Reconfiguration of Minimum Steiner Trees via Vertex Exchanges

Haruka Mizuta
Graduate School of Information Sciences, Tohoku University, Sendai, Japan
haruka.mizuta.s4@dc.tohoku.ac.jp

Tatsuhiko Hatanaka
Graduate School of Information Sciences, Tohoku University, Sendai, Japan
hatanaka@ecei.tohoku.ac.jp

Takehiro Ito
Graduate School of Information Sciences, Tohoku University, Sendai, Japan
takehiro@ecei.tohoku.ac.jp

Xiao Zhou
Graduate School of Information Sciences, Tohoku University, Sendai, Japan
zhou@ecei.tohoku.ac.jp

Abstract

In this paper, we study the problem of deciding if there is a transformation between two given minimum Steiner trees of an unweighted graph such that each transformation step respects a prescribed reconfiguration rule and results in another minimum Steiner tree of the graph. We consider two reconfiguration rules, both of which exchange a single vertex at a time, and generalize the known reconfiguration problem for shortest paths in an unweighted graph. This generalization implies that our problems under both reconfiguration rules are PSPACE-complete for bipartite graphs. We thus study the problems with respect to graph classes, and give some boundaries between the polynomial-time solvable and PSPACE-complete cases.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Combinatorial reconfiguration, Graph algorithms, Steiner tree

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.79

Funding *Haruka Mizuta*: Partially supported by JSPS KAKENHI Grant Number JP19J10042, Japan.
Takehiro Ito: Partially supported by JST CREST Grant Number JPMJCR1402, and JSPS KAKENHI Grant Numbers JP18H04091 and JP19K11814, Japan.
Xiao Zhou: Partially supported by JSPS KAKENHI Grant Number JP19K11813, Japan.

1 Introduction

Recently, *combinatorial reconfiguration* [7] has been extensively studied in the field of theoretical computer science. In combinatorial reconfiguration, we are given two feasible solutions of a combinatorial search problem, and are asked to determine whether we can transform one into the other by repeatedly applying a specified reconfiguration rule so that all intermediate results are also feasible. Such problems are called *reconfiguration problems* and have been studied intensively for several combinatorial search problems. (See, e.g., surveys [6, 11].) For example, the Shortest Path Reconfiguration problem (SPR, for short) is defined as follows [1, 2, 8, 13]: We are given two shortest paths between two vertices s and t in an unweighted graph, and are asked to determine whether or not we can transform one into the other by exchanging a single vertex in a shortest path at a time so that all intermediate results remain shortest paths between s and t. The problem is known to be

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).
Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 79; pp. 79:1–79:11
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

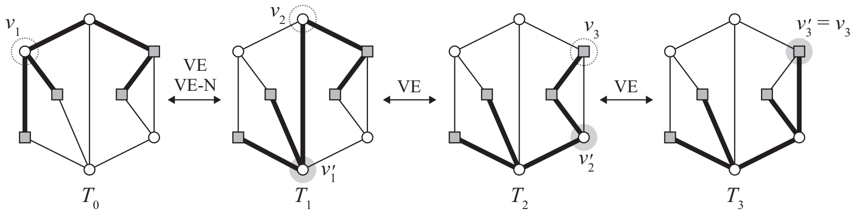


Figure 1 A sequence of minimum Steiner trees, where the terminals are depicted by gray squares, non-terminals by white circles, the edges in Steiner trees by thick lines.

PSPACE-complete [1, 13], and solvable in polynomial time for several graph classes [1, 2, 13]. Interestingly, the polynomial-time solvable cases include planar graphs [2], although many reconfiguration problems remain PSPACE-complete for planar graphs.

In this paper, we introduce and study reconfiguration problems for minimum Steiner trees in a graph, as generalizations of SPR.

1.1 Our problems

For an unweighted graph G and a vertex subset $S \subseteq V(G)$, called a *terminal set*, a *Steiner tree* of G for S is a subtree of G which contains all vertices in S. A Steiner tree of G for S is *minimum* if it has the minimum number of edges among all Steiner trees of G for S. For example, Figure 1 illustrates four minimum Steiner trees of the same graph G for the same terminal set S. Note that minimum Steiner trees can be seen as a generalization of shortest paths, because any shortest path in G between two vertices s and t forms a minimum Steiner tree of G for $S = \{s, t\}$. We use the terms *node* for Steiner trees and *vertex* for input graphs.

In this paper, we introduce following two reconfiguration rules, which define slightly different adjacency relations on minimum Steiner trees of a graph G for a terminal set S. Both rules exchange a single node v in a minimum Steiner tree T for a single vertex in G (possibly v itself) so that it results in another minimum Steiner tree T' of G for S. (Formal definitions will be given in Section 2.)

