Reachability between Steiner Trees in a Graph

Haruka Mizuta[1,2,a)] Takehiro Ito[1,3,b)] Xiao Zhou[1,c)]

Abstract: In this paper, we study the reachability between Steiner trees in a graph: Given two Steiner trees of an unweighted graph, we wish to transform one into the other via Steiner trees by exchanging a single edge at a time. This decision problem is PSPACE-complete in general. In this paper, we give a linear-time algorithm to solve the problem when restricted to interval graphs.

1. Introduction

The Steiner tree problem on graphs is one of the most well-known NP-complete problems [3]. Let G be an unweighted graph with vertex set $V(G)$ and edge set $E(G)$. For a vertex subset $S \subseteq V(G)$, a *Steiner tree* of G for S is a subtree of G which includes all vertices in S; each vertex in S is called a *terminal*. For example, Fig. 1 illustrates four Steiner trees of the same graph G for the same terminal set S. Given an unweighted graph G, a terminal set $S \subseteq V(G)$, and an integer $k \geq 0$, the Steiner tree problem is to determine whether there exists a Steiner tree T of G for S such that T consists of at most k edges.

In this paper, we study the following problem: Suppose that we are given two Steiner trees of a graph G for a terminal set $S \subseteq V(G)$ (e.g., the leftmost and rightmost ones in Fig. 1), and we are asked whether we can transform one into the other via Steiner trees for S such that each intermediate Steiner tree can be obtained from the previous one by exchanging a single edge, that is, two consecutive Steiner trees T and T' in the transformation satisfy both $|E(T) \setminus E(T')| = 1$ and $|E(T') \setminus E(T)| = 1$. We call this decision problem the Steiner tree reconfiguration problem. For the particular instance of Fig. 1, the answer is yes as illustrated in the figure.

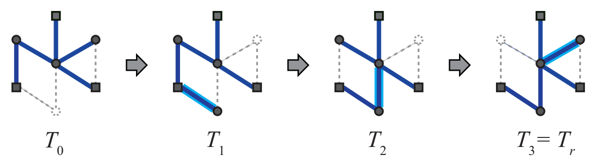

Fig. 1 A sequence $\langle T_0, T_1, \ldots, T_3 \rangle$ of Steiner trees of the same graph G for the same terminal set S, where the terminals are depicted by squares, non-terminals by circles, the edges in Steiner trees by thick lines.

1 Graduate School of Information Sciences, Tohoku University, Aobayama 6-6-05, Sendai, 980-8579, Japan.
2 JST, ERATO, Kawarabayashi Large Graph Project, c/o Global Research Center for Big Data Mathematics, NII, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.
3 CREST, JST, 4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan.
a) haruka.mizuta.s4@dc.tohoku.ac.jp
b) takehiro@ecei.tohoku.ac.jp
c) zhou@ecei.tohoku.ac.jp

Similar problems have been extensively studied under the reconfiguration framework [8], which arises when we wish to find a step-by-step transformation between two feasible solutions of a combinatorial (search) problem such that all intermediate solutions are also feasible. The reconfiguration framework has been applied to several well-studied problems, including satisfiability [4], [14], independent set [2], [7], [8], [12], vertex cover [8], [9], [15], clique [8], [10], dominating set [5], [6], [16] shortest path [1], [11], and so on. (See also a survey [17].)

Ito et al. [8] studied the spanning tree reconfiguration problem, which can be seen as Steiner tree reconfiguration when restricted to the case where all vertices in a given graph are terminals. They showed that any instance of spanning tree reconfiguration is a yes-instance, that is, there always exists a desired transformation between two spanning trees in any graph.

In this paper, we study the complexity status of Steiner tree reconfiguration. We can show that this problem is PSPACE-complete in general. Thus, in this paper, we prove that the problem is solvable in linear time for interval graphs. To do so, we first give a sufficient condition and a necessary condition for the existence of a desired transformation between two Steiner trees; we emphasize that these conditions hold for any graph. We then show that our necessary condition is indeed a necessary and sufficient condition for interval graphs.

