Optim Lett
DOI 10.1007/s11590-010-0194-x

ORIGINAL PAPER

Approximations for node-weighted Steiner tree in unit disk graphs

X. Xu · Y. Wang · H. Du · P.-J. Wan · F. Zou · X. Li · W. Wu

Received: 20 June 2009 / Accepted: 12 April 2010

Abstract Given a node-weighted connected graph and a subset of terminals, the problem node-weighted Steiner tree (NWST) seeks a lightest tree connecting a given set of terminals in a node-weighted graph. While NWST in general graphs are as hard as Set Cover, NWST restricted to unit-disk graphs (UDGs) admits

X. Xu, H. Du, P.-J. Wan were supported in part by NSF under grant CNS-0831831.
Y. Wang was supported in part by the National Basic Research Program of China Grants 2007CB807900 and 2007CB807901, NSFC under Grant 60604033, and the Hi-Tech Research Development Program of China Grant 2006AA10Z216.

X. Xu · H. Du · P.-J. Wan (✉)
Department of Computer Science, Illinois Institute of Technology, Chicago, USA
e-mail: wan@cs.iit.edu

X. Xu
e-mail: xxu23@iit.edu

H. Du
e-mail: hongwei.ddu@gmail.com

Y. Wang
Institute of Theoretical Computer Science, Tsinghua University, Beijing, People's Republic of China
e-mail: wangyuexuan@tsinghua.edu.cn

F. Zou · W. Wu
Department of Computer Science, University of Texas at Dallas, Richardson, USA
e-mail: phenix.zou@student.utdallas.edu

W. Wu
e-mail: weiliwu@utdallas.edu

X. Li
School of Mathematics and Statistics, Lanzhou University, Lanzhou, People's Republic of China
e-mail: lixianyue@lzu.edu.cn

Published online: 05 May 2010

constant-approximations. Recently, Zou et al. (Lecture notes in computer science, vol 5165, COCOA, 2008, pp 278–285) showed that any μ-approximation algorithm for the classical edge-weighted Steiner tree problem can be used to produce 2.5 μ-approximation algorithm for NWST restricted to UDGs. With the best known approximation bound 1.55 for the classical Steiner tree problem, they obtained an approximation bound 3.875 for NWST restricted to UDGs. In this paper, we present three approximation algorithms for NWST restricted to UDGs, the k-Restricted Relative Greedy Algorithm whose approximation bound converges to $1 + \ln 5 \approx 2.$ 61 as $k \to \infty$, the 3-Restricted Greedy Algorithm with approximation bound $4\frac{1}{3}$, and the k-Restricted Variable Metric Algorithm whose approximation bound converges to 3.9334 as $k \to \infty$.

Keywords Node-weighted Steiner tree · Unit-disk graph · Approximation algorithm

1 Introduction

Let $G = (V, E; c)$ be a node-weighted connected graph with $c \in \mathbb{R}_+^V$. The weight of any subgraph H of G, denoted by $c(H)$, is the total weight of all nodes in H. The weight of any subset U of nodes, denoted by $c(U)$, is the total weight of all nodes in H. Given a subset S of nodes in G which are referred to as *terminals*, the problem node-weighted Steiner tree (NWST) is to find a tree of minimum weight in G which interconnects all terminals. Alternatively, the objective of NWST is to seek a subset $U \subseteq V \setminus S$ of minimum weight satisfying that the subgraph of G induced by $S \cup U$ is connected. Without loss of generality, we assume that all terminals have zero weight. Under this assumption, the weight of a minimum-weighted tree in G which interconnects all terminals is equal to the weight of a minimum-weighted subset $U \subseteq V \setminus S$ satisfying that the subgraph of G induced by $S \cup U$ is connected, and such minimum weight is denoted by $smt(S)$.

NWST in general graphs is at least as hard as Set Cover (see ref. in [5]). Consequently, it admits no polynomial $o(\ln |S|)$-approximation unless $P = NP$. Klein and Ravi [5] presented an polynomial $2 \ln |S|$-approximation algorithm, and Guha and Khuller [4] made an further improvement to achieve an approximation bound $(1.35 + \varepsilon) \ln |S|$ for any fixed positive ε. On the other hand, NWST restricted to unit-disk graphs admits polynomial constant-approximations. A graph $G = (V, E)$ is a unit-disk graph (UDG) if all nodes in V lie in a plane and there is an edge in E between two nodes u and v in V if and only if the Euclidean distance between u and v is at most one. Zou et al. [11] presented an reduction from NWST restricted to UDGs to the classical (edge-weighted) Steiner tree problem, and showed that any μ-approximation for the classical Steiner tree problem gives rise to a 2.5μ-approximation algorithm for NWST restricted to UDGs. With the best known approximation bound 1.55 due to Robins and Zelikovsky [7] for the classical Steiner tree problem, they obtained an approximation bound 3.875 for NWST restricted to UDGs.