- **Vertex Exchange** (VE, for short):
  We say that two minimum Steiner trees T and T' of G for S are *adjacent under* VE if there exist two vertices $v \in V(T)$ and $v' \in V(T')$ such that their removal results in the common subgraph of T and T'. For example, any two consecutive minimum Steiner trees in Figure 1 are adjacent under VE. It should be noted that v and v' can be the same vertex; in such a case, only edges incident to $v = v'$ may be changed between T and T'. (See T_2 and T_3 in Figure 1 as an example.)
- **Vertex Exchange without changing Neighbors** (VE-N, for short):
  We say that two minimum Steiner trees T and T' of G for S are *adjacent under* VE-N if there exist two vertices $v \in V(T)$ and $v' \in V(T')$ such that (a) their removal results in the common subgraph of T and T', and (b) the neighborhood of v in T is equal to that of v' in T'. In Figure 1, only two Steiner trees T_0 and T_1 are adjacent under VE-N. It can hold also under this rule that $v = v'$, but then $T = T'$ holds.

We now define our problems. Given two minimum Steiner trees T_0 and T_r of a graph G for a terminal set S, *the* Minimum Steiner Tree Reconfiguration *problem* (MSTR, for short) *under* VE (resp., VE-N) asks to determine whether or not we can transform one into the other via adjacent minimum Steiner trees under VE (resp., VE-N). For example, when we

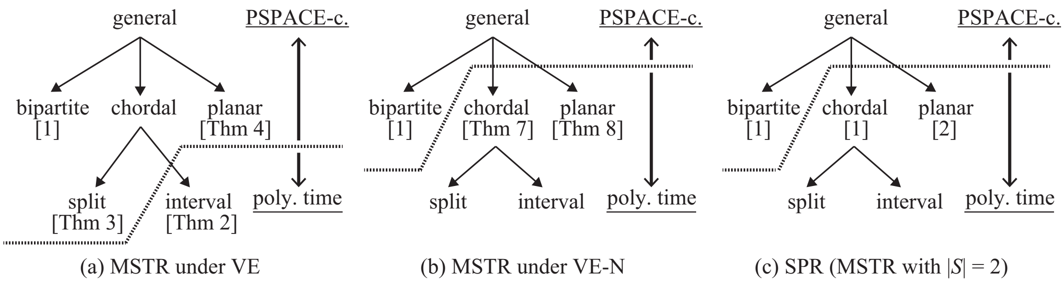


Figure 2 Our results for MSTR and some known results for SPR, where each arrow between graph classes represents their inclusion relationship: $A \to B$ represents that the graph class B is properly included in the graph class A.

are given two minimum Steiner trees T_0 and T_3 in Figure 1, the answer is yes under VE as illustrated in the figure; while it is a no-instance under VE-N. We note that if $|S| = 2$, then any minimum Steiner tree for S forms a shortest path between the two terminals, and both the rules VE and VE-N are equivalent to the reconfiguration rule of SPR. Therefore, MSTR under both rules VE and VE-N are generalizations of SPR.

1.2 Known and related work

There are several reconfiguration problems for subtrees in an unweighted graph [1, 2, 4, 8, 10, 12, 13]. However, to the best of our knowledge, there is no direct relationship between our problems and these known problems, except for the following two reconfiguration problems.

It is known that SPR is PSPACE-complete even for bipartite graphs [1], and for bounded bandwidth (and hence bounded pathwidth) graphs [13]. Since MSTR under VE and VE-N are generalizations of SPR, they are also PSPACE-complete for bipartite graphs and for bounded bandwidth graphs. As positive results, there are polynomial-time algorithms to solve SPR for planar graphs [2], for chordal graphs [1], and for claw-free graphs [1]. (Figure 2(c) shows a part of these results.)

There is another reconfiguration problem for Steiner trees [10], but it is not a generalization of SPR; its reconfiguration rule is exchanging edges (not vertices). As we have seen in the example of Figure 1, the existence of a transformation often changes according to the choice of reconfiguration rules. However, we will show in Section 3 that some known results for this edge-variant [10] can be converted to our MSTR under VE.

1.3 Our contribution

In this paper, we study the computational complexity of MSTR under VE and VE-N with respect to graph classes. (Figure 2 shows all our results for MSTR.)

We first show that MSTR under VE is solvable in polynomial time for interval graphs, while is PSPACE-complete even for split graphs and for planar graphs. Recall that SPR is solvable in polynomial time for planar graphs and for chordal graphs (and hence split graphs). We next show that MSTR under VE-N is solvable in polynomial time for chordal graphs and for planar graphs; these results generalize the known results for SPR [1, 2].

Notice that there are interesting contrasts between the reconfiguration rules VE and VE-N when we focus on planar graphs, chordal graphs, and split graphs: MSTR is PSPACE-complete under VE, while is solvable in polynomial time under VE-N.

