Anchored Alignment Problem for Rooted Labeled Trees

Yuma Ishizaka[1], Takuya Yoshino[1], and Kouichi Hirata[2](✉)

[1] Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Kawazu 680-4, Iizuka 820-8502, Japan
{y_ishizaka,yoshino}@dumbo.ai.kyutech.ac.jp

[2] Department of Artificial Intelligence, Kyushu Institute of Technology, Kawazu 680-4, Iizuka 820-8502, Japan
hirata@dumbo.ai.kyutech.ac.jp

Abstract. An *anchored alignment tree* between two rooted labeled trees with respect to a mapping that is a correspondence between nodes in two trees, called an *anchoring*, is an alignment tree which contains a node labeled by a pair of labels for every pair of nodes in the anchoring. In this paper, we formulate an *anchored alignment problem* as the problem, when two rooted labeled trees and an anchoring between them are given as input, to output an anchored alignment tree if there exists; to return "no" otherwise. Then, we show that the anchored alignment problem can be solved in $O(h\alpha^2 + n + m)$ time and in $O(h\alpha)$ space, where n is the number of nodes in a tree, m is the number of nodes in another tree, h is the maximum height of two trees and α is the cardinality of an anchoring.

1 Introduction

An *anchored alignment tree* between two rooted labeled trees (trees, for short) with respect to a mapping that is a correspondence between nodes in two trees, called an *anchoring*, has been introduced by Schiermer and Giegerich [5] in the context of forest alignments in bioinformatics. Then, the anchored alignment tree is an alignment tree which contains a node labeled by a pair of labels for every pair of nodes in the anchoring. By using an anchoring whose number is α, we can obtain the anchored alignment tree in α times faster than the case without using an anchoring [5].

Note first that an arbitrary anchoring between two trees does not always provide an anchored alignment tree; If an anchoring is not less-constrained [4], then there exists no alignment tree between them, because the less-constrained mapping coincides with an alignable mapping [3], and then we can construct an alignment tree from no less-constrained anchoring. Then, in this paper, we deal with the following *anchored alignment problem*.

This work is partially supported by Grant-in-Aid for Scientific Research 24240021, 24300060, 25540137, 26280085 and 26370281 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

T. Murata et al. (Eds.): JSAI-isAI 2014 Workshops, LNAI 9067, pp. 296–309, 2015.
DOI: 10.1007/978-3-662-48119-6_22

AnchoredAlignment
Instance: Two trees T_1 and T_2, and a mapping $M \subseteq V(T_1) \times V(T_2)$, called an *anchoring*.
Solution: Find an *anchored alignment tree* $\mathcal{T}$ of T_1 and T_2 such that $\mathcal{T}$ contains a node labeled by $(l(v), l(w))$ for every $(v, w) \in M$ if $\mathcal{T}$ exists; return "no" otherwise.

Note that the anchored alignment tree as output is not necessary to be optimum in the sense of the alignment distance or the minimum cost alignment [2]; it is just an alignment tree between two trees containing nodes labeled by every pair of labels in an anchoring.

In order to solve the anchored alignment problem, in this paper, we provide an alternative proof that a less-constrained mapping coincides with an alignable mapping [3]. In this proof, first we introduce the cover sequences consisting of nodes of complete subtrees from a node in a mapping to the root. Then, we show that a mapping is less-constrained if and only if, for every pair of nodes in the mapping, the cover sequence of a tree and one in another tree are comparable. By using this property, we can prove the above theorem, according to following algorithm to solve the problem of AnchoredAlignment. Here, n is the number of nodes in T_1, m is the number of nodes in T_2, h is the maximum height of T_1 and T_2 and α is the cardinality of an anchoring M.

First, we compute cover sequences of an anchoring and determine whether or not they are comparable in $O(h\alpha)$ time and space. If so, then next we construct an alignment subtree by aligning these cover sequences and by merging them in $O(h^2\alpha)$ time. Finally, we complete an anchored alignment tree by adding appropriate alignment subtrees to the merged alignment subtree in $O(n + m)$ time. Hence, we can solve the problem of AnchoredAlignment in $O(h\alpha^2 + n + m)$ time and in $O(h\alpha)$ space.

Schiermer and Giegerich [5] have introduced the anchoring to divide the dynamic programming to compute the alignment distance [2] into α parts and claimed to reduce the time complexity from $O(nmD^2)$ time [2] to $O(nmD^2/\alpha)$ time, where D is the maximum degree of two trees. However, since the anchoring is not always less-constrained, we cannot guarantee that the division is correct. On the other hand, this paper determines whether or not the anchoring is less-constrained and, if so, then uses it to find the anchored alignment tree directly and correctly in $O(h\alpha^2 + n + m)$ time. When $n \geq m$, we can roughly estimate that $O(nmD^2/\alpha) = O(h\alpha^2 + n + m) = O(n^3)$. Hence, we can find the anchored alignment tree as fast as [5] even if an anchoring is less-constrained.

2 Preliminaries

A *tree* is a connected graph without cycles. For a tree $T = (V, E)$, we denote V and E by $V(T)$ and $E(T)$, respectively. We sometimes denote $v \in V(T)$ by $v \in T$. We denote an empty tree by $\emptyset$.

A *rooted tree* is a tree with one node r chosen as its *root*. We denote the root of a rooted tree T by $r(T)$. For each node v in a rooted tree with the root r, let

$UP_r(v)$ be the unique path from v to r. If $UP_r(v)$ has exactly k edges, then we say that the *depth* of v is k and denote it by $d(v) = k$. The *height* of T, denoted by $h(T)$, is defined as $\max\{dep(v) \mid v \in T\}$. The *parent* of $v(\neq r)$, which we denote by $par(v)$, is its adjacent node on $UP_r(v)$ and the *ancestors* of $v(\neq r)$ are the nodes on $UP_r(v) - \{v\}$. We say that u is a *child* of v if v is the parent of u. In this paper, we use the ancestor orders $<$ and $\leq$, that is, $u < v$ if v is an ancestor of u and $u \leq v$ if $u < v$ or $u = v$. In particular, we denote neither $u \leq v$ nor $v \leq u$ by $u \# v$. We say that w is the *least common ancestor* of u and v, denoted by $u \sqcup v$, if $u \leq w$, $v \leq w$ and there exists no w' such that $w' < w$, $u \leq w'$ and $v \leq w'$. A (*complete*) *subtree of* $T = (V, E)$ *rooted at* v, denoted by $T[v]$, is a tree $T' = (V', E')$ such that $r(T') = v$, $V' = \{u \in V \mid u \leq v\}$ and $E' = \{(u, w) \in E \mid u, w \in V'\}$.

A rooted tree is *labeled* if every node is labeled by some alphabet. A rooted tree is *ordered* if a left-to-right order among siblings is fixed; *unordered* otherwise. In particular, for nodes u and v in an ordered tree, u is *to the left of* v, denoted by $u \preceq v$, if $pre(u) \leq pre(v)$ and $post(u) \leq post(v)$ for the preorder number *pre* and the postorder number *post*. In this paper, we call a rooted labeled tree a *tree* simply. If it is necessary to distinguish, we call either ordered trees or unordered trees.