In this paper, we present the three approximation algorithms for NWST restricted to UDGs adapted from the algorithms in [2,9,10] for the classical Steiner tree problem, and derive their approximation bounds. The k-Restricted Relative Greedy Algorithm has approximation bound converging to $1+\ln 5 \approx 2.61$ as $k \to \infty$. The 3-Restricted Greedy Algorithm has approximation bound $4\frac{1}{3}$. The k-Restricted Variable Metric Algorithm has approximation bound converging to 3.9334 as $k \to \infty$.

We conclude this section with some standard notations and definitions. A tree in G without non-terminal leaf is referred to as a *Steiner tree*. A tree in G is said to be *full* if all its leaves are terminals and none of its internal nodes is a terminal. Note that any Steiner tree ST can be decomposed into full subtrees at terminals with degree greater than one, and these full subtrees decomposed from ST are called as the *full components* of ST. A full tree is said to be *k-restricted* for some integer $k \geq 2$ if it contains at most k terminals. The *length* of a simple path in G is the sum of the weights of all its internal nodes. The *distance* between two nodes in G is the length of the shortest path between them, and can be computed in polynomial time. The *distance graph* of G is the complete graph K on V in which the length of each edge uv is the distance between u and v in G. Consider an undirected graph H. The subgraph of H induced by a subset of vertices U is denoted by $H[U]$. If H is edge-weighted, we use $MST(H)$ to denote a minimum-weighted spanning tree of H, and use $mst(H)$ to denote the weight of $MST(H)$. A set of nodes in H are said to be independent if there are mutually non-adjacent in H. A node of H is said to be a *relay* node of H if its degree in H is equal to 2, or a *branching* node of T if its degree is at least three. The *contraction* of a relay node v in H is the operation which removes v from H and then add an edge between the two neighbors of v in H.

2 *k*-Restricted Steiner cover

The *cost* of a subset τ of at least two terminals, denoted by $cost(\tau)$, is the weight of a minimum-weighted subset $U \subseteq V \backslash \tau$ satisfying that $G[\tau \cup U]$ is connected. In particular, $cost(S) = smt(S)$. A collection $\mathcal{C}$ of subsets $\tau_1, \tau_2, \ldots, \tau_l$ of S is said to be a *Steiner cover* of S if $|\tau_i| \geq 2$ for $1 \leq l \leq 2$ and the union of the l complete graphs on $\tau_1, \tau_2, \ldots, \tau_l$, respectively, is a connected graph spanning S. The *cost* of $\mathcal{C}$ is defined as $cost(\mathcal{C}) = \sum_{i=1}^{l} cost(\tau_i)$. $\mathcal{C}$ is said to be *k-restricted* if $2 \leq |\tau_i| \leq k$ for $1 \leq l \leq 2$. The cost of a minimum-cost k-restricted Steiner cover of S is denoted by $smt_k(S)$. A collection $\mathcal{T}$ of full trees $T_1, T_2, \ldots, T_l$ is said to be a *full-tree cover* of S if terminal sets of $T_1, T_2, \ldots, T_l$ form a Steiner cover of S. Each tree T_i is called a full component of $\mathcal{T}$. The *weight* of $\mathcal{T}$ is defined as $c(\mathcal{T}) = \sum_{i=1}^{l} c(T_i)$. $\mathcal{T}$ is said to be *k-restricted* if each T_i is k-restricted.

For any integer $k \geq 2$, denote

$$\rho_k = 1 + \frac{4}{\lfloor \log_2 k \rfloor - 1 + k/2^{\lfloor \log_2 k \rfloor}}.$$

Clearly, ρ_k converges to one as $k \to \infty$. However, the rate of the convergence is slow. Table 1 lists some values of ρ_k.

Lemma 2.1 *Suppose that T is a full tree in G and τ is the set of terminals in T. If the maximum degree of T is at most 5, then T contains a k-restricted full-tree cover of τ whose weight is at most $\rho_k c(T)$.*

Table 1 Some values of the Steiner ratio ρ_k

k	2	3	4	5	8	16	32	64	2^{40}	2^{400}
ρ_k	5	$3\frac{2}{3}$	3	$2\frac{7}{9}$	$2\frac{1}{3}$	2	1.75	$1\frac{2}{3}$	1.1	1.01