We omit proofs for the claims marked with $(*)$ from this extended abstract.

2 Preliminaries

In this paper, we consider only simple and unweighted graphs. For a graph G, we denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G, respectively. For a graph G and its vertex subset $S \subseteq V(G)$, we denote by $G[S]$ the subgraph of G induced by S. For a vertex $v \in V(G)$, let $N_G(v)$ be the set of all neighbors of v in G, that is, $N_G(v) = \{w \in V(G) \mid vw \in E(G)\}$.

We now formally define our reconfiguration rules. Let T and T' be two minimum Steiner trees of a graph G for a terminal set S. Then, T and T' are *adjacent under* VE if there exist two vertices $v \in V(T)$ and $v' \in V(T')$ such that

- $T[V(T) \setminus \{v\}] = T'[V(T') \setminus \{v'\}]$.

Recall that v and v' can be the same vertex; in such a case, only edges incident to $v = v'$ may be changed between T and T'. On the other hand, T and T' are *adjacent under* VE-N if there exist two vertices $v \in V(T)$ and $v' \in V(T')$ such that

- $T[V(T) \setminus \{v\}] = T'[V(T') \setminus \{v'\}]$; and
- $N_T(v) = N_{T'}(v')$.

Note that the first condition for VE-N is the same as the condition for VE. It can hold also under VE-N that $v = v'$, but then $T = T'$ holds.

For two minimum Steiner trees T and T', a *reconfiguration sequence* between T and T' under VE (resp., VE-N) is a sequence $\langle T = T_0, T_1, \ldots, T_\ell = T' \rangle$ of minimum Steiner trees such that T_i and T_{i+1} are adjacent under VE (resp., VE-N) for each $i \in \{0, 1, \ldots, \ell - 1\}$. We write $T \overset{\mathsf{VE}}{\leftrightsquigarrow} T'$ (resp., $T \overset{\mathsf{VE\text{-}N}}{\leftrightsquigarrow} T'$) if there exists a reconfiguration sequence between T and T' under VE (resp., VE-N). Then, we formally define the MINIMUM STEINER TREE RECONFIGURATION problem (MSTR, for short) under VE (resp., VE-N) as follows:

MSTR under VE (resp., VE-N)
Input: A graph G, a terminal set $S \subseteq V(G)$, and two minimum Steiner trees T_0 and T_{r} of G for S.
Task: Determine whether $T_0 \overset{\mathsf{VE}}{\leftrightsquigarrow} T_{\mathrm{r}}$ (resp., $T_0 \overset{\mathsf{VE\text{-}N}}{\leftrightsquigarrow} T_{\mathrm{r}}$) or not.

We denote by a 4-tuple $(G, S, T_0, T_{\mathrm{r}})$ an instance of the problems. Throughout the paper, we assume without loss of generality that $|V(T_0)| = |V(T_{\mathrm{r}})|$ holds; otherwise it is clearly a no-instance.

3 Minimum Steiner Tree Reconfiguration under VE

In this section, we show that MSTR under VE is solvable in polynomial time for interval graphs, while it is PSPACE-complete for split graphs and for planar graphs. To this end, we use the concept of "Steiner sets" and their reconfiguration, which was introduced by [10].

3.1 Steiner sets and their reconfiguration

For a graph G and a terminal set S, a *Steiner set* of G for S is a vertex subset $F \subseteq V(G)$ such that $S \subseteq F$ and $G[F]$ is connected. Notice that if a subtree T of G is a Steiner tree for S, then $V(T)$ is a Steiner set of G for S. Conversely, if F is a Steiner set of G for S, then any spanning tree of $G[F]$ is a Steiner tree for S. A Steiner set F of G for S is *minimum* if the cardinality of F is minimum among all Steiner sets of G for S.

For two Steiner sets F and F' of G for S, a sequence $\langle F = F_0, F_1, \ldots, F_\ell = F' \rangle$ of Steiner sets of G for S is called a *Steiner set sequence* between F and F' if $|F_i \setminus F_{i+1}| = |F_{i+1} \setminus F_i| = 1$ holds for each $i \in \{0, 1, \ldots, \ell - 1\}$. Note that all Steiner sets in the sequence have the same cardinality. The following lemma shows that, in some sense, we do not need to care the tree structure property (but need to care only a connectivity) when we want to check the existence of a reconfiguration sequence under VE.

▶ **Lemma 1** (∗). *Let T and T' be Steiner trees of a graph G for a terminal set S. Then, $T \overset{\mathsf{VE}}{\leftrightsquigarrow} T'$ holds if and only if there exists a Steiner set sequence between two Steiner sets $V(T)$ and $V(T')$ of G for S.*

The concept of Steiner sets was introduced for the reconfiguration of Steiner trees via edge exchanges [10]. Lemma 1 allows us to convert two known results for this edge-exchange variant [10] to our MSTR under VE.