We say that two sets A and B are *incomparable* if none of $A \subset B$, $A = B$ and $B \subset A$ holds, that is, there exist both $a \in A \setminus B$ and $b \in B \setminus A$; *comparable* otherwise. Also we say that two sequences $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ of sets are *incomparable* if there exist i and j ($1 \leq i \leq n$, $1 \leq j \leq m$) such that A_i and B_j are incomparable; *comparable* otherwise. Furthermore, we call a sequence $A_1, \ldots, A_n$ of sets such that $A_i \subseteq A_{i+1}$ ($1 \leq i \leq n-1$) *increasing*.

3 Less-Constrained Mapping

In this section, we introduce a less-constrained mapping and characterize it as cover sequences.

Definition 1 (Mapping [6]). Let T_1 and T_2 be trees and $M \subseteq V(T_1) \times V(T_2)$. We say that a triple (M, T_1, T_2) is a *Tai mapping* between T_1 and T_2 if every pair (v_1, w_1) and (v_2, w_2) in M satisfies the following conditions.

1. $v_1 = v_2$ iff $w_1 = w_2$ (one-to-one condition).
2. $v_1 \leq v_2$ iff $w_1 \leq w_2$ (ancestor condition).
3. $v_1 \preceq v_2$ iff $w_1 \preceq w_2$ (sibling condition).

For unordered trees, the condition 3 is omitted. We will use M instead of (M, T_1, T_2) when there is no confusion. Furthermore, we denote the set $\{v \in T_1 \mid (v, w) \in M\}$ by $M|_1$ and the set $\{w \in T_2 \mid (v, w) \in M\}$ by $M|_2$.

Definition 2 (Less-Constrained Mapping [3,4]). Let T_1 and T_2 be trees. We say that a mapping M between T_1 and T_2 is a *less-constrained mapping* if M satisfies the following condition.

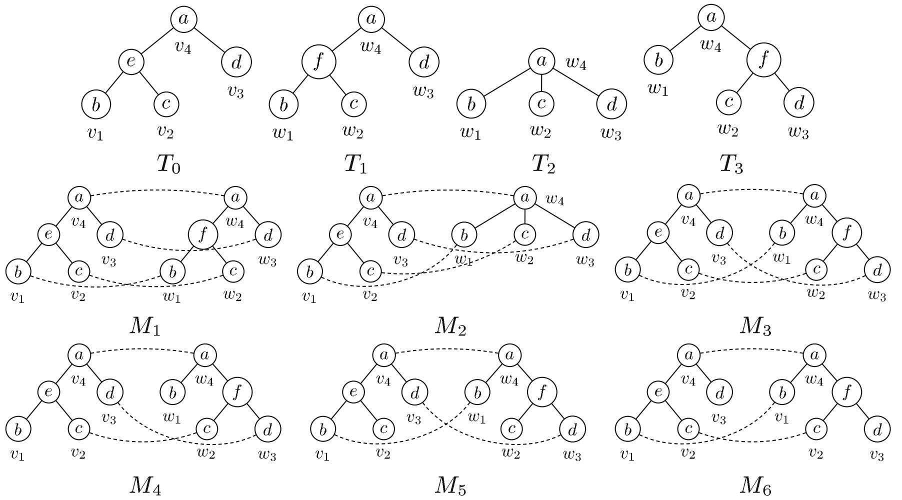

Fig. 1. Trees T_0, T_1, T_2 and T_3 (upper), mappings M_1, M_2 and M_3 (center) and mappings M_4, M_5 and M_6 (lower) in Example 1.

$$\forall (v_1, w_1), (v_2, w_2), (v_3, w_3) \in M \Big(v_1 \sqcup v_2 < v_1 \sqcup v_3 \Longrightarrow w_2 \sqcup w_3 = w_1 \sqcup w_3 \Big).$$

Or equivalently [3]:

$$\forall (v_1, w_1), (v_2, w_2), (v_3, w_3) \in M \Big(w_1 \sqcup w_2 < w_1 \sqcup w_3 \Longrightarrow v_2 \sqcup v_3 = v_1 \sqcup v_3 \Big).$$

Example 1. Consider trees T_0, T_1, T_2 and T_3 in Fig. 1 (upper). Also suppose that M_i is a mapping $\{(v_1, w_1), (v_2, w_2), (v_3, w_3), (v_4, w_4)\}$ between T_0 and T_i ($i = 1, 2, 3$) in Fig. 1 (center). Then, M_1 and M_2 are less-constrained mapping, while M_3 is not, because $v_1 \sqcup v_2 < v_1 \sqcup v_3$ but $w_2 \sqcup w_3 < w_1 \sqcup w_3$.

Furthermore, let $M_4 = M_3 - \{(v_1, w_1)\}$, $M_5 = M_3 - \{(v_2, w_2)\}$ and $M_6 = M_3 - \{(v_3, w_3)\}$ in Fig. 1 (lower). Then, we can show that M_4, M_5 and M_6 are less-constrained.

Definition 3 (Cover Set and Cover Sequence). Let T be a tree with the root r, v a node in T and U a set of nodes in T. Also suppose that $UP_r(v)$ is $v = v_1, \ldots, v_n = r$.

Then, we call a set $\{w \in T[v] \mid w \in U\}$ (or equivalently, $T[v] \cap U$) the *cover set* of v in T w.r.t. U and denote it by $C_T(v, U)$. Also we call a sequence $C_1, \ldots, C_n$ such that $C_i = C_T(v_i, U)$ for every i ($1 \leq i \leq n$) the *cover sequence* of v in T w.r.t. U and denote it by $S_T(v, U)$.

In particular, we use the cover sequences concerned with a mapping M between T_1 and T_2, that is, $S_{T_1}(v, M|_1)$ and $S_{T_2}(w, M|_2)$ for $(v, w) \in M$. For $r_1 = r(T_1)$ and $r_2 = r(T_2)$, we call $UP_{r_1}(v)$ and $UP_{r_2}(w)$ *paths* of $S_{T_1}(v, M|_1)$ and $S_{T_2}(w, M|_2)$, respectively, and denote them by $P_{T_1}(v)$ and $P_{T_2}(w)$, respectively.