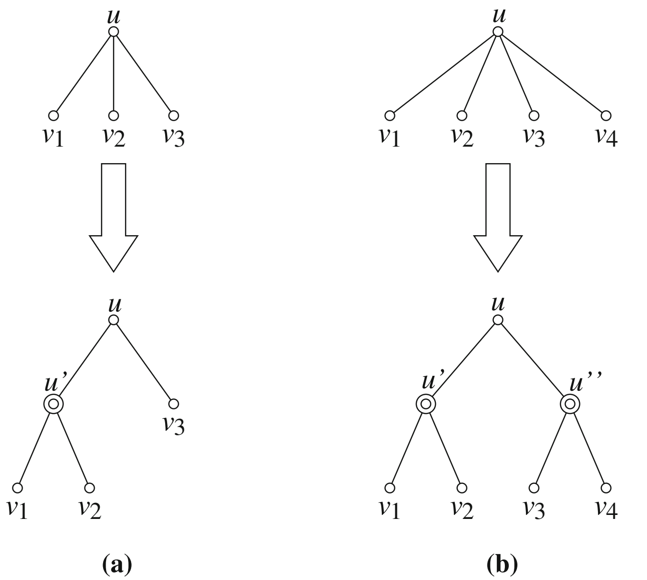

Fig. 1 Local snapshot of the construction of T^+ from T

Proof The lemma holds trivially if $|\tau| \leq k$. So we assume that $|\tau| > k$. We choose a root in the middle of an edge to convert T into a rooted tree. For each internal node u with three children v_1, v_2 and v_3, we introduce a duplicate u' of u as the new parent of v_1 and v_2 and as a new child of u (see Fig. 1a). For each internal node u with four children v_1, v_2, v_3 and v_4, we introduce two duplicates u' and u'' of u. The duplicate u' becomes the new parent of v_1 and v_2, the duplicate u'' becomes the new parent of v_3 and v_4, and both of them become the new children of u (see Fig. 1b). The resulting tree, denoted by T^+, is referred to as the *expansion* of T. Note that T^+ is a binary tree, and a node other than the root is a relay node in T^+ if and only it is a relay node in T.

Now, we construct a mapping from all the branching nodes to the terminals (i.e., leaves), and a labelling of the root and all branching nodes in T^+. We contract all the relaying nodes except the root. The resulting tree, denoted by T^-, is a regular binary tree. The set of internal nodes of T^- consists of exactly the root and all the branching nodes of T^+. Let f be the one-to-one mapping from internal vertices to leaves in T^- given in [1]. It has the following two properties: (1) for any internal vertex v, $f(v)$ is a descendent of v; (2) all paths $P^-(v)$ from v to $f(v)$ are edge-disjoint. For each branching node v, let $P^+(v)$ denote the tree path from v to $f(v)$ in T^+. Then, all paths $P^+(v)$ for all branching nodes of T^+ are edge-disjoint. It's easy to show that these paths are also internally node-disjoint.

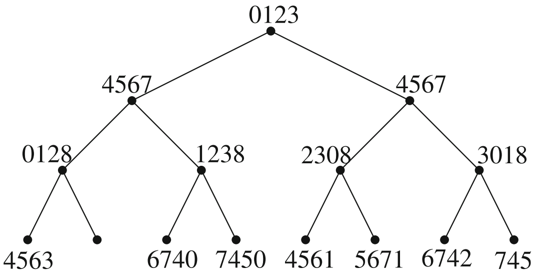

Fig. 2 Labeling a binary tree when $k = 5$

Let $r = \lfloor \log_2 k \rfloor$ and $s = k - 2^{\lfloor \log_2 k \rfloor}$. Then, $k = 2^r + s, 0 \leq s < 2^r$, and

$$\rho_k = 1 + \frac{2^{r+2}}{r2^r + s}.$$

We label the internal nodes of T^- inductively with sets of size exactly 2^r chosen from the first $r2^r + s$ non-negative integers as in [1]. The labelling of a node is determined inductively by the labelling of its r immediate ancestors. All internal vertices at level i, for $0 \leq i \leq r - 1$, are labeled with the set of 2^r numbers $\{i2^r, i2^r + 1, \ldots, (i + 1)\, 2^r - 1\}$. Assume that the first $i \geq r$ levels have been labelled. Consider an internal vertex v at level i. Let w be the ancestor of v at the level $i - r$ and $\{\ell_0, \ell_1, \ldots, \ell_{2^r-1}\}$ be its label set with $\ell_0 < \ell_1 < \cdots < \ell_{2^r-1}$. Suppose that v is the jth descendent of w at the level i with $0 \leq j < 2^r$. Then, v inherits from w the $2^r - s$ labels $\left\{\ell_j, \ell_{j+1}, \ldots, \ell_{j+2^r-s-1}\right\}$ where the algebraic operations within the subscripts are with respect to mod 2^r. The remaining labels of v are those labels not used by any of its $r - 1$ immediate ancestors. See Fig. 2 for an example of this labeling process when $k = 5$. Clearly, this labeling maintains the following *disjointedness property*: the label sets of up to r consecutive internal vertices on a path up the tree are disjoint. If ℓ is a label of some vertex w at level i, then ℓ is a label of at most $2^r - s$ descendants of w at the level $i + r$ and is not a label of at most s descendent of w at the level $i + r$. If ℓ is not a label of some descendant v of w at the level $i + r$, then ℓ must be a label of each child of v, if there is any. Therefore, the number of immediate descendents of w with label ℓ is at most $(2^r - s) + 2s = 2^r + s = k$.

