Multi-rooted Greedy Approximation of Directed Steiner Trees with Applications*

Tomoya Hibi** and Toshihiro Fujito***

Department of Computer Science and Engineering
Toyohashi University of Technology
Tempaku, Toyohashi 441-8580 Japan
hibi@algo.cs.tut.ac.jp,
fujito@cs.tut.ac.jp

Abstract. We present a greedy algorithm for the directed Steiner tree problem (DST), where any tree rooted at any (uncovered) terminal can be a candidate for greedy choice. It will be shown that the algorithm, running in polynomial time for any constant l, outputs a directed Steiner tree of cost no larger than $2(l-1)(\ln n+1)$ times the cost of the minimum l-restricted Steiner tree. We derive from this result that 1) DST for a class of graphs, including quasi-bipartite graphs, in which the length of paths induced by Steiner vertices is bounded by some constant can be approximated within a factor of $O(\log n)$, and 2) the tree cover problem on directed graphs can also be approximated within a factor of $O(\log n)$.

1 Introduction

The Steiner tree (in graphs) problem is one of the most well-known combinatorial optimization problems with a long and rich history of being a subject for mathematical and computational studies. The problem is of fundamental importance especially in the areas of network design, network routing such as multicasting, and so on, where it is required to find a minimum cost tree, in a given edge-costed graph, spanning all the vertices specified as *terminals*. The problem is, however, one of the Karp's original *NP*-complete problems [9], and various approximation algorithms as well as heuristics have been developed for it. The case of undirected graphs has been and continues to be actively studied, and after the basic result of a factor 2 approximation by the minimum spanning tree based approach, the best approximation factors have been renewed several times [21,2,10,18], culminating with the recent breakthrough result with a performance ratio of $\ln(4) + \epsilon < 1.39$ [1]. It is *NP*-hard, on the other hand, to guarantee solutions of cost less than 96/95 times the optimal cost [3].

* Supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Science, Sports and Culture of Japan.

** Currently at NTT corporation.

*** Corresponding author. Also affiliated with Intelligent Sensing System Research Center, Toyohashi Univ. of Tech.

M.C. Golumbic et al. (Eds.): WG 2012, LNCS 7551, pp. 215–224, 2012.

The case of directed graphs in contrast has seen much less progress. The *directed Steiner tree problem* (DST), the main subject of the current paper, is to find a minimum cost subgraph T, given a directed graph $G = (V, A)$ with arc costs $c(a)$ ($\forall a \in A$), a root vertex $r \in V$, and a subset X of vertices called *terminals*, such that T contains a path starting at r and leading to every terminal, where the cost of a subgraph is defined to be the total cost of arcs in it. Those non-terminals ($\in V \setminus X$) are called *Steiner vertices*, and a (directed) tree in this paper is assumed to be the one in which every arc is directed away from its root towards a leaf. A *directed Steiner tree* ("*dst*" for short) is a tree spanning all the terminals, and DST is then equivalently defined to be the problem of computing a minimum cost dst rooted at r. The first nontrivial approximation algorithm for DST was developed by Charikar et al., achieving a performance ratio of $l^2(l-1)|X|^{1/l}$ in time $O(n^l|X|^{2l})$ for any $l > 1$ [4,5]. The algorithm thus approximates DST within a factor of $O(n^\epsilon)$ for any $\epsilon > 0$ in polynomial time and at the same time within a factor of $O(\log^3 |X|)$ in quasi-polynomial time of $n^{O(\log |X|)}$, raising a conjecture that a polylogarithmic approximation of DST might be possible. It has, in fact, been attempted to improve the Charikar et al.'s approximation bound [22,12,15], and without success, however, the $O(n^\epsilon)$ factor of Charikar et al. [4] remains as the best performance ratio known today, and the polylogarithmic approximability is still wide open for DST.