2. Preliminaries

In this section, we first define some basic terms and notation. Then, we introduce a sufficient condition and a necessary condition for the existence of a reconfiguration sequence between two Steiner trees.

2.1 Definitions

In this paper, we assume without loss of generality that graphs are simple and connected. For a graph G, we denote by $V(G)$ and $E(G)$ the vertex set and edge set of G, respectively. For a vertex subset $V' \subseteq V(G)$, we denote by $G[V']$ the subgraph of G induced by V'. We simply write $G \setminus V' = G[V(G) \setminus V']$.

For a graph G and a terminal set $S \subseteq V(G)$, a subtree T of G is a *Steiner tree* for S if $S \subseteq V(T)$ holds. For convenience, although T is not a rooted tree, we call each degree-1 vertex of T a *leaf* of T. We say that a leaf v_f of T is *free* if it is a non-terminal, that is, $v_f \in V(T) \setminus S$. Thus, $T \setminus \{v_f\}$ is also a Steiner tree for S, and hence a Steiner tree having a free leaf is not minimal.

For a graph G and a terminal set S, we say that two Steiner trees T and T' for S are *adjacent* if both $|E(T) \setminus E(T')| = 1$ and $|E(T') \setminus E(T)| = 1$ hold; we write $T \leftrightarrow T'$ in this case. For two Steiner trees T_p and T_q for S, a sequence $\langle T_0, T_1, \ldots, T_\ell \rangle$ of Steiner trees for S is called a *reconfiguration sequence* between T_p and T_q if $T_0 = T_p$, $T_\ell = T_q$, and $T_{i-1} \leftrightarrow T_i$ holds for each $i \in \{1, 2, \ldots, \ell\}$. Note that any reconfiguration sequence is *reversible*, that is, $\langle T_\ell, T_{\ell-1}, \ldots, T_0 \rangle$ is a reconfiguration sequence between T_q and T_p. We write $T_p \leftrightsquigarrow T_q$ if there is a reconfiguration sequence between T_p and T_q. Then, the Steiner tree reconfiguration problem is defined as follows:

Input: An unweighted graph G, a terminal set $S \subseteq V(G)$, and two Steiner trees T_0 and T_r for S

Question: Determine whether $T_0 \leftrightsquigarrow T_r$ or not.

We denote by a 4-tuple (G, S, T_0, T_r) an instance of Steiner tree reconfiguration. Note that Steiner tree reconfiguration is a decision problem, and hence it does not ask for an actual reconfiguration sequence.

2.2 Sufficient condition and necessary condition

In this subsection, we give a sufficient condition and a necessary condition for the existence of a reconfiguration sequence between two Steiner trees. These conditions will play important roles in this paper to prove our results, and we emphasize that they hold for any graph.

We first give a sufficient condition, as follows.

Theorem 1. *Let (G, S, T_0, T_r) be an instance of* Steiner tree reconfiguration. *If $V(T_0) = V(T_r)$, then it is a* yes*-instance.*

Proof. Suppose that $V(T_0) = V(T_r)$ holds. Then, we have $G[V(T_0)] = G[V(T_r)]$. Therefore, both T_0 and T_r form spanning trees of $G[V(T_0)] = G[V(T_r)]$. It is known that any two spanning trees are reconfigurable each other by exchanging a single edge at a time [8], and hence the theorem follows. □

Theorem 1 says that any two Steiner trees are reconfigurable each other as long as they consist of the same vertex set. On the other hand, since we can exchange only a single edge at a time, two adjacent Steiner trees having different vertex sets satisfy a special property, as in the following proposition.