Next, we use the mapping and labeling constructed above to produce $r2^r + s$ k-restricted full-tree covers of τ. Each label ℓ determines a unique k-restricted full-tree cover $\mathcal{T}_\ell^+$ of τ in T^+. Each full component of $\mathcal{T}_\ell^+$ is a rooted subtree of T^+ whose root is either the root of T^+ or some branching node labelled with ℓ. The full component rooted at a node v is defined in two cases:

- Case 1: No descendent of v is labeled by ℓ. Then the component rooted at v is the subtree of T^+ induced by v and all its descendents.
- Case 2: At least one descendent of v is labeled by ℓ. Let U be the set of immediate descendents u of v labeled with ℓ. Then the tree component rooted at v is the union of the minimal subtree of T^+ which spans v and U and the paths $P^+(u)$ for all u in U. Equivalently, it is the the minimal subtree of T^+ which spans v and the nodes $f(u)$ for all u in U. The nodes in U are called the intermediate leaves of T_ℓ^+.

By the argument in [1], $\mathcal{T}_\ell^+$ is a full-tree cover of τ in T^+.

Each $\mathcal{T}_\ell^+$ is further converted to a k-restricted full-tree cover of τ in T in two steps. We first remove all duplicates from each full component of $\mathcal{T}_\ell^+$ as follows. We observe that if a duplicate but not its original appears in a full component, then such duplicate must be the component root. In this case, we simply replace the duplicate by its original. If a duplicate and its original appear in the same full component, we remove the duplicate and put all its children as the children of its parent (i.e., its original). After all duplicates have been removed, we contract the root. The resulting k-restricted full-tree cover of τ in T is denoted by $\mathcal{T}_\ell$.

Finally, we show that at one of the $r2^r + s$ k-restricted full-tree covers $\mathcal{T}_\ell$ for $0 \leq \ell < r2^r + s$ has weight at most $\rho_k c(T)$. For each label ℓ, we compute $c(\mathcal{T}_\ell) - C(T)$ by introducing a charging scheme on the internal nodes of T. For each node v with label ℓ other than the root of T^+, we make the following three types of charges.

1. Each internal node u of the path $P^+(v)$ which is not a duplicate is charged with its weight $c(u)$.
2. If v is not a duplicate, then v is charged with its weight $c(v)$.
3. If v is a duplicate of some original node u, then u is charged with its weight $c(u)$.

We remark that an original node may get charged multiple times. Then, $c(\mathcal{T}_\ell) - c(T)$ is exactly the total charges on all internal nodes of T from this charging.

The above charging scheme for all labels leads to an upper bound on $\sum_{\ell=0}^{r2^r+s-1}(c(\mathcal{T}_\ell) - c(T))$. Note that $\sum_{\ell=0}^{r2^r+s-1}(c(\mathcal{T}_\ell) - c(T))$ is the total charges on all internal nodes of T from all $r2^r + s$ chargings. We first claim that each internal node u receives at most 2^r charges of Type 1. Indeed, if u is not in the internal node of any path $P^+(v)$, then u does not receive any charge of Type 1. Otherwise, there is a unique internal node v of T^+ such that u is an internal node of $P^+(v)$, and u receives exactly 2^r charges of Type 1 since v has 2^r labels. So, our claim holds. We further claim that u receives at most 2^r charges of Type 2 and at most 2^{r+1} charges of Type 3. If u is not a branching node, then u does not receive any charge of Type 2 or Type 3. So we assume that u is a branching node. Since u has exact 2^r labels, u receives exactly 2^r charges of Type 2. Since each duplicate of u also has exactly 2^r labels, it introduces exactly 2^r charges of Type 3. As u has at most two duplicates, u receives at most $2 \cdot 2^r = 2^{r+1}$ charges of Type 3. So, our second claim holds. These two claims imply that the total number of charges on each internal node of T is at most $2^r + 2 \cdot 2^r + 2^r = 4 \cdot 2^r = 2^{r+2}$. Therefore,

$$\sum_{\ell=0}^{r2^r+s-1}(c(\mathcal{T}_\ell) - c(T)) \leq 2^{r+2} c(T).$$

Let ℓ^* be the label such that $c(\mathcal{T}_{\ell^*}) - c(T)$ is the smallest. Then

$$c(\mathcal{T}_{\ell^*}) - c(T) \leq \frac{2^{r+2}}{r2^r + s} c(T).$$

Hence,

$$c\left(\mathcal{T}_{\ell^*}\right) \leq \left(1+\frac{2^{r+2}}{r2^r+s}\right) c\left(T\right) = \rho_k c\left(T\right).$$