1.1 Greedy Approaches for Approximating DST

It is natural to consider DST to be a generalization of the set cover problem by representing the notion of coverage by "reachability" from the root. Here, any trees rooted at r are the subsets for covering elements and terminals are those elements to be covered. The greedy set cover algorithm repeatedly selects into a solution a most "cost-effective" subset until all the elements become covered. Here, the cost-effectiveness of a subset is measured by the ratio of its cost to the number of yet uncovered elements in it, and it is the *density* $d(T)$ of a tree T rooted at r in DST defined to be the ratio of its cost to the number of terminals in it not yet reachable from r, i.e., $d(T) = c(T)/(\#$ of terminals in T not reachable from $r)$. If it were possible to compute a rooted tree with minimum density in polynomial time, it would lead to an $O(\log n)$-approximation for DST as is for the set cover problem (or in more general, a factor of α approximation of the minimum density tree yields an $O(\alpha \log n)$ approximation of DST). It is hard to compute it exactly, however, and this is why all the greedy approaches for DST including ours have had to settle for trees with approximately lowest density in their greedy choices.

Definition 1. – *An* l-level *tree is a tree in which no leaf is more than l arcs away from the root.*
- *A Steiner tree in which all the terminals are at leaves (or at the root) is called a* full Steiner tree. *Any Steiner tree can be decomposed into arc-disjoint full Steiner trees (full components) by splitting all the non-leaf terminals, each of them into a leaf of one tree and a root of the other tree.*
- *A Steiner tree is* l-restricted *if every full component in it is an l-leveled tree.*

The algorithm of Charikar et al. uses an l-level tree (in the metric closure of an original graph) of which density is a factor of at most $l-1$ away from that of the minimum density l-level tree [4]. Zosin and Khuller show that a tree of density bounded by $D+1$ times the minimum density (of any tree rooted at r) is polynomially computable if $V-X$ induces a tree of depth D [22]. Either algorithm considers such trees rooted at a fixed root r only. Zelikovsky's algorithm, based on a different type of density function, considers any full Steiner tree, and it computes (and adopts) a tree of which density is a factor of at most $(2+\ln|X|)^{l-2}$ away from that of the minimum density l-restricted Steiner tree [20].

1.2 Our Approach and Contributions

The greedy algorithm designed in this paper iteratively chooses a full Steiner tree rooted at either r or any "uncovered" terminal in a way similar to Zelikovsky's [20]. It uses a density function different from Zelikovsky's and naturally from Charikar et al.'s and Zosin-Khuller's as the notion of coverage cannot be represented by "reachability from r" in our setting. A terminal in a full Steiner tree T rooted at either r or any terminal is considered "covered by T" if it is not the root of T, and the density $d(T)$ is then redefined to be the ratio of its cost to the number of yet uncovered non-root terminals in T. The main theorem (Theorem 1) of the paper states that this algorithm computes a dst in polynomial time of which cost is no larger than $2(l-1)(\ln|X|+1)$ times the cost of the minimum l-restricted Steiner tree. It is interesting to compare this with the Zelikovsky's [20] and Charikar et al.'s [4] algorithms; the former outputs a dst of cost no larger than $(\ln|X|+2)^{l-1}$ times the cost of the minimum l-restricted Steiner tree, whereas the latter outputs an l-level dst of cost at most $l(l-1)$ times the cost of the minimum l-level dst.

The main result described above does not lead to an improved performance ratio for DST per se, yet some new approximation results can be derived from it. One is the case of DST for a class of graphs G where Steiner vertices induce no path of length longer than l in G. A *quasi-bipartite* graph belongs to such a class with $l=0$, and DST is known to be hard to approximate better than $O(\log n)$ even when inputs are restricted to quasi-bipartite graphs. It can be shown from the main theorem that our greedy algorithm approximates DST for such a case within a factor of $O(\log|X|)$ for any constant l. When combined with the $\Omega(\log^{2-\epsilon} n)$ approximation hardness of DST on general graphs [8], this separates the approximability of DST between the cases of quasi-bipartite graphs and general graphs. It has been repeatedly observed, in case of undirected graphs, that the Steiner tree problem is easier to approximate on quasi-bipartite graphs than on general graphs since [17], and it is here proven to be true in case of directed graphs.

