An Approximation Algorithm for the Covering Steiner Problem

Goran Konjevod* R. Ravi†

Abstract

The covering Steiner tree problem is a common generalization of the k-MST and the group Steiner problems: Given an edge weighted graph, with subsets of vertices called the groups, and a requirement for each group which is an integer of value at most the size of the group, the problem is to find a minimum-weight tree such that for each group, at least as many nodes as its requirement are included in the tree. When all requirements are equal to 1, we get the group Steiner problem, while if there is only group whose node set is all the vertices in the graph, and this group's requirement value is the integer k, the problem reduces to finding a minimum-weight tree containing k vertices.

We present a polylogarithmic approximation algorithm for this problem which uses an integer linear programming formulation, and rounds the optimal fractional solution iteratively. One interesting feature of our algorithm is that even though the optimal fractional value of the original LP formulation may be a very bad estimate of the optimal integral solution value, at least one of the formulations arising in one of the iterations of our rounding estimates the optimal integral solution value well.

1 Introduction

1.1 Statement of the problem. Let $G = (V, E)$ be an undirected graph with a cost function $c: E \to \mathbb{R}_+$ defined on the edges. Let $g_1, \ldots, g_m \subseteq V$ be subsets of vertices. We call the sets g_i *groups*. For each group g_i a nonnegative integer k_i, called the *requirement* of the group is specified - this value for group i is at most $|g_i|$. The *covering Steiner problem* on G is the problem of finding a minimum-cost connected subgraph of G which contains at least k_i vertices of group g_i for all $i \in \{1, \ldots m\}$. We denote the size of the largest group by N, and the largest requirement of a group by K. We call the group vertices *terminals*.

The covering Steiner problem generalizes two different NP-hard network design problems that have been studied recently, namely the k-MST problem (see Ravi, Sundaram, Marathe, Rosenkrantz and Ravi [16], Fischetti, Hamacher, Jørnsten and Maffioli [9], Blum, Ravi and Vempala [6], and Garg [10]), and the group Steiner problem (see Reich and Widmayer [17], Garg, Konjevod and Ravi [12] and Charikar, Chekuri, Goel and Guha [7]).

The k-MST problem is that of finding a minimum-cost connected subgraph that contains at least k nodes in an undirected graph with nonnegative costs on edges. The covering Steiner problem reduces to the k-MST problem when there is only one group and when all the vertices in V belong to this group. This problem is NP-hard and the best-known approximation ratio for this problem currently is 2, using an algorithm of Garg [11], but see also papers of Arya and Ramesh [3] and Arora and Karakostas [2]. This problem is solvable in polynomial time in some special cases, e.g. if the underlying graph is a tree.

The group Steiner problem is the restriction of the covering Steiner problem where all group requirements are 1. This problem is at least as hard to approximate as the set cover problem, because even the special case where the underlying graph is a star-tree contains the set cover problem (Klein and Ravi [14]). Approximation algorithms for the group Steiner problem with polylogarithmic approximation ratios were presented in [12] (randomized) and [7] (deterministic).

1.2 Results. Our central result is a randomized approximation algorithm for the covering Steiner problem on a tree with an approximation guarantee of $O(\log N \log m \log K)$ Here, N denotes the maximum size of a group (which in turn is at most n, the total number of nodes), K denotes the maximum requirement value of any group (which in turn is at most N) and m denotes the number of groups. For the group Steiner problem where $K = 1$, this approximation ratio matches the best-known ratio. We can transform the problem in any metric to one on a tree, with a worsening in the performance ratio, by using the technique of Bartal [5] and Charikar, Chekuri, Goel and Guha [7]. This then leads to an approximation algorithm for the general covering Steiner problem with approximation guarantee of $O(\log n \log \log n \log N \log m \log K)$. Using im-

*Dept. of Math. Sciences, Carnegie Mellon University, Pittsburgh, PA 15213-3890. Work partly done at Los Alamos National Laboratory. Supported by the NSF CAREER grant CCR-9625297 and the DOE Contract W-7405-ENG-36. konjevod@andrew.cmu.edu

†GSIA, Carnegie Mellon University, Pittsburgh, PA 15213-3890. Work partly done while visiting IBM SRC, New Delhi. Supported by the NSF CAREER grant CCR-9625297. ravi@cmu.edu

proved metric approximations for graphs that exclude $K_{s,s}$ as a minor (such as planar graphs, points in the plane), we can get an improved approximation guarantee for the problem on such graphs. Our algorithms can be derandomized by using ideas from [7].