So, the lemma holds. □

Theorem 2.2 *If G is a unit-disk graph, then $smt_k\left(S\right) \leq \rho_k \cdot smt\left(S\right)$.*

Proof Consider a minimum-weighted Steiner tree for S in G with the largest number of full components $T_1, T_2, \ldots, T_m$. For each $1 \leq i \leq m$, let V_i denote the set of nodes in T_i, and τ_i denote the set of terminals in T_i. Then, any spanning tree of $G\left[V_i\right]$ must be full. Let T_i^* be an Euclidean minimum spanning tree of V_i whose maximum degree is at most five. Then, T_i^* is a full tree contained in $G\left[V_i\right]$. By Lemma 2.1, there is a k-restricted full tree cover $\mathcal{T}_i$ of τ_i in T_i^* satisfying that $c\left(\mathcal{T}_i\right) \leq \rho_k c\left(T_i^*\right)$. Let $\mathcal{T}$ be the union of $\mathcal{T}_i$ for $1 \leq i \leq m$. Then, $\mathcal{T}$ is a k-restricted full-tree cover of S and

$$c\left(\mathcal{T}\right) = \sum_{i=1}^{m} c\left(\mathcal{T}_i\right) \leq \rho_k \sum_{i=1}^{m} c\left(T_i^*\right) = \rho_k \sum_{i=1}^{m} c\left(T_i\right) = \rho_k \cdot smt\left(S\right).$$

It is obvious that $smt_k\left(S\right) \leq c\left(\mathcal{T}\right)$. So, the theorem holds. □

3 Approximation algorithms

In this section, we present the three approximation algorithms and derive their approximation bounds.

3.1 Proprocessing

In the proprocessing step, we compute $cost\left(\tau\right)$ and construct a subset $SMN\left(\tau\right) \subseteq V \setminus S$ for each subset τ of at most k terminals. The following lemma provides a way to computing $cost\left(\tau\right)$.

Lemma 3.1 *For any $\tau \subseteq S$,*

$$cost\left(\tau\right) = \min\left\{c\left(W\right) + mst\left(K\left[\tau \cup W\right]\right) : W \subseteq V \backslash \tau, \left|W\right| \leq \left|\tau\right| - 2\right\}.$$

Proof We first prove $cost\left(\tau\right)$ is no less than the right term in the equality. Let W be a subset of at most $\left|\tau\right| - 2$ nodes in $V \backslash \tau$ which minimizes $c\left(W\right) + mst\left(K\left[\tau \cup W\right]\right)$. We construct a set W' as follows. W' is initially empty. For each edge uv in $MST\left(K\left[\tau \cup W\right]\right)$, we add to W' all the internal nodes of a shortest path between u and v in. Finally, we remove all nodes in τ from W'. Let $U = W \cup W'$. By the construction, $c\left(W'\right) \leq mst\left(K\left[\tau \cup W\right]\right)$. Thus,

$$c\left(U\right) \leq c\left(W\right) + c\left(W'\right) \leq c\left(W\right) + mst\left(K\left[\tau \cup W\right]\right).$$

In addition, $U \subseteq V \backslash \tau$ and $G[\tau \cup U]$ is connected. Thus, $cost(\tau) \leq c(U)$. So,

$$cost(\tau) \leq \min\{c(W) + mst(K[\tau \cup W]) : W \subseteq V \backslash \tau, |W| \leq |\tau| - 2\}.$$

Next, we show that $cost(\tau)$ is no more than the right term in the equality. Consider a subset $U \subseteq V \backslash \tau$ such that $G[\tau \cup U]$ is connected and $c(U) = cost(\tau)$. Let T be a spanning tree of $G[\tau \cup U]$. Let W (respectively, W') be the set of nodes in U which are branching (respectively, relay) nodes of T. We contract all relay nodes to obtain a tree T' in $K[\tau \cup W]$. It's easy to show that the edge-weight of T' is at most $c\left(W'\right)$. Thus,

$$c\left(W'\right) \geq mst(K[\tau \cup W]),$$

and hence

$$cost(\tau) = c(U) = c(W) + c\left(W'\right) \geq c(W) + mst(K[\tau \cup W]).$$

Furthermore,

$$\begin{aligned} 2\left(|W| + \left|W'\right| + |\tau| - 1\right) = &\sum_{w \in W} \deg(w) + \sum_{w \in W'} \deg(w) \\ &+ \sum_{t \in \tau} \deg(t) \geq 3|W| + 2\left|W'\right| + |\tau|, \end{aligned}$$

which implies $|W| \leq |\tau| - 2$. Therefore,

$$cost(\tau) \geq \min\{c(W) + mst(K[\tau \cup W]) : W \subseteq V \backslash \tau, |W| \leq |\tau| - 2\}.$$

So, the lemma holds. □

We compute $cost(\tau)$ and construct $SMN(\tau)$ as follows. We first compute by exhaustive search a set W of at most $|\tau| - 2$ nodes in $V \backslash \tau$ which minimizes $c(W) + mst(K[\tau \cup W])$. A set U is initialized to W. For each edge uv in $MST(K[\tau \cup W])$, we add to U all the internal nodes of a shortest path between u and v in G. Set $SMN(\tau)$ to $U \backslash S$. Then, $cost(\tau) = c(SMN(\tau))$.