Another application of the main theorem presented is in approximation of the *Directed Tree Cover problem (DTC)*. It is required in DTC, given an arc-costed directed graph G and a root vertex r, to compute a minimum directed tree T rooted at r such that there exists a path in T from r to every arc in G. In case of undirected graphs, the tree cover problem is known to be approximable within a factor of 2, by a simple algorithm for the uniform costs [19] and by a not so simple one for general costs [7]. The approximability of DTC, on the other hand, has remained wide open as mentioned in [11] [1]. It will be shown, by reducing general DTC to DST on bipartite graphs with terminal-Steiner bipartition, that DTC can be approximated within a factor of $O(\log n)$, again matching the known approximation lower bound of $\Omega(\log n)$ for DTC.

2 Algorithm

Let $G = (V, A)$ be a directed graph with non-negative arc cost $c(a)$ for each arc $a \in A$, a node r designated as a root, and a set $X \subseteq V$ of terminals. The greedy algorithm presented below grows a subgraph P of G in sequence, initially consisting of r and all the terminals only (no arcs), by iteratively adding trees in G rooted either at the "real" root r or at some terminals not yet covered. Here any terminal becomes "covered" whenever a tree containing it as a "non-root" is added to P by the algorithm (or equivalently, we may contract such a tree into a single vertex). The algorithm repeats this as long as uncovered terminals remain in G, and eventually ends up with a subgraph of G composed of all the trees added with all the terminals covered by some of them. As it contains a path from r to every terminal in X, a dst spanning all the terminals can be found within it, and it will be output by the algorithm.

Algorithm 1: Multi-Rooted Greedy

Input: $G = (V, A)$, root $r \in V$, and terminal set $X \subseteq V$
Output: a Steiner tree rooted at r spanning all terminals in X

1 Initialize: $P = (X \cup \{r\}, \emptyset)$ and $C = \emptyset$;
2 **while** *there remain (uncovered) terminals in* $X \setminus C$ **do**
3   Compute a tree T of low density rooted at any vertex in $\{r\} \cup (X \setminus C)$;
4   Set $P = (V_P \cup V(T), A_P \cup A(T))$;
5   Letting u be the root of T, add all terminals in T but u to C by setting $C = C \cup (X(T) - u)$;
6   Reset $c(a) = 0$ for all $a \in A(T)$ and recompute the metric closure of G;
7 **end**
8 Compute and output any tree within P rooted at r spanning all terminals in X.

It remains to elaborate on how to compute a small density tree T in step 3, and we use the algorithm developed by Kortsarz, Peleg, and Charikar et al. [13,4], assuming that we are working with the metric closure of the current graph in what follows. Let $d^*_l(k, v, X)$ denote the minimum density of l-level trees T_l rooted at v containing any k terminals from X. It was shown that good approximation of the minimum density tree among l-level trees is possible when l is a constant [4]:

[1] In fact an $O(\log n)$-approximation of DTC was claimed in error [14].

Lemma 1 ([4]). *For any $v \in V, X \subseteq V, 1 \leq k \leq |X|,$ and $l \geq 2$, an l-level tree T_l can be found in time $O(n^l k^{2l-1})$ such that $d(T_l) \leq (l-1)d_l^*(k, v, X)$.*

Let $d_l^*(X)$ denote the minimum density of l-level trees T_l rooted at any vertex in $\{r\} \cup X$ containing any number of terminals from X. It follows immediately from Lemma 1, by running the algorithm used in it for each $1 \leq k \leq |X|$ and $v \in \{r\} \cup X$, that

Lemma 2. *For any $X \subseteq V$ and $l \geq 2$, an l-level tree T_l rooted at some vertex in $\{r\} \cup X$ can be found in time $O(n^l |X|^{2l+1})$ such that $d(T_l) \leq (l-1)d_l^*(X)$.*

In the next section the approximation performance of Algorithm 1 is analyzed assuming that trees T are computed in step 3 by the algorithm of Lemma 2.

3 Analysis

Definition 2. *Let $P = (V_P, A_P)$ be a subgraph of G and $X_P \subseteq X$. $(P, \{r\} \cup X_P)$ is called a* partial Steiner tree (PST) *for (G, r, X) if*

- *$\{r\} \cup X \subseteq V_P$, and*
- *every vertex in V_P is reachable within P from some vertex in $\{r\} \cup X_P$.*

Lemma 3. *Let $P = (V_P, A_P)$ and C be a subgraph of G and a set of covered terminals, respectively, computed at any iteration of the while-loop during the execution of Algorithm 1. Then, $(P, \{r\} \cup (X \setminus C))$ with $X_P = X \setminus C$ is a PST for (G, r, X).*

Proof. Initially, $P = (\{r\} \cup X, \emptyset)$ is clearly a PST for (G, r, X). Suppose at some iteration a tree T is added to P and let u be its root. Then, all the terminals in T but u become covered (and leave X_P), but all the vertices in T are reachable from u within T. So, any vertex reachable from those newly covered terminals before addition of T becomes reachable now from u after addition of T.