Example 2. Consider mapping M_1, M_2 and M_3 in Example 1. For mapping M_i $(i = 1, 2, 3)$, we identify $v_j \in M_i|_1$ with $w_j \in M_i|_2$ $(j = 1, 2, 3, 4)$ and both of them are denoted by the index j. Then, the cover sequences $S_{T_0}(j, M_i|_1)$ and $S_{T_i}(j, M_i|_2)$ are described as follows.

$$
\begin{array}{ll}
S_{T_0}(1, M_i|_1) = \{1\}, \{1, 2\}, \{1, 2, 3, 4\}. & S_{T_2}(1, M_2|_2) = \{1\}, \{1, 2, 3, 4\}. \\
S_{T_0}(2, M_i|_1) = \{2\}, \{1, 2\}, \{1, 2, 3, 4\}. & S_{T_2}(2, M_2|_2) = \{2\}, \{1, 2, 3, 4\}. \\
S_{T_0}(3, M_i|_1) = \{3\}, \{1, 2, 3, 4\}. & S_{T_2}(3, M_2|_2) = \{3\}, \{1, 2, 3, 4\}. \\
S_{T_0}(4, M_i|_1) = \{1, 2, 3, 4\}. & S_{T_2}(4, M_2|_2) = \{1, 2, 3, 4\}. \\
S_{T_1}(1, M_1|_2) = \{1\}, \{1, 2\}, \{1, 2, 3, 4\}. & S_{T_3}(1, M_3|_2) = \{1\}, \{1, 2, 3, 4\}. \\
S_{T_1}(2, M_1|_2) = \{2\}, \{1, 2\}, \{1, 2, 3, 4\}. & S_{T_3}(2, M_3|_2) = \{2\}, \{2, 3\}, \{1, 2, 3, 4\}. \\
S_{T_1}(3, M_1|_2) = \{3\}, \{1, 2, 3, 4\}. & S_{T_3}(3, M_3|_2) = \{3\}, \{2, 3\}, \{1, 2, 3, 4\}. \\
S_{T_1}(4, M_1|_2) = \{1, 2, 3, 4\}. & S_{T_3}(4, M_3|_2) = \{1, 2, 3, 4\}.
\end{array}
$$

Since $\{1, 2\}$ and $\{2, 3\}$ are incomparable, so are $S_{T_0}(2, M_3|_1)$ and $S_{T_3}(2, M_3|_2)$. On the other hand, $S_{T_0}(j, M_i|_1)$ and $S_{T_i}(j, M_i|_2)$ are comparable for $(i, j) \in \{1, 2, 3\} \times \{1, 2, 3, 4\} - \{(3, 2)\}$.

Furthermore, consider mappings M_4, M_5 and M_6 in Example 1. Then, the cover sequences $S_{T_0}(j, M_i|_1)$ and $S_{T_2}(j, M_i|_2)$ are described as follows, where $j \in I_i$ and $I_4 = \{2, 3, 4\}$, $I_5 = \{1, 3, 4\}$ and $I_6 = \{1, 2, 4\}$. All of them are comparable.

$$
\begin{array}{ll}
S_{T_0}(2, M_4|_1) = \{2\}, \{2\}, \{2, 3, 4\}. & S_{T_3}(2, M_4|_2) = \{2\}, \{2, 3\}, \{2, 3, 4\}. \\
S_{T_0}(3, M_4|_1) = \{3\}, \{2, 3, 4\}. & S_{T_3}(3, M_4|_2) = \{3\}, \{2, 3\}, \{2, 3, 4\}. \\
S_{T_0}(4, M_4|_1) = \{2, 3, 4\}. & S_{T_3}(4, M_4|_2) = \{2, 3, 4\}. \\
S_{T_0}(1, M_5|_1) = \{1\}, \{1\}, \{1, 3, 4\}. & S_{T_3}(1, M_5|_2) = \{1\}, \{1, 3, 4\}. \\
S_{T_0}(3, M_5|_1) = \{3\}, \{1, 3, 4\}. & S_{T_3}(3, M_5|_2) = \{3\}, \{3\}, \{1, 3, 4\}. \\
S_{T_0}(4, M_5|_1) = \{1, 3, 4\}. & S_{T_3}(4, M_5|_2) = \{1, 3, 4\}. \\
S_{T_0}(1, M_6|_1) = \{1\}, \{1, 2\}, \{1, 2, 4\}. & S_{T_3}(1, M_6|_2) = \{1\}, \{1, 2, 4\}. \\
S_{T_0}(2, M_6|_1) = \{2\}, \{1, 2\}, \{1, 2, 4\}. & S_{T_3}(2, M_6|_2) = \{2\}, \{2\}, \{1, 2, 4\}. \\
S_{T_0}(4, M_6|_1) = \{1, 2, 4\}. & S_{T_3}(4, M_6|_2) = \{1, 2, 4\}.
\end{array}
$$

Theorem 1. *Let T_1 and T_2 be trees. Also let M be a mapping between T_1 and T_2. Then, M is not a less-constrained mapping between T_1 and T_2 if and only if there exists a pair $(v, w) \in M$ such that $S_{T_1}(v, M|_1)$ and $S_{T_2}(w, M|_2)$ are incomparable.*

Proof. Suppose that there exists a pair $(v_1, w_1) \in M$ such that cover sets $C_1 \in S_{T_1}(v_1, M|_1)$ and $C_2 \in S_{T_2}(w_1, M|_2)$ are incomparable. Then, there exist $v_2 \in M|_1$ and $w_3 \in M|_2$ such that $v_2 \in C_1 - C_2$ and $w_3 \in C_2 - C_1$. Let v^* and w^* be nodes $v_1 \sqcup v_2$ and $w_1 \sqcup w_3$, respectively. Then, we can assume that $C_1 = C_{T_1}(v^*, M|_1)$ and $C_2 = C_{T_2}(w^*, M|_2)$. Also consider v_3 and w_2.

Since $v_3 \notin C_1$, it holds that $v^* < v_1 \sqcup v_3$. Since $w_2 \notin C_2$, it holds that $w^* < w_2 \sqcup w_3$. Hence, even if $v^* = v_1 \sqcup v_2 < v_2 \sqcup v_3$, it holds that $w^* = w_1 \sqcup w_3 < w_2 \sqcup w_3$, which implies that M is not a less-constrained mapping.

Conversely, suppose that M is not a less-constrained mapping. Then, there exist $v_1, v_2, v_3 \in M|_1$ and $w_1, w_2, w_3 \in M|_2$ such that (1) $v_1 \sqcup v_2 < v_1 \sqcup v_3$ holds and either (2) $w_2 \sqcup w_3 < w_1 \sqcup w_3$ or (3) $w_2 \sqcup w_3 > w_1 \sqcup w_3$ holds.

By the condition (1), the cover sequences $S_{T_1}(v_1, M|_1)$ and $S_{T_1}(v_2, M|_1)$ contain a cover set C_1 such that $\{v_1, v_2\} \subseteq C_1$ and $v_3 \notin C_1$. On the other hand, by the condition (2), the cover sequence $S_{T_2}(w_2, M|_2)$ contains a cover set C_2 such that $\{w_2, w_3\} \subseteq C_2$ and $w_1 \notin C_2$, which implies that C_1 and C_2 are incomparable. Also, by the condition (3), the cover sequence $S_{T_2}(w_1, M|_2)$ contains a cover set C_3 such that $\{w_1, w_3\} \subseteq C_3$ and $w_2 \notin C_3$, which implies that C_1 and C_3 are incomparable. □

Corollary 1. *Let T_1 and T_2 be trees. Also let M be a mapping between T_1 and T_2. Then, M is a less-constrained mapping between T_1 and T_2 if and only if, for every pair $(v, w) \in M$, $S_{T_1}(v, M|_1)$ and $S_{T_2}(w, M|_2)$ are comparable.*

4 Alignable Mapping and Alignment Tree

Let T_1 and T_2 be trees. We say that I is a *root-preserving mapping* from T_1 to T_2 if I is a mapping between T_1 and T_2 and $r(T_1) \in I|_1$ always holds. In particular, for $v \in T_1$, we denote the node $w \in T_2$ such that $(v, w) \in I$ by $I(v)$. Note that T_2 is not necessary to be labeled.

