Approximation Algorithms for Steiner Connected Dominating Set

Ya-Feng Wu (吴亚峰), Yin-Long Xu (许胤龙), and Guo-Liang Chen (陈国良)

National High Performance Computing Center at Hefei, Department of Computer Science and Technology University of Science and Technology of China, Hefei 230027, P.R. China

E-mail: yfwu@ustc.edu; {ylxu, glchen}@ustc.edu.cn

Received July 21, 2004; revised January 17, 2005.

Abstract Steiner connected dominating set (*SCDS*) is a generalization of the famous *connected dominating set* problem, where only a specified set of required vertices has to be dominated by a connected dominating set, and known to be NP-hard. This paper firstly modifies the *SCDS* algorithm of Guha and Khuller and achieves a worst case approximation ratio of $(2+1/(m-1))H(\min(\Delta,k))+O(1)$, which outperforms the previous best result $(c+1)H(\min(\Delta,k))+O(1)$ in the case of $m \geqslant 1+1/(c-1)$, where c is the best approximation ratio for Steiner tree, Δ is the maximum degree of the graph, k is the cardinality of the set of required vertices, m is an optional integer satisfying $0 \leqslant m \leqslant \min(\Delta,k)$ and H is the harmonic function. This paper also proposes another approximation algorithm which is based on a greedy approach. The second algorithm can establish a worst case approximation ratio of $2\ln(\min(\Delta,k))+O(1)$, which can also be improved to $2\ln k$ if the optimal solution is greater than $\frac{c\cdot e^{2c+1}}{2(c+1)}$.

Keywords approximation algorithm, Steiner connected dominated set, graph algorithm, NP-hard

1 Introduction

Connected dominating set (*CDS*) is a classical problem in combinatorial optimization, which is defined as the following: given a graph $G=(V,E)$, find the smallest subset S of vertices that induce a connected subgraph and each vertex in *V-S* is adjacent to at least one vertex in S. It is known to be NP-hard[1] and much effort has been devoted to the study of approximation algorithms for its solution[1−4]. A generalization of *CDS* is the *Steiner connected dominating set* (*SCDS*) problem where only a specified set $R \subseteq V$ of required vertices (terminals) has to be dominated by a connected dominating set. Considering that the specified set R can be V in some cases, *SCDS* is also known to be NP-hard[4]. While *CDS* is applied to support efficient broadcasting in wireless ad hoc networks[2,5,6], one of *SCDS* important applications is multicasting and anycasting in wireless ad hoc network. When all group members are viewed as terminals of R, only the vertices in *SCDS* are responsible for relaying multicast or anycast packets[7]. Therefore, how to construct the minimum *SCDS* is a key issue of energy-efficient multicast/anycast in wireless ad hoc networks.

Despite of a lot of approximation algorithms for *CDS*, few heuristics have been proposed for *SCDS*. One reason for the dearth of results on *SCDS* may be that it is harder than *CDS*. Indeed, Guha and Khuller proved that when vertices have weights, the *SCDS* problem is at least as hard as the notorious *TSP* problem on graphs for which no non-trivial approximation algorithm is known[4]. They also proposed the first non-trivial polynomial time approximation algorithm for the unweighted *SCDS* problem, with a worst case approximation ratio of $(c+1)H(\min(\Delta,k))+O(1)$, where c is the best approximation ratio for *Steiner tree* (currently $c=1.55$[8]), Δ is the maximum degree of the graph, k is the cardinality of the set of required vertices and H is the harmonic function.

This paper focuses on designing approximation algorithms for the *SCDS* problem. After analyzing the complexity of the problem, two approximation algorithms are proposed for the unweighted *SCDS* problem. One is a modification of Guha and Khuller's algorithm, and establishes an approximation ratio of $(2+1/(m-1))H(\min(\Delta,k))+O(m)$, which outperforms Guha and Khuller's by choosing $m \geqslant 1+1/(c-1)$. The analysis also shows that the algorithm has the same time complexity as the Guha and Khuller's algorithm. The other algorithm is based on a greedy approach with an approximation ratio of $2\ln(\min(\Delta,k))+O(1)$, and can be further improved to a worst case approximation ratio of $2\ln k$, if the optimal solution is greater than $\frac{c\cdot e^{2c+1}}{2(c+1)}$.

The rest of the paper is organized as follows. Section 2 gives some preliminary knowledge and definitions. Sections 3 and 4 present our first and second algorithms for *SCDS* with the analysis of approximation ratio, respectively. Finally, Section 5 concludes this paper.

2 Preliminaries

Given a graph $G=(V,E)$ and a set $R \subseteq V$ of required vertices (terminals), let $n=|V|$, $k=|R|$, Δ the maximum degree of G, $N(v)$ the set of neighbors of vertex v, and δ the size of the largest subset of R adjacent to one vertex in G ($\delta \leqslant \min(\Delta,k)$). For a subset V' of V, $G(V')$ is the subgraph of G induced by V'.

$G=(V,E)$ is a *unit disk graph*, if $E=\{(u,v)|$ the distance between u and v is at most one unit$\}$.

Steiner connected dominating set of R is a subset S of V that induces a connected subgraph and each vertex of *R-S* is adjacent to at least one vertex on S.

Short Paper

Supported by the National Natural Science Foundation of China under Grant No. 60173048 on "Research on Routing and Wavelength Assignment in WDM All-optical Networks".

For a given *SCDS* problem, we use $OPT(R)$ to denote the optimal solution of the problem, namely the minimum *Steiner connected dominating set* of terminals, and $OPT = |OPT(R)|$.

Note that in the case of $R = V$, *SCDS* is actually *CDS*. With the results about the complexity of *CDS* in [4], we have the following Theorem 1.

Theorem 1. *The SCDS problem is NP-hard in general graphs, even in UDGs. For general graphs, the approximation ratio* $H(\Delta)$ *will be hard to improve unless* $NP \subseteq DTIME[n^{O(\log \log n)}]$.

3 First Approximation Algorithm

This section presents our first algorithm *SCDS*1, which is a modification of Guha and Khuller's algorithm (*GKA*)[4]. We begin with an overview of *GKA,* presented in Fig.1. Step 1 and Step 2 apply a modified greedy *set cover* algorithm[9,p.977] to find a small dominating set C_1 of R, and then Step 4 chooses a terminal as the representative for each vertex of C_1. Finally, Step 5 finds a Steiner tree of all representative terminals to connect all vertices of C_1.

Algorithm *GKA*
Input: graph $G = (V, E)$ and a set $R \subseteq V$ of terminals.
Output: a Steiner connected dominating set of R.
Begin
1. For each $v \in V$, construct $S_v = \{u | u \in (R \cap N(v))\}$. Let $S = \{S_v | v \in V\}$.
2. Run the function Greedy-set cover, with R as the ground set and S as the set of subsets[9,p.977], until no subset contains more than one element. Let U denote the output of the function, and $C_1 = \{v | S_v \in U\}$.
3. Let $B' = \{u | u \in R$, and u is not covered by $U\}$.
4. For each $v \in C_1$, select a terminal $r(v) \in S_v$ as the representative of v. Let B denote the set of all representative terminals.
5. Apply a (edge weighted) Steiner tree algorithm for the set $(B \cup B')$, with all edges having unit weight. Let T denote the vertex-set of the output tree.
6. **Return** $C_1 \cup T$.

Fig.1. Algorithm *GKA*.

Note that after Step 4 of *GKA,* there are still some *extra* vertices in graphs, each of which connects two or more representative terminals of B. The algorithm directly connects the representative vertices with the Steiner tree at Step 5, ignoring these *extra* nodes. In some cases, these representative terminals are connected by two or more vertices in the Steiner tree T at Step 5, which leads to a large size of the Steiner tree as well as the output *SCDS.* On the other hand, adding the *extra* vertices into the dominating set and removing the corresponding dominated representative terminals of B will make the vertex set of the final Steiner tree smaller, and thus improve the approximation performance. With this consideration, we present an *SCDS* algorithm, a modification of *GKA*, in Fig.2.

Algorithm *SCDS*1 introduces new phases (from Step 5 to Step 13) to *GKA.* In each iteration from Step 7 to Step 12, a vertex u that connects more than m terminals is added into C_2. Then terminals dominated by u in B are added into $C(B)$ and removed from B except the representative $r(u)$, which is used to connect other terminals later.