3.2 *k*-Restricted relative greedy algorithm

The k-Restricted Relative Greedy Algorithm presented in this section is adapted from the algorithm in [10] for the classical Steiner tree problem. It is also an application of a general greedy algorithm for Minimum Submodular Cover, which is described below. Consider a ground set Γ and a real function f defined on 2^{Γ}. f is *increasing* if for $X \subset Y$, $f(X) \leq f(Y)$. f is *submodular* if for any two subsets X and Y of Γ,

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y).$$

Table 2 Greedy algorithm for Minimum Fractional Submodular Cover (GSC)

```
$X \leftarrow \emptyset$;
while $f(X) < f(\Gamma)$ do
    select $e \in \Gamma$ with maximum $\Delta_e f(X)/c(e)$;
    $X \leftarrow X \cup \{e\}$;
output $X$.
```

A submodular and increasing function f is called a *polymatroid function* if $f(\emptyset) = 0$. The *gain* of an element $e \in \Gamma$ with respect to $X \subseteq E$ is defined by

$$\Delta_e f(X) = f(X \cup \{e\}) - f(X).$$

Suppose that f is a polymatroid function on 2^{Γ}. Then, a set $X \subseteq \Gamma$ is said to be a *submodular cover* of (Γ, f) if $f(X) = f(E)$. Consider a cost function $c \in \mathbb{R}_{+}^{\Gamma}$. The minimization problem

$$\min \left\{ \sum_{x \in X} c(x) : f(X) = f(\Gamma), X \subseteq \Gamma \right\}$$

is known as a Minimum Submodular Cover (MSC). A greedy approximation GSC for MSC is described in Table 2.

The following result was a special case of a more general theorem established in [8].

Theorem 3.2 *Suppose that $f(\Gamma) \geq opt$ where opt is the cost of a minimum submodular cover of (Γ, f). If in each iteration of the GSC, the selected e always satisfies that $\Delta_e f(X) \geq c(e)$, then the greedy solution is a $(1 + \ln(f(\Gamma)/opt))$-approximation.*

Now, we formulate the problem of finding a minimum-cost k-restricted Steiner cover into a special case of Minimum Submodular Cover. For any $\tau \subseteq S$, denote $K_0(\tau)$ be the complete graph on τ with all edges having *zero* length. Denote by Π_0 the set of pairs of adjacent terminals in G, and by Γ_k the collection of subsets τ of terminals satisfying that $2 \leq |\tau| \leq k$ and $cost(\tau) > 0$. Then, for each $\tau \in \Pi_0$, $K_0(\tau) \subseteq K[S]$. We define a set function f on 2^{Γ_k} as follows. For any $\Pi \subseteq \Gamma_k$, denote $K_0(\Pi) = \cup_{\tau \in \Pi} K_0(\tau)$ and set

$$f(\Pi) = mst(K[S]) - mst(K[S] \cup (\cup_{\tau \in \Pi} K_0(\tau))).$$

Then f is polymatroidal on 2^{Γ}, and a subset $\Pi \subseteq \Gamma_k$ is a submodular cover of (Γ_k, f) if and only if $\Pi \cup \Pi_0$ is a k-restricted Steiner cover of S. Note that $\sum_{\tau \in \Pi} cost(\tau)$ is the exactly the cost of $\Pi \cup \Pi_0$. Therefore, if Π is a minimum-cost submodular cover of (Γ_k, f), then $\Pi \cup \Pi_0$ is a minimum-cost k-restricted Steiner cover of S. Suppose a subset $\Pi \subseteq \Gamma_k$ is not a submodular cover of (Γ_k, f), i.e. $f(\Pi) < mst(K[S])$. Then, $MST(K[S] \cup (\cup_{\tau \in \Pi} K_0(\tau)))$ contains at least one edge uv with positive weight. Let $\tau = \{u, v\}$. Then, $\Delta_\tau f(\Pi) = cost(\tau)$. Clearly,