For any PST $(P = (V_P, A_P), \{r\} \cup X_P)$ for (G, r, X) every vertex $v \in V_P$ is reachable from some vertex in $\{r\} \cup X_P$ within P. If v is reachable from more than one vertex in $\{r\} \cup X_P$, choose v itself if $v \in \{r\} \cup X_P$, but choose any one of them otherwise, and denote it by $r(v)$. Then, V_P is partitioned into a family of disjoint subsets, each of them consisting of vertices $v \in V_P$ with a common representative vertex $r(v) \in \{r\} \cup X_P$, and the subset v belongs to is referred to as $V(r(v))$ for any $v \in V_P$.

Fix one l-restricted Steiner tree $T_{(l)}$ for (G, r, X). Since $X_P \subseteq X$, each vertex in X_P is contained in $T_{(l)}$. Denote by $s(v)$ for $v \in X_P$ the lowest ancestor of v within $T_{(l)}$ such that $r(v) \neq r(s(v))$ (thus, $V(r(v)) \cap V(r(s(v))) = \emptyset$). As $T_{(l)}$ is rooted at r and $v \neq r$, $s(v)$ exists for any $v \in X_P$. Consider the set of $s(v)$-v paths for all $v \in X_P$, and denote it by $\mathcal{T}_0$, i.e., $\mathcal{T}_0 = \{s(v)\text{-}v \text{ path} \mid v \in X_P\}$. The following properties of the paths in $\mathcal{T}_0$ can be verified by recalling the choice of $s(v)$ for $v \in X_P$, that all the paths in $\mathcal{T}_0$ are parts of tree $T_{(l)}$ and that $V(r(u)) \cap V(r(v)) = \emptyset$ for any $u, v \in X_P$ if $u \neq v$.

Lemma 4. *The paths in $\mathcal{T}_0$ possess the following properties:*

1. *On any $s(v)$-v path all the vertices but $s(v)$ come from $V(r(v)) \cup (V \setminus V_P)$.*
2. *Suppose two paths, $s(u)$-u and $s(v)$-v, in $\mathcal{T}_0$ overlap.*
   (a) *If $s(u) = s(v)$, $s(u)$-u and $s(v)$-v paths overlap only in their initial segments, i.e., the subpaths starting at $s(u) = s(v)$ followed by a sequence of vertices in $V \setminus V_P$ only.*
   (b) *If $s(u) \neq s(v)$,*
      i. *either $s(u)$ is a proper ancestor of $s(v)$ in $T_{(l)}$, or the other way around, and*
      ii. *if $s(u)$ is a proper ancestor of $s(v)$ (the other case is similar), u and $s(v)$ must belong to the same set $V(r(u)) = V(r(s(v)))$, and therefore, they can overlap only in the initial segment of $s(v)$-v path consisting of $s(v)$ and vertices in $V \setminus V_P$ only.*

For any $v \in X_P$, collect all the $s(u)$-u paths in $\mathcal{T}_0$ with $s(u) = s(v)$, and merge them into a single tree rooted at the common starting vertex $s(v)$ (if $s(u) \neq s(v)$ for any $u \in X_P - \{v\}$, $s(v)$-v path is such a tree by itself). Call a subtree of $T_{(l)}$ thus constructed from some paths in $\mathcal{T}_0$ and rooted at $s(v)$ as $s(v)$-*tree*, and denote by $\mathcal{T}_1$ the collection of all $s(v)$-trees.