1.3 Overview and contributions. We solve the covering Steiner problem on the tree by successively rounding solutions to linear programming relaxations of $\log K$ different covering Steiner subproblems that arise from our original problem. Our formulations capture the multiplicity requirement for the groups by using a different "commodity" to denote different unit requirements. Intuitively, each group is required to participate (include one of its vertices) in as many commodities as its requirement value. All vertices included in a commodity must be connected to the root via a subtree of edges that is chosen by the solution. The objective is to minimize the total cost of the edges chosen to support the trees for all commodities. Since a rounding step can only ensure that a constant fraction of each group's current requirement is met with high probability, we set up a new LP to satisfy the remaining requirement and proceed iteratively until all requirements are met.

One surprising outcome is that the initial LP relaxation that we use for the original problem *does not* necessarily provide a good lower bound on the cost of the optimal covering Steiner tree (See Section 2.3). Even for the special case with only one group whose requirement is K, the gap between the optimal integral and fractional solutions can be $\Omega(K)$. However, it follows from our proof that at least one of the $O(\log K)$ LP relaxations we use in the rounding provides a good lower bound (within polylogarithmic factor of optimal).

2 Linear programming formulation

First, we make several assumptions that do not reduce the generality of the problem, but make it easier to formulate.

2.1 Assumptions. We assume the following.
(1) The graph G is a weighted tree (we can use the results of Bartal [5] on probabilistic approximation of general metrics by tree metrics to reduce the original problem to the problem in a weighted tree with a slight loss in the performance ratio — the details are in Section 4.1);
(2) The groups are disjoint (a vertex belonging to several groups may be expanded into a star with edges of cost 0 and each leaf belonging to exactly one of the groups);
(3) Every vertex belonging to a group has degree 1 (similar to the construction in **(2)**);
(4) Every vertex of degree 1 belongs to some group (otherwise it may be removed from the graph);
(5) It suffices to consider the rooted problem where we know one vertex called the *root* that belongs to the optimal solution (the algorithm may be run for each possible choice of this vertex).

2.2 ILP Formulation. We formulate the covering Steiner problem on a tree as an integer linear program. Let the indicator variable x_e denote whether the edge e is contained in the solution, Then the cost of a solution x equals $\sum_{e\in E} c_e x_e$. In addition to x we use variables y^i for $i \in \{1,\ldots,K\}$. We may think of the variables y^i as supporting flows of different commodities from the root to the terminals. In a covering Steiner tree, there are k_i vertices of the group g_i, so we require the commodities $c_1,\ldots,c_{k_i}$ to each send a unit of flow from the root to some unique terminal in group g_i, for all i. The final solution x must majorize every y^i. In addition, we must require that no terminal in a group be used by more than one commodity for this group. To ensure this, for each edge e incident on a terminal (pendant edge), we require that

$$x_e \geq \sum_i y^i_e.$$

This constraint (or rather the fact that it can only be enforced for pendant edges) necessitates another set of constraints: $x_e \geq x_f$ for all successor edges f of e in the tree, for all e. This in turn makes our formulation possible only when G is a tree.

The complete linear programming formulation follows. Here, ∂S denotes the set of edges with exactly one endpoint in S, and $y^i(E')$ denotes $\sum_{e\in E'} y^i(e)$ for any subset of edges E'.

$$\begin{aligned}
&\min \sum_{e\in E} c_e x_e \\
&y^i(\partial S) \geq 1 \\
&\qquad \text{for all } S \subset V \text{ such that } r \in S \text{ and } i \\
&\qquad \text{such that for some group } g, \\
&\qquad S \cap g = \emptyset \text{ and } k_g \geq i \\
&x_e \geq y^i_e \\
&\qquad \text{for all non-pendant edges } e \\
&x_e \geq \sum_i y^i_e \\
&\qquad \text{for all pendant edges } e \\
&x_e \geq x_f \\
&\qquad \text{for all } (e,f) \text{ where } e \text{ is the parent of } f \\
&0 \leq x_e \leq 1\ \forall e.
\end{aligned} \tag{2.1}$$

2.3 Integrality gap. Even for the version of the problem with a single group, this linear formulation does not give a tight relaxation. For instance, let G be the tree in Fig. 1, that consists of a star with $k-1$ leaves, whose center is connected to another star with k leaves by a single edge. Let the center of the first star be the root of the tree. Denote the first star by A, the second by B, and the edge joining them by e_0. The single group consists of all the leaves of G and has requirement value k.