Definition 4 (Alignable Mapping [3]). Let T_1 and T_2 be trees. We say that M is an *alignable* mapping between T_1 and T_2 if there exist a tree $\mathcal{T}$ (not necessary to be labeled) and root-preserving mappings I_1 from T_1 to $\mathcal{T}$ and I_2 from T_2 to $\mathcal{T}$ satisfying that $I_1(v) = I_2(w)$ for every $(v, w) \in M$. In particular, we call the tree $\mathcal{T}$ an *aligned tree* between T_1 and T_2 and the root-preserving mappings I_1 and I_2 *side mappings* of M from T_1 and T_2, respectively.

Let M be an alignable mapping between T_1 and T_2 and I_i a side mapping of M from T_i ($i = 1, 2$). Then, it holds that $M|_1 = \{v \in T_1 \mid I_1(v) = I_2(w)\}$ and $M|_2 = \{w \in T_2 \mid I_1(v) = I_2(w)\}$. For an aligned tree $\mathcal{T}$, we denote the inverse image of I_i from $V(\mathcal{T})$ to $V(T_i)$ by I_i^{-1}. In particular, when no $v \in T_i$ such that $I_i(v) = u$ exists for a node $u \in \mathcal{T}$, we denote $I_i^{-1}(u)$ by $\emptyset$ and $l(I_i^{-1}(u))$ by ε.

Definition 5 (Alignment Tree [2]). Let M be an alignable mapping between T_1 and T_2, I_i a side mapping of M from T_i ($i = 1, 2$) and $\mathcal{T}$ an aligned tree between T_1 and T_2. Then, we call the tree obtained by replacing every label of $u \in \mathcal{T}$ with $(l(I_1^{-1}(u)), l(I_2^{-1}(u)))$ an *alignment tree* between T_1 and T_2. For an alignment tree $\mathcal{T}$, we denote a mapping M between T_1 and T_2 constructed from $\mathcal{T}$ such that $(v, w) \in M$ iff $(l(v), l(w)) \in \mathcal{T}$ by $M_{\mathcal{T}}$.

Example 3. Consider mappings M_i ($i = 4, 5, 6$) in Example 1 (Fig. 1). Then, every mapping M_i is an alignable mapping. Also, the tree $\mathcal{T}_i$ in Fig. 2 is an alignment tree between T_0 and T_3 in Example 1 corresponding to M_i.

Let $\varepsilon \notin \Sigma$ denote a special *blank* symbol and define $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$. Then, we define a *cost function* $\gamma : (\Sigma_\varepsilon \times \Sigma_\varepsilon - \{(\varepsilon, \varepsilon)\}) \mapsto \mathbf{R}^+$ on pairs of labels. We constrain γ to be a *metric*, that is, $\gamma(l_1, l_2) \geq 0$, $\gamma(l_1, l_1) = 0$, $\gamma(l_1, l_2) = \gamma(l_2, l_1)$ and $\gamma(l_1, l_3) \leq \gamma(l_1, l_2) + \gamma(l_2, l_3)$. In particular, the *unit cost function* μ such

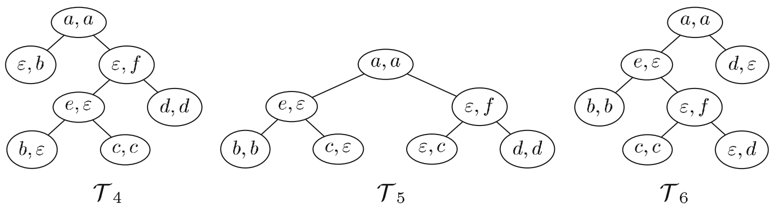

Fig. 2. The alignment trees $\mathcal{T}_i$ between T_0 and T_3 in Example 1.

that $\mu(a,b)=0$ if $a=b$ and $\mu(a,b)=1$ if $a\neq b$ is the most famous cost function. The *cost* of an alignment tree $\mathcal{T}$ under γ, denoted by $\gamma(\mathcal{T})$, is the sum of the costs of all labels in $\mathcal{T}$. The minimum cost of all the possible alignment trees is known to an alignment distance [2].

5 An Alternative Proof of Theorem 2

In this section, by using the cover sequence, Theorem 1 and Corollary 1, we give an alternative proof of the following Theorem 2.

Theorem 2 (Kuboyama [3]). *Let T_1 and T_2 be trees and M a mapping between T_1 and T_2. Then, M is less-constrained if and only if M is alignable.*

First, we show the if-direction of Theorem 2.

Lemma 1. *Let T_1 and T_2 be trees and M an alignable mapping between T_1 and T_2. Then, M is also a less-constrained mapping.*

Proof. For an alignable mapping M, there exists an alignment tree $\mathcal{T}$ such that $M=M_{\mathcal{T}}$. Also suppose that M is not a less-constrained mapping. By Theorem 1, there exists a pair $(v,w)\in M$ such that cover sets $C_1\in S_{T_1}(v,M|_1)$ and $C_2\in S_{T_2}(w,M|_2)$ are incomparable. Then, there exist $v_1\in M|_1$ and $w_2\in M|_2$ such that $v_1\in C_1-C_2$ and $w_2\in C_2-C_1$. Let v' and w' denote $v\sqcup v_1$ and $w\sqcup w_2$. Then, we can assume that $C_1=C_{T_1}(v',M|_1)$ and $C_2=C_{T_2}(w',M|_2)$. Also consider w_1 and v_2 such that $(v_1,w_1)\in M$ and $(v_2,w_2)\in M$.

Since $M=M_{\mathcal{T}}$, both $(l(v_1),l(w_1))$ and $(l(v_2),l(w_2))$ occur in $\mathcal{T}$. Since $v_1\in C_1$, it holds that $(l(v),l(w))<(l(v_1),l(w_1))$ in $\mathcal{T}$. Since $w_2\in C_2$, it holds that $(l(v),l(w))<(l(v_2),l(w_2))$ in $\mathcal{T}$. Furthermore, since $v_1\in C_1-C_2$ and $w_2\in C_2-C_1$, we can show that $(l(v_1),l(w_1))\ \#\ (l(v_2),l(w_2))$ in $\mathcal{T}$ as follows. Note here that we identify v_i with w_i $(i=1,2)$.

If $(l(v_1),l(w_1))<(l(v_2),l(w_2))$ in $\mathcal{T}$, then it holds that $v<v_1<v_2$ in T_1 and $w<w_1<w_2$ in T_2. Then, since $v'=v_1$ and $w'=w_2$, it holds that $C_1=C_{T_1}(v',M|_1)\subseteq C_{T_1}(v_2,M|_1)=C_{T_2}(w',M|_2)=C_2$. If $(l(v_2),l(w_2))<(l(v_1),l(w_1))$ in $\mathcal{T}$, then it holds that $v<v_2<v_1$ in T_1 and $w<w_2<w_1$ in T_2. Then, since $v'=v_1$ and $w'=w_2$, it holds that $C_2=C_{T_2}(w',M|_2)\subseteq C_{T_2}(w_1,M|_2)=C_{T_1}(v',M|_1)=C_1$. These imply a contradiction that C_1 and C_2 are incomparable.

Hence, it holds that $(l(v), l(w)) < (l(v_1), l(w_1))$, $(l(v), l(w)) < (l(v_2), l(w_2))$ and $(l(v_1), l(w_1)) \# (l(v_2), l(w_2))$ in $\mathcal{T}$ for $(l(v), l(w)), (l(v_1), l(w_1)), (l(v_2), l(w_2)) \in \mathcal{T}$, which is a contradiction that T is a tree. □

In order to show the only-if-direction of Theorem 2, we start the following lemma.

Lemma 2. *Let T_1 and T_2 be trees and M a less-constrained mapping between T_1 and T_2. Then, for $(v, w) \in M$, both $S_{T_1}(v, M|_1)$ and $S_{T_2}(w, M|_2)$ are comparable increasing such that the last element of $S_{T_1}(v, M|_1)$ (resp., $S_{T_2}(w, M|_2)$) is $M|_1$ (resp., $M|_2$).*

Let M be a mapping between T_1 and T_2. Then, for every $(v_j, w_j) \in M$, we sometimes identify $v_j \in M|_1$ with $w_j \in M|_2$ and both of them are denoted by the index j. Under such an identification, we can regard that $M|_1 = M|_2$. Then, we introduce the following *aligned sequence* and *aligned path* for comparable increasing sequences of sets.

Definition 6 (Aligned Sequence, Aligned Path). Let $S_1 = A_1, \ldots, A_n$ and $S_2 = B_1, \ldots, B_m$ be comparable increasing sequences of sets such that $A_n = B_m$. Then, we call the sequences $S'_1 = A'_1, \ldots, A'_k$ and $S'_2 = B'_1, \ldots, B'_k$ obtained from S_1 and S_2 by the procedure AlnSq in Algorithm 1 *aligned sequences* of S_1 and S_2. Furthermore, for the aligned sequences $S'_1 = A'_1, \ldots, A'_k$ and $S'_2 = B'_1, \ldots, B'_k$ of S_1 and S_2, we define the *aligned path of* S_1 and S_2 as a rooted labeled path $P = (V, E)$ such that $V = \{p_1, \ldots, p_k\}$, $E = \{(p_i, p_{i+1}) \mid 1 \leq i \leq k-1\}$, the root of P is p_1 and the label of p_i is (A'_{k-i+1}, B'_{k-i+1}) for $1 \leq i \leq k$. We sometimes denote such a path by $[p_1, \ldots, p_k]$.

Example 4. Consider a mapping M_2 in Example 1. By Lemma 2, $S_{T_0}(j, M_2|_1)$ and $S_{T_2}(j, M_2|_2)$ in Example 2 are comparable increasing. Then, we can obtain the aligned sequences $S'_{T_0}(j, M_2|_1)$ and $S'_{T_2}(j, M_2|_2)$ illustrated in Fig. 3 (upper). Also, for an aligned path $P_{M_2}(j) = [p_1, p_2, p_3]$ of $S_{T_0}(j, M_2|_1)$ and $S_{T_2}(j, M_2|_2)$,

procedure AlnSq(S_1, S_2)
/* $S_1 = A_1, \ldots, A_n$, $S_2 = B_1, \ldots, B_m$ */
$i \leftarrow 1$; $j \leftarrow 1$; $k \leftarrow 1$;
while $i \leq n+1$ **and** $j \leq m+1$ **do**
  if $i = n+1$ **then** $A'_k \leftarrow \lambda$; $B'_k \leftarrow B_j$; j++;
  else if $j = m+1$ **then** $A'_k \leftarrow A_i$; $B'_k \leftarrow \lambda$; i++;
  else if $A_i = B_j$ **then** $A'_k \leftarrow A_i$; $B'_k \leftarrow B_j$; i++; j++;
  else if $A_i \subset B_j$ **then** $A'_k \leftarrow A_i$; $B'_k \leftarrow \lambda$; i++;
  else if $A_i \supset B_j$ **then** $A'_k \leftarrow \lambda$; $B'_k \leftarrow B_j$; j++;
  k++;
return $S'_1 = A'_1, \ldots, A'_k$ and $S'_2 = B'_1, \ldots B'_k$;

Algorithm 1. AlnSq.

$S'_{T_0}(1, M_2|_1) = \{1\}, \{1, 2\}, \{1, 2, 3, 4\}$. $S'_{T_2}(1, M_2|_2) = \{1\}, \lambda, \{1, 2, 3, 4\}$.
$S'_{T_0}(2, M_2|_1) = \{2\}, \{1, 2\}, \{1, 2, 3, 4\}$. $S'_{T_2}(2, M_2|_2) = \{2\}, \lambda, \{1, 2, 3, 4\}$.
$S'_{T_0}(3, M_2|_1) = \{3\}, \{1, 2, 3, 4\}$. $S'_{T_2}(3, M_2|_2) = \{3\}, \{1, 2, 3, 4\}$.
$S'_{T_0}(4, M_2|_1) = \{1, 2, 3, 4\}$. $S'_{T_2}(4, M_2|_2) = \{1, 2, 3, 4\}$.

$P_{M_2}(j)$	$l(p_1)$	$l(p_2)$	$l(p_3)$
$P_{M_2}(1)$	$(\{1,2,3,4\}, \{1,2,3,4\})$	$(\{1,2\}, \lambda)$	$(\{1\}, \{1\})$
$P_{M_2}(2)$	$(\{1,2,3,4\}, \{1,2,3,4\})$	$(\{1,2\}, \lambda)$	$(\{2\}, \{2\})$
$P_{M_2}(3)$	$(\{1,2,3,4\}, \{1,2,3,4\})$	$(\{3\}, \{3\})$	
$P_{M_2}(4)$	$(\{1,2,3,4\}, \{1,2,3,4\})$		

Fig. 3. The aligned sequences $S'_{T_0}(j, M_2|_1)$ and $S'_{T_2}(j, M_2|_2)$ of $S_{T_0}(j, M_2|_1)$ and $S_{T_2}(j, M_2|_2)$ (upper) and the labels in aligned path $P_{M_2}(j)$ of $S_{T_0}(j, M_2|_1)$ and $S_{T_i}(j, M_2|_2)$ (lower) in Example 4.

every label $l(p_i)$ of a vertex p_i ($i = 1, 2, 3$) in $P_{M_2}(j)$ is illustrated in Fig. 3 (lower).

Consider mappings M_4, M_5 and M_6 in Example 1. By Lemma 2, $S_{T_0}(j, M_i|_1)$ and $S_{T_2}(j, M_i|_2)$ ($i = 4, 5, 6$, $j \in I_i$) in Example 2 are comparable increasing. Then, we can obtain the aligned sequences $S'_{T_0}(j, M_i|_1)$ and $S'_{T_2}(j, M_i|_2)$ illustrated in Fig. 4 (upper). Also, for an aligned path $P_{M_i}(j) = [p_1, p_2, p_3, p_4]$ of $S_{T_0}(j, M_i|_1)$ and $S_{T_2}(j, M_i|_2)$, every label $l(p_i)$ of a vertex p_i in $P_{M_i}(j)$ are illustrated in Fig. 4 (lower).

Let M be a less-constrained mapping between T_1 and T_2, where $r_1 = r(T_1)$ and $r_2 = r(T_2)$. By Lemma 2, for $S_{T_1}(v, M|_1) = A_1, \ldots, A_n$ and $S_{T_2}(w, M|_2) = B_1, \ldots, B_m$ for every $(v, w) \in M$, there exist paths $P_{T_1}(v) = v_1, \ldots, v_n$ and $P_{T_2}(w) = w_1, \ldots, w_m$ such that $v_1 = v$, $w_1 = w$, $v_n = r_1$ and $w_m = r_2$. Also, by identifying $v' \in M|_1$ with $w' \in M|_2$ for $(v', w') \in M$, it holds that $A_1 = B_1 = \{v\} = \{w\}$ and $A_n = B_m = M|_1 = M|_2$.

Furthermore, suppose that the aligned sequences $S'_{T_1}(v, M|_1)$ of $S_{T_1}(v, M|_1)$ and $S'_{T_2}(w, M|_2)$ of $S_{T_2}(w, M|_2)$ are of the forms $A'_1, \ldots, A'_k$ and $B'_1, \ldots, B'_k$, respectively. Then, we denote the corresponding path of $S'_{T_1}(v, M|_1)$ in T_1 including λ by $P'_{T_1}(v) = v'_1, \ldots, v'_k$ such that $v'_i = \lambda$ if $A'_i = \lambda$ and $v'_i = v_{i'}$ otherwise, where $i' = |\{l \mid 1 \leq l \leq i, v'_l \neq \lambda\}|$. Also we denote the corresponding path of $S_{T_2}(v, M|_1)$ in T_2 including λ by $P'_{T_2}(w) = w'_1, \ldots, w'_k$ such that $w'_j = \lambda$ if $B'_j = \lambda$ and $w'_j = w_{j'}$ otherwise, where $j' = |\{l \mid 1 \leq l \leq j, w'_l \neq \lambda\}|$.

Lemma 3. *Let M be a less-constrained mapping between T_1 and T_2. Also, for $(v_1, w_1), (v_2, w_2) \in M$, let:*
$S'_{T_1}(v_1, M|_1) = A'_1, \ldots, A'_k$, $S'_{T_2}(w_1, M|_2) = B'_1, \ldots, B'_k$,
$S'_{T_1}(v_2, M|_1) = C'_1, \ldots, C'_h$, $S'_{T_2}(w_2, M|_2) = D'_1, \ldots, D'_h$.

Then, for the maximum indices i and j $(2 \leq i \leq k, 2 \leq j \leq h)$ such that $(A'_i, B'_i) = (C'_j, B'_j)$ and $(A'_{i-1}, B'_{i-1}) \neq (C'_{j-1}, B'_{j-1})$, it holds that $(A'_a, B'_a) \neq (C'_b, B'_b)$ for every a $(1 \leq a \leq i-1)$ and b $(1 \leq b \leq j-1)$.

$S'_{T_0}(2, M_4|_1) = \{2\}, \{2\}, \lambda, \{2, 3, 4\}.$
$S'_{T_0}(3, M_4|_1) = \{3\}, \lambda, \{2, 3, 4\}.$
$S'_{T_0}(4, M_4|_1) = \{2, 3, 4\}.$
$S'_{T_0}(1, M_5|_1) = \{1\}, \{1\}, \{1, 3, 4\}.$
$S'_{T_0}(3, M_5|_1) = \{3\}, \lambda, \{1, 3, 4\}.$
$S'_{T_0}(4, M_5|_1) = \{1, 3, 4\}.$
$S'_{T_0}(1, M_6|_1) = \{1\}, \{1, 2\}, \{1, 2, 4\}.$
$S'_{T_0}(2, M_6|_1) = \{2\}, \lambda, \{1, 2\}, \{1, 2, 4\}.$
$S'_{T_0}(4, M_6|_1) = \{1, 2, 4\}.$

$S'_{T_3}(2, M_4|_2) = \{2\}, \lambda, \{2, 3\}, \{2, 3, 4\}.$
$S'_{T_3}(3, M_4|_2) = \{3\}, \{2, 3\}, \{2, 3, 4\}.$
$S'_{T_3}(4, M_4|_2) = \{2, 3, 4\}.$
$S'_{T_3}(1, M_5|_2) = \{1\}, \lambda, \{1, 3, 4\}.$
$S'_{T_3}(3, M_5|_2) = \{3\}, \{3\}, \{1, 3, 4\}.$
$S'_{T_3}(4, M_5|_2) = \{1, 3, 4\}.$
$S'_{T_3}(1, M_6|_2) = \{1\}, \lambda, \{1, 2, 4\}.$
$S'_{T_3}(2, M_6|_2) = \{2\}, \{2\}, \lambda, \{1, 2, 4\}.$
$S'_{T_3}(4, M_6|_2) = \{1, 2, 4\}.$

$P_{M_i}(j)$	$l(p_1)$	$l(p_2)$	$l(p_3)$	$l(p_4)$
$P_{M_4}(2)$	$(\{2,3,4\},\{2,3,4\})$	$(\lambda,\{2,3\})$	$(\{2\},\lambda)$	$(\{2\},\{2\})$
$P_{M_4}(3)$	$(\{2,3,4\},\{2,3,4\})$	$(\lambda,\{2,3\})$	$(\{3\},\{3\})$	
$P_{M_4}(4)$	$(\{2,3,4\},\{2,3,4\})$			
$P_{M_5}(1)$	$(\{1,3,4\},\{1,3,4\})$	$(\{1\},\lambda)$	$(\{1\},\{1\})$	
$P_{M_5}(3)$	$(\{1,3,4\},\{1,3,4\})$	$(\lambda,\{3\})$	$(\{3\},\{3\})$	
$P_{M_5}(4)$	$(\{1,3,4\},\{1,3,4\})$			
$P_{M_6}(1)$	$(\{1,2,4\},\{1,2,4\})$	$(\{1,2\},\lambda)$	$(\{1\},\{1\})$	
$P_{M_6}(2)$	$(\{1,2,4\},\{1,2,4\})$	$(\{1,2\},\lambda)$	$(\{1,2\},\{2\})$	$(\{2\},\{2\})$
$P_{M_6}(4)$	$(\{1,2,4\},\{1,2,4\})$			

Fig. 4. The aligned sequences $S'_{T_0}(j, M_i|_1)$ and $S'_{T_2}(j, M_i|_2)$ of $S_{T_0}(j, M_i|_1)$ and $S_{T_2}(j, M_i|_2)$ (upper) and the labels in aligned path $P_{M_i}(j)$ of $S_{T_0}(j, M_i|_1)$ and $S_{T_2}(j, M_i|_2)$ (lower) in Example 4.

Proof. Let $A = A'_1 \cup \cdots \cup A'_{i-1} - \{\lambda\}$, $B = B'_1 \cup \cdots \cup B'_{i-1} - \{\lambda\}$, $C = C'_1 \cup \cdots \cup C'_{j-1} - \{\lambda\}$ and $D = D'_1 \cup \cdots \cup D'_{j-1} - \{\lambda\}$. Then, we show that $A \cap C = \emptyset$ and $B \cap D = \emptyset$.

Suppose that $A \cap C \neq \emptyset$. Then, there exist a vertex $v \in T_1$ such that $v \in A \cap C$. Then, it holds that $v \in P'_{T_1}(v_1)$ and $v \in P'_{T_1}(v_2)$. Since T_1 is a rooted tree, $v_1 \sqcup v$ and $v_2 \sqcup v$ satisfy one of the statements of $v_1 \sqcup v < v_2 \sqcup v$, $v_2 \sqcup v < v_1 \sqcup v$ and $v_1 \sqcup v = v_2 \sqcup v$. For $P'_{T_1}(v_1) = p'_1, \ldots, p'_k$ and $P'_{T_1}(v_2) = q'_1, \ldots, q'_h$ such that $p'_1 = v_1$, $q'_1 = v_2$, $p'_k = q'_h = r(T_1)$, let $v^* = p'_{i'+1} = q'_{j'+1} \in T_1$. Then, it holds that $v_1 \sqcup v_2 = v^*$.

If $v_1 \sqcup v < v_2 \sqcup v$ holds, then it holds that $v_1 \sqcup v < v^*$, which means that $v \notin C$. If $v_2 \sqcup v < v_1 \sqcup v$ holds, then it holds that $v_2 \sqcup v < v^*$, which means that $v \notin A$. If $v_1 \sqcup v = v_2 \sqcup v$, then it holds that $v^* \leq v_1 \sqcup v = v_2 \sqcup v$, which means that $v \notin A \cap C$. Hence, it holds that $A \cap C = \emptyset$.

By the same way, we can show that $B \cap D = \emptyset$. Hence, the statement holds. □

Definition 7 (Merged Graph). For a less-constrained mapping M between T_1 and T_2, let $\mathcal{P}_M$ be the set of all aligned paths concerned with M. Then, we define the *merged graph* $\mathcal{G}_M$ of M as a rooted graph obtained by identifying vertices with the same labels in $\mathcal{P}_M$, where the label of the root is $(M|_1, M|_2)$.

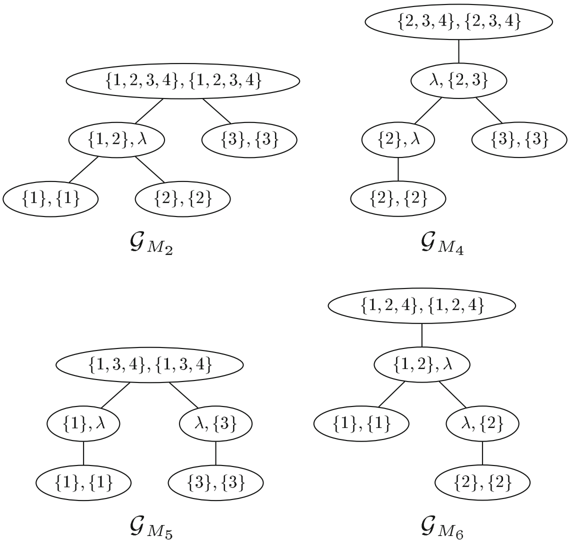

Fig. 5. The merged graphs $\mathcal{G}_{M_i}$ $(i = 2, 4, 5, 6)$ in Example 5.

Lemma 4. *Let T_1 and T_2 be trees and M a less-constrained mapping between T_1 and T_2. Then, the merged graph $\mathcal{G}_M$ of M is a rooted labeled tree.*

Example 5. Consider the mappings M_2, M_4, M_5 and M_6 in Example 1. By Example 4, it holds that $\mathcal{P}_{M_2} = \{P_{M_2}(1), P_{M_2}(2), P_{M_2}(3), P_{M_2}(4)\}$ and $\mathcal{P}_{M_i} = \{P_{M_i}(j) \mid j \in I_i\}$ $(i = 4, 5, 6)$. Then, the merged graphs $\mathcal{G}_{M_i}$ of $\mathcal{P}_{M_i}$ are illustrated in Fig. 5.

Let T_1 and T_2 be trees, M a less-constrained mapping between T_1 and T_2 and $\mathcal{G}_M$ the merged graph of M. Then, we denote the tree obtained by removing all of the labels in $\mathcal{G}_M$ by $\mathcal{G}_M^-$. Also, for a vertex $u \in \mathcal{G}_M$, the label of u is of the form (A, B), where $A \subseteq M_1$ or $A = \lambda$ and $B \subseteq M_2$ or $B = \lambda$. When $A \neq \lambda$ (*resp.*, $B \neq \lambda$), there exists a unique vertex in T_1 corresponding to A (*resp.*, in T_2 corresponding to B), which we denote such a vertex by $v_{T_1}(A)$ (*resp.*, $v_{T_2}(B)$).

For every vertex $u \in \mathcal{G}_M$, consider to replace the label (A, B) of u with $(l(v_{T_1}(A)), l(v_{T_2}(B)))$ if $A \neq \lambda$ and $B \neq \lambda$; $(\varepsilon, l(v_{T_2}(B)))$ if $A = \lambda$ and $B \neq \lambda$; $(l(v_{T_1}(A)), \varepsilon)$ if $A \neq \lambda$ and $B = \lambda$. We denote the tree obtained by this replacement of labels in every $u \in \mathcal{G}_M$ from $\mathcal{G}_M$ by $\mathcal{G}_M^*$.

Lemma 5. *Let T_1 and T_2 be trees, M a less-constrained mapping between T_1 and T_2 and $\mathcal{G}_M$ the merged graph of M. Then, $\mathcal{G}_M^-$ is a subtree of the aligned tree between T_1 and T_2, and $\mathcal{G}_M^*$ is a subtree of the alignment tree between T_1 and T_2.*

Proof. Note that the labels of vertices in $\mathcal{G}_M$ are of the form (A, B). Then, by the definition of $\mathcal{G}_M$ and Lemma 4, we obtain a subtree of T_1 (*resp.*, T_2) by first connecting $v_{T_1}(A)$ (*resp.*, $v_{T_2}(B)$) for every A (*resp.*, B) and then by deleting λ and, for a vertex v whose label is λ, connecting the children of v to the parent of v. Hence, the statement holds. $\square$

Let $\mathcal{P}_1$ be the set of all rooted maximal paths in $T_1 - \{P_{T_1}(v) \mid v \in M|_1\}$ and $\mathcal{P}_2$ the set of all rooted maximal paths in $T_2 - \{P_{T_2}(w) \mid w \in M|_2\}$. For every $P = [p_1, \dots, p_k] \in \mathcal{P}_1$ such that $r(P) = p_1$, there exists a vertex $v \in T_1$ such that v is a parent of p_1 in T_1, which we denote by $par_{T_1}(P)$. Similarly, for every $Q = [q_1, \dots, q_k] \in \mathcal{P}_2$ such that $r(Q) = q_1$, there exists a vertex $v \in T_2$ such that v is a parent of q_1 in T_2, which we denote by $par_{T_2}(Q)$.

Furthermore, for every $P = [p_1, \dots, p_k] \in \mathcal{P}_1$, we denote a labeled path obtained by replacing $l(p_i)$ with $(l(p_i), \varepsilon)$ by $\langle P, \varepsilon \rangle$, and, for every $Q = [q_1, \dots, q_k] \in \mathcal{P}_2$, we denote a labeled path obtained by replacing $l(q_i)$ with $(\varepsilon, l(q_i))$ by $\langle \varepsilon, Q \rangle$.

Lemma 6. *Let T_1 and T_2 be trees and M a less-constrained mapping between T_1 and T_2. Then, M is also an alignable mapping.*

Proof. It is sufficient to construct an alignment tree between T_1 and T_2 from M. By Lemma 5, $\mathcal{G}_M^*$ is a subtree of the alignment tree between T_1 and T_2. In order to complete the alignment tree, it is necessary to insert the paths not "covered by" M, which are denoted by the above $\mathcal{P}_1$ and $\mathcal{P}_2$. Hence, by inserting paths $\langle P, \varepsilon \rangle$ to the appropriate child of $par_{T_1}(P)$ in $\mathcal{G}_M^*$ for every $P \in \mathcal{P}_1$ and $\langle \varepsilon, Q \rangle$ to the appropriate child of $par_{T_1}(P)$ in $\mathcal{G}_M^*$ for every $Q \in \mathcal{P}_2$, we can obtain the alignment tree between T_1 and T_2. $\square$

It is not necessary for Theorem 2 to distinguish that trees are ordered or unordered.

6 Anchored Alignment Problem

Finally, we discuss the *anchored alignment problem* introduced in Sect. 1.

Theorem 3. *Let $n = |T_1|$, $m = |T_2|$, $h = \max\{h(T_1), h(T_2)\}$ and $\alpha = |M|$. Then, the problem of* AnchoredAlignment *can be solved in $O(h\alpha^2 + n + m)$ time and in $O(h\alpha)$ space for both ordered and unordered trees.*

Proof. Since the correctness is shown in Sect. 5, it is sufficient to show the time complexity. We use α-bits $\{0, 1\}$ vectors for set operations in totally $O(h\alpha)$ space, which we can prepare in $O(h\alpha)$ time.

For an anchoring M, first check whether or not M is less-constrained by using Corollary 1. If M is not less-constrained, then return "no," which runs in $O(h\alpha)$ time. Otherwise, construct a partial alignment tree $\mathcal{T}$ between T_1 and T_2 by using the algorithm AlnSq in Algorithm 1. We can check whether $A_i = B_j$, $A_i \subset B_j$ or $A_i \supset B_j$ in Algorithm 1 in $O(\alpha)$ time, so the running time of Algorithm 1 is $O(h\alpha)$ and the total running time in this process is $O(h\alpha^2)$. Next, construct the merged graph $\mathcal{G}_M$ and the replacement $\mathcal{G}_M^*$ of $\mathcal{G}_M$, which runs in $O(h\alpha)$ time for ordered trees (just checking adjacent nodes in postorder) and in $O(h\alpha^2)$ time for unordered trees. Finally, add $\langle P, \varepsilon \rangle$ and $\langle \varepsilon, Q \rangle$ to $\mathcal{T}$ according to Lemma 6, which runs in $O(n+m)$ time.

Hence, the time complexity is $O(h\alpha)+O(h\alpha^2)+O(h\alpha)+O(n+m) = O(h\alpha^2+n+m)$ for ordered trees and $O(h\alpha)+O(h\alpha^2)+O(h\alpha^2)+O(n+m) = O(h\alpha^2+n+m)$ for unordered trees. □

7 Conclusion

In this paper, first we have provided an alternative proof that a mapping is less-constrained iff it is alignable, by using cover sequences and merged graphs. Then, we have formulated the problem of AnchoredAlignment and then shown that we can solve it in $O(h\alpha^2+n+m)$ time and in $O(h\alpha)$ space for both ordered and unordered trees. Note that, if a given anchoring is optimum, that is, the cost of a given anchoring is minimum [2], then the problem of AnchoredAlignment is corresponding to the traceback of the alignment.

As stated in Sect. 1, an anchored alignment tree as output is not necessary to be optimum; it is just an alignment tree between two trees containing nodes labeled by pairs of labels in an anchoring. For example, consider the trees T_0 and T_3 in Fig. 1 and let M_7 in Fig. 6 (left) be an anchor between T_0 and T_3. Then, the anchored alignment tree of M_7 is $\mathcal{T}_7$ in Fig. 6 (right) such that $\mu(\mathcal{T}_7) = 8$ under a unit cost function μ. On the other hand, $\mathcal{T}_5$ in Fig. 2 is the optimum alignment tree between T_0 and T_3 such that $\mu(\mathcal{T}_5) = 4$.

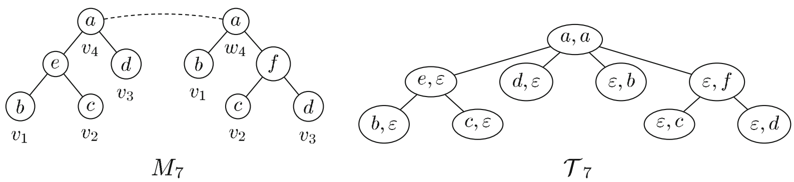

Fig. 6. A mapping M_7 and the anchored alignment tree of M_7.

Then, it is a future work to discuss the problem of AnchoredAlignment such that the anchored alignment tree is optimum. Also, it is a future work to investigate whether or not we can improve the time complexity to find the alignment distance for ordered trees, by using the algorithm to solve the problem of AnchoredAlignment. Furthermore, it is a future work to discuss the relationship between the results of this paper and the maximum agreement supertrees [1].

References

1. Berry, V., Nicolas, F.: Maximum agreement and comaptible supertrees. J. Discrete Algorothms **5**, 564–591 (2007)
2. Jiang, T., Wang, L., Zhang, K.: Alignment of trees - an alternative to tree edit. Theoret. Comput. Sci. **143**, 137–148 (1995)
3. Kuboyama, T.: Matching and learning in trees. Ph.D. thesis, University of Tokyo (2007)
4. Lu, C.L., Su, Z.-Y., Tang, C.Y.: A new measure of edit distance between labeled trees. In: Wang, J. (ed.) COCOON 2001. LNCS, vol. 2108, pp. 338–348. Springer, Heidelberg (2001)
5. Schiermer, S., Giegerich, R.: Forest alignment with affine gaps and anchors, applied in RNA structure comparision. Theoret. Comput. Sci. **483**, 51–67 (2013)
6. Tai, K.-C.: The tree-to-tree correction problem. J. ACM **26**, 422–433 (1979)