Lemma 5. *$\mathcal{T}_1$ satisfies the following properties:*

1. *Every vertex in X_P occurs at a leaf of exactly one tree in $\mathcal{T}_1$.*
2. *No arc of $T_{(l)}$ occurs in more than two trees of $\mathcal{T}_1$.*
3. *For any tree with multiple leaves in $\mathcal{T}_1$, any branching occurs within the distance of $l-1$ from the root.*

Proof. 1. This is clear from the construction of $\mathcal{T}_0$ and $\mathcal{T}_1$.

2. Suppose an arc (y, z) of $T_{(l)}$ is shared by three trees, $s(v_1)$-, $s(v_2)$-, and $s(v_3)$-trees, from $\mathcal{T}_1$. Then, it must be the case that no two of $s(v_1), s(v_2),$ and $s(v_3)$ can coincide, and that all of $s(v_1), s(v_2),$ and $s(v_3)$ are ancestors of y in $T_{(l)}$. Then, $s(v_1)$-z, $s(v_2)$-z, and $s(v_3)$-z paths are all the initial segments of distinct three paths in $\mathcal{T}_0$, all of them lying on the r-z path of $T_{(l)}$. There is no way, however, that they can satisfy property 2(b)ii. of Lemma 4.
3. Recall that any paths running from the root to leaves in a tree of $\mathcal{T}_1$ come from $\mathcal{T}_0$, and hence, any two of them must satisfy property 2(a) of Lemma 4. Recall also that $T_{(l)}$ is an l-restricted tree, and hence, the length of a consecutive run of Steiner vertices on any path is bounded by $l-2$ in $T_{(l)}$. Therefore, any two paths starting at the same vertex must branch out within the distance of $l-1$ from the starting vertex.

Let us assume henceforth that PST $(P, \{r\} \cup X_P)$ for (G, r, X) is the one generated during the execution of Algorithm 1 (Lemma 3); i.e., $P = (V_P, A_P)$ and C are a subgraph of G and a set of covered terminals, respectively, computed at any iteration of the while-loop, and $X_P = X \setminus C$. For any $s(v)$-tree T in $\mathcal{T}_1$, we do the following operations:

1. When a path is followed from $s(v)$ to a leaf w, no branching occurs after passing the $(l-1)$st vertex u (property 3 in Lemma 5). As we are working with the metric closure of the current graph, there exists an arc (u, w) of cost no larger than that of the subpath running from u to w. So, replacing such a subpath by such an arc on any path leading to a leaf if it is longer than l, T becomes an l-level tree of no larger cost.
2. Recall that $s(v)$ is reachable from $r(s(v))$ within P, where every arc has a zero cost (due to step 6 of Algorithm 1). Hence, by connecting $r(s(v))$ directly to each child of $s(v)$ by an arc, the root of $s(v)$-tree can be replaced by $r(s(v))$ without increasing its cost nor its levels.

Let us denote by $\mathcal{T}_2$ the set of l-level trees resulting from applications of the operations above to the $s(v)$-trees in $\mathcal{T}_1$. The next lemma is a key to our main theorem and we prove it by examining properties of $\mathcal{T}_2$:

Lemma 6. *Let $T_{(l)}$ be any l-restricted Steiner tree rooted at r in G. Suppose $(P, \{r\} \cup X_P)$ is a PST for (G, r, X) generated by Algorithm 1 during its execution. Then, there exists an l-level tree T_l in G rooted at some vertex in $\{r\} \cup X_P$ such that $d(T_l) \leq 2c(T_{(l)})/|X_P|$.*

Proof. Consider $\mathcal{T}_2$. Each l-level tree in it is rooted at some vertex in $\{r\} \cup X_P$ (due to operation 2), and every vertex in X_P occurs at a leaf of some tree in $\mathcal{T}_2$ (Lemma 5.1). Therefore, all the terminals in X_P can be covered by using all the trees in $\mathcal{T}_2$. The total cost of trees in $\mathcal{T}_2$ is no larger than that of those trees in $\mathcal{T}_1$. The latter can be bounded by $2c(T_{(l)})$ because of Lemma 5.2. Therefore, those trees in $\mathcal{T}_2$ can jointly cover $|X_P|$ uncovered terminals, and it costs at most $2c(T_{(l)})$ to do so. Hence, there must be a tree T_l in $\mathcal{T}_2$ of density no larger than $2c(T_{(l)})/|X_P|$.