Consider the solution to the linear program where each commodity sends $1/k$ of flow from the root to each of the $k-1$ leaves of A and each of the k different commodities sends $1/k$ of flow to a distinct leaf of B. The packing constraints force $x_e = 1$ for all edges e in A, but since only one commodity is served by every edge of B, $x_f = 1/k$ for all edges f of B and for the edge e_0.

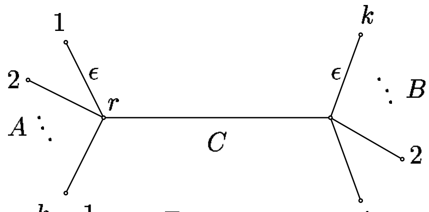

Figure 1.

Let the cost of the edges belonging to the stars A and B be ϵ, and the cost of edge e_0 be C. Clearly, the cost of the optimal tree that contains the root and covers k terminals is at least $C + k\epsilon$, since at least one of the leaves of B must be included. However, the relaxed solution described above costs only $C/k + k\epsilon$. Thus, the ratio between the optimal integral and fractional solutions of the formulation (2.1) can be as large as k.

3 The Iterative Rounding Algorithm

The algorithm runs in $\log K$ phases where K is the maximum requirement value. In each phase, a linear program is solved and the solution rounded to a set of edges of G. The rounded solution halves the residual requirement for every group and is then used to modify the problem whose LP formulation is solved in the next phase.

The rounding procedure in each phase succeeds with probability at least $1 - 1/2\log K$, and so the whole algorithm succeeds with probability at least $1/2$.

3.1 Basic rounding step. Suppose we are given an optimal solution to the linear program (2.1). We use the rounding procedure described in [12], namely, an edge e is included in the set of edges chosen with probability $x_e/x_{p(e)}$, where $p(e)$ denotes the predecessor of e on the path from the root. This experiment is performed for each edge of G independently. Let H denote the subgraph of G induced by the chosen edges. We discard all connected components of H except the one containing the root, and denote the resulting tree by T. It is not difficult to see that the expected cost of a tree obtained in this experiment is equal to the cost of the initial linear programming solution. An edge e is included in T iff all the edges in the path from r to e, say $e_1, \ldots, e_p, e$ are picked in their respective independent random experiments, the probability of which is

$$\frac{x_{e_1}}{1} \cdot \frac{x_{e_2}}{x_{e_1}} \cdots \frac{x_e}{x_{e_p}} = x_e.$$

Garg, Konjevod and Ravi [12] further show that for each group g, the probability that a vertex of g is included in T is $\Omega(1/\log|g|)$. For this, they use an inequality of Janson [13]. By a similar argument, and using a generalization of the same inequality, we show that for each group, the probability that at least *half* of its requirement is satisfied by T is $\Omega(1/\log|g|)$.

THEOREM 3.1. *Let T be the tree arising from the random experiment described above. Then there exists a constant C such that for any group g of size $|g|$ and with requirement k_g,*

$$\Pr[T \text{ contains at least } k_g/2 \text{ vertices of } g] \geq C\frac{1}{\log|g|}.$$

In what follows we consider the experiment for a single group g with requirement k. First we state the probabilistic inequality: let Ω be a universal set, and $R \subseteq \Omega$ determined by the experiment in which each element $r \in \Omega$ is independently included in R with probability p_r. Let A_i be subsets of Ω, and denote by B_i the event that $A_i \subseteq R$. Write $i \sim j$ if B_i and B_j are not independent. Define $\Delta = \sum_{i \sim j} \Pr[B_i \cap B_j]$ (the sum is over ordered pairs). Let $X = \sum_i X_i$, where X_i is an indicator variable for the event B_i, and let $\mu = \mathbf{E}[X] = \sum_i \Pr[B_i]$.

