A polylogarithmic approximation algorithm for the group Steiner tree problem

Naveen Garg* Goran Konjevod† R. Ravi‡

Abstract

The group Steiner tree problem is a generalization of the Steiner tree problem where we are given several subsets (groups) of vertices in a weighted graph, and the goal is to find a minimum-weight connected subgraph containing at least one vertex from each group. The problem was introduced by Reich and Widmayer and finds applications in VLSI design. The group Steiner tree problem generalizes the set covering problem, and is therefore at least as hard.

We give a randomized $O(\log^3 n \log k)$-approximation algorithm for the group Steiner tree problem on an n-node graph, where k is the number of groups. The best previous performance guarantee was $(1+\frac{\ln k}{2})\sqrt{k}$ (Bateman, Helvig, Robins and Zelikovsky).

Noting that the group Steiner problem also models the network design problems with location-theoretic constraints studied by Marathe, Ravi and Sundaram, our results also improve their bicriteria approximation results. Similarly, we improve previous results by Slavík on a tour version, called the errand scheduling problem.

We use the result of Bartal on probabilistic approximation of finite metric spaces by tree metrics to reduce the problem to one in a tree metric. To find a solution on a tree, we use a generalization of randomized rounding. Our approximation guarantees improve to $O(\log^2 n \log k)$ in the case of graphs that exclude small minors by using a better alternative to Bartal's result on probabilistic approximations of metrics induced by such graphs (Konjevod, Ravi and Salman) – this improