

On the clustered Steiner tree problem

Bang Ye Wu · Chen-Wan Lin

Published online: 15 July 2014
© Springer Science+Business Media New York 2014

Abstract We investigate the *Clustered Steiner tree* problem on metric graphs, which is a variant of the Steiner minimum tree problem. In this problem, the required vertices are partitioned into clusters, and the subtrees spanning different clusters must be disjoint in a feasible clustered Steiner tree. In this paper, it is shown that the problem is NP-hard even if the inter-cluster tree and all the local topologies are given, where a local topology specifies the tree structure of required vertices in the same cluster. We show that the Steiner ratio of this problem is lower and upper bounded by three and four, respectively. We also propose a $(\rho + 2)$ -approximation algorithm, where ρ is the approximation ratio for the Steiner minimum tree problem, and the approximation ratio can be improved to $\rho + 1$ if the local topologies are given. Two variants of this problem are also studied. When the goal is to minimize the inter-cluster cost and ignore the cost of local trees, the problem can be solved in polynomial time. But it is NP-hard if we ask for the minimum cost of local trees among all solutions with minimum inter-cluster cost.

Keywords Approximation algorithm · Steiner tree · NP-hard · Graph theory

1 Introduction

For a simple undirected graph $G = (V, E, c)$ and a *required vertex* set $R \subseteq V$, a *Steiner tree* is a connected and acyclic subgraph of G that spans all the vertices in R . Due to the large number of applications, Steiner tree problems are extensively studied. The Steiner Minimum Tree (SMT) problem is a classical and well-known NP-hard problem which involves finding a Steiner tree with minimum total edge cost

B. Y. Wu (✉) · C.-W. Lin
National Chung Cheng University, ChiaYi 621, Taiwan, ROC
e-mail: bangye@cs.ccu.edu.tw

(Garey and Johnson 1979; Karp 1972). The best approximation ratio ρ on general metrics achieved in polynomial time is an important properties for many graph problems. From the first non-trivial result $11/6$ (Zelikovsky 1993), it has been improved several times (Robins and Zelikovsky 2005; Byrka et al. 2010). The current best approximation ratio is 1.39 (Byrka et al. 2010). Numerous variants of the SMT problem have been studied, for example, the versions on the Euclidean metric (Garey et al. 1977) and the rectilinear metric (Garey and Johnson 1977), the Steiner forest problem (Agrawal et al. 1995), the group Steiner tree problem Garg et al. (1998), the terminal Steiner tree problem (Chen et al. 2003; Drake 2004; Lin and Xue 2002; Lu et al. 2003; Martinez et al. 2007), the internal-selected Steiner tree problem (Hsieh and Yang 2007; Huang et al. 2013; Li et al. 2010), and many others (Ding and Xue 2014; Hsu et al. 2005; Zou et al. 2009).

An undirected complete graph $G = (V, E, c)$ is a *metric graph* if all edge costs are nonnegative and satisfy the triangle inequality, i.e., $c(u, v) + c(v, x) \geq c(u, x)$ for all $u, v, x \in V$. In this paper, we consider another variant of SMT, the *Clustered Steiner tree* (CLUSTEINER) problem. In addition to a metric graph $G = (V, E, c)$ and required vertex set R , we are also given a partition $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ of R . A Steiner tree T is a clustered Steiner tree for \mathcal{R} if all the vertices in the same cluster (R_i) are *clustered together* in T . More formally, a Steiner tree T is a clustered Steiner tree if all the *local trees* are mutually disjoint, where the local tree of a cluster R_i in T is the minimal subtree of T spanning R_i . That is, T can be cut into k subtrees by removing $k - 1$ edges such that each subtree is a Steiner tree for one cluster R_i . The CLUSTEINER problem is formally defined as follows.

Clustered Steiner Tree problem (CLUSTEINER)

INSTANCE: A metric graph $G = (V, E, c)$, required vertices $R \subseteq V$, and a partition $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ of R .

GOAL: Find a minimum-cost clustered Steiner tree for \mathcal{R} .

An equivalent definition is that for $s_i, t_i \in R_i$ and $s_j, t_j \in R_j$ the unique $s_i t_i$ -path and $s_j t_j$ -path in T are disjoint whenever $i \neq j$. If there is only one cluster or each required vertex is itself a cluster, the problem degenerates to the original Steiner minimum tree problem. The *local cost* of a clustered Steiner tree is the total cost of all edges in its local trees, and the cost of the remaining edges is called *inter-cluster cost*. In addition to CLUSTEINER, we also study two variants. The first one is the minimum inter-cluster Steiner tree problem (INTERCLUSTER), in which we want to find a cluster Steiner tree with minimum inter-cluster cost and ignore the local cost. The second variant is the minimum local-cost Steiner tree problem (LOCALCLUSTER) which asks for the minimum local cost among all clustered Steiner trees with minimum inter-cluster cost. Figure 1 illustrates the definitions.

The contribution of this paper is as follows.

- When $R = V$, or equivalently no Steiner vertices can be used, the problem can be simply solved in polynomial time.
- The Steiner ratio for CLUSTEINER is lower and upper bounded by three and four, respectively, where the Steiner ratio is defined as the largest possible ratio of the minimal cost without using any Steiner vertex to the optimal cost.

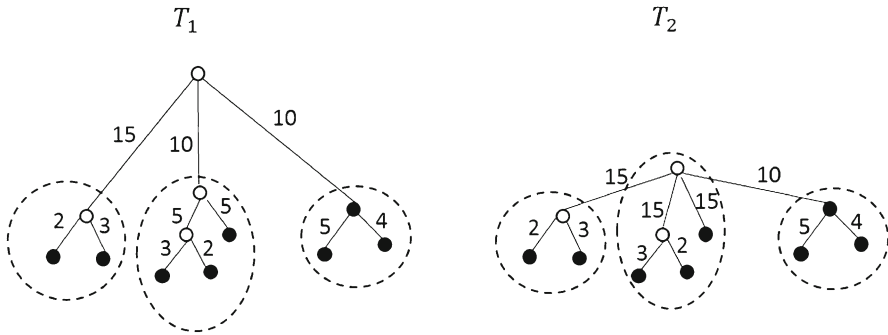


Fig. 1 Both T_1 and T_2 are clustered Steiner trees of the same input, in which white and black vertices are Steiner and required vertices, respectively. The subtrees circled by dashed lines are local trees of three clusters. The inter-cluster costs of T_1 and T_2 are 35 and 25, respectively, while their local costs are 29 and 49, respectively. For CLUSTEINER, T_1 has smaller total cost than T_2 . But for INTERCLUSTER and LOCALCLUSTER, T_2 is better than T_1 since the inter-cluster cost of T_2 is smaller

- CLUSTEINER remains NP-hard even if the inter-cluster tree and the local topologies of all clusters are given, where a local topology specifies the tree structure of a local tree and will be formally defined in the next section.
- CLUSTEINER can be $(\rho + 2)$ -approximated in polynomial time, where ρ is the best approximation ratio for the Steiner minimum tree problem. The ratio can be improved to $\rho + 1$ if all the local topologies are given.
- INTERCLUSTER can be solved in polynomial time when at least one cluster contains more than one required vertex.
- LOCALCLUSTER is NP-hard even if at least one cluster contains more than one required vertex.

A possible application of CLUSTEINER is as follows. When designing transportation or computer networks, the links are usually divided into two levels: inter-cluster or intra-cluster, possibly with different costs, qualities, and capacities. Also, after the network is built, the communications between nodes in the same cluster should be routed locally rather than globally for the sake of capacity consideration or the simpleness of routing protocols. Another application is for the case that all the local topologies are given. In this case the task is to design the inter-cluster topology, as well as the possible insertion of local Steiner vertices without violating their topologies. A similar consideration was also studied for the *traveling salesperson problem* (TSP), named *clustered TSP* problem (Bao and Liu 2012; Guttmann-beck et al. 2000). For this problem, the goal is to find a minimum cost Hamiltonian path such that the vertices of each cluster are visited consecutively.

The rest of the paper is organized as follows. In Sect. 2, we give some notation and definitions. In Sect. 3, we discuss some properties of the Steiner ratio. In Sect. 4, we show the NP-hardness of CLUSTEINER. The approximation algorithms are shown in Sect. 5. In Sects. 6 and 7, we discuss the two variants INTERCLUSTER and LOCALCLUSTER, respectively. Finally some future work and open questions are given in Sect. 8.

2 Notation and definitions

For a graph $G = (V, E, c)$, V and E are the vertex and the edge sets, respectively, and c is the edge cost. In this paper we consider only undirected graph, and an (undirected) edge incident to vertices u and v is denoted by (u, v) . The cost of (u, v) is denoted by $c(u, v)$. For a subgraph T of G , $c(T)$ denotes the total cost of all edges of T . For a graph G , $V(G)$ and $E(G)$ denote the vertex and the edge sets, respectively. For a vertex subset U , the subgraph of G induced by U is denoted by $G[U]$. By $\text{smt}(G, R)$, we denote a Steiner minimum tree with instance (G, R) and also its cost. We use $\text{mst}(G, R)$ to denote a minimum spanning tree (MST), and also its cost, of $G[R]$. A path with end vertices s and t is called an st -path. For a set S , a collection \mathcal{S} of subsets of S is a *partition* of S if the subsets are mutually disjoint and their union is exactly S .

Definition 1 For a tree T spanning S , i.e., $S \subseteq V(T)$, the *local tree* of S on T is the minimal subtree of T spanning all vertices in S . In other words, if Y is the local tree of S , then $S \subseteq V(Y)$ and all leaves of Y are in S .

Definition 2 Let $\mathcal{R} = \{R_i \mid 1 \leq i \leq k\}$ be a partition of R . A Steiner tree T for R is a *clustered Steiner tree* for \mathcal{R} if the local trees of all $R_i \in \mathcal{R}$ are mutually disjoint, i.e., there exists a cut set $C \subseteq E(T)$ with $|C| = k - 1$ such that each component of $T - C$ is a Steiner tree T_i for R_i for all $1 \leq i \leq k$.

The CLUSTERED STEINER problem is formally defined as follows.

Clustered Steiner Tree problem (CLUSTERED STEINER)

INSTANCE: A metric graph $G = (V, E, c)$, required vertices $R \subseteq V$, and a partition $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ of R .

GOAL: Find a minimum-cost clustered Steiner tree for \mathcal{R} .

A vertex not in R is a *Steiner vertex*. In the remainder of this paper, we assume that (G, \mathcal{R}) is the instance of the problem, where $G = (V, E, c)$ and $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ is a partition of R . We also use $n = |V|$ and note that $|E| \in \Theta(n^2)$ since G is a complete graph.

An Eulerian tour is a cycle traveling all the edges exactly once. A graph is Eulerian if there exists an Eulerian tour. A connected undirected graph is Eulerian if and only if all the vertex degrees are even. A Hamiltonian path is a path visiting each vertex exactly once.

For a graph H , *contraction* of an edge (u, v) replaces u, v with a new vertex w . For any other vertex s , the edge cost is set to $c(s, w) = \min\{c(s, u), c(s, v)\}$. For a subgraph S , contracting S in H means contracting all the edges $E(S)$ in an arbitrary order, and the resulting graph is denoted by H/S . For convenience, we also use H/S to denote $H/H[S]$ when S is a vertex subset. Let G/\mathcal{R} denote the graph resulted from contracting every R_i in \mathcal{R} , i.e., each cluster of required vertices is shrunk into a vertex.

For a graph T and $(u, r), (r, v) \in E(T)$, “taking a shortcut between u, v ” means we replace edges (u, r) and (r, v) with (u, v) . Similarly, for a uv -path, taking a shortcut between u, v replaces the path with edge (u, v) .

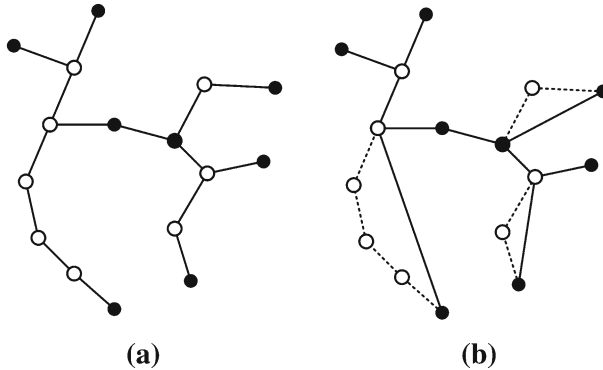


Fig. 2 Topology of a local tree. White vertices are Steiner vertices and *black* ones are required vertices. **a** The original local tree. **b** The topology (*solid lines*). Note that the degree-2 Steiner vertices are not vertices of the topology

Definition 3 For a local tree T_i of R_i , the *topology* of T_i is the tree obtained by repeatedly taking shortcuts between the two neighbors of Steiner vertices with degree two and removing such Steiner vertices in T_i until there are no such vertices.

For simplicity, the topology of a local tree will also be called “local topology”. Figure 2 depicts an example of the topology of a local tree. For a clustered Steiner tree T , contracting all the local trees results in a tree, denoted by T/\mathcal{R} , called the *inter-cluster tree* of T . The topology of the inter-cluster tree is also called “inter-cluster topology”. Since there is always an optimal solution tree without degree-two vertices in the inter-cluster tree, the topology of an inter-cluster tree is itself. For a clustered Steiner tree T , let $\alpha(T)$ denote the total cost of all its local trees and $\beta(T) = c(T) - \alpha(T)$ the cost of its inter-cluster topology, i.e., $\beta(T) = c(T/\mathcal{R})$. The two variants of CLUSTERSTEINER we study in this paper are formally defines as follows. The first variant looks for a clustered Steiner tree minimizing the inter-cluster cost.

Minimum Inter-cluster Steiner Tree (INTERCLUSTER)

INSTANCE: A metric graph $G = (V, E, c)$, required vertices $R \subseteq V$, and a partition $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ of R .

GOAL: Find a clustered Steiner tree minimizing $\beta(T)$.

In the second variant, we want to first minimize the inter-cluster cost $\beta(T)$, and among those with minimum inter-cluster cost $\beta(T)$ we ask for the one with minimum local cost.

Minimum Local-Cost Clustered Steiner Tree (LOCALCLUSTER)

INSTANCE: A metric graph $G = (V, E, c)$, required vertices $R \subseteq V$, and a partition $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ of R .

GOAL: Find a clustered Steiner tree T with minimum $\alpha(T)$ subject to $\beta(T)$ is minimum.

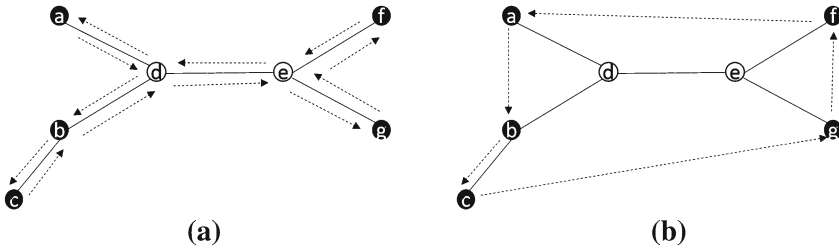


Fig. 3 An example of constructing a Hamiltonian path/cycle from a tree. **a** By doubling the tree edges, we obtain an Eulerian cycle $Y = (a, d, b, c, b, d, e, g, e, f, e, d, a)$, in which d, e are Steiner vertices and the others are required vertices. **b** Starting from a , since the next vertex d in Y is not required, we make a shortcut to the next unvisited required vertex b . Next we visit c . After c , since vertex b has been visited and vertices d and e are not required, the next unvisited required vertex is g . Therefore we make a shortcut from c to g . Continue this process and we can obtain a Hamiltonian cycle (a, b, c, g, f, a) of the required vertices

3 Steiner ratio of CLUSTEINER

Possibly the simplest way to approximate the SMT is by MST. The *Steiner ratio* (for the classic SMT problem) is the largest possible ratio between the cost of an MST and the cost of an SMT. The inequality (1) is well-known, see for example [Wu and Chao \(2004\)](#), which shows the Steiner ratio is two for general metric spaces.

$$\text{mst}(G, R) \leq 2 \cdot \text{smt}(G, R). \tag{1}$$

The inequality can be simply shown as follows. Let $T = \text{smt}(G, R)$. By doubling $E(T)$, we can obtain an Eulerian multigraph and therefore an Eulerian tour Y with $c(Y) = 2c(T) = 2\text{smt}(G, R)$. Travelling along the Eulerian tour and taking shortcuts between consecutive unvisited required vertices, we can obtain a Hamiltonian path of $G[R]$ with cost at most $c(Y)$ because of the triangle inequality. An example is shown in Fig. 3. Since MST is the cheapest way to connect R , we have that $\text{mst}(G, R) \leq c(Y)$ and the inequality follows.

When $R = V$, i.e., the minimum clustered spanning tree problem, the problem is equivalent to the case that no Steiner vertices are allowed. We now show a simple algorithm for this variant. Since no Steiner vertices are allowed, the local tree of R_i in the optimal tree is an $\text{mst}(G, R_i)$ for each i . Similarly the inter-cluster topology is an MST of G/R . The next result is simple, in which an MST can be found in $O(n \log n + |E|) = O(n^2)$ time for a metric graph ([Cormen et al. 2001](#)).

Proposition 1 *The minimum clustered spanning tree problem can be solved with the same asymptotic time complexity as the MST problem.*

However, we found that the Steiner ratio 2 does not hold for CLUSTEINER. Figure 4 gives a simple example. The left tree (a) is a minimum clustered Steiner tree with cost $p(3 + \epsilon)$, where $p = |R_1|$. If no Steiner vertices can be used, the right tree (b) is the best and the cost is $6(p - 1) + p(2 + \epsilon) \approx 8p$. The ratio is about $8/3 > 2$. Note that an MST consists of a path connecting required vertices not in R_1 and linking vertices

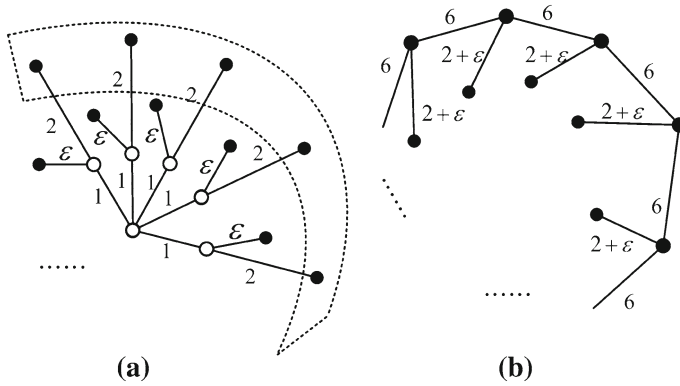


Fig. 4 An example for Steiner ratio larger than 2 for CLUSTEINER. *Black vertices* are required vertices and *white ones* are Steiner vertices. The required vertices circled by *dotted line* are in one cluster R_1 , and all the other clusters are singletons. ϵ is an arbitrarily small positive number. For any vertices u, v , $c(u, v)$ is the same of the cost of the uv -path in the tree on the left. **a** A clustered Steiner tree. **b** The best clustered Steiner tree without any Steiner vertex

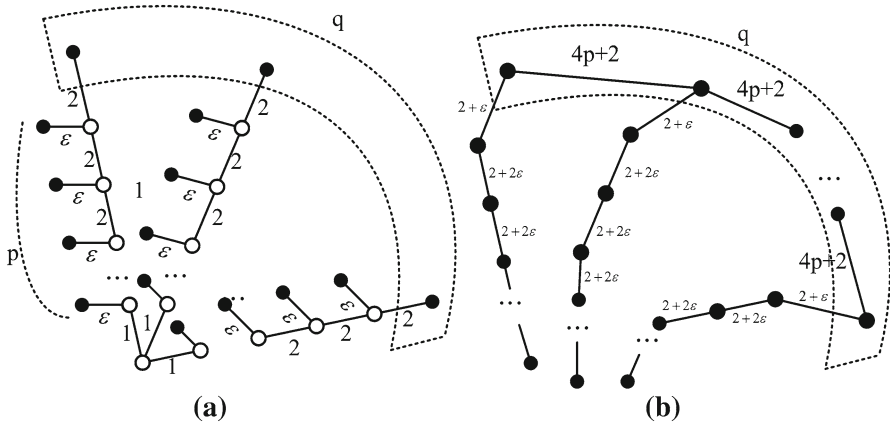


Fig. 5 An example with Steiner ratio three. The setting is similar to Fig. 4. R_1 consists of the q required vertices circled by *dotted line*. As indicated, each path has p internal Steiner vertices. **a** The optimal solution. **b** The best one without any Steiner vertex

in R_1 to the path individually. However, it is not feasible for CLUSTEINER since R_1 is not clustered together, i.e., the local tree contains other required vertices.

The above example shows that the Steiner ratio of CLUSTEINER is at least $8/3$. Figure 5 shows an even worse example. The optimal tree (a) has cost $q(p(2+\epsilon)+1) \approx 2pq + 2q$. The right tree (b) is the best possible without any Steiner vertex, and its cost is $(q - 1)(4p + 2) + qp(2 + \epsilon)$. The ratio is asymptotically three when pq is large. By this example, we have the lower bound in Lemma 1.

Lemma 1 *The Steiner ratio of CLUSTEINER is at least three.*

4 NP-hardness for fixed topologies

At the first glance, the hardness of CLUSTERING seems from determining the best local and inter-cluster topologies. In this section we shall show that NP-hardness remains even when the topologies are given. A *caterpillar* is a tree of which all the internal vertices form a path. An *st-caterpillar* is a caterpillar with leaves s and t adjacent to the two endpoints of the path, respectively. Consider the following problem.

STEINER CATERPILLAR

INSTANCE: A metric graph $G = (V, E)$, required vertices $R \subset V$, and two vertices $x, y \in R$.

GOAL: Find a minimum-cost xy -caterpillar spanning R such that all internal vertices of the xy -path are not in R .

The (1, 2)-STEINER CATERPILLAR is the special version in which all the edge costs are either one or two. Note that a complete graph with all edge costs of one or two is a metric graph. We show the NP-hardness of (1, 2)-STEINER CATERPILLAR by transforming from the following well-known NP-complete problem.

DOMINATING SET

INSTANCE: A simple undirected graph H and an integer h .

QUESTION: Is there a dominating set of size at most h , i.e., a set $S \subseteq V(H)$ with $|S| \leq h$ such that for all $u \notin S$ there exists $v \in S$ for which $(u, v) \in E$?

Lemma 2 (1, 2)-STEINER CATERPILLAR is NP-hard.

Proof We reduce DOMINATING SET to (1, 2)-STEINER CATERPILLAR, and then the result follows from the NP-completeness of DOMINATING SET (Garey and Johnson 1979). Let (H, h) be an instance of DOMINATING SET. We construct an instance (G, \mathcal{R}) of STEINER CATERPILLAR as follows. Let $V(H) = \{v_i \mid 1 \leq i \leq p\}$. For each $v_i \in V(H)$, we create a Steiner vertex s_i . Let $S = \{s_i \mid 1 \leq i \leq p\}$ and $R = V(H) \cup \{x, y\}$, where $x, y \notin V(H)$ are two added vertices. Let $V(G) = R \cup S$ and the edge costs are as follows.

$$\begin{cases} c(s_i, s_j) = 1 & \text{for } 1 \leq i < j \leq p \\ c(s_i, x) = c(s_i, y) = 1 & \text{for } 1 \leq i \leq p \\ c(s_i, v_j) = 1 & \text{for } (v_i, v_j) \in E(H) \text{ or } i = j \end{cases}$$

The cost of any other edge is two. We now claim that H has a dominating set of size h if and only if there is an xy -caterpillar of cost $p + h + 1$.

First, suppose that D is a dominating set of H and $|D| = h$. W.l.o.g. let $D = \{v_i \mid 1 \leq i \leq h\}$. Construct an xy -caterpillar with internal vertices $S' = \{s_i \mid 1 \leq i \leq h\}$ which are exactly those Steiner vertices corresponding to D . The order of the internal vertices is irrelevant. Since D is a dominating set, for each v_i there is an internal vertex $s_j \in S'$ such that $c(v_i, s_j) = 1$. Since the xy -path has h internal vertices and all its edges are of cost one, the total cost is $p + h + 1$.

Conversely, suppose that there is an xy -caterpillar T of cost $p + h + 1$. For any v_i , if there does not exist an internal vertex s in T such that $c(s, v_i) = 1$, we can add s_i

to the xy -path and then connect v_i to s_i . Since the cost of any pair of Steiner vertices is one, the total cost is not increased. Therefore we can obtain an xy -caterpillar of the same cost such that all leaves except for x, y are connected to the xy -path by an edge of cost one. Since the total cost is $p + h + 1$, the number of internal vertices is h , and its corresponding vertex subset in $V(H)$ is a dominating set of H . \square

Theorem 1 *CLUSTERED STEINER is NP-hard even when all the local topologies and the inter-cluster topology are given.*

Proof By Lemma 2, it is sufficient to show that (1, 2)-STEINER CATERPILLAR is a special case. For an instance $(G = (V, E, c), R, x, y)$ of (1, 2)-STEINER CATERPILLAR, we transform it into an instance $(G' = (V, E, c'), \mathcal{R})$ of CLUSTERED STEINER. Let $\mathcal{R} = \{R_i \mid 1 \leq i \leq k\}$, where $k = |R| - 1$, $R_1 = \{x, y\}$ and $|R_i| = 1$ for $i \geq 2$. Since every cluster contains at most two vertices, the local topologies are fixed. The given inter-cluster topology is the star centered at R_1 . Let $c'(x, y) = 2L$. Let $c'(x, v) = c(x, v) + L$ and $c'(y, v) = c(y, v) + L$ for all $v \neq x, y$, where $L = 5$. All the other edge costs remain the same. We assume that $k \geq 3$ since (1, 2)-STEINER CATERPILLAR can be trivially solved in polynomial time if the number of required vertices is a constant.

First we show that G' is also a metric graph. It is sufficient to show the triangle inequality for any three vertices involving x or y . For any vertex $v \notin \{x, y\}$, $c'(x, y) + c'(x, v) > c'(y, v)$ since $c'(x, y)$ is the largest edge cost, and $c'(x, v) + c'(v, y) = c(x, v) + c(v, y) + 2L > c'(x, y) = 2L$. For $u, v \notin \{x, y\}$, $c'(x, u) + c'(u, v) = L + c(x, u) + c(u, v) \geq L + c(x, v) = c'(x, v)$.

Let T be a minimum clustered Steiner tree. Since the inter-cluster topology is a star centered at R_1 , if there are no Steiner vertices on T , the cost is larger than $(k + 1)L$ since an edge connecting any vertex to x or y has cost more than L . But adding any Steiner vertex to subdivide (x, y) and connecting all the required vertices to it reduces the cost to at most $2L + (k + 1)2 < (k + 1)L$ since $L > 4$ and $k \geq 3$. Recall that $c'(u, v) \leq 2$ for vertices $u, v \notin \{x, y\}$. Since there exists at least one Steiner vertex in T , we claim that no required vertices are connected to x or y . Otherwise, re-connecting this vertex to any Steiner vertex reduces the total cost. We conclude that x and y are leaves in T .

Therefore the optimal solution of the CLUSTERED STEINER problem is the same as the one of the STEINER CATERPILLAR except for the additional cost $2L$. \square

5 Approximation algorithms

By Algorithm 1, we shall show that any clustered Steiner tree T can be transformed into a clustered Steiner tree T' of which the local trees have no Steiner vertices. Fig. 6 illustrates an example.

In fact, we replace each local tree with a Hamiltonian path and cut some edges to break cycles. Since the Hamiltonian path consists of shortcuts of the cycle Y , the next result follows from the triangle inequality.

Claim 1 Each local tree T_i is replaced with a Hamiltonian path of R_i with cost at most $2c(T_i)$.

Algorithm 1

Input: a clustered Steiner tree T .

Output: a clustered Steiner tree T' .

- 1: $T' \leftarrow T$;
- 2: **for all** local tree T_i of T' **do**
- 3: construct a multigraph H_i by doubling the edges of T_i ;
- 4: construct an Eulerian tour Y on H_i ;
- 5: pick any required vertex r in T_i ;
- 6: **while** there is a non-visited required vertex in Y **do** ▷ travelling along Y
- 7: let r' be the next non-visited required vertex in Y ;
- 8: let s be the previous vertex of r' in Y ;
- 9: replace (s, r') with (r, r') in T_i ; ▷ no change if $s = r$
- 10: $r \leftarrow r'$;
- 11: **end while**
- 12: **end for**
- 13: output T' .

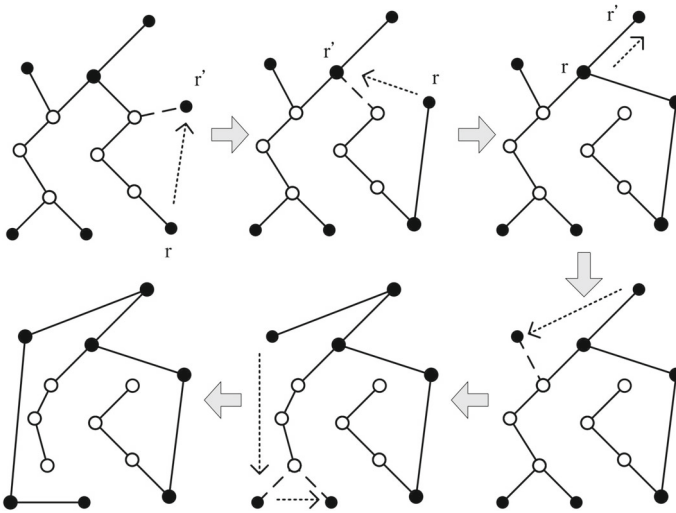


Fig. 6 An example illustrates Algorithm 1. Here only one local tree is shown, in which white and black vertices are Steiner and required vertices, respectively. For each step, r and r' are the tail and head of the arrow, respectively. The dashed edge is the one to be removed

After the transformation, each local tree is the added Hamiltonian path, and cutting the edges makes its topology a part of the inter-cluster topology. Since no other edges are added, the cost increment is at most the original cost of the local trees. Recall that $\alpha(T)$ is the local cost and $\beta(T) = c(T) - \alpha(T)$ is inter-cluster cost.

Claim 2 $\beta(T') \leq \beta(T) + \alpha(T) = c(T)$.

The next lemma comes from the above two claims.

Lemma 3 *There exists a clustered Steiner tree T' with no Steiner vertices in its local trees and $\beta(T') \leq \beta(T^*) + \alpha(T^*) = c(T^*)$, where T^* is a minimum clustered Steiner tree.*

Algorithm 2 Approximating the minimum clustered Steiner tree

Input: an instance (G, \mathcal{R}) of the problem.

Output: a clustered Steiner tree T^a .

- 1: construct $T_i^a \leftarrow \text{mst}(G, R_i)$ for each i ;
- 2: construct G/\mathcal{R} and let $R' = \{r'_i \mid 1 \leq i \leq k\}$, where r'_i is the vertex resulted from the contraction of R_i ;
- 3: construct a ρ -approximation T_0^a of $\text{smt}(G/\mathcal{R}, R')$;
- 4: replace r'_i with T_i^a to obtain a clustered Steiner tree T^a ;
- 5: output T^a .

Theorem 2 *Algorithm 2 reports a $(2 + \rho)$ -approximation for CLUSTEINER in $O(n \log n + f(n))$ time, where ρ and $f(n)$ are the approximation ratio and the time complexity of any approximation algorithm for Steiner minimum tree on a metric graph with n vertices.*

Proof Let T^* be a minimum cluster Steiner tree. Let T^a be the tree constructed by Algorithm 2 and T' be a tree satisfying Lemma 3. We have that $\alpha(T^a) \leq 2\alpha(T^*)$. Since T' has no Steiner vertices in its local tree, we have $\beta(T') \geq \text{smt}(G/\mathcal{R}, R')$, and therefore $\beta(T^a) \leq \rho\beta(T')$. Since $\beta(T') \leq \alpha(T^*) + \beta(T^*)$ by Lemma 3, then

$$\begin{aligned} c(T^a) &= \alpha(T^a) + \beta(T^a) \\ &\leq 2\alpha(T^*) + \rho(\alpha(T^*) + \beta(T^*)) \\ &\leq (2 + \rho)\alpha(T^*) + \rho\beta(T^*) \leq (2 + \rho)c(T^*). \end{aligned}$$

□

In Algorithm 2, if we use $\text{mst}(G, R')$ instead of the Steiner tree T_0^a , we obtain the best clustered Steiner tree without any Steiner vertex. Let Y be the tree. Since $\text{mst}(G, R') \leq 2\beta(T')$ by (1), we have that $\beta(Y) \leq 2\beta(T') \leq 2(\alpha(T^*) + \beta(T^*))$, and therefore $c(Y) \leq 4\alpha(T^*) + 2\beta(T^*)$.

Corollary 1 *The Steiner ratio for CLUSTEINER is at most four.*

Next we consider the case when the local topologies are given. Let Y_i be the local topology for R_i . Clearly $R_i \subseteq V(Y_i)$. Let $S_i = V(Y_i) - R_i$. The vertices in S_i are the Steiner vertices with degree at least three in the local tree. We may assume that $S_i \cap S_j = \emptyset$ for $i \neq j$. Otherwise there are no solutions. We can modify Algorithm 3 such that the shortcuts are taken between vertices in $V(Y_i)$ but not only R_i . The local tree T_i of the optimal tree T^* is now transformed to Y_i . Figure 7 illustrates an example. Similar to Lemma 3, we have the next result.

Corollary 2 *Suppose that T^* is a minimum cluster Steiner tree and the local topology Y_i is given for each i . There exists a clustered Steiner tree T' with $\beta(T') \leq \beta(T^*) + \alpha(T^*) = c(T^*)$ such that the vertex set of each local tree is exactly $V(Y_i)$.*

Theorem 3 *When the local topologies are given, the problem CLUSTEINER can be $(1 + \rho)$ -approximated in $O(n \log n + f(n))$ time.*

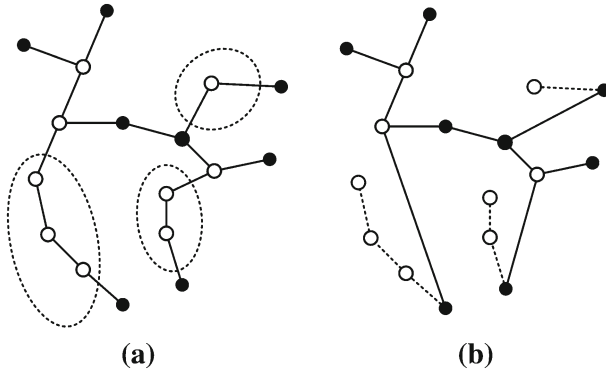


Fig. 7 Transformation from T_i to its topology Y_i . **a** A local tree in which the white vertices are Steiner vertices. The paths circled by dotted line will be cut into the inter-cluster tree. **b** After the transformation. The dotted lines are now a part of the inter-cluster tree

Proof Let $S' = V - R - \bigcup_i S_i$ be the possible Steiner vertices in the inter-cluster tree. The approximation algorithm is similar to Algorithm 2 except that we use Y_i as the local tree T_i^a and construct the ρ -approximation of $\text{smt}(G[R' \cup S'], R')$ as the inter-cluster tree. Therefore $\beta(T') \geq \text{smt}(G[R' \cup S'], R')$, and then $\beta(T^a) \leq \rho\beta(T')$. By the triangle inequality, the cost of a tree is at least the cost of its topology, and we have that $\alpha(T^a) \leq \alpha(T^*)$. By Corollary 2, $\beta(T') \leq \alpha(T^*) + \beta(T^*)$. In summary,

$$\begin{aligned} c(T^a) &= \alpha(T^a) + \beta(T^a) \\ &\leq \alpha(T^*) + \rho(\alpha(T^*) + \beta(T^*)) \\ &\leq (1 + \rho)\alpha(T^*) + \rho\beta(T^*) \leq (1 + \rho)c(T^*). \end{aligned}$$

□

6 The INTERCLUSTER problem

A cluster R_i is called “singleton cluster” if it contains only one vertex. In this section we shall first show a greedy algorithm for INTERCLUSTER without singleton clusters, and then show how to deal with the case with singleton clusters. Algorithm 3 is a procedure called by Algorithm 4 to find $k - 1$ cheapest edges such that there exists a clustered Steiner tree with these edges as its inter-cluster edges. The remaining steps of Algorithm 4 constructs an optimal solution tree. Figure 8 illustrates an example.

As Algorithm 4 is running, a *pending Steiner vertex* is a Steiner vertex not belonging to any local tree. Let T_i be a local tree with at least one edge and s be a pending Steiner vertex. By “inserting s into T_i ”, we denote the following operation:

$$T_i \leftarrow T_i \cup \{(s, u), (s, v)\} - \{(u, v)\},$$

where (u, v) is an arbitrary edge in T_i . We note that T_i remains a local tree after the operation and that s becomes a Steiner vertex in the local tree.

Algorithm 3

Input: a metric graph G and a partition $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ of R .
 1: $G' \leftarrow G/\mathcal{R}$; ▷ each vertex is either a shrunk cluster or a Steiner vertex
 2: let H be the graph with $V(H) = V(G')$ and $E(H) = \emptyset$;
 3: let Q be a priority queue containing $E(G')$;
 4: **while** $|E(H)| < k - 1$ **do**
 5: $e \leftarrow \text{extract-min}(Q)$;
 6: **if** $H + e$ has no cycles **then**
 7: add e into H ;
 8: **end if**
 9: **end while**

Algorithm 4

1: construct any spanning tree T_i of R_i as the local tree for each i ;
 2: $Y \leftarrow \bigcup_i T_i$;
 3: run Algorithm 3 to obtain $E(H)$;
 4: for each edge in $E(H)$, add the corresponding edge into Y ;
 5: **while** there is a pending Steiner vertex s in Y **do**
 6: pick a local tree T_i such that s and T_i are on different components of Y ;
 7: insert s into T_i ;
 8: **end while**
 9: output Y .