Proposition 1. *Suppose that $T \leftrightarrow T'$ holds for two Steiner trees T and T' of a graph G with a terminal set S. If $V(T) \neq V(T')$, then*

- *$V(T) \setminus V(T')$ contains exactly one vertex v_f, and v_f is a free leaf in T; and*
- *$V(T') \setminus V(T)$ contains exactly one vertex v'_f, and v'_f is a free leaf in T'.*

Proof. Suppose for a contradiction that $V(T) \setminus V(T')$ contains more than one vertex. (The proof is symmetric for the case where $V(T') \setminus V(T)$ contains more than one vertex.) Then, notice that $T \setminus T'$ contains at least one edge joining vertices in $V(T) \setminus V(T')$. Since $T \leftrightarrow T'$ and hence both $|V(T)| = |V(T')|$ and $|E(T) \setminus E(T')| = 1$ hold, the edge in $E(T') \setminus E(T)$ must join a vertex in $V(T) \setminus V(T')$ and a vertex in $V(T) \cap V(T')$. Therefore, the resulting Steiner tree T' consists of the same vertex set $V(T)$; this contradicts the assumption that $V(T) \neq V(T')$.

In this way, we have verified that $V(T) \setminus V(T')$ contains exactly one vertex v_f, and hence it is a leaf in T. Since both T and T' are Steiner trees for S, we know $V(T) \triangle V(T') = (V(T) \setminus V(T')) \cup (V(T') \setminus V(T)) \subseteq V(G) \setminus S$. Thus, v_f is free. □

We now give a sufficient condition for no-instance; by taking a contrapositive, this yields a necessary condition for yes-instance.

Theorem 2. *Let (G, S, T_0, T_r) be an instance of* Steiner tree reconfiguration. *Then, it is a* no*-instance if the following conditions* (*a*) *and* (*b*) *hold*:

(*a*) *$V(T_0) \neq V(T_r)$; and*

(*b*) *at least one of $G[V(T_0)]$ and $G[V(T_r)]$ has no Steiner tree for S with a free leaf.*

Proof. Suppose for a contradiction that (G, S, T_0, T_r) is a yes-instance even though it satisfies both Conditions (a) and (b). Then, there exists a reconfiguration sequence $\mathcal{T}$ between T_0 and T_r. Let T_{i+1} be the first Steiner tree in $\mathcal{T}$ such that $V(T_{i+1}) \neq V(T_0)$; such a Steiner tree exists since $V(T_0) \neq V(T_r)$. Then, the Steiner tree T_i in $\mathcal{T}$ satisfies $T_i \leftrightarrow T_{i+1}$ and $V(T_i) = V(T_0)$. By Proposition 1, $V(T_i) \setminus V(T_{i+1})$ contains exactly one vertex v_f which is a free leaf in T_i. Since $V(T_i) = V(T_0)$, we can conclude that $G[V(T_0)]$ has a Steiner tree T_i for S with a free leaf v_f. By the symmetric arguments, $G[V(T_r)]$ has a Steiner tree for S with a free leaf, too. This contradicts the assumption that Condition (b) holds. □

3. Algorithm for Interval Graphs

A graph G with $V(G) = \{v_1, v_2, \ldots, v_n\}$ is an *interval graph* if there exists a set $\mathcal{I}$ of (closed) intervals $I_1, I_2, \ldots, I_n$ such that $v_i v_j \in E(G)$ if and only if $I_i \cap I_j \neq \emptyset$ for each $i, j \in \{1, 2, \ldots, n\}$. We call the set $\mathcal{I}$ of intervals an *interval representation* of the graph. For a given graph G, it can be determined in linear time whether G is an interval graph, and if so obtain an interval representation of G [13].

In this section, we prove that Steiner tree reconfiguration is solvable in linear time for interval graphs. The key is the following theorem, whose proof will be given in the remainder of this section.

Theorem 3. *Let (G, S, T_0, T_r) be an instance of* Steiner tree reconfiguration *such that G is an interval graph. Then, it is a* yes*-instance if and only if the following conditions* (*a*) *or* (*b*) *hold*:

(*a*) *$V(T_0) = V(T_r)$; or*

(*b*) *each of $G[V(T_0)]$ and $G[V(T_r)]$ has a Steiner tree for S with a free leaf.*

Then, we have the following corollary.

Corollary 1. Steiner tree reconfiguration *can be solved in linear time for interval graphs.*

Proof. It suffices to show that Conditions (a) and (b) of Theorem 3 can be checked in linear time. We can clearly check Condition (a) in linear time. Thus, we show that Condition (b) can be checked in linear time, as follows.

Notice that, for a non-terminal vertex $v \in V(T_0) \setminus S$, if the induced graph $G[V(T_0) \setminus \{v\}]$ is connected, then any spanning tree T of $G[V(T_0) \setminus \{v\}]$ is a Steiner tree for S; by adding the non-terminal vertex v to T as a leaf, we can obtain a Steiner tree with a free leaf. The same holds for T_r, too.

We now check in linear time whether such a non-terminal vertex $v \in V(T_0) \setminus S$ exists or not. Since $G[V(T_0)]$ is an interval graph, we first obtain its interval representation in linear time [13]. Then, by traversing the interval representation from left to right, we can enumerate all cut-vertices in $G[V(T_0)]$ in linear time, and hence the existence of a desired non-terminal vertex $v \in V(T_0) \setminus S$ can be checked in linear time. (The same is applied to T_r, too.) $\square$

We give a proof of Theorem 3 in the remainder of this section. The only-if direction is immediate from Theorem 2 (by taking a contrapositive). In addition, when Condition (a) holds, the if direction is also immediate from Theorem 1. Therefore, it suffices to prove that (G, S, T_0, T_r) is a yes-instance if both $V(T_0) \neq V(T_r)$ and Condition (b) hold.

Let (G, S, T_0, T_r) be a given instance of Steiner tree reconfiguration such that G is an interval graph, $V(T_0) \neq V(T_r)$, and Condition (b) of Theorem 3 holds. Then, $G[V(T_0)]$ has a Steiner tree for S with a free leaf, and by Theorem 1 there exists a reconfiguration sequence between T_0 and the Steiner tree with a free leaf; the same holds for T_r. Therefore, we assume without loss of generality that two given Steiner trees T_0 and T_r have free leaves. We will construct a reconfiguration sequence between T_0 and T_r.

Let $\mathcal{I}$ be an interval representation of G. For an interval $I_i \in \mathcal{I}$, we denote by $l(I_i)$ and $r(I_i)$ the left and right coordinates of I_i, respectively; we sometimes call the values $l(I_i)$ and $r(I_i)$ the *l-value* and *r-value* of I_i, respectively. We may assume without loss of generality that all l-values and r-values are distinct. For notational convenience, we sometimes identify a vertex $v_i \in V(G)$ with its corresponding interval $I_i \in \mathcal{I}$, and simply write $l(v_i) = l(I_i)$ and $r(v_i) = r(I_i)$. We say that a path P in G is *r-increasing* if the r-values of the vertices along P are increasing. Let $s_{\rm left}$ be the terminal in S which has the minimum l-value, that is, $l(s_{\rm left}) = \min\{l(v) : v \in S\}$, while let $s_{\rm right}$ be the terminal in S which has the maximum r-value, that is, $r(s_{\rm right}) = \max\{r(v) : v \in S\}$. Note that $s_{\rm left} = s_{\rm right}$ may hold. Then, we say that a Steiner tree F for S is in *standard form* if

- the unique path P in F from $s_{\rm left}$ to $s_{\rm right}$ is r-increasing; and
- every terminal in $S \setminus V(P)$ is a leaf in F which is adjacent to some vertex in P.

(See Fig. 2(c) as an example.)

Lemma 1. *For any Steiner tree T of an interval graph G, there exists a Steiner tree F of G such that F is in standard form, all free leaves in T are free leaves in F, $V(F) = V(T)$, and $T \leftrightsquigarrow F$.*

Proof. Let V_F be the set of all free leaves in T, and let T' be the subtree of T obtained by deleting the vertices in V_F. (See Fig. 2(a) in which T' is illustrated by the thick dotted lines.) We first prove the existence of a Steiner tree F' in standard form for S such that $V(F') = V(T')$.

Consider the induced subgraph $G[V(T')]$ of G. (See Fig. 2(b).)

