The Steiner ratio of the clustered Steiner tree problem is three

Bang Ye Wu, Chen-Wan Lin and Li-Hsuan Chen
Department of Computer Science and Information Engineering
National Chung Cheng University, ChiaYi, Taiwan 621,R.O.C.
bangye@ccu.edu.tw, {lcw101p,clh100p}@cs.ccu.edu.tw

Abstract

The Clustered Steiner tree problem is a variant of Steiner minimum tree problem. The required vertices are partitioned into clusters, and the subtrees spanning different clusters must be disjoint in a feasible clustered tree. In this paper we show that the Steiner ratio of the cluster Steiner tree problem is three, where the Steiner ratio is defined as the largest possible ratio of the minimal cost without using any Steiner vertex to the optimal cost.

1 Introduction

Over the years Steiner tree problems are extensively studied, it is widely used in the telecommunication networks, design of VLSI, or optimal networks routing, etc.

Given 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. 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 [9, 14]. On general metrics, the best approximation ratio ρ achieved in polynomial time is an important parameters for many graph problems. From the first non-trivial result 11/6 [21], it has been improved several times [6, 2]. The current best approximation ratio is 1.39 [2]. A large number of variants of the SMT problem have been studied, for example, the versions on the Euclidean metric [7] and the rectilinear metric [8], the Steiner forest problem [1], the group Steiner tree problem [10], the terminal Steiner tree problem [3, 5, 16, 17, 18], the internal-selected Steiner tree problem [11, 13, 15], and many others [4, 12, 22].

The *Clustered Steiner tree* (CluSteiner) problem was proposed in [20]. In addition to a metric graph $G = (V, E, c)$ and required vertex set R, we are also given a partition $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$ 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. 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. A formal definition will be given in Section 2. If there is only one cluster or each required vertex is itself a cluster, the problem degenerates to the original Steiner minimum tree problem.

When no Steiner vertices can be used, that is we want to find the *minimum clustered spanning tree*, the problem can be simply solved in polynomial time [20]. For an instance $(G, \mathcal{R})$, let $\mathrm{MCST}(G, \mathcal{R})$ and $\mathrm{CSMT}(G, \mathcal{R})$ denote the minimum costs of a clustered spanning tree and a clustered Steiner tree, respectively. It is interesting to know the largest ratio of

$$\frac{\mathrm{MCST}(G, \mathcal{R})}{\mathrm{CSMT}(G, \mathcal{R})}$$

among all possible instances. Analogous to the original Steiner minimum tree problem, we call the ratio "Steiner ratio" of the clustered Steiner tree problem. In [20], it was shown that

$$\frac{\mathrm{MCST}(G, \mathcal{R})}{\mathrm{CSMT}(G, \mathcal{R})} \leq 4$$

and there exist instances with ratio three. In this paper, we show that the Steiner ratio is three by giving an algorithm which transform a minimum clustered Steiner tree T into a cluster spanning tree with cost at most three times of T.

The rest of the paper is organized as follows. In Section 2, we give some notation, definitions and some properties used in this paper. In Section 3, we shown the Steiner ratio for CluSteiner. Finally some remarks are given in Section 4.

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. An edge between vertices u and v is denoted by (u, v), and its cost 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 mst(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. An undirected complete graph $G = (V, E, c)$ is a *metric graph* if

- $c(u, u) = 0$ for each $u \in V$;
- $c(u, v) \geq 0$ for all $u, v \in V$;
- $c(u, v) = c(v, u)$ for all $u, v \in V$; and
- $c(u, v) + c(v, x) \geq c(u, x)$ for all $u, v, x \in V$ (triangle inequality).

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 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 $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$ 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 $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$ is a partition of R. We also use $n = |V|$ and note that $|E| \in \Theta(n^2)$ since G is a complete graph.

A odd vertex is a vertex with odd degree, or otherwise an even vertex. An Eulerian path is a path traveling all the edges exactly once. A connected undirected graph has an Eulerian path if and only if there are exactly two odd vertices. An Eulerian cycle is an Eulerian path starting and ending at the same vertex. A connected undirected graph has an Eulerian cycle if and only if all vertices are even. A Hamiltonian path is a path traveling all the vertices exactly once.

For a graph H, *contraction* of $(u, v) \in E(H)$ replaces u, v with a new vertex s. For any other vertex w, the edge cost is set to $c(s, w) = \min\{c(u, w), c(v, w)\}$. For a subgraph S, contracting S in H means contracting all the edges $E(S)$ in H, and the resulting graph is denoted by H/S. For convenience, for a vertex set S, we also use H/S to denote the graph obtained by contracting all *vertices* in S even when $H[S]$ is disconnected. That is, we shrink S into a new vertex s and $c(s, w) = \min_{v \in S}\{c(v, w)\}$ for any vertex $w \notin S$. Let $G/\mathcal{R}$ denote the graph resulted from contracting all R_i for all $R_i \in \mathcal{R}$.

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).

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. Since a Steiner vertex with degree two in an inter-cluster tree is meaningless, the topology of an inter-cluster tree is itself.

The Steiner ratio for general metric spaces is bounded by 2. The inequality (1) is well-known, see for example [19].

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

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 $c(Y) = 2c(T) = 2\text{smt}(G, R)$. Traveling 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. Since MST is the cheapest way to connect R, we have that $\text{mst}(R) \leq c(Y)$ 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. For this case, the next lemma is shown in [20].

Lemma 1: The minimum clustered spanning tree problem can be solved in $O(n^2)$ time [20].

Figure 1 was given in [20], which shows the Steiner ratio for CluSteiner is at least three.

3 Steiner ratio of CluSteiner

Definition 3: Let T be a tree. For any $u, v \in V(T)$, the unique path between u and v on T is denoted by $T[u, v]$. When $u = v$, $T[u, v]$ is null path with only one vertex but no edges.

Lemma 2: Let Y be a tree on a metric graph G with edge length function c. For any $U \subseteq V(Y)$ and $x, y \in V(Y)$, there exists an xy-path Q on G such that $U \subseteq V(Q)$ and $c(Q) \leq 2c(Y) - c(Y[x, y])$.

Proof: Construct a multi-graph M by doubling all edges in Y except for the edges in the path $Y[x, y]$. We have that $c(M) = 2c(Y) - c(Y[x, y])$. Furthermore, x and y are odd vertices and all the other in $V(M)$ are even vertices. Therefore there exists an Eulerian xy-path P of M. Traveling along P and taking short-cuts between the first visiting of every vertex in U, we can obtain a path visiting each vertex in U exactly once. By the triangle inequality, the path length is at most $c(M)$. □

Let $\text{dia}(Y)$ denote the diameter of a tree Y.

Algorithm 1 Path-Partition

Input: a local Steiner tree T_i rooted at v.
Output: a set of disjoint paths on T_i containing all vertices of T_i.

1: $F = \emptyset$;
2: **for** each Steiner vertex p in $V(T_i)$ **do**
3: insert an arbitrary edge between p and its children into F;
4: **end for**
5: output the subgraph $(V(T_i), F)$.

In the Path-Partition of a local tree T_i, the spine of T_i is defined as the path containing the root of T_i.

Corollary 3: Let Y be a tree on a metric graph with edge length function c. If $U \subseteq V(Y)$, then $\text{mst}(U) \leq 2c(Y) - \text{dia}(Y)$.

Corollary 4: Let Y be a tree on a metric graph. For any $U \subseteq V(Y)$ and $x, y \in V(Y)$, we have that $\text{mst}(U) \leq 2c(Y) - c(Y[x, y])$.

Corollary 5: Let T_i be a local tree of an optimal clustered Steiner tree. For any $x, y \in V(T_i)$, we have that $\text{mst}(R_i) \leq 2c(T_i) - c(T_i[x, y])$. Particularly, $\text{mst}(R_i) \leq 2c(T_i) - c(S_i)$, where S_i is the spine of T_i in any path partition.

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., $\beta(T) = c(T/\mathcal{R})$. A *local edge* is an edge of a local tree and a *local Steiner vertex* is a Steiner vertex in a local tree. Note that a local Steiner vertex is incident to at least two local edges.

We define a procedure Path-Partition by Algorithm 1. An example is shown in Figure 2. The next lemma can be easily shown and the proof is omitted.

Lemma 6: Let $\mathcal{P} = (V(T_i), F)$ be the output of Path-Partition. Let $\mathcal{P} = \{P_0, P_1, \ldots\}$ be the collection of the connected components of $\mathcal{P}$. The following properties hold.

- Each connected component P_j is a path (possibly a null path).
- Each P_j contains exactly one required vertex, and the required vertex is one of its endpoints.

For a given clustered Steiner tree T, Algorithm 2 computes a clustered spanning tree T', i.e., without using any Steiner vertex. We shall show that $c(T') \leq 3c(T)$.

Lemma 7: In Algorithm 2, $\bigcup_j \text{mst}(U_j)/\mathcal{R}$ is an inter-cluster tree and all endpoints of the edges are required vertices. $\bigcup_i \text{mst}(R_i) \cup \bigcup_j \text{mst}(U_j)$ in Algorithm 2 is also a tree. Furthermore, $\sum_j \text{mst}(U_j) \leq 2\beta(T) + \alpha(T) + \sum_{1 \leq i \leq k} c(S_i)$, where S_i is the spine of the local tree T_i.

Proof: Each inter-cluster edge of T is in one component C_j after path-partitioning all local trees. Apparently $\bigcup_j C_j/\mathcal{R}$ is still the original inter-cluster tree of T. The clusters intersecting with U_j is exactly the same as C_j, and therefore $\bigcup_j \text{mst}(U_j)/\mathcal{R}$ is an inter-cluster tree. Since all U_j contain only required vertices, all endpoints of the edges in $\bigcup_j \text{mst}(U_j)$ are required vertices.

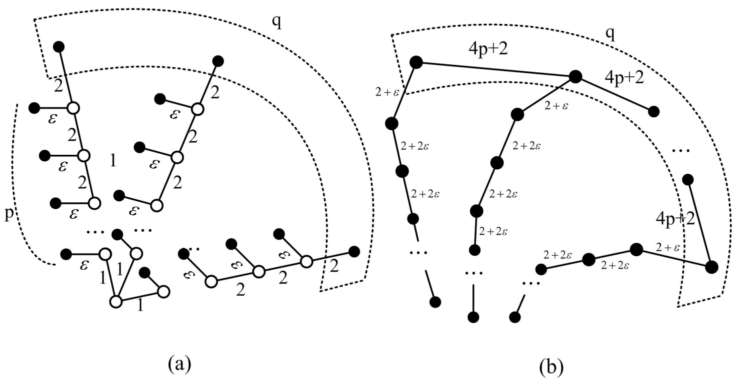

Figure 1: An example with Steiner ratio three. R_1 consists of the q required vertices circled by dotted line. As indicated, each path has p internal Steiner vertices. (a) The optimal solution. (b) The best one without any Steiner vertex. The optimal tree (a) has cost $q(p(2+\epsilon)+1) \approx 2pq+2q$. The right tree (b) is the best possible without Steiner vertex, and its cost is $(q-1)(4p+2)+qp(2+\epsilon)$. The ratio is asymptotically three when pq is large.

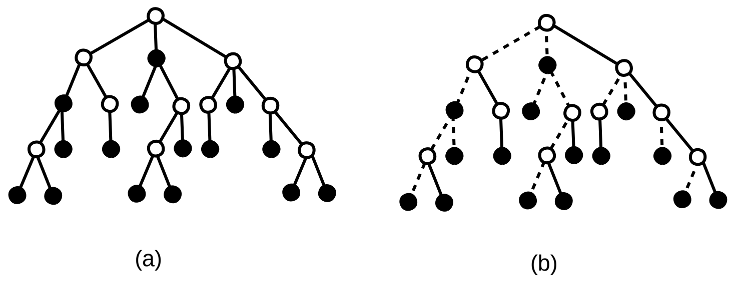

Figure 2: An example of path-partition algorithm. (a) A local Steiner tree. (b) An Output of the path-partition algorithm. Dotted lines indicate the deleted edges, and solid lines indicate the paths after applying path-partition algorithm.

