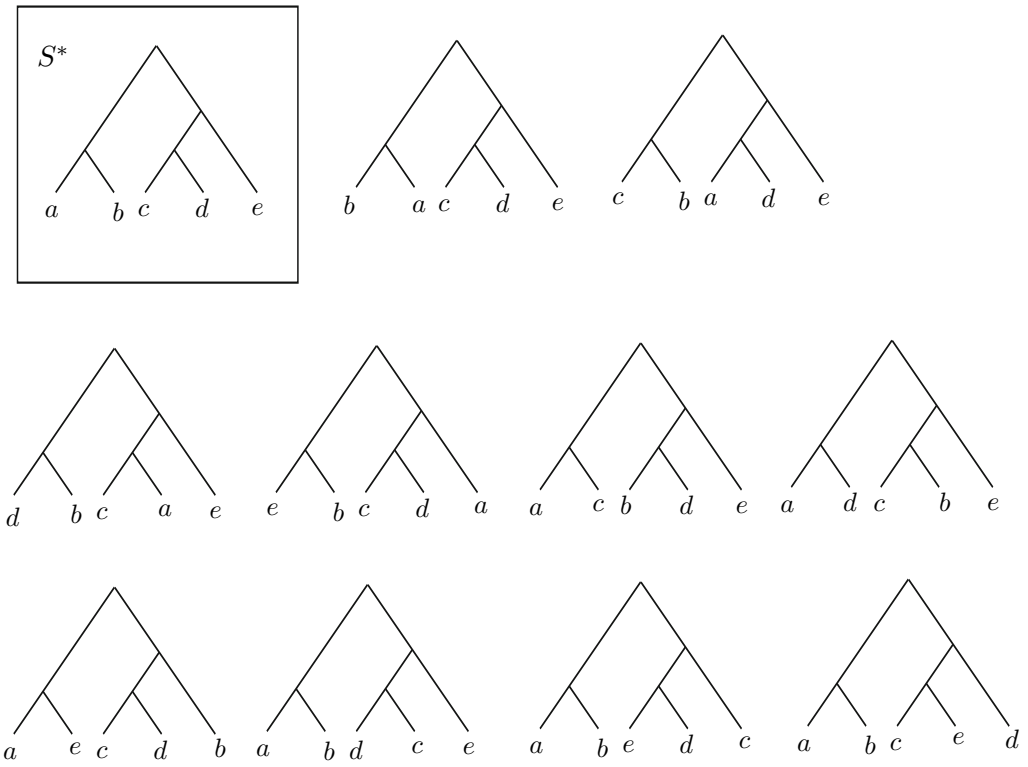

Fig. 5. Supertree S^* and ten gene trees that are input to the algorithm.

Table

$V = \{x, y\}$	t	$c(V)$	$V = \{x, y\}$	t	$c(V)$
a, b	6	24	b, d	8	32
c, d	6	24	a, e	9	36
a, c	8	32	b, e	9	36
a, d	8	32	c, e	9	36
b, c	8	32	d, e	9	36

clade and $\{c, d, e\}$ not a clade for two gene trees; (2) $\{c, d, e\}$ is a clade and $\{c, d\}$ not a clade for two gene trees; (3) $\{c, d\}$ and $\{c, d, e\}$ are not clades for four gene trees. Finally, for the standard partition we obtain $c(V, V_1, V_2) = 8 + 10 + 24 + c(\{c, d\}) + c(\{e\}) = 42 + 24 + 0 = 66$, since by induction we have $c(\{c, d\}) = 24$ and $c(\{e\}) = 0$. Now consider a nonstandard partition of the same set V into $V_1 = \{c, e\}$ and $V_2 = \{d\}$, one of two symmetric partitions. Let us compute $c(V, V_1, V_2)$. Again we consider three cases: (1) $\{c, d\}$ is a clade and $\{c, d, e\}$ not a clade for two trees; (2) $\{c, d, e\}$ is a clade and $\{c, e\}$ not a clade for three trees; (3) neither of the sets $\{c, d\}$, $\{c, e\}$, and $\{d, e\}$ is a clade for four trees. Finally, for the nonstandard partition we obtain $c(V, V_1, V_2) = 4 + 15 + 24 + c(\{c, e\}) + c(\{d\}) = 43 + 36 + 0 = 79$, since by induction we had $c(\{c, e\}) = 36$ and $c(\{d\}) = 0$. We have examined all cases of partitioning V into two parts from P, and we choose the partition with the smallest value (equal to 66) of the functional (2); in our case, this is the standard partition. Therefore, the tree $S(\{c, d, e\})$ coincides with a subtree in S^*. Similarly, we compute $c(V_0, V_1, V_2)$ for the set $V_0 = \{a, b, c, d, e\}$ of all species and its standard partition into $V_1 = \{a, b\}$ and $V_2 = \{c, d, e\}$ (it equals 128) and for its nonstandard partitions (the smallest of these values is 143). Thus, the algorithm outputs the tree $S(V_0)$ with cost 128, which coincides with S^*.

REFERENCES

1. *Phylogenetic Supertrees. Combining Information to Reveal the Tree of Life*, Bininda-Emonds, O.R.P., Ed., Dordrecht: Kluwer, 2004.

2. Guigo, R., Muchnik, I., and Smith, T.F., Reconstruction of Ancient Molecular Phylogeny, *Mol. Phylogenet. Evol.*, 1996, vol. 6, no. 2, pp. 189–213.

3. Gorbunov, K.Yu. and Lyubetsky, V.A., Reconstructing the Evolution of Genes along the Species Tree, *Molekulyarnaya Biol.*, 2009, vol. 43, no. 5, pp. 946–958 [*Molecular Biol.* (Engl. Transl.), 2009, vol. 43, no. 5, pp. 881–893].

4. Gorbunov, K.Yu. and Lyubetsky, V.A., On An Algorithm for Gene/Species Trees Reconciliation Taking into Account Duplications, Losses, and Horizontal Gene Transfers, *Inform. Protsessy*, 2010, vol. 10, no. 2, pp. 140–144.

5. Gorbunov, K.Yu. and Lyubetsky, V.A., Fast Algorithm for Building Species Supertrees Given a Set of Protein Trees, to appear in *Molekulyarnaya Biol.*, 2011.

6. Gorbunov, K.Yu. and Lyubetsky, V.A., Identification of Ancestral Genes That Introduce Incongruence between Protein- and Species Trees, *Molekulyarnaya Biol.*, 2005, vol. 39, no. 5, pp. 847–858 [*Molecular Biol.* (Engl. Transl.), 2005, vol. 39, no. 5, pp. 741–751].

7. Doyon, J.P., Scornavacca, C., Gorbunov, K.Yu., Szeollosi, G.J., Ranwez, V., and Berry, V., An Efficient Algorithm for Gene/Species Trees Parsimonious Reconciliation with Losses, Duplications and Transfers, *Comparative Genomics (Proc. Int. Workshop RECOMB-CG 2010, Ottawa, Canada, 2010)*, Tannier, E., Ed., Lect. Notes Comp. Sci., vol. 6398, Lecture Notes in Bioinformatics, Berlin: Springer, 2010, pp. 93–108.