THEOREM 3.2. *(Janson's inequality [13].) With the notation as above,*

$$\Pr\big[X \leq (1-\gamma)\mu\big] \leq e^{-\gamma^2\mu/(2+\frac{\Delta}{\mu})}.$$

In our application, $\Omega = E(T')$, and $p_e = x_e/x_{p(e)}$ where $p(e)$ is the parent or predecessor of e on the path from r. The subsets A_i are edge-sets of paths from r to leaves belonging to a fixed group g, and X is the number of vertices of the group chosen in the experiment.

The following lemma was used in [12].

LEMMA 3.3. *If T and T' are trees that differ only in the capacity of one edge e, so that $x_T(e) \geq x_{T'}(e)$, then for any group, g, the probability of including a vertex from g is no greater in T' than in T.*

This lemma still holds in the covering Steiner problem, if the probability of including a vertex from g is replaced by the probability of including at least $k/2$ vertices of g, where k is the requirement of g.

Another useful observation is the following.

LEMMA 3.4. *Let x be the solution of the linear program (2.1) where a flow of 1 can be sent to vertices of group g by each of k different commodities. Then the expected number of vertices of g picked in one rounding step is at least k.*

In [12] it is shown (in the proof of their Theorem 3.2) that, given a solution x of the linear programming relaxation of the group Steiner problem, for every group g, there is a tree T^g with capacities x^g such that
(1) the probability of success for g when rounding according to x^g (that is, the probability of including a vertex of g) is $\Omega(1/\log|g|)$ and
(2) the probability of success for g when rounding according to x^g is no more than the probability of success when rounding according to x.

Essentially, T^g is obtained by rounding down the capacities to powers of 2, discarding the edges with capacity less than $1/(2n)$ and contracting the edges preceded by edges of equal capacities (except for the pendant edges). In the resulting tree, there is still a flow of $1/4$ from the root to every group.

We need a slightly different result now, because we must guarantee to cover at least a half of a group's requirement. To be able to apply Theorem 3.2, we must keep the expected number of vertices of a group covered by a single rounding step a constant fraction above $1/2$. We arrange the constants used in the first part of the argument so that the expected number of vertices of g covered in a single rounding step is $2k/3$, where k is the requirement of the group g.

Now we can describe the proof of Theorem 3.1.

Proof. (of Theorem 3.1.) Let g be a group with requirement k. Consider an optimal solution to the linear program (2.1) and its support tree. Let T be the tree spanned by the paths between the root and the vertices of g, and let (x, y) be the restriction of the optimal LP solution to the edges of T.

For every $i \in \{1, \ldots, k\}$ and every $e \in E(T)$ let

$$v_e^i = (10/11)^\ell,$$

where ℓ is chosen so that

$$(10/11)^\ell \leq y_e^i < (10/11)^{\ell+1}$$

(i.e., v_e^i is y_e^i rounded down to the nearest power of $10/11$).

Then define w, starting from the pendant edges and going up the tree T. For a pendant edge e, let

$$w_e = \sum_{i=1}^{k} v_e^i.$$

If w has been defined for all children of a non-pendant edge f, let

$$\phi_h^1 = \max\{w_e \mid f \text{ child of } e\},$$
$$\phi_h^2 = \max_{1 \leq i \leq k} v_e^i$$

and

$$w_h = \max\{\phi_h^1, \phi_h^2\}.$$

Next, for every edge e, let $u_e = (10/11)^\ell$ where ℓ is chosen so that $(10/11)^\ell \leq w_e < (10/11)^{\ell+1}$ (i.e., u_e is w_e rounded down to the nearest power of $10/11$). For every edge e such that $0 < u_e < 1/(121n)$, define $u_e = 0$ and thus effectively remove e from T. Finally, for every pair (e, f) of edges such that e is a parent of f, f is not pendant and $u_e = u_f$, contract f. Call the resulting tree T'.

We use the following properties of T'.
(1) In one rounding step performed according to the values of u, the probability of picking at least $k/2$ vertices of g is no greater than the probability of picking at least $k/2$ vertices of g when rounding according to x.
(2) The expected number of vertices of g picked in one rounding step done according to u is at least $3k/4$.
(3) The depth of T' is $d \leq \lceil \log_{11/10} n \rceil$.