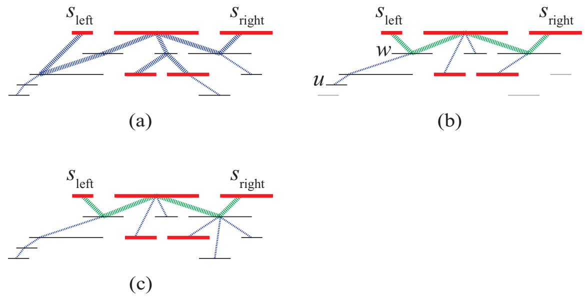

Fig. 2 (a) Steiner tree T of an interval graph G, (b) Steiner tree F' of $G[V(T')]$ in a standard form, and (c) Steiner tree F of G in a standard form. In the figure, graphs are illustrated by their interval representations; each terminal in S is depicted by thick (red) line, and each non-terminal by thin (black) line. Steiner trees are depicted by dotted lines on the interval representations. In (b) and (c), the thick (green) dotted lines represent the paths from $s_{\rm left}$ to $s_{\rm right}$.

Since T' is connected, $G[V(T')]$ is also connected. Therefore, we can greedily find an r-increasing path P in $G[V(T')]$ from $s_{\rm left}$ to $s_{\rm right}$. By the choice of $s_{\rm left}$ and $s_{\rm right}$, every terminal s in $S \setminus V(P)$ intersects with at least one vertex in P; we arbitrarily choose such a vertex in P, and connect s with it.

To finish the construction of F', we now claim that every vertex in $V(T') \setminus S$ is either on P or has a path to a vertex w in P which consists of only non-terminal vertices except for w. (See the vertex u in Fig. 2(b) as an example for the latter case.) Then, the terminals in $S \setminus V(P)$ remain leaves in F', as required in standard form. Suppose for a contradiction that a vertex u in $V(T') \setminus S$ does not have such a path. If both $l(u) < r(s_{\rm right})$ and $l(s_{\rm left}) < r(u)$ hold, then u intersects with some vertex in P. Thus, u must satisfy either $r(s_{\rm right}) < l(u)$ or $r(u) < l(s_{\rm left})$. Consider the case where $r(u) < l(s_{\rm left})$ holds; the other case is symmetric. Then, since $G[V(T')]$ is connected but u has no desired path to any vertex in P, there must exist a terminal $s \in S$ such that $l(s) < l(s_{\rm left})$; this contradicts the definition of $s_{\rm left}$.

In this way, there exists a Steiner tree F' in standard form such that $V(F') = V(T')$. Then, since $G[V(T)]$ is connected and every vertex u with $l(u) < r(s_{\rm right})$ and $l(s_{\rm left}) < r(u)$ intersects with a vertex in P, we can add the vertices in V_F to F' as leaves so that the terminals in $S \setminus V(P)$ remain leaves in F'; let F be the resulting tree. (See Fig. 2(b) and (c).)

Therefore, we have verified the existence of a Steiner tree F in standard form such that $V(F) = V(T)$ and all free leaves in T are free leaves also in F. Then, since $V(F) = V(T)$ holds, Theorem 1 yields that $T \leftrightsquigarrow F$. $\square$

Recall that a given instance (G, S, T_0, T_r) is assumed to satisfy Condition (b) of Theorem 3. Then, to verify that $T_0 \leftrightsquigarrow T_r$ holds, by Lemma 1 it suffices to construct a reconfiguration sequence between two Steiner trees T'_0 and T'_r such that $V(T'_0) = V(T_0)$, $V(T'_r) = V(T_r)$, both T'_0 and T'_r are in standard form and have free leaves. Thus, the following lemma completes the proof of Theorem 3.