We first consider 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 $v_iv_j \in E(G)$ if and only if $I_i \cap I_j \neq \emptyset$ for each $i, j \in \{1, 2, \ldots, n\}$. For a given graph G, it can be determined in linear time whether G is an interval graph or not [9].

▶ **Theorem 2.** *Let $(G, S, T_0, T_{\mathrm{r}})$ be a given instance of* MSTR *under* VE *such that G is an interval graph. Then, $T_0 \overset{\mathsf{VE}}{\leftrightsquigarrow} T_{\mathrm{r}}$ holds.*

Proof. It is known that if a given graph is an interval graph, then there always exists a Steiner set sequence between any pair of Steiner sets of the same cardinality [10]. Thus, the theorem follows from Lemma 1. ◀

We then consider split graphs. A graph is a *split graph* if its vertex set can be partitioned into a clique and an independent set. The following theorem can be obtained by a polynomial-time reduction which is similar to that for Theorem 3 of [10].

▶ **Theorem 3** (∗). MSTR *under* VE *is PSPACE-complete for split graphs.*

3.2 PSPACE-completeness for planar graphs

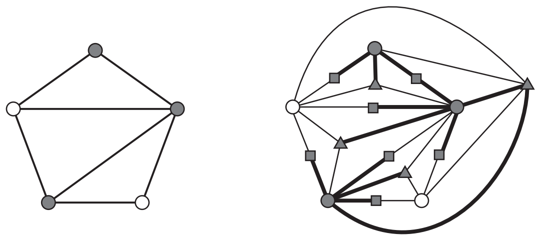
In this subsection, we consider planar graphs, and give the following theorem.

▶ **Theorem 4.** MSTR *under* VE *is PSPACE-complete for planar graphs.*

We note that MSTR under VE is in PSPACE. Therefore, in the remainder of this subsection, we will prove that the problem is PSPACE-hard for planar graphs. To this end, we construct a polynomial-time reduction from the MINIMUM VERTEX COVER RECONFIGURATION problem.

Recall that a *vertex cover* C of a graph G is a vertex subset of G which contains at least one of the two endpoints of every edge in G. A vertex cover C of G is *minimum* if the cardinality of C is minimum among all vertex covers of G. Given a graph G and two minimum vertex covers C_0 and C_{r} of G, the MINIMUM VERTEX COVER RECONFIGURATION problem (MVCR, for short) asks to determine whether or not there exists a sequence $\langle C_0, C_1, \ldots, C_\ell = C_{\mathrm{r}} \rangle$ of minimum vertex covers of G such that $|C_i \setminus C_{i+1}| = |C_{i+1} \setminus C_i| = 1$ holds for any $i \in \{0, 1, \ldots, \ell - 1\}$; we call such a sequence a *vertex cover sequence* between C_0 and C_{r}. We denote by a triple (G, C_0, C_{r}) an instance of MVCR. This problem is known to be PSPACE-complete for planar graphs [5].[1]

[1] Precisely, Hearn and Demaine [5] showed the PSPACE-completeness for the reconfiguration of maximum independent sets. However, it immediately yields the PSPACE-completeness of MVCR.



(a) G' (b) G

Figure 3 (a) An input graph G' of MVCR with a minimum vertex cover C', where the vertices in C' are depicted by gray vertices, and (b) its corresponding planar graph G of MSTR together with the minimum Steiner tree corresponding to C', where face-terminals are depicted by triangles, edge-terminals by squares, and the minimum Steiner tree by thick lines.

Reduction

Let (G', C'_0, C'_r) be a given instance of MVCR such that G' is a planar graph. We fix a planar embedding of G' arbitrarily, and denote by $F(G')$ the set of all faces (including the outer face) of G'. We construct the corresponding instance (G, S, T_0, T_r) of MSTR under VE, as follows. (See Figure 3 as an example.)

We first construct the corresponding graph G from G'. For each face $f \in F(G')$, we add a new vertex w_f, and join w_f and all vertices on the boundary of f by adding new edges. Then, we subdivide each original edge $e = uv \in E(G')$ by adding a new vertex w_e. Let G be the resulting graph. Then, G is a planar graph.

We then define the corresponding terminal set S as the set of all newly added vertices, that is, $S = \{w_f \mid f \in F(G')\} \cup \{w_e \mid e \in E(G')\}$; each w_f is called a *face-terminal*, while each w_e is called an *edge-terminal*.

We finally define the corresponding minimum Steiner trees T_0 and T_r. We will prove later in Lemma 5 that both $C'_0 \cup S$ and $C'_r \cup S$ form minimum Steiner sets of G for S. Then, we choose arbitrary spanning trees of $G[C'_0 \cup S]$ and $G[C'_r \cup S]$ as T_0 and T_r, respectively.