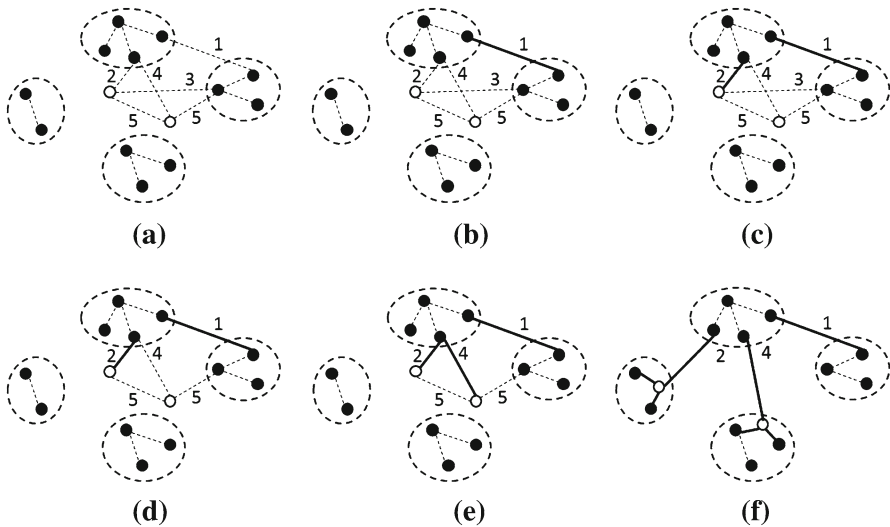


Fig. 8 An example of Algorithms 3 and 4. White vertices are Steiner vertices and *black* ones are required vertices. Each *circle* is for one cluster. The selected edges in solution are drawn by *solid* lines, and the edges with large costs are not shown

Theorem 4 *When there are no singleton clusters, Algorithm 4 finds an optimal solution of INTERCLUSTER in $O(n^2 \log n)$ time.*

Proof Let β^* denote total cost of the edges found by Algorithm 3. We shall show that β^* is the necessary and sufficient inter-cluster cost to construct a clustered Steiner tree. First, the necessity is similar to the optimality of Kruskal’s algorithm for finding

a minimum spanning tree. Since we pick the edges with costs from small to large and edges are discarded only because it make a cycle, any clustered Steiner tree has inter-cluster cost at least β^* . It remains to show that β^* is sufficient. That is, Algorithm 4 reports a feasible solution with inter-cluster cost β^* .

Suppose that there are t pending Steiner vertices after step 4. Since there are $k - 1$ “non-local” edges in Y , the number of components is $k + t - (k - 1) = t + 1$. Each execution of step 7 simultaneously reduces the number of components and the number of pending Steiner vertices by one. We conclude that the while-loop at steps 5–8 can be successfully executed. Consequently there is exactly one component when the algorithm terminates. By step 6, no cycles will be formed and therefore the algorithm returns a tree. Since a Steiner vertex is inserted into exactly one local tree, it results in a valid clustered Steiner tree. To see the inter-cluster cost, we observe that the inter-cluster edges are exactly those found by Algorithm 3 since all the Steiner vertices are inserted into local trees with more than one required vertex.

In worst case, the time complexity of Algorithm 3 is the same as sorting all edges. Since a metric graph is a complete graph, the time complexity is $O(n^2 \log n)$. All the other steps of Algorithm 4 can be done in linear time. \square

Next we consider the case with singleton clusters. First we note that if all clusters are singletons, the problem degenerates to the original Steiner minimum tree and thus NP-hard. Therefore we focus on the case that at least one cluster is not a singleton cluster.

The difficulty here is that, at steps 6–7 of Algorithm 4, we cannot insert a Steiner vertex into a local tree with only one required vertex. Simply connecting the Steiner vertex to the required vertex will result in an additional inter-cluster edge. To deal with this case, we modify Algorithm 3–5 by inserting a while-loop at steps 3–5. This while-loop is similar to Prim’s algorithm for minimum spanning tree. It ensures that every singleton cluster is connected, directly or indirectly, to some Steiner vertex or non-singleton cluster.

Algorithm 5

```

1:  $G' \leftarrow G/\mathcal{R}$ ; ▷ each vertex is either a shrunk cluster or a Steiner vertex
2: let  $H$  be the graph with  $V(H) = V(G')$  and  $E(H) = \emptyset$ ;
3: while there is a component  $C$  consisting of only singleton clusters do
4:   add into  $E(H)$  a minimum edge connecting  $C$  to another component;
5: end while
6: let  $Q$  be a priority queue containing  $E(G')$ ;
7: while  $|E(H)| < k - 1$  do
8:    $e \leftarrow \text{extract-min}(Q)$ ;
9:   if  $H + e$  has no cycles then
10:    add  $e$  into  $H$ ;
11:   end if
12: end while

```

Theorem 5 *When at least one cluster is not a singleton cluster, INTERCLUSTER can be solved in $O(n^2 \log n)$ time.*

Proof The entire algorithm for this case is similar except that we use Algorithm 5 instead of Algorithm 3 to find the $k - 1$ edges. Let β^* denote total cost of the edges found by Algorithm 5. Similar to Theorem 4, we need to show the necessity of β^* . The $k - 1$ edges are picked at steps 4 and 10. We shall show step 4 since step 10 is the same as in Algorithm 3. Suppose that e is the edge added at step 4 and at that moment C is the component consisting only singleton clusters but no Steiner vertices. Since e is an edge crossing the cut $(V(C), V(G') - V(C))$, there must be an optimal solution containing e . Note that initially every Steiner vertex is a component, and therefore in step 4 “Another component” includes the possibility of any Steiner vertex and any shrunk cluster not in C .

To complete the proof, we shall show that since no components contain only singleton clusters, at step 6 of Algorithm 4, we can always choose a non-singleton cluster. Suppose by contradiction that at some iteration we pick a pending Steiner vertex s but any other component contains no edges of a local tree. Let $t + 1$ be the number of components at this moment. As shown in the proof of the previous theorem, the number of pending Steiner vertices must be t . Observe that after step 4 each component without edges of a local tree contains exactly one pending Steiner vertex. So the number of pending Steiner vertices is at least $t + 1$, a contradiction. \square

7 The LOCALCLUSTER problem

In this section we show the NP-hardness of the LOCALCLUSTER problem. If each required vertex is itself a cluster, i.e., all clusters are singleton clusters, then LOCALCLUSTER degenerates to the original Steiner minimum tree problem and is therefore NP-hard by definition. In the following, we shall show that the NP-hardness remains even if there exists a non-singleton cluster, which is the same condition that INTERCLUSTER can be solved in polynomial time.

Theorem 6 LOCALCLUSTER is NP-hard even if there exists a non-singleton cluster and the local and inter-cluster topologies are given.

Proof We transform from the Hamiltonian path problem which asks if there exists a Hamiltonian path in an undirected graph and is NP-complete (Garey and Johnson 1979). Suppose that H is an instance of the Hamiltonian path problem. Let $V(H) = \{v_i \mid 1 \leq i \leq n\}$. We construct a metric graph $G = (V, E, c)$ as follows.