We are now ready to bound the cost of a dst output by Algorithm 1:

Theorem 1. *Let $\mathrm{OPT}_{(l)}$ denote the cost of the minimum l-restricted Steiner tree for (G, r, X). Algorithm 1 computes a dst of cost no larger than $2(l-1)H(|X|)\mathrm{OPT}_{(l)}$, in time $O(n^l|X|^{2l+2})$, where $H(k)$ is the kth harmonic number and $H(k) = 1 + 1/2 + \cdots + 1/k$.*

Proof. The running time is dominated by that consumed in step 3, which is executed in total $O(|X|)$ times.

Suppose T is the tree computed in step 3 at any iteration of the while-loop. Assign $d(T)$ to each of the terminals newly covered by T. Total value assigned in one iteration of the while-loop coincides with the cost of T chosen during the iteration by definition of density $d(T)$. By doing this at every iteration, each terminal in X gets assigned with some density exactly once, and hence, total cost of trees chosen by Algorithm 1 can be recovered by collecting all the density values assigned to the terminals in X.

The density $d(T)$ of T can be bounded by $2(l-1)\mathrm{OPT}_{(l)}/|X \setminus C|$ according to Lemmas 2 and 6. Order the terminals in X in the order of becoming covered by Algorithm 1, and let x_i be the ith terminal covered by Algorithm 1 for

$1 \le i \le |X|$. As there remain at least $|X|-(i-1)$ uncovered terminals when x_i is covered, the density x_i receives is no larger than $2(l-1)\mathrm{OPT}_{(l)}/(|X|-(i-1))$. Therefore, the total density assigned to all the terminals in X is bounded by

$$\sum_{i=1}^{|X|} \frac{2(l-1)\mathrm{OPT}_{(l)}}{|X|-(i-1)} = 2(l-1)\mathrm{OPT}_{(l)} \sum_{i=1}^{|X|} \frac{1}{i} = 2(l-1)\mathrm{OPT}_{(l)} H(|X|).$$

The output tree is a subgraph of PST P, of which cost is bounded as above, and the claim follows.

4 Applications

A graph $G = (V, A)$ is called *quasi-bipartite* (with respect to terminal set X) when the set of Steiner vertices ($= V \setminus X$) induces no arc in G. It is easy to confirm the following corollary of Theorem 1 by observing that every Steiner tree is $(l+2)$-restricted in such special inputs as given below:

Corollary 1. *When inputs $(G = (V, A), r, X)$ are limited to those in which $V \setminus X$ induces no path of length longer than l, Algorithm 1 approximates DST within a factor of $2(l+1)H(|X|) = O(l \log |X|)$, running in polynomial time for any constant l. In particular, when inputs are restricted to quasi-bipartite graphs, DST can be approximated within a factor of $2H(|X|) \le 2 \ln |X| + 2$.*

The set cover problem can be embedded in DST on bipartite graphs $G = (X \cup (V \setminus X), A)$. Because of $\Omega(\log n)$ lower bound for set cover approximation [16,6], it can be said that Algorithm 1 yields an optimal approximation for such special cases as given in Corollary 1 for any constant l.

Let us turn our attention to the directed tree cover problem (DTC). The set cover problem can be embedded in DTC by the almost same construction as in DST, and hence, the same approximation hardness of $\Omega(\log n)$ lower bound holds. For the upper bound, we use the following reduction:

Lemma 7. *DTC on general graphs is reducible to DST on bipartite graphs with terminal-Steiner bipartition in an approximation preserving manner.*

Proof. Let $(G = (V, A), r, c)$ be an instance of DTC. For each arc $a = (u, v) \in A$, introduce a new vertex x_a as a terminal for DST. Each arc $a = (u, v) \in A$ is replaced by three arcs, $(u, x_a), (x_a, v),$ and (v, x_a), and the costs of these arcs are set equal to $0, c(a),$ and 0, respectively. An instance $(G' = (V', A'), r, X, c')$ of DST is constructed from a DTC instance (G, r, c) in this way such that $X = \{x_a \mid a \in A\}, V' = V \cup X, A' = \{(u, x_a), (x_a, v), (v, x_a) \mid a = (u, v) \in A\},$ and $\forall a \in A, c'(u, x_a) = c'(v, x_a) = 0, c'(x_a, v) = c(a)$. It is not hard to verify that a tree cover of any cost exists in (G, r, c) if and only if a dst of the same cost exits in (G', r, X, c'). It is also clear that $G' = (V \cup X, A')$ constructed from $G = (V, A)$ in the reduction is a bipartite graph for any G.