Algorithm *SCDS*1
Input: graph $G = (V, E)$, a set $R \subseteq V$ of terminals, and an integer m, where $0 \leqslant m \leqslant \Delta$.
Output: a Steiner connected dominating set of R.
Begin
*SCDS*1 is the same as the *GKA* from Step 1 to Step 4.
5. $C_2 \leftarrow \emptyset$
6. $C(B) \leftarrow \emptyset$
7. **while** $\exists u \in V$ that satisfies $|N(u) \cap (B \cup B')| \geqslant m$
8. **do** $C_2 \leftarrow C_2 \cup \{u\}$
9. $B' \leftarrow B' - (N(u) \cap B')$
10. select a terminal $r(u) \in (N(u) \cap B)$ as the representative of u
11. $C(B) \leftarrow C(B) \cup (N(u) \cap B) - \{r(u)\}$
12. $B \leftarrow B - (N(u) \cap B) \cup \{r(u)\}$
13. Apply a (edge weighted) Steiner tree algorithm for the set $(B \cup B')$, with all edges having unit weight. Let T denote the vertex-set of the output tree.
14. **Return** $C_1 \cup C_2 \cup C(B) \cup T$

Fig.2. Algorithm *SCDS*1.

Lemma 1. *The output of SCDS*1 *is a Steiner connected dominating set of R.*

Proof. First, the output of the algorithm dominates all terminals of R. Each terminal of $R - B'$ is dominated by a vertex of C_1 at Step 2, and the terminals of B' are dominated by vertices of C_2 or vertices in tree T.

Next, we will show that the output induces a connected subgraph of G. Clearly, $G(T)$ is connected. For any $u \in C_2$, suppose $r(u)$ is the representative terminal of u selected by Step 10. If $r(u)$ remains in B at Step 13, there is an edge $(u, r(u))$ connecting u and $G(T)$. Otherwise $r(u)$ must be dominated by another vertex $u_1 \in C_2$, and removed from B. Similarly, if $r(u_1)$ remains in B at Step 13, there is a path $(u, r(u), u_1, r(u_1))$ connecting u and $G(T)$. Since C_2 is a finite set, there is a sequence of vertices $u_1, u_2, \ldots, u_i$, such that u_j dominates $r(u_{j-1})$ $(1 < j \leqslant i)$ and the representative terminal $r(u_i)$ of u_i remains in B at Step 13. So there is a path $(u, r(u), u_1, r(u_1), \ldots, u_i, r(u_i))$ connecting u and T. In addition, each terminal of $C(B)$ connects with a vertex of C_2. So $G(C_2 \cup C(B) \cup T)$ is connected. By Step 4, each vertex of C_1 has a connected representative terminal in B, and then connects with $G(C_2 \cup C(B) \cup T)$. Therefore $G(C_1 \cup C_2 \cup C(B) \cup T)$ is connected.

Above all, the output of *SCDS*1 is a *Steiner connected dominating set of R.* □

Theorem 2. *Given a graph G, a subset R of terminals and an integer m, where* $0 \leqslant m \leqslant \Delta$, *SCDS*1 *computes an SCDS of R, with an approximation ratio* $(2 + 1/(m-1))H(\delta) + c(m+1) - 2 + 1/(m-1)$.

Proof. From Step 1 to Step 4, it has been proved in [4] that $|C_1| = |B| = |U| \leqslant (H(\delta) - 1)OPT$ and $|B'| \leqslant OPT$. In each iteration from Step 7 to Step 12, one vertex is added into C_2, with at least $m-1$ terminals added into $C(B)$, so $(m-1)|C_2| \leqslant |C(B)|$. Obviously, all terminals of $C(B)$ come from B that is computed by Step 4, so we get

$$|C(B)| \leqslant |B| \leqslant (H(\delta) - 1)OPT$$

and

$$|C_2| \leqslant (H(\delta) - 1)OPT/(m-1).$$

In Step 13, since $OPT(R)$ cannot dominate any m terminals of $B \cup B'$ by one vertex, $|B \cup B'| \leqslant m \cdot OPT$. Considering that connected $OPT(R)$ dominates all terminals of $B \cup B'$, the spanning tree of $G(B \cup B' \cup OPT(R))$ is a Steiner tree of $B \cup B'$. Hence by Step 13, applying the edge weighted Steiner tree heuristic[8] with all edges having unit weight, *SCDS*1 finds a Steiner tree, the number of whose edges is no more than $c(|B \cup B' \cup OPT(R)| - 1)$ (c is the approximation ratio of the Steiner heuristics, the same as that in the following). Therefore