This completes the construction of (G, S, T_0, T_r). The construction can be done in polynomial time.

Correctness

We start with showing that both $C'_0 \cup S$ and $C'_r \cup S$ form minimum Steiner sets of G for S, and hence T_0 and T_r are indeed minimum Steiner trees of G for S.

▶ **Lemma 5** (∗). *Let C' be a vertex subset of $V(G')$. Then, C' is a minimum vertex cover of G' if and only if $C' \cup S$ is a minimum Steiner set of G for S.*

The following lemma completes the proof of Theorem 4.

▶ **Lemma 6.** *(G', C'_0, C'_r) is a* yes-*instance of* MVCR *if and only if (G, S, T_0, T_r) is a* yes-*instance of* MSTR.

Proof. By Lemma 1, it suffices to show that there exists a vertex cover sequence on G' between C'_0 and C'_r if and only if there exists a Steiner set sequence on G between $V(T_0)$ and $V(T_r)$.

First, suppose that there exists a Steiner set sequence $\langle V(T_0) = F_0, F_1, \ldots, F_\ell = V(T_r) \rangle$ between $V(T_0)$ and $V(T_r)$. Then, Lemma 5 implies that the sequence $\langle C'_0 = F_0 \setminus S, F_1 \setminus S, \ldots, F_\ell \setminus S = C'_r \rangle$ is a vertex cover sequence on G' between C'_0 and C'_r.

Second, suppose that there exists a vertex cover sequence $\langle C'_0, C'_1, \ldots, C'_{\ell'} = C'_{\mathrm{r}}\rangle$ between C'_0 and C'_{r}. Then, Lemma 5 implies that the sequence $\langle V(T_0) = C'_0 \cup S, C'_1 \cup S, \ldots, C'_{\ell'} \cup S = V(T_{\mathrm{r}})\rangle$ is a Steiner set sequence on G between $V(T_0)$ and $V(T_{\mathrm{r}})$. ◀

4 Minimum Steiner Tree Reconfiguration under VE-N

In this section, we show that MSTR under VE-N is solvable in polynomial time for chordal graphs and for planar graphs.

We first consider chordal graphs. A graph is *chordal* if G has no induced cycle of length at least four [3]. We use a well-known characterization of chordal graphs, called perfect elimination orderings [3], and give the following theorem.

▶ **Theorem 7** ($*$). MSTR *under* VE-N *is solvable in polynomial time for chordal graphs.*

We then consider planar graphs. Recall that MSTR under VE is PSPACE-complete for planar graphs (as shown in Theorem 4). In contrast, we give the following theorem.

▶ **Theorem 8.** MSTR *under* VE-N *is solvable in polynomial time for planar graphs.*

As a proof of the Theorem 8, we construct a polynomial-time algorithm to solve MSTR under VE-N for planar graphs. Roughly speaking, our idea is to decompose a given instance of MSTR under VE-N into several SPR instances for planar graphs. Then, we can solve each SPR instance by using the polynomial-time algorithm for SPR on planar graphs [2]. Finally, we combine the answers to SPR instances, and output the answer to the original MSTR instance under VE-N. To this ends, we introduce the concept of Steiner tree embeddings and their reconfiguration, which gives a necessary condition for the existence of a reconfiguration sequence under VE-N.

4.1 Steiner tree embeddings and their reconfiguration

We first introduce the concept of Steiner tree embeddings. Let T be a Steiner tree of a graph G for a terminal set S. An injection $\varphi\colon V(T) \to V(G)$ is called a T*-embedding* into G if the following two conditions hold:

- $\varphi(x)\varphi(y) \in E(G)$ if $xy \in E(T)$; and
- $\varphi(s) = s$ holds for each $s \in S \subseteq V(T)$.

Thus, a T-embedding φ *defines* a Steiner tree T_φ of G for S. Observe that no two distinct T-embeddings define the same Steiner tree. A Steiner tree T' is said to be T*-embeddable* if there exists a T-embedding φ which defines T'. Note that T itself is T-embeddable. We now give the following lemma.

▶ **Lemma 9** ($*$). *Let T_a and T_b be any two minimum Steiner trees of a graph G for a terminal set S. If $T_a \overset{\text{VE-N}}{\leftrightsquigarrow} T_b$, then T_b is T_a-embeddable.*

By taking a contrapositive of Lemma 9, we can conclude that a given instance $(G, S, T_0, T_{\mathrm{r}})$ is a no-instance if T_{r} is not T_0-embeddable; this can be checked in polynomial time. Thus, in the remainder of this section, we assume without loss of generality that T_{r} is T_0-embeddable.