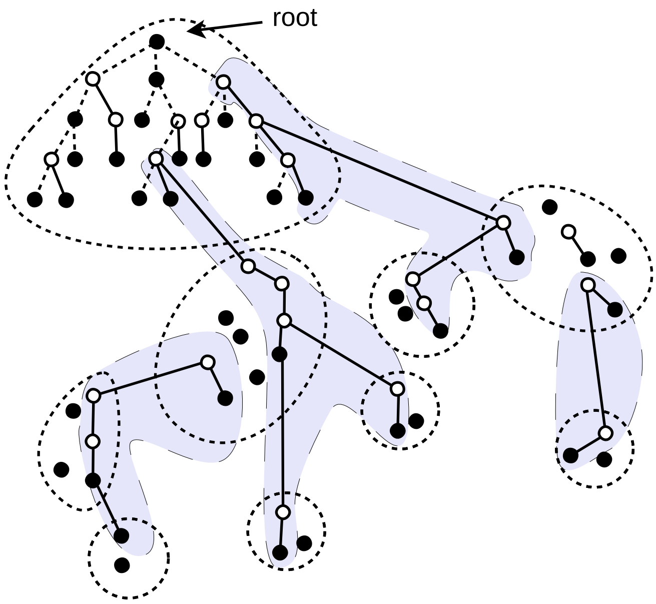

Figure 3: The clustered Steiner tree after applying the path-partition algorithm. The shaded areas indicate the connected components C.

Algorithm 2 CONSTRUCTION

Input: a clustered Steiner tree T.
Output: a clustered spanning tree T'.

root T at a required vertex r;
for all local tree T_i **do**
    call PATH-PARTITION(T_i)
end for
for all connected components C_j containing an inter-cluster edge **do**
    $U_j \leftarrow V(C_j) \cap R$;
    construct $\text{mst}(U_j)$;
end for
construct $\text{mst}(R_i)$ for all i.
output $\bigcup_i \text{mst}(R_i) \cup \bigcup_j \text{mst}(U_j)$.

Let I and L denote the sets of inter-cluster and local edges of T, respectively. For any C_j, let I_j be the set of inter-cluster edges in C_j and $D_j = \{h \mid R_h \cap C_j \neq \emptyset, h = 1, 2, ..., k\}$ be the set of indexes of clusters intersecting with C_j. Let $|D_j| = \eta_j$ and $f_j : [1, \eta_j] \mapsto D_j$ be an arbitrary labeling. By Lemma 6, the edge set of C_j can be written as

$$E(C_j) = I_j \cup \bigcup_{1 \leq h \leq \eta_j} E(P_{jh}), \tag{2}$$

where P_{jh} is a path in local tree $T_{f_j(h)}$ and these paths come from different local trees, i.e., f_j is bijection. Among these paths, there is exactly one path which is not a spine. We assume that P_{j1} is not a spine. By Lemma 2, we have that $\text{mst}(U_j) \leq 2(C_j) - c(P_{j1})$. By (2),

$$\begin{aligned} & \sum_j \text{mst}(U_j) \\ \leq\; & 2\sum_j \left(c(I_j) + \sum_{h=1}^{\eta_j} c(P_{jh}) \right) - \sum_j c(P_{j1}) \\ =\; & 2\sum_j c(I_j) + 2\sum_j \sum_{h=1}^{\eta_j} c(P_{jh}) - \sum_j c(P_{j1}) \\ =\; & 2\sum_j c(I_j) + \sum_j \sum_{h=1}^{\eta_j} c(P_{jh}) + \sum_j \sum_{h=2}^{\eta_j} c(P_{jh}) \end{aligned} \tag{3}$$

Since each inter-cluster edge is in exactly one component, we have

$$2\sum_j c(I_j) = 2c(I) = 2\beta(T). \tag{4}$$

Since the components C_j for all j are pairwise disjoint, the second term in (4) is exactly the total cost of the local edges in all components, i.e.,

$$\begin{aligned} & \sum_j \sum_{h=1}^{\eta_j} c(P_{jh}) \\ =\; & \sum_j c(E(C_j) \cap L) \end{aligned}$$

$$\leq \sum_{i=1}^{k} c(T_i) = \alpha(T). \quad (5)$$

Recall that the spine of a local tree is the path containing its root. For each j and $h > 1$, P_{jh} is a spine of some local tree. On the other hand, the spine of any local tree is in exactly one component. Therefore,

$$\sum_j \sum_{h=2}^{\eta_j} c(P_{jh}) = \sum_{i=1}^{k} c(S_i). \quad (6)$$

By (3–6), we have

$$\sum_j \mathrm{mst}(U_j) \leq 2\beta(T) + \alpha(T) + \sum_{i=1}^{k} c(S_i)$$

□

Theorem 8 : The Steiner ratio of clustered Steiner trees is three.

Proof: First, it was shown in [20] that the Steiner ratio is lower bounded by three. Let T be an optimal cluster Steiner tree. To complete the proof, we shall show that Algorithm 2 always outputs a clustered spanning tree with cost at most $3c(T)$.