- $R = \{r_i \mid 1 \leq i \leq n\} \cup \{x, y\}$ is the required vertex set.
- $V(G) = R \cup S$, where $S = \{s_i \mid 1 \leq i \leq n\}$ is the Steiner vertex set.
- $R_0 = \{x, y\}$ and $R_i = \{r_i\}$ for $1 \leq i \leq n$.
- The edge costs are set to
 - $c(r_i, s_i) = 1$ for all i ;
 - $c(s_i, s_j) = 1$ if $(v_i, v_j) \in E(H)$;
 - $c(x, s_i) = c(y, s_i) = 1$ for all i ; and
 - all the other edge costs are 2.

First we note that all the edge costs are either one or two, and therefore G is a metric graph. Since there are $n + 1$ clusters, the inter-cluster cost is at least n for any clustered

Steiner tree. Meanwhile, inserting all s_i into the local tree of R_0 and connect v_i to s_i for all i gives us a clustered Steiner tree with inter-cluster cost n . To complete the proof, we claim that

there is a Hamiltonian path in H if and only if there is a solution T with $\alpha(T) = n + 1$ and $\beta(T) = n$.

Suppose, w.l.o.g., that (v_1, v_2, \dots, v_n) is a Hamiltonian path in H . We construct a solution T as follows. The local tree of R_0 is $(x, s_1, s_2, \dots, s_n, y)$, and $(r_i, s_i) \in E(T)$ for all i . It can be easily verified that $\alpha(T) = n + 1$ and $\beta(T) = n$.

Conversely suppose that T is a clustered Steiner tree with $\alpha(T) = n + 1$ and $\beta(T) = n$. First we note that there cannot be any Steiner vertex in the inter-cluster tree of T . Otherwise, the inter-cluster tree has more than $n + 1$ vertices and therefore has cost more than n . Let T_0 be the local tree of R_0 in T . Since all clusters other than R_0 are singleton clusters, all the Steiner vertices of T are in T_0 . For each r_i , there is only one edge (r_i, s_i) of cost one incident to r_i . Since $\alpha(T) = n + 1$, all s_i must be in T_0 . Since the topology of T_0 is limited to be an xy -path and $c(T_0) = n + 1$, we have that all the edges in T_0 are of cost one, and therefore the order of the internal vertices of T_0 corresponds to a Hamiltonian path in H . Finally, we note that the local and the inter-cluster topologies are fixed in the transformed instance. \square

8 Concluding remarks

In this paper, we show that the Steiner ratio for CLUSTER is lower and upper bounded by three and four, respectively. It is interesting to improve the gap between the two bounds. Open questions and possible future work also include the approximabilities of CLUSTER and LOCALCLUSTER. Finally we propose another variant of CLUSTER. In applications to network design, there may be two cost functions. An edge (u, v) in a local tree has cost $c(u, v)$ and costs $c'(u, v)$ if it is in the inter-cluster tree. Usually $c'(u, v) > c(u, v)$. Now the problem is to design a clustered Steiner tree with minimum total cost.

Acknowledgments This work was supported in part by NSC 100-2221-E-194-036-MY3 and NSC 101-2221-E-194-025-MY3 from the National Science Council, Taiwan, R.O.C.

References

- Agrawal A, Klein P, Ravi R (1995) When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J Comput* 24(3):440–456
- Bao X, Liu Z (2012) An improved approximation algorithm for the clustered traveling salesman problem. *Inform Process Lett* 112(23):908–910
- Byrka J, Grandoni F, Rothvoß T, Sanità L (2010) An improved LP-based approximation for Steiner tree. *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC10ACM, New York, USA*, pp 583–592
- Chen YH, Lu CL, Tang CY (2003) On the full and bottleneck full Steiner tree problems. In: Warnow T, Zhu B (eds) *Computing and Combinatorics, COCOON03*, vol 2697., *Lecture Notes in Computer Science* Springer, Berlin Heidelberg, pp 122–129

- Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to algorithms, 2nd edn. The MIT Press, Cambridge
- Ding W, Xue G (2014) Minimum diameter cost-constrained Steiner trees. *J Comb Optim* 27(1):32–48
- Drake DE, Hougardy S (2004) On approximation algorithms for the terminal Steiner tree problem. *Inform Process Lett* 89:15–18
- Garey MR, Graham RL, Johnson DS (1977) The complexity of computing Steiner minimal trees. *SIAM J Appl Math* 32(4):835–859
- Garey MR, Johnson DS (1977) The rectilinear Steiner tree problem is NP-Complete. *SIAM J Appl Math* 32(4):826–834
- Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman & Co, New York
- Garg N, Konjevod G, Ravi R (1998) A polylogarithmic approximation algorithm for the group Steiner tree problem. Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA98SIAM, Philadelphia, PA, USA, pp 253–259
- Guttmann-beck N, Hassin R, Khuller S, Raghavachari B (2000) Approximation algorithms with bounded performance guarantees for the clustered traveling salesman problem. *Algorithmica* 28:422–437
- Hsieh SY, Yang SC (2007) Approximating the selected-internal Steiner tree. *Theor Comput Sci* 381(1–3): 288–291
- Hsu TS, Tsai KH, Wang DW, Lee DT (2005) Two variations of the minimum Steiner problem. *J Comb Optim* 9(1):101–120
- Huang CW, Lee CW, Gao HM, Hsieh SY (2013) The internal Steiner tree problem: hardness and approximations. *J Complex* 29(1):27–43
- Karp R (1972) Reducibility among combinatorial problems. In: Miller R, Thatcher J (eds) Complexity of computer computations. Plenum Press, New York, pp 85–103
- Li X, Zou F, Huang Y, Kim D, Wu W (2010) A better constant-factor approximation for selected-internal Steiner minimum tree. *Algorithmica* 56(3):333–341
- Lin GH, Xue G (2002) On the terminal Steiner tree problem. *Inform Process Lett* 84(2):103–107
- Lu CL, Tang CY, Lee RCT (2003) The full Steiner tree problem. *Theor Comput Sci* 306:55–67
- Martinez F, de Pina J, Soares J (2007) Algorithms for terminal Steiner trees. *Theor Comput Sci* 389(1–2): 133–142
- Robins G, Zelikovsky A (2005) Tighter bounds for graph Steiner tree approximation. *SIAM J Discret Math* 19:122–134
- Wu BY, Chao KM (2004) Spanning trees and optimization problems. Chapman and Hall, Boca Raton
- Zelikovsky A (1993) An $11/6$ -approximation algorithm for the network Steiner problem. *Algorithmica* 9(5):463–470
- Zou F, Li X, Gao S, Wu W (2009) Node-weighted Steiner tree approximation in unit disk graphs. *J Comb Optim* 18(4):342–349