Property **(1)** follows from Lemma 3.3. Let p be the probability of success (picking at least $k/2$ vertices of g) in one step of rounding according to x. Let p_1 and p_2 be the respective probabilities of success when the rounding is done according to w and u. Then Property **(1)** claims that

$$p \geq p_2.$$

Since $v \leq y$ (v is just y rounded down), and since w and x are obtained respectively from v and y by the same monotone process (compare LP (2.1) to the definition of w above), it follows that $w \leq x$. Thus $p \geq p_1$.

To prove that $p_1 \geq p_2$, note that removing some subtrees (consisting of edges of very small value) only decreases the success probability, and that the contractions described above do not change the probability of picking any vertex in a rounding step. Thus we have

$$p \geq p_1 \geq p_2.$$

To see that Property **(2)** holds, notice that converting y to v reduces the flow of each commodity by at most

1/11. Furthermore, a fraction 1/11 of this flow may be lost in rounding w to u, leaving behind at least $(10/11)^2$ flow for each commodity. Finally, deleting edges with u-value at most $1/(121n)$ reduces this flow by at most 1/121, leaving at least $(10/11)^2 - 1/121 > 3/4$ flow for each commodity. Property **(2)** then follows from linearity of expectation.

Property **(3)** is true because $u_e \geq 11u_f/10$ whenever e is a parent edge of f, except possibly when f is a pendant edge.

Denote by X the random variable that counts the vertices of g included in T after rounding. To apply Theorem (3.2), we need an upper bound on the value Δ.

Let $a(e,f)$ denote the least common ancestor of the pair of pendant edges e and f, both belonging to group g. For an edge f, let $i(f)$ denote the index i for which $\max_i v_f^i$ is achieved. Let d be the depth of the support tree of x. Then,

$$\begin{aligned}\Delta &= \sum_e \sum_{f\sim e} \frac{u_e u_f}{u_{a(e,f)}} \leq \sum_e u_e \sum_{f\sim e} \frac{k \max_i v_f^i}{u_{a(e,f)}} \\ &\leq k \sum_e u_e \sum_{f\sim e} \frac{v_f^{i(f)}}{v_{a(e,f)}^{i(f)}} \leq \frac{11}{10} kd \sum_e u_e \\ &\leq \frac{11}{10} k^2 \log_{11/10} |g| \leq 200k^2 \log |g|.\end{aligned}$$

The first inequality follows since f is a pendant edge going to a terminal node of group g whose requirement value is k, so that $u_f = \sum_{i=1}^k v_f^i \leq k \max_i v_f^i$. The second inequality uses the LP inequality $u_h \geq v_h^i$ for a non-pendant edge h and any commodity i.

The third inequality follows from the proof of Theorem 3.2 in [12]. The most important observation is that

$$\sum_{f\in E(e,a)} \frac{v_f^i}{v_a^i} \leq \frac{11}{10},$$

where the $E(e,a)$ is the set of all pendant edges f such that the least common ancestor of f and e is a. This follows because $\sum_{f\in E(e,a)} y_f^i / y_a^i \leq 1$ for all commodities i, and because v^i was obtained from y^i (which satisfied the flow-conservation constraints) by rounding down to powers of 10/11.

By Property **(3)**, the expected number of vertices of g covered in one rounding step is at least $\mu = 3k_g/4$. We need to cover $k/2$ vertices, so let $\gamma = 1/3$. Now substituting the upper bound for Δ in Theorem 3.2 gives

$$\begin{aligned}&\Pr\big[\text{at least } k/2 \text{ vertices of } g \text{ are covered}\big] \\ &= \Pr[X \geq (1-1/3)\cdot 3k/4] \\ &> 1 - \exp\left(\frac{\frac{1}{9}\mu}{2 + \frac{200k^2\log|g|}{\mu}}\right) \\ &= 1 - \exp\left(\frac{\frac{1}{9}}{\frac{2}{\mu} + \frac{800k^2\log|g|}{\mu^2}}\right) \\ &= 1 - \exp\left(\frac{\frac{1}{9}}{\frac{8}{3k} + \frac{12800\log|g|}{9}}\right) \\ &= 1 - \exp\left(\frac{1}{\frac{24}{k} + 12800\log|g|}\right) \\ &> \frac{C}{\log|g|},\end{aligned}$$

for a suitably chosen C. □

3.2 Amplification. After the basic rounding step described above, with probability at least $C/\log|g|$ we have satisfied a half of g's requirement. If we perform $C\log|g|$ rounding steps independently, with probability at least $1/e$, at least one of them will succeed.