By lemma 7, Algorithm 2 outputs a clustered spanning tree T' with $\beta(T') \leq 2\beta(T) + \alpha(T) + \sum_{1 \leq i \leq k} c(S_i)$, where S_i is the spine of the local tree T_i. The local trees of T' are minimum spanning trees of the clusters, and we have

$$\alpha(T') = \sum_{i=1}^{k} \mathrm{mst}(R_i).$$

By Corollary 5, $\mathrm{mst}(R_i) \leq 2c(T_i) - c(S_i)$, and therefore

$$\begin{aligned} & c(T') \\ = \; & \alpha(T') + \beta(T') \\ \leq \; & \sum_{1 \leq i \leq k} (2c(T_i) - c(S_i)) \\ & +2\beta(T) + \alpha(T) + \sum_{1 \leq i \leq k} c(S_i) \\ = \; & 2\beta(T) + 3\alpha(T) \leq 3c(T). \end{aligned}$$

□

4 Conclusion

In this paper, we show that the Steiner ratio for CluSteiner is three. The result also provides a 3-approximation for CluSteiner. How to improve the approximation ratio is an interesting future work.

References

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

[2] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. An improved LP-based approximation for Steiner tree. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, pages 583–592, New York, USA, 2010. ACM.

[3] Y.-H. Chen, C.-L. Lu, and C.-Y. Tang. On the full and bottleneck full Steiner tree problems. In *Computing and Combinatorics*, volume 2697 of *Lecture Notes in Computer Science*, pages 122–129. 2003.

[4] W. Ding and G. Xue. Minimum diameter cost-constrained Steiner trees. *J. Comb. Optim.*, 27(1):32–48, 2014.

[5] D. E. Drake and H. Stefan. On approximation algorithms for the terminal Steiner tree problem. *Inform. Process. Lett.*, pages 15–18, 2004.

[6] R. Gabriel and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM J. Discrete Math.*, 19:122–134, 2005.

[7] M. R. Garey, R. L. Graham, and D. S. Johnson. The complexity of computing Steiner minimal trees. *SIAM J. Appl. Math.*, 32(4):835–859, 1977.

[8] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-Complete. *SIAM J. Appl. Math.*, 32(4):826–834, June 1977.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1979.

[10] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '98, pages 253–259, Philadelphia, PA, USA, 1998. SIAM.

[11] S.-Y. Hsieh and S.-C. Yang. Approximating the selected-internal Steiner tree. *Theor. Comput. Sci.*, 381(1-3):288–291, 2007.

[12] T.-S. Hsu, K.-H. Tsai, D.-W. Wang, and D.-T. Lee. Two variations of the minimum Steiner problem. *J. Comb. Optim.*, 9(1):101–120, 2005.

[13] C.-W. Huang, C.-W. Lee, H.-M. Gao, and S.-Y. Hsieh. The internal Steiner tree problem: Hardness and approximations. *J. Complexity*, 29(1):27–43, 2013.

[14] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[15] X. Li, F. Zou, Y. Huang, D. Kim, and W. Wu. A better constant-factor approximation for selected-internal Steiner minimum tree. *Algorithmica*, 56(3):333–341, March 2010.

[16] G.-H. Lin and G. Xue. On the terminal Steiner tree problem. *Inform. Process. Lett.*, 84(2):103–107, 2002.

[17] C.-L. Lu, C.-Y. Tang, and R. C. T. Lee. The full Steiner tree problem. *Theoret. Comput. Sci.*, 306:55 – 67, 2003.

[18] F.V. Martinez, J.C. de Pina, and J. Soares. Algorithms for terminal Steiner trees. *Theoret. Comput. Sci.*, 389(1-2):133–142, 2007.

[19] B.-Y. Wu and K.-M. Chao. *Spanning Trees and Optimization Problems*. Chapman and Hall, 2004.

[20] BangYe Wu. On the clustered steiner tree problem. In *Combinatorial Optimization and Applications*, volume 8287 of *Lecture Notes in Computer Science*, pages 60–71. Springer Berlin Heidelberg, 2013.

[21] A.Z. Zelikovsky. An 11/6-approximation algorithm for the network Steiner problem. *Algorithmica*, 9(5):463–470, 1993.

[22] F. Zou, X.-Y. Li, S. Gao, and W. Wu. Node-weighted Steiner tree approximation in unit disk graphs. *J. Comb. Optim.*, 18(4):342–349, 2009.