Due to this lemma, the following optimal approximation for DTC follows from Theorem 1 as in Corollary 1:

Theorem 2. *DTC can be approximated by Algorithm 1 within a factor of* $2H(|A|) \le 2\ln|A| + 2$.

References

1. Byrka, J., Grandoni, F., Rothvoß, T., Sanità, L.: An improved LP-based approximation for Steiner tree. In: Proc. 42nd STOC, pp. 583–592 (2010)
2. Berman, P., Ramaiyer, V.: Improved approximations for the Steiner tree problem. In: Proc. 3rd SODA, pp. 325–334 (1992)
3. Chlebík, M., Chlebíková, J.: The Steiner tree problem on graphs: Inapproximability results. Theory Comput. Syst. 406(3), 207–214 (2008)
4. Charikar, M., Chekuri, C., Cheung, T., Dai, Z., Goel, A., Guha, S., Li, M.: Approximation algorithms for directed Steiner tree problems. J. Algorithms 33, 73–91 (1999)
5. Calinescu, G., Zelikovsky, A.: The polymatroid Steiner problems. J. Comb. Opt. 9, 281–294 (2005)
6. Feige, U.: A threshold of $\ln n$ for approximating set cover. J. ACM 45(4), 634–652 (1998)
7. Fujito, T.: How to Trim an MST: A 2-Approximation Algorithm for Minimum Cost Tree Cover. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 431–442. Springer, Heidelberg (2006)
8. Halperin, E., Krauthgamer, R.: Polylogarithmic inapproximability. In: Proc. 35th STOC, pp. 585–594 (2003)
9. Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of Computer Computations, pp. 85–103. Plenum Press, New York (1972)
10. Karpinski, M., Zelikovsky, A.Z.: New approximation algorithms for the Steiner tree problem. J. Comb. Opt. 1, 47–65 (1997)
11. Könemann, J., Konjevod, G., Parekh, O., Sinha, A.: Improved Approximations for Tour and Tree Covers. In: Jansen, K., Khuller, S. (eds.) APPROX 2000. LNCS, vol. 1913, pp. 184–193. Springer, Heidelberg (2000)
12. Konjevod, G.: Directed Steiner trees, linear programs and randomized rounding, 8 pages (2005) (manuscript)
13. Kortsarz, G., Peleg, D.: Approximating the weight of shallow Steiner trees. Discrete Applied Mathematics 93, 265–285 (1999)
14. Nguyen, V.H.: Approximation Algorithm for the Minimum Directed Tree Cover. In: Wu, W., Daescu, O. (eds.) COCOA 2010, Part II. LNCS, vol. 6509, pp. 144–159. Springer, Heidelberg (2010)
15. Rothvoß, T.: Directed Steiner tree and the Lasserre hierarchy. ArXiv e-prints (November 2011)
16. Raz, R., Safra, S.: A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In: Proc. 29th STOC, pp. 475–484 (1997)
17. Rajagopalan, S., Vazirani, V.V.: On the bidirected cut relaxation for the metric Steiner tree problem. In: Proc. 10th SODA, pp. 742–751 (1999)
18. Robins, G., Zelikovsky, A.: Tighter bounds for graph Steiner tree approximation. SIAM J. Discrete Math. 19, 122–134 (2005)

19. Savage, C.: Depth-first search and the vertex cover problem. Inform. Process. Lett. 14(5), 233–235 (1982)
20. Zelikovsky, A.: A series of approximation algorithms for the acyclic directed Steiner tree problem. Algorithmica 18, 99–110 (1997)
21. Zelikovsky, A.: An 11/6-approximation algorithm for the network Steiner problem. Algorithmica 9, 463–470 (1993)
22. Zosin, L., Khuller, S.: On directed Steiner trees. In: Proc. 13th SODA, pp. 59–63 (2002)