3.3 Union. To ensure that we cover at least a half of *every* group's requirement, and that our iterative algorithm succeeds with a reasonable probability, we repeat the rounding a few more times. The goal is to bring down the probability of failure (to cover a half of the required number of vertices) for any group to $1/(2m\log K)$. Then by the union bound, we will have covered a half of *every* group's requirement with probability at least $1/(2\log K)$.

Recall that we write N for the size of the largest group and K for the maximum requirement. We repeat the basic rounding step $\beta C \log N \log 2m$ times, where β satisfies

$$\left(1 - \frac{C}{\log N}\right)^{(\beta \log 2m \log N)/C} < \frac{1}{2m\log K}.$$

Since $1/e < 1/2$, the above equation is implied by

$$\left(\frac{1}{2m}\right)^{\beta} < \frac{1}{2m\log K},$$

which is true whenever

$$\beta > \frac{\log 2m + \log\log K}{\log 2m}.$$

This gives

$$\frac{1}{C}\log N(\log 2m + \log\log K)$$

as the number of basic rounding steps needed. We may assume that $\log\log K < \log 2m$. Indeed, otherwise we have $2m < \log K$ so there are no more than $O(\log n)$ groups and just taking a 2-optimal k-MST [11] for each of them gives an $O(\log n)$ approximation algorithm for the overall problem.

Thus we repeat the basic rounding step

$$O(\log N \log 2m)$$

times and ensure that a half of every group's requirement is covered with probability at least $1-1/(2\log K)$.

3.4 Iteration. To achieve the required coverage for every group, we can repeat the above rounding algorithm on a sequence of subproblems of the original covering problem, at each step halving the residual requirement.

Suppose the first rounding phase succeeded in covering a half of the requirement for every group. Then we modify the graph by contracting the chosen edges and reducing the number of commodities that support group g to at most $k_g/2$ for every g. Note that any integral solution to the original problem contains as a subgraph an integral solution to this residual problem. Therefore, the cost of the optimal fractional solution to the residual problem is a lower bound on the cost of the optimal solution to the original problem.

In this way, we solve a sequence of $\log K$ subproblems, and then form the solution of the original problem as the union of the solutions to the subproblems.

If any of the $\log K$ steps fails, we stop the algorithm and say the algorithm fails. Since the probability of success of each step is at least $1 - 1/2\log K$, and the events are independent, the algorithm succeeds with probability at least $1/\sqrt{e} > 1/2$.

3.5 Cost. Overall, the basic rounding step is performed $\alpha = O(\log N \log K \log m)$ times, and so by Markov's inequality,

$$\Pr[\text{solution costs more than } 4\alpha \cdot OPT] < 1/4,$$

where OPT denotes the minimum cost of a covering Steiner tree. Thus, with probability more than 1/2 both "good" events happen, that is we get a feasible solution of cost at most $4\alpha \cdot OPT$.

THEOREM 3.5. *There is a randomized polynomial time algorithm that, with probability at least 1/2, finds a covering Steiner tree on an underlying graph which is a tree, of cost $O(\log N \log m \log K)$ times the minimum. Here, N denotes the maximum size of a group (which is at most the number of nodes in the tree), K denotes the maximum requirement value of any group (which in turn is at most N) and m denotes the number of groups.*

4 Extensions

4.1 General metrics.

DEFINITION 4.1. *A set of metric spaces $\mathcal{S}$ over V is said to α-probabilistically approximate a metric space M over V, if* **(1)** *for all $x, y \in V$ and $S \in \mathcal{S}$, $d_S(x,y) \geq d_M(x,y)$, and* **(2)** *there exists a probability distribution D over metric spaces in $\mathcal{S}$ such that for all $x, y \in V$, $\mathbf{E}[d_D(x,y)] \leq \alpha d_M(x,y)$.*

Bartal [4, 5] proved the following theorem.