Lemma 2. *Let T_A and T_B be any two Steiner trees for S which are in standard form and have free leaves. Then, $T_A \leftrightsquigarrow T_B$.*

Proof. Let $P_A = (a_1, a_2, \ldots, a_{\ell_A})$ and $P_B = (b_1, b_2, \ldots, b_{\ell_B})$ be the paths from $s_{\rm left}$ to $s_{\rm right}$ in T_A and T_B, respectively; and hence

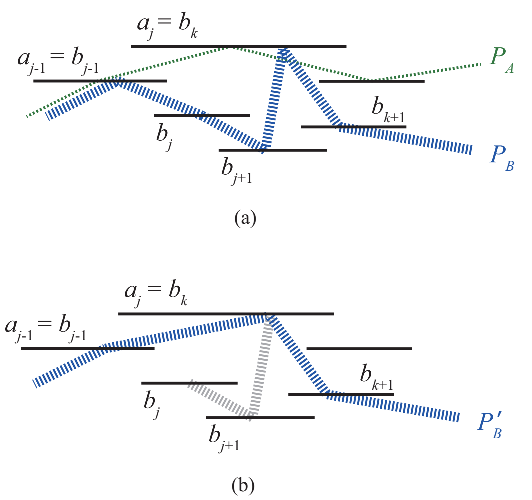

Fig. 3 Illustration for Case (i).

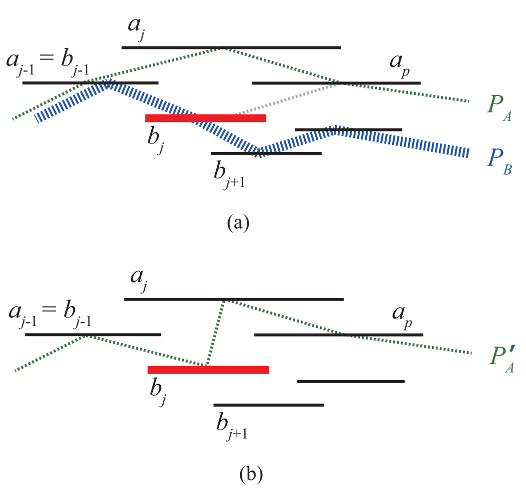

Fig. 4 Illustration for Case (ii).

$a_1 = b_1 = s_{\rm left}$ and $a_{\ell_A} = b_{\ell_B} = s_{\rm right}$. We prove the lemma by induction on the number of vertices in $V(P_A) \triangle V(P_B)$.

First, consider the case where $V(P_A) \triangle V(P_B) = \emptyset$. Since both T_A and T_B are in standard form, we know $P_A = P_B$ and all terminals in $S \setminus V(P_A)$ are leaves and adjacent to vertices in P_A. Thus, by greedily exchanging the edges in $E(T_A) \triangle E(T_B)$, we can obtain a reconfiguration sequence between T_A and T_B.

Second, consider the case where $V(P_A) \triangle V(P_B) \neq \emptyset$. Let j be the first index such that $a_j \neq b_j$. (See Fig. 3(a).) Since both P_A and P_B are r-increasing, a_j and b_j intersect with each other and hence $a_j b_j \in E(G)$. Assume without loss of generality that $r(b_j) < r(a_j)$ holds, as illustrated in Fig. 3(a). (The other case is symmetric.) Then, we have $a_j b_{j+1} \in E(G)$. We deal with this case according to the following three sub-cases.

Case (i): a_j appears in P_B. (See Fig. 3.)

Let k be the index such that $b_k = a_j$. Since $r(b_j) < r(a_j) = r(b_k)$ and P_B is r-increasing, we know that $k > j$ holds. Therefore, we simply exchange the edge $b_{j-1} b_j \in E(P_B)$ with the edge $b_{j-1} b_k \in E(G) \setminus E(T_B)$, and obtain a Steiner tree T'_B for S with the path $P'_B = (b_1, b_2, \ldots, b_{j-1}, b_k, b_{k+1}, \ldots, b_{\ell_B})$ from $b_1 = s_{\rm left}$ to $b_{\ell_B} = s_{\rm right}$. (See Fig. 3(a) and (b).) Then, P'_B is r-increasing. Since $b_{j-1} b_j \in E(P_B)$, neither b_{j-1} nor b_j is a free leaf in T_B. Thus, free leaves in T_B remain free leaves also in T'_B. If needed, we can transform T'_B into a Steiner tree T''_B in standard form with keeping the free leaves, as in the proof of Lemma 1. Then, since $|V(P_A) \triangle V(P'_B)| < |V(P_A) \triangle V(P_B)|$, we can apply the induction hypothesis to T_A and T''_B. Therefore, we have $T_B \leftrightarrow T'_B \leftrightsquigarrow T''_B \leftrightsquigarrow T_A$.

Case (ii): b_j is a terminal in S. (See Fig. 4.)

Since P_A is r-increasing and we have assumed without loss of generality that $r(b_j) < r(a_j)$ holds, b_j does not appear in P_A. Then, since T_A is in standard form, b_j must be a leaf in T_A which is adjacent to a vertex a_p in P_A for some index p. If $p \neq j$, then we first exchange the edge $a_p b_j \in E(T_A)$ with the edge $a_j b_j \in E(G) \setminus E(T_A)$. We then exchange the edge $a_{j-1} a_j \in E(P_A)$ with the edge $a_{j-1} b_j \in E(G) \setminus E(T_A)$, and obtain a Steiner tree T'_A for S with the path $P'_A = (a_1, a_2, \ldots, a_{j-1}, b_j, a_j, a_{j+1}, \ldots, a_{\ell_A})$ from $a_1 = s_{\rm left}$ to $a_{\ell_A} = s_{\rm right}$. (See Fig. 4(a) and (b).) Note that, since $r(a_{j-1}) = r(b_{j-1}) < r(b_j) < r(a_j)$ holds, P'_A is r-increasing. Since b_j is a terminal, it is not a free leaf. In addition, since $a_{j-1} a_j \in E(P_A)$, neither a_{j-1} nor a_j is a free leaf in T_A. Thus, free leaves in T_A remain free leaves also in T'_A. Then, by similar arguments as in Case (i), we thus have $T_A \leftrightsquigarrow T'_A \leftrightsquigarrow T_B$.

Case (iii): b_j is not a terminal in S. (See Fig. 5.)

If a_j appears in P_B, then we apply Case (i) above. We now consider the case where a_j does not appear in P_B. Let b_q be any vertex in P_B such that $l(b_q) < r(a_j) < r(b_q)$; by the definitions of $s_{\rm left}$ and $s_{\rm right}$, such a vertex b_q always exists. Recall that $a_j b_{j+1} \in E(G)$ holds, and hence we know that $q \geq j+1$. If $a_j \notin V(T_B)$, then we exchange an arbitrary chosen edge $e_f \in E(T_B)$ incident to a free leaf with the edge $b_q a_j \in E(G) \setminus E(T_B)$. Otherwise, we pick the first edge on the path in T_B from a_j to a vertex in P_B, and exchange it with the edge $b_q a_j \in E(G) \setminus E(T_B)$. We then exchange the edge $b_{j-1} b_j \in E(P_B)$ with the edge $b_{j-1} a_j$, and obtain a Steiner tree T'_B for S with the path $P'_B = (b_1, b_2, \ldots, b_{j-1}, a_j, b_q, b_{q+1}, \ldots, b_{\ell_B})$ from $b_1 = s_{\rm left}$ to $b_{\ell_B} = s_{\rm right}$. (See Fig. 5(a) and (b).) By the choice of b_q, P'_B is r-increasing. In addition, since $q \geq j+1$ and b_j is not a terminal, b_j is a free leaf in T'_B. By similar arguments as in Case (i), we thus have $T_B \leftrightsquigarrow T'_B \leftrightsquigarrow T_A$. □

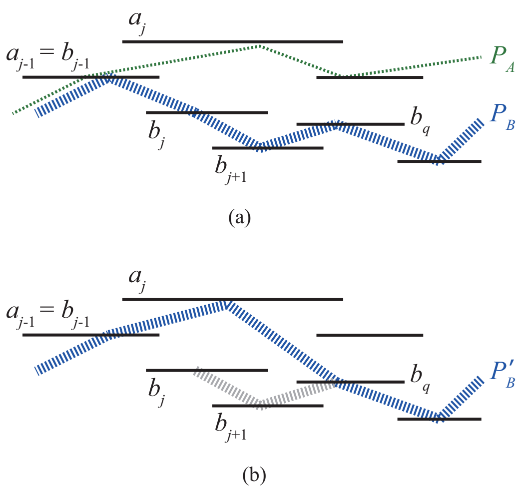

Fig. 5 Illustration for Case (iii).