$$|T| \leqslant c[(m+1)OPT - 1] + 1 \leqslant c(m+1)OPT.$$

Above all, the size of *SCDS*1 output is no more than

$$[(2 + 1/(m-1))H(\delta) + c(m+1) - 2 + 1/(m-1)]OPT.$$

So, the theorem holds. □

GKA achieves an approximation ratio of $(c+1)H(\delta) + c - 1$, while *SCDS*1 can achieve a better approximation ratio in the case of $m \geqslant 1/(c-1) + 1$. On the other hand, *SCDS*1 only introduces to *GKA* the iteration from Step 7 to Step 12. The iteration executes no more than k/m times, hence the complexity of the iteration is $O(kn/m)$. So the time complexity of *SCDS*1 is still $O(kn^2)$, the same as *GKA*.

4 Second Approximation Algorithm

The *SCDS*1 algorithm modifies *GKA* and achieves an approximation ratio of $(2 + 1/(m-1))H(\delta) + O(1)$, better than $(c+1)H(\delta) + O(1)$ of *GKA* in the case of $m \geqslant 1/(c-1) + 1$. This section proposes our second *SCDS* algorithm, which further improves approximation performance of *SCDS*1. The algorithm runs in two phases. In the first phase, the algorithm iteratively picks the core (a core is a vertex that connects at least two terminals) which connects maximum terminals and selects one of its dominated terminals as a *connecting point* that waits to be dominated by another core, until no core exists. All cores and *connecting points* are added into the solution. In the second phase, the algorithm runs a Steiner tree algorithm to connect left terminals, with each edge having unit weight. The detail of *SCDS*2 appears in Fig.3. In the algorithm, the set S consists of all cores while the set C consists of all *connecting points*.

Lemma 2. *The output of SCDS2 is a Steiner connected dominating set of R.*

Proof. First, the output of the algorithm dominates all terminals of R. Each terminal of R-A is dominated by a vertex u of S at Step 2, and the terminals of A are dominated by vertices in tree T.

Next, we will show that the output induces a connected subgraph of G. Clearly, $G(T)$ is connected. For any $u \in S$, suppose $c(u)$ is the connecting point of u selected by Step 6. If $c(u)$ remains in A at Step 9, there is an edge $(u, c(u))$ connecting u and $G(T)$. Otherwise $c(u)$ must be dominated by another core $u_1 \in C_2$, and removed from R. Similarly, if $c(u_1)$ remains in A at Step 9, there is a path $(u, c(u), u_1, c(u_1))$ connecting u and $G(T)$. Since S is a finite set, there is a sequence of vertices $u_1, u_2, \ldots, u_i$, such that u_j dominates $c(u_{j-1})$ $(1 < j \leqslant i)$ and the connecting point $c(u_i)$ of u_i remains in A at Step 9. So there is a path $(u, c(u), u_1, c(u_1), \ldots, u_i, c(u_i))$ connecting u and T. So, any core in S connects with T in the subgraph induced by the solution. Also note that any connecting point connects with its core. So the output of *SCDS*2 is a Steiner connected dominating set of R. □

Algorithm *SCDS*2
Input: graph $G = (V, E)$ and a set $R \subseteq V$ of terminals.
Output: a Steiner connected dominating set of R.
Begin:
1. $C \leftarrow \emptyset$
2. $S \leftarrow \emptyset$
3. **while** $\exists u \in V$ that satisfies $|N(u) \cap R| \geqslant 2$
4. **do** select a vertex u that maximizes $|N(u) \cap R|$
5. $S \leftarrow S \cup \{u\}$
6. arbitrarily select a vertex $v \in (N(u) \cap R)$ as a *connecting point*
7. $C \leftarrow C \cup \{v\}$
8. $R \leftarrow R - (N(u) \cap R) \cup \{v\}$
9. $A \leftarrow R$
10. Apply a (edge weighted) Steiner tree algorithm for the set A, with all edges having unit weight. Let T denote the vertex-set of the output tree.
11. **Return** $S \cup C \cup T$

Fig.3. Algorithm *SCDS*2.

Theorem 3. *Algorithm SCDS2 finds a solution to the SCDS problem with an approximation ratio of* $2\ln\delta + c + 4$.