THEOREM 4.2. *Every weighted connected graph G on n vertices can be α-probabilistically approximated by a set of weighted trees, where $\alpha = O(\log n \log\log n)$. Moreover, the probability distribution can be computed in polynomial time.*

The trees that we get from Bartal's algorithm are not subtrees of the original graph. Only their leaves are the original vertices of G. To solve the covering Steiner tree problem on a general graph G, first find a set of trees and the distribution on them that $O(\log n \log\log n)$-approximates G. Then pick a tree from the distribution and solve the covering Steiner tree problem on it. Now this solution subtree must be transformed into a subgraph of G, and this can be done by simply taking the tour that visits all the leaves of the solution tree, as in the classical 2-approximation for the metric TSP. The distances in the tree are greater than those in the original graph, so this tour will at most double the cost of the solution tree. The expected cost of this tree is $O(\log n \log\log n \log N \log m \log K)$ times the optimum. By using Markov's inequality, we finally get the following theorem.

THEOREM 4.3. *The algorithm described above with high probability finds a covering Steiner tree of cost $O(\log n \log\log n \log N \log m \log K)$ times the cost of the optimal tree.*

4.2 Improved metric approximations. The following improvement of Bartal's result to graphs that exclude small minors is presented by Konjevod et al [15].

THEOREM 4.4. *Let G be an n-node graph that excludes $K_{s,s}$ as a minor. Then G can be α-probabilistically approximated by a set of weighted trees, where $\alpha = O(s^3 \log n)$. Moreover, the probability distribution can be computed in polynomial time.*

This improved result (for constant s) applies, e.g., to planar graphs, which exclude $K_{3,3}$ as a minor. This theorem, together with the arguments from the previous section, then gives an improved approximation ratio of $O(\log n \log N \log m \log K)$ for such graphs.

Since distances in the Euclidean plane can be approximated to within a factor of 2 by a planar graph [8], the improvements also apply to this case. More formally, if the edge lengths of the resulting planar graph can be assumed to be integers in a polynomial range, then we can probabilistically approximate the original distances by trees with only a logarithmic loss. Even if these assumptions cannot be made, by identifying some points we can assume the distances to be in $\{1, \ldots, O(n^2)\}$. This can be done so that the optimum value of a covering Steiner tree only changes by a factor of $1 + \epsilon$ for any constant ϵ as in [1].

4.3 Derandomization. It is possible to derandomize our main procedure by using ideas from [7] to obtain the same guarantees. We defer the details to a full version of the paper.

Acknowledgements. Thanks to Naveen Garg, Madhav Marathe and Neal Young for helpful conversations.

References

[1] S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 2–11, 1996.

[2] S. Arora and G. Karakostas. A $2 + \epsilon$ approximation algorithm for the k-mst problem. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2000.

[3] S. Arya and H. Ramesh. A 2.5-factor approximation algorithm for the k-mst problem. *Information Processing Letters*, 65:117–118, 1998.

[4] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 184–193, October 1996.

[5] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 161–168, 1998.

[6] A. Blum, R. Ravi, and S. Vempala. A constant-factor approximation for the k-mst problem. In *Proceedings of the 28thAnnual ACM Symposium on Theory of Computing*, pages 442–448, 1996.

[7] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for group steiner trees and k-median. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 114–123, 1998.

[8] L. P. Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205–219, Oct. 1989.

[9] M. Fischetti, H. W. Hamacher, K. Jørnsten, and F. Maffioli. Weighted k-cardinality trees: complexity and polyhedral structure. *Networks*, 24:11–21, 1994.

[10] N. Garg. A 3-approximation for the minimum tree spanning k vertices. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 302–309, Oct. 1996.

[11] N. Garg. Personal communication, September 1999.

[12] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 253–259, 1998.

[13] S. Janson. Poisson approximations for large deviations. *Randoms Structures and Applications*, 1:221–230, 1990.

[14] P. N. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted Steiner tree. *J. Algorithms*, 19:104–115, 1995.

[15] G. Konjevod, R. Ravi, and F. S. Salman. On approximating planar metrics by tree metrics. manuscript, Jul. 1997.

[16] R. Ravi, R. Sundaram, M. V. Marathe, D. Rosenkrantz, and S. S. Ravi. Spanning trees—short or small. *SIAM J. Discrete Math.*, 9:178–200, 1996.

[17] G. Reich and P. Widmayer. Beyond Steiner's problem: A VLSI oriented generalization. In *Graph-Theoretic Concepts in Computer Science WG89*, volume 411 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 1990.

This research was sponsored in part by National Science Foundation (NSF) grant no. CCR-0122581.