Table 3 *k*-Restricted Relative Greedy Algorithm

```
$\Pi \leftarrow \emptyset, U \leftarrow \{v \in V \setminus S : c(v) = 0\}$;
while $f(\Pi) < mst(K[S])$ do
    select $\tau \in \Gamma_k$ with maximum $\Delta_\tau f(\Pi) / cost(\tau)$;
    $\Pi \leftarrow \Pi \cup \{\tau\}, U \leftarrow U \cup SMN(\tau)$;
output a Steiner tree $T$ for $S$ in $G[S \cup U]$.
```

$f(\Gamma_k) = mst(K[S]) \geq smt_k(S)$. So, the application of GSC to (Γ_k, f) would produce a solution whose cost is at most

$$\begin{aligned}
\left(1 + \ln \frac{mst(K[S])}{smt_k(S)}\right) smt_k(S) &= \left(1 + \ln \frac{smt_2(S)}{smt_k(S)}\right) smt_k(S) \\
&\leq \left(1 + \ln \frac{\frac{smt_2(S)}{smt(S)}}{\frac{smt_k(S)}{smt(S)}}\right) \frac{smt_k(S)}{smt(S)} smt(S) \\
&\leq \left(1 + \ln \frac{\rho_2}{\frac{smt_k(S)}{smt(S)}}\right) \frac{smt_k(S)}{smt(S)} smt(S) \\
&\leq \left(1 + \ln \frac{\rho_2}{\rho_k}\right) \rho_k smt(S) \\
&= \left(1 + \ln \frac{5}{\rho_k}\right) \rho_k smt(S).
\end{aligned}$$

The *k*-Restricted Relative Greedy Algorithm is described in Table 3. It follows the same framework of GSC, but introduces another variable U to store the Steiner nodes. The output is an arbitrary Steiner tree T for S in $G[\tau \cup U]$, instead of Π. Since

$$c(T) \leq \sum_{\tau \in \Pi} c(SMN(\tau)) = \sum_{\tau \in \Pi} cost(\tau),$$

we have

$$c(T) \leq \left(1 + \ln \frac{5}{\rho_k}\right) \rho_k smt(S).$$

Therefore, we have the following approximation bound of *k*-Restricted Relative Greedy Algorithm.

Theorem 3.3 *The approximation ratio of k-Restricted Relative Greedy Algorithm is at most* $\left(1 + \ln \frac{5}{\rho_k}\right) \rho_k$.

The approximation bound $\left(1 + \ln \frac{5}{\rho_k}\right) \rho_k$ converges to $1 + \ln 5 \approx 2.61$ when $k \to \infty$. However, the rate of the convergence is slow.

Table 4 3-Restricted Greedy Algorithm

```
$\Pi \leftarrow \emptyset, U \leftarrow \{v \in V \setminus S : c(v) = 0\}$;
while $f(\Pi) < mst(K[S])$ do
    select $\tau \in \Gamma_3$ with maximum $\Delta_\tau f(\Pi) - cost(\tau)$;
    $\Pi \leftarrow \Pi \cup \{\tau\}, U \leftarrow U \cup SMN(\tau)$;
output a Steiner tree $T$ for $S$ in $G[S \cup U]$.
```

3.3 3-Restricted greedy algorithm

The 3-Restricted Greedy Algorithm presented in this section is adapted from the algorithm in [9] for the classical Steiner tree problem. It is described in described in Table 4. It differs from 3-Restricted Relative Greedy Algorithm only in the selection criteria in the iterations. Instead of selecting a subset $\tau \in \Gamma_3$ with maximum *gain-cost ratio* $\Delta_\tau f(\Pi)/cost(\tau)$ in each iteration of 3-Restricted Relative Greedy Algorithm, the 3-Restricted Greedy Algorithm selects a subset $\tau \in \Gamma_3$ with maximum *net-gain* $\Delta_\tau f(\Pi) - cost(\tau)$ in each iteration.

Using the same proof arguments given in [3], we can prove the following approximation bound of 3-Restricted Greedy Algorithm.

Theorem 3.4 *The approximation ratio of* 3*-Restricted Greedy Algorithm is at most* $(\rho_2 + \rho_3)/2 = 4\frac{1}{3}$.

3.4 *k*-Restricted variable metric algorithm

The k-Restricted Variable Metric Algorithm presented in this section is adapted from the algorithm in [2] for the classical Steiner tree problem. we introduce some notations. Let M be an edge-weighted tree on S, and τ be a subset of S. Let $Cut_\tau(M)$ be a heaviest subset C of edges in M satisfying that each connected component of $M - C$ contains exactly one node in τ. $Cut_\tau(M)$ can be computed in polynomial time and must consists of exactly $|\tau| - 1$ edges. The total weight of the edges in $Cut_\tau(M)$ is denoted by $cut_\tau(M)$. We define the *net-gain* of τ w.r.t. M by

$$g_\tau(M) = cut_\tau(M) - cost(\tau).$$

Each $e \in Cut_\tau(M)$ links exactly two connected components of $M - Cut_\tau(M)$. Let u and v be the two nodes in τ which belong to such two components, respectively. We create an *artificial* edge $e' = uv$ with weight equal to the weight of e minus $g_\tau(M)$. Let

$$A_\tau(M) = \left\{e' : e \in Cut_\tau(M)\right\}.$$

The k-Restricted Variable Metric Algorithm consists of two phases: evaluation and construction. The *evaluation* phase begins with a minimum-weighted spanning tree M of G and considers all possible subsets of terminals of size between 3 and k,