4. Conclusion

In this paper, we have shown that the Steiner tree reconfiguration problem is solvable in linear time for interval graphs.

Acknowledgment We are grateful to Tatsuhiko Hatanaka for valuable discussions with him.

References

[1] Bonsma, P.: The complexity of rerouting shortest paths. Theoretical Computer Science 510, pp. 1–12 (2013)
[2] Demaine, E.D., Demaine, M.L., Fox-Epstein, E., Hoang, D.A., Ito, T., Ono, H., Otachi, Y., Uehara, R., Yamada, T.: Linear-time algorithm for sliding tokens on trees. Theoretical Computer Science 600, pp. 132–142 (2015)
[3] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
[4] Gopalan, P., Kolaitis, P.G., Maneva, E.N., Papadimitriou, C.H.: The connectivity of Boolean satisfiability: computational and structural dichotomies. SIAM J. Computing 38, pp. 2330–2355 (2009)
[5] Haas, R., Seyffarth, K.: The *k*-dominating graph. Graphs and Combinatorics 30, pp. 609–617 (2014)
[6] Haddadan, A., Ito, T., Mouawad, A.E., Nishimura, N., Ono, H., Suzuki, A., Tebbal, Y.: The complexity of dominating set reconfiguration. Proc. of WADS 2015, LNCS 9214, pp. 398–409 (2015)
[7] Hearn, R.A., Demaine, E.D.: PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. Theoretical Computer Science 343, pp. 72–96 (2005)
[8] Ito, T., Demaine, E.D., Harvey, N.J.A., Papadimitriou, C.H., Sideri, M., Uehara, R., Uno, Y.: On the complexity of reconfiguration problems. Theoretical Computer Science 412, pp. 1054–1065 (2011)
[9] Ito, T., Nooka, H., Zhou, X.: Reconfiguration of vertex covers in a graph. IEICE Transactions on Information and Systems E99-D, pp. 598–606 (2016)
[10] Ito, T., Ono, H., and Otachi, Y.: Reconfiguration of cliques in a graph. Proc. of TAMC 2015, LNCS 9076, pp. 212–223 (2015)
[11] Kamiński, M., Medvedev, P., Milanič, M.: Shortest paths between shortest paths. Theoretical Computer Science 412, pp. 5205–5210 (2011)
[12] Kamiński, M., Medvedev, P., Milanič, M.: Complexity of independent set reconfigurability problems. Theoretical Computer Science 439, pp. 9–15 (2012)
[13] Korte, N., Möhring, R.: An incremental linear-time algorithm for recognizing interval graphs. SIAM J. Computing 18, pp. 68–81 (1989)
[14] Mouawad, A.E., Nishimura, N., Pathak, V., Raman, V.: Shortest reconfiguration paths in the solution space of Boolean formulas. Proc. of ICALP 2015, LNCS 9134, pp. 985–996 (2015)
[15] Mouawad, A.E., Nishimura, N., Raman, V.: Vertex cover reconfiguration and beyond. Proc. of ISAAC 2014, LNCS 8889, pp. 452–463 (2014)
[16] Suzuki, A., Mouawad, A.E., Nishimura, N.: Reconfiguration of dominating sets. Proc. of COCOON 2014, LNCS 8591, pp. 405–416 (2014)
[17] van den Heuvel, J.: The complexity of change. Surveys in Combinatorics 2013, London Mathematical Society Lecture Notes Series 409 (2013).