We then introduce the reconfiguration of Steiner tree embeddings. Let T be a Steiner tree of a graph G for a terminal set S. We say that two T-embeddings φ and φ' are *adjacent* if exactly one node in T is mapped into different vertices between φ and φ', that is, $|\{x \in V(T) \mid \varphi(x) \neq \varphi'(x)\}| = 1$ holds. For two T-embeddings φ and φ', an *embedding sequence* between φ and φ' is a sequence $\langle \varphi = \varphi_0, \varphi_1, \ldots, \varphi_\ell = \varphi'\rangle$ of T-embeddings such that φ_i and φ_{i+1} are adjacent for each $i \in \{0, 1, \ldots, \ell - 1\}$. We write $\varphi \overset{\text{emb}}{\leftrightsquigarrow} \varphi'$ if there exists an embedding sequence between φ and φ'. Then, we have the following lemma.

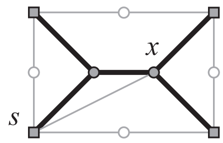


Figure 4 The subtree T depicted by thick lines is a minimum Steiner tree. However, the subpath on T between s and x is not a shortest path in the underlying graph.

▶ **Lemma 10** (∗). *Suppose that T_a and T_b are any two minimum Steiner trees of a graph G for a terminal set S such that T_b is T_a-embeddable. Let φ_a and φ_b be T_a-embeddings which define T_a and T_b, respectively. Then, $T_a \overset{\mathsf{VE\text{-}N}}{\leftrightsquigarrow} T_b$ if and only if $\varphi_a \overset{\mathsf{emb}}{\leftrightsquigarrow} \varphi_b$.*

By Lemmas 9 and 10, MSTR under VE-N can be rephrased to the following problem: Given a graph G, a terminal set S, a minimum Steiner tree T (actually T_0), and two T-embeddings φ_0 and φ_{r} into G, we are asked to determine whether or not there exists an embedding sequence between φ_0 and φ_{r}. Therefore, we also denote by $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$ an instance of MSTR under VE-N.

4.2 Layers for Steiner trees

We here introduce one more important concept, called layers, which was originally introduced by Bonsma [1] for SPR. We generalize the concept to Steiner trees.

Let T be a minimum Steiner tree of a graph G for a terminal set S. For each $x \in V(T)$, let $L_T(x) = \{\varphi(x) \in V(G) \mid \varphi$ is a T-embedding into $G\}$; we call $L_T(x)$ the *layer* of x. Notice that $L_T(s) = \{s\}$ holds for each $s \in S$. We write $L_T(V') = \bigcup_{x \in V'} L_T(x)$ for any node subset $V' \subseteq V(T)$. Then, we have the following property, which says that the layers are disjoint.

▶ **Lemma 11** (∗). *Let T be any minimum Steiner tree of a graph G for a terminal set S. Then, $L_T(x) \cap L_T(y) = \emptyset$ holds for any two distinct nodes $x, y \in V(T)$.*

We call a node $x \in V(T)$ a *branching node* of T if $|N_T(x)| \geq 3$. Let $B(T)$ be the set of all branching nodes of T. Then, we show that a layer of each node in $B(T)$ contains at most two vertices if a given graph is planar.

▶ **Lemma 12** (∗). *Let T be any minimum Steiner tree of a graph G for a terminal set S. If G is planar, then $|L_T(x)| \leq 2$ holds for every branching node $x \in B(T)$.*

We now explain how to compute the layers for a Steiner tree. In SPR [1], we can easily find the layers for a shortest path by computing the distances from the two terminals to each vertex in the underlying graph. This is because the subpath between each node and each terminal is always a shortest path in the underlying graph. On the other hand, this property does not always hold if $|S| \geq 3$, and hence it is difficult to find the layers simply by computing the distances. (For example, see Figure 4.)

Our idea is to compute the "refined" layers for a Steiner tree, instead of computing the layers completely. Let $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$ be a given instance of MSTR under VE-N. Then, for all nodes $x \in V(T)$, it suffices to find vertex subsets $L'_T(x)$ such that

(a) $L'_T(x) \subseteq L_T(x)$; and

(b) $\varphi(x) \in L'_T(x)$ holds for any T-embedding φ satisfying $\varphi_0 \overset{\mathsf{emb}}{\leftrightsquigarrow} \varphi$ or $\varphi_{\mathrm{r}} \overset{\mathsf{emb}}{\leftrightsquigarrow} \varphi$.

To avoid a confusion, we call such a vertex subset $L'_T(x)$ the *refined-layer* of x, while call the (original) layer $L_T(x)$ the *complete-layer* of x. We know that the vertices in $L_T(x) \setminus L'_T(x)$

are useless when we want to check if $\varphi_0 \overset{\mathsf{emb}}{\leftrightsquigarrow} \varphi_{\mathrm{r}}$ or not. The following lemma says that the refined-layers can be found in polynomial time.

▶ **Lemma 13** (∗)**.** *Let $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$ be a given instance of* MSTR *under* VE-N *such that G is a planar graph. Then, there exists a polynomial-time algorithm to compute the refined-layers for all nodes in T.*

Given an instance $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$ of MSTR under VE-N, we can compute refined-layers in polynomial time by Lemma 13. Since the vertices in $L_T(x) \setminus L'_T(x)$, $x \in V(T)$, are useless, we can remove such useless vertices from G. In this way, we can assume without loss of generality that each vertex in G belongs to exactly one (complete-)layer, and we indeed know the layer $L_T(x)$ for each node $x \in V(T)$.

4.3 Decomposition of an MSTR instance into SPR instances

Suppose that $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$ is an instance of MSTR under VE-N, and that we have the layer $L_T(x)$ for each node $x \in V(T)$. We say that two nodes $x, y \in B(T) \cup S$ are *close* if the unique path on T between x and y contains no vertex in $(B(T) \cup S) \setminus \{x, y\}$. To avoid the duplication of $\{x, y\}$ and $\{y, x\}$, we choose one of the ordered pairs (x, y) and (y, x) arbitrarily for each pair of close nodes, and define the set $C(T)$ of all ordered pairs (x, y) of close nodes x, y in $B(T) \cup S$; we call each pair in $C(T)$ a *close pair*.

For each close pair (x, y) in $C(T)$, we now construct the corresponding instance $\mathrm{SPR}(x, y) = (G', S', T', \varphi'_0, \varphi'_{\mathrm{r}})$ such that $|S'| = 2$, as follows. Let P be the unique path on T between x and y. Note that by the definition of close pairs, P is a shortest path on G between x and y. Consider the subgraph of G induced by the vertices in $L_T(V(P))$. We add two new vertices s_x and t_y to the subgraph so that s_x is joined to all vertices in $L_T(x)$ and t_y is joined to all vertices in $L_T(y)$; note that each of s_x and t_y is indeed adjacent to one or two vertices. Let G' be the resulting graph, and let $S' = \{s_x, t_y\}$. We then define T' as the path on G' between s_x and t_y obtained by adding s_x and t_y to P. Note that T' is a shortest path on G' between s_x and t_y. We finally define φ'_0 as a T'-embedding into G' such that $\varphi'_0(s_x) = s_x$, $\varphi'_0(t_y) = t_y$, and $\varphi'_0(x) = \varphi_0(x)$ for each $x \in V(P)$. Similarly, we define φ'_{r} as a T'-embedding into G' such that $\varphi'_{\mathrm{r}}(s_x) = s_x$, $\varphi'_{\mathrm{r}}(t_y) = t_y$, and $\varphi'_{\mathrm{r}}(x) = \varphi_{\mathrm{r}}(x)$ for each $x \in V(P)$. This completes the construction of $\mathrm{SPR}(x, y)$. The corresponding instance $\mathrm{SPR}(x, y)$ can be obtained in polynomial time, and satisfies the following property.

▶ **Lemma 14** (∗)**.** *If G is planar, then G' is also planar.*

By Lemma 14 we can solve the instance $\mathrm{SPR}(x, y)$ for each close pair $(x, y) \in C(T)$ by the polynomial-time algorithm for SPR on planar graphs [2]. We can immediately conclude that the given instance $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$ of MSTR under VE-N is a no-instance if there exists at least one instance $\mathrm{SPR}(x, y)$ whose answer is no. However, even if the answers are yes to all instances $\mathrm{SPR}(x, y)$, $(x, y) \in C(T)$, it is not always possible to extend their embedding sequences to a whole embedding sequence between φ_0 and φ_{r} for the original instance $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$. To check this, we introduce further notion.

Consider an embedding sequence $\mathcal{R} = \langle \varphi = \varphi_0, \varphi_1, \ldots, \varphi_\ell = \varphi' \rangle$ between two T-embeddings φ and φ'. For each node $x \in V(T)$, we say that $\mathcal{R}$ is *x-touching* if the assignment of x is changed by $\mathcal{R}$ at least once; otherwise it is *x-untouching*. Note that if $\varphi(x) \neq \varphi'(x)$ for a node $x \in V(T)$, then any embedding sequence between φ and φ' must be x-touching. On the other hand, if $|L_T(x)| = 1$, then any embedding sequence must be x-untouching. For each close pair $(x, y) \in C(T)$ and its corresponding instance $\mathrm{SPR}(x, y) = (G', S', T', \varphi'_0, \varphi'_{\mathrm{r}})$, we define the set $\mathsf{Touch}(x, y) \subseteq \{(\mathsf{u}, \mathsf{u}), (\mathsf{u}, \mathsf{t}), (\mathsf{t}, \mathsf{u}), (\mathsf{t}, \mathsf{t})\}$, as follows:

- $(\mathsf{u}, \mathsf{u}) \in \mathsf{Touch}(x, y)$ if and only if there exists an embedding sequence between φ_0' and φ_{r}' which is x-untouching and y-untouching;
- $(\mathsf{u}, \mathsf{t}) \in \mathsf{Touch}(x, y)$ if and only if there exists an embedding sequence between φ_0' and φ_{r}' which is x-untouching and y-touching;
- $(\mathsf{t}, \mathsf{u}) \in \mathsf{Touch}(x, y)$ if and only if there exists an embedding sequence between φ_0' and φ_{r}' which is x-touching and y-untouching; and
- $(\mathsf{t}, \mathsf{t}) \in \mathsf{Touch}(x, y)$ if and only if there exists an embedding sequence between φ_0' and φ_{r}' which is x-touching and y-touching.

Note that $\mathsf{Touch}(x, y) = \emptyset$ if there is no embedding sequence between φ_0' and φ_{r}'. Then, we have the following lemma.

▶ **Lemma 15** (∗). *For each close pair $(x, y) \in C(T)$, $\mathsf{Touch}(x, y)$ can be computed in polynomial time.*

We finally solve the given instance $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$ of MSTR under VE-N. Assume that $\mathrm{SPR}(x, y)$ are yes-instances for all close pairs $(x, y) \in C(T)$, and hence $\mathsf{Touch}(x, y) \neq \emptyset$; otherwise $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$ is a no-instance. Consider an assignment $\alpha : B(T) \cup S \to \{\mathsf{u}, \mathsf{t}\}$. Then, we say that α is *synchronizing* if $(\alpha(x), \alpha(y)) \in \mathsf{Touch}(x, y)$ holds for every close pair $(x, y) \in C(T)$. The following lemma completes the proof of Theorem 8.

▶ **Lemma 16** (∗). *Suppose that $(G, S, T, \varphi_0, \varphi_{\mathrm{r}})$ is an instance of* MSTR *under* VE-N *such that G is a planar graph. Then, it is a* yes*-instance if and only if there exists a synchronizing assignment α. Furthermore, the existence of a synchronizing assignment can be checked in polynomial time.*

5 Conclusion

In this paper, we have introduced the Minimum Steiner Tree Reconfiguration (MSTR) problems under two reconfiguration rules VE and VE-N. As summarized in Figure 2, we have studied the polynomial-time solvability of the problems with respect to graph classes, and shown several interesting contrasts. In particular, when we focus on planar graphs, chordal graphs, and split graphs, MSTR is PSPACE-complete under VE, while is solvable in polynomial time under VE-N. It would give us a deeper understanding of the problems if there is a graph class such that MSTR is solvable in polynomial time under VE but is intractable under VE-N.

References

1 P. Bonsma. The complexity of rerouting shortest paths. *Theoretical Computer Science*, 510:1–12, 2013.

2 P. Bonsma. Rerouting shortest paths in planar graphs. *Discrete Applied Mathematics*, 231:95–112, 2017.

3 A. Brandstädt, V.B. Le, and J.P. Spinrad. *Graph Classes: A Survey*. SIAM, Philadelphia, PA, 1999.

4 T. Hanaka, T. Ito, H. Mizuta, B. Moore, N. Nishimura, V. Subramanya, A. Suzuki, and K. Vaidyanathan. Reconfiguring spanning and induced subgraphs. In *Proceedings of the 24th International Computing and Combinatorics Conference (COCOON 2018)*, volume 10976 of *Lecture Notes in Computer Science*, pages 428–440, 2018.

5 R.A. Hearn and E.D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science*, 343(1–2):72–96, 2005.

6 J. van den Heuvel. The complexity of change. In *Surveys in Combinatorics 2013*, volume 409 of *London Mathematical Society Lecture Note Series*, pages 127–160. Cambridge University Press, 2013.

7 T. Ito, E.D. Demaine, N.J.A. Harvey, C.H. Papadimitriou, M. Sideri, R. Uehara, and Y. Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12–14):1054–1065, 2011.

8 M. Kamiński, P. Medvedev, and M. Milanič. Shortest paths between shortest paths. *Theoretical Computer Science*, 412(39):5205–5210, 2011.

9 N. Korte and R.H. Möhring. An Incremental Linear-time Algorithm for Recognizing Interval Graphs. *SIAM Journal on Computing*, 18(1):68–81, 1989.

10 H. Mizuta, T. Ito, and X. Zhou. Reconfiguration of Steiner trees in an unweighted graph. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E100-A(7):1532–1540, 2017.

11 N. Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018.

12 K. Wasa, K. Yamanaka, and H. Arimura. The Complexity of Induced Tree Reconfiguration Problems. *IEICE Transactions on Information and Systems*, 102-D(3):464–469, 2019.

13 M. Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *Journal of Computer and System Sciences*, 93:1–10, 2018.