Table 5 Some values of $5-\sum_{j=3}^{k}\frac{\rho_{j-1}-\rho_j}{j-1}$

k	3	4	5	2^3	2^4	2^{20}
$5-\sum_{i=3}^{k}\frac{\rho_{i-1}-\rho_i}{i-1}$	$4\frac{1}{3}$	$4\frac{1}{9}$	$4\frac{1}{18}$	3.9785	3.9472	3.9334

in the increasing order of size. At a generic step, if a subset τ offers positive net-gain in the current M, the tuple $(\tau, A_\tau(M), Cut_\tau(M))$ is pushed onto a stack and M is modified by replacing $Cut_\tau(M)$ with $A_\tau(M)$. The *construction* phase maintains two spanning trees M and T and a subset U of Steiner nodes. Both M and T are initially the tree at the end of the evaluation phase, and U is initially empty. When the tuple $(\tau, A_\tau(M), Cut_\tau(M))$ is popped off the stack, M is replaced by $M - A_\tau(M) \cup Cut_\tau(M)$. If the whole tree $A_\tau(M)$ is intact in T, then we add $SMN(\tau)$ to U. Otherwise, whatever edges e' of $A_\tau(M)$ remaining in T is replaced by the lightest edge in M linking the two pieces of $T - e'$. When the stack is empty, for each edge uv in T which is not artificial, we add to U all the internal non-terminal nodes of the shortest path between u and v in G. Finally, we output an arbitrary Steiner tree for S in $G[S \cup U]$.

Following the same proof in [2], we can derive the following approximation bound of k-Restricted Variable Metric Algorithm.

Theorem 3.5 *The approximation ratio of k-Restricted Variable Metric Algorithm is at most* $5-\sum_{j=3}^{k}\frac{\rho_{j-1}-\rho_j}{j-1}$.

Clearly, $5-\sum_{j=3}^{k}\frac{\rho_{j-1}-\rho_j}{j-1}$ strictly decreases with k. Table 5 lists some values of $5-\sum_{j=3}^{k}\frac{\rho_{j-1}-\rho_j}{j-1}$. When $k=2^{20}$, $5-\sum_{j=3}^{k}\frac{\rho_{j-1}-\rho_j}{j-1}$ becomes less than 3.9334.

References

1. Borchers, A., Du, D.-Z.: The k-Steiner ratio in graphs. SIAM J. Comput. **26**(3), 857–869 (1997)
2. Berman, P., Ramaiyer, V.: Improved approximations for the Steiner tree problem. J. Algorithms **17**(3), 381–408 (1994)
3. Gröpl, C., Hougardy, S., Nierhoff, T., Prömel, H.J. : Approximation algorithms for the Steiner tree problem in graphs. In: Cheng, X., Du, D.-Z. (eds.) Steiner trees in Industry., pp. 235–279. Kluwer, Dordrecht (2001)
4. Guha, S., Khuller, S.: Improved methods for approximating node weighted Steiner trees and connected dominating sets. Inf. Comput. **150**(1), 57–74 (1999)
5. Klein, P., Ravi, R.: A nearly best-possible approximation algorithm for node-weighted Steiner trees. J. Algorithms **19**(1), 104–115 (1995)
6. Papadimitriou, C.H., Vazirani, U.V.: On two geometric problems relating to the traveling salesman problem. J. Algorithms **5**, 231–246 (1984)
7. Robins, G., Zelikovsky, A.: Tighter bounds for graph Steiner tree approximation. SIAM J. Discrete Math. **19**(1), 122–134 (2005)
8. Wan, P.-J., Du, D.-Z., Pardalos, P., Wu, W.: Greedy approximations for minimum submodular cover with submodular cost. Comput. Optim. Appl. (2009) (to appear)
9. Zelikovsky, A.: The 11/6-approximation algorithm for the Steiner problem on networks. Algorithmica **9**, 463–470 (1993)
10. Zelikovsky, A.: Better approximation bounds for the network and Euclidean Steiner tree problems. Technical Report CS-96-06, Department of Computer Science, University of Virginia (1996)
11. Zou, F., Li, X., Kim, D., Wu, W.: Two Constant Approximation Algorithms for Node-Weighted Steiner tree in Unit Disk Graphs. Lecture Notes in Computer Science, vol. 5165, pp. 278–285. COCOA (2008)