Proof. Clearly, A only consists of terminals left in R at Step 9. Considering that the connected $OPT(R)$ dominates all terminals of A, the spanning tree of $G(A \cup OPT(R))$ is a Steiner tree of A. So there is a Steiner tree of A with at most $|A| + OPT$ vertices. When a (edge weighted) Steiner tree algorithm with all edges having unit weight is applied by Step 9, *SCDS*2 computes a tree with at most $c(|A| + OPT - 1) + 1$ vertices.

Let r denote the number of iterations. Since at most one core and one connecting point are selected in each iteration, $|C| \leqslant |S| \leqslant r$. Hence the size of the final solution is at most $2r + c(|A| + OPT)$. Since no vertex connects two terminals of A, we get the bound of A, $|A| \leqslant OPT$.

We now give a bound on r. Let a_i denote the number of terminals left in R after the i-th iteration and initially $a_0 = k$. Since all vertices of $OPT(R)$ dominate all terminals, there must be a core that dominates at least $\lceil \frac{a_{i-1}}{OPT} \rceil$ terminals in the i-th iteration. So

$$a_i \leqslant a_{i-1} - \left\lceil \frac{a_{i-1}}{OPT} \right\rceil + 1 \leqslant a_{i-1}(1 - 1/OPT) + 1. \quad (1)$$

It can be concluded that

$$a_i \leqslant a_0(1 - 1/OPT)^i + \sum_{j=0}^{i-1}(1 - /OPT)^j$$

by summing up (1). When $i \geqslant OPT \cdot \ln(a_0/OPT)$, the first term $a_0(1 - 1/OPT)^i \leqslant OPT$. The second term is a geometric series that sums up to at most OPT. So there are at most 2 OPT vertices left in R after $OPT \cdot \ln(a_0/OPT)$ iterations. If more OPT iterations are executed to select cores, the number of terminals left in R will be less than OPT, because at least one terminal is removed from R in one iteration.

Suppose more a_f iterations are further executed before invoking the edge weighted Steiner tree algorithm, then the number of iterations r is at most $OPT \cdot \ln(a_0/OPT) + OPT + a_f$, and $A \leqslant OPT - a_f$. So the final solution has at most

$$2(OPT \cdot \ln(a_0/OPT) + OPT + a_f) + c(OPT - a_f OPT)$$

terminals. Since there are at most OPT terminals left in R after $OPT \ln(a_0/OPT) + OPT$ iterations, $a_f \leqslant OPT$. Since each vertex can be adjacent to at most δ terminals, $\delta \cdot OPT \geqslant a_0$. Finally, *SCDS*2 gets an approximation ratio of $2\ln\delta + c + 4$. □

We also give the following modification of *SCDS*2, which can achieve an approximation ratio of $2\ln k$ if the optimal solution is great enough. This modification is based on the idea of [10], which is used to solve the *node Steiner tree* problem.

The modification of *SCDS*2 is presented in Fig.4, in which $\lambda = 2c + 1$.

```
Algorithm Modification of SCDS2
Input: graph G = (V, E) and a set R ⊆ V of terminals.
Output: a Steiner connected dominating set of R.
Begin:
    C ← ∅
    S ← ∅
    i ← 0
    while ∃u ∈ V that satisfies |N(u) ∩ R| ⩾ 2) and
    |R| ⩾ i/(ln k−λ) + e^λ)
       do select a vertex u that maximizes |N(u) ∩ R|
          S ← S ∪ {u}
          arbitrarily select a vertex v ∈ (N(u) ∩ R) as the
          connecting point
          C ← C ∪ {v}
          R ← R − (N(u) ∩ R) ∪ {v}
          i ← i + 1
    A ← R
    Apply a (edge weighted) Steiner tree algorithm for the
    set A, with all edges having unit weight. Let T denote
    the vertex-set of the output tree.
    Return S ∪ C ∪ T
```

Fig.4. Modification of *SCDS*2.

The modified *SCDS*2 iteratively picks a core, until no core exists or the number of terminals left in R is less than $\frac{i}{\ln k - \lambda} + e^{\lambda}$ before the i-th iteration.

Theorem 4. *The modification of SCDS*2 *finds a solution to the SCDS problem with an approximation ratio of* $2\ln k$*, when the optimal solution is greater than* $\frac{c \cdot e^{2c+1}}{2(c+1)}$.

Proof. As before, let r denote the total number of iterations, a_i denote the number of terminals left in R after the i-th iteration and initially $a_0 = k$. The size of the final solution is at most $2r + c(|A| + OPT)$. We also have

$$a_i \leqslant a_0 \cdot (1 - 1/OPT)^i + \textstyle\sum_{j=0}^{i-1}(1 - 1/OPT)^j.$$

When $i \geqslant (\ln k - \lambda)OPT$, the first term $a_0 \cdot (1 - 1/OPT)^i \leqslant a_0 \cdot e^{\lambda - \ln k} \leqslant e^{\lambda}$ and the second term is at most OPT. So after $(\ln k - \lambda)OPT$ iterations the number of terminals $a_i \leqslant e^{\lambda} + OPT \leqslant \frac{i}{\ln k - \lambda} + e^{\lambda}$, which means that the iteration must be stopped after at most $(\ln k - \lambda)OPT$ iterations. This guarantees the bound of the total number of iterations,

$$r \leqslant (\ln k - \lambda)OPT.$$

If the iteration stops because no core exists, no vertex connects two terminals of A, so $|A| \leqslant OPT$. Else the iteration stops because the number of terminals is smaller than $\frac{i}{\ln k - \lambda} + e^{\lambda}$, then

$$|A| \leqslant \frac{1}{\ln k - \lambda} + e^{\lambda} \leqslant OPT + e^{\lambda}$$

because $r \leqslant (\ln k - \lambda) \cdot OPT$. So we get $|A| \leqslant OPT + e^{\lambda}$ in both cases. And the size of the final solution is at most

$$2 \cdot (\ln k - \lambda) \cdot OPT + c \cdot (OPT + e^{\lambda} + OPT) =$$
$$[2\ln k \cdot c \cdot e^{\lambda} - 2(\lambda - c)]OPT.$$

Since $\lambda = 2c + 1$, our solution gives at most $2\ln k \cdot OPT$ vertices in the case of $OPT \geqslant \frac{c \cdot e^{2c+1}}{2(c+1)}$. □

5 Conclusion

In this paper, we make an in-depth investigation on the *SCDS* problem, which is a generalization of *CDS* problem. After analyzing the complexity of the problem, we propose two approximation algorithms, both of which achieve better approximation performance than previous results.

For the problem in arbitrary graphs, our algorithms achieve an approximation ratio of $2\ln(\min(\Delta, k)) + O(1)$. Considering no ratio better than $H(\Delta)$ for *SCDS* exists unless $NP \subseteq DTIME[n^{O(\log\log n)}]$, our algorithms are with quite good performance. An open question is whether the problem can be approximated within a ratio less than $2\ln(\min(\Delta, k))$ or not?

References

[1] Garey M R, Johnson D S. Computers and Intractability. A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, 1979.

[2] Alzoubi K M, Wan P-J, Frieder O. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proc. 35th HICSS,* Hawaii, Jan. 7–10, 2002, pp.3849–3855.

[3] Jia L, Rajaraman R, Suel T. An efficient distributed algorithm for constructing small dominating sets. In *Proc. the Annual ACM Symposium on Principles of Distributed Computing,* 2001, pp.33–42.

[4] Guha S, Khuller S. Approximation algorithms for connected dominating sets. *Algorithmica*, April 1998, 20(4): 374–387.

[5] Wu J, Li H L. On calculating connected dominating set for networks. In *Proc. the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications,* 1999, pp.7–14.

[6] Bharghavan V, Das B. Routing in ad hoc networks using minimum connected dominating sets. In *Proc. IEEE Int. Conf. Communications'97,* Montreal, June 1997, pp.376–380.

[7] Wu Y-F, Xu Y-X, Chen G-L, Wang K. On the Construction of virtual multicast backbone for wireless ad hoc networks. In *Proc. the First IEEE Int. Conf. Mobile Ad Hoc and Sensor Systems,* Florida, USA, October, 2004, pp.294–303.

[8] Karpinsky M, Zelikovsky A. New approximation algorithms for the Steiner tree problem. Technical Report, Electronic Colloquium on Computational Complexity: TR95–030, 1995.

[9] Corman T, Leiserson C, Rivest R. Introduction to Algorithm. The MIT Press, 1989.

[10] Berman P, Ramaiyer V. Improved approximation algorithms for the Steiner tree problem. *J. Algorithm,* 1994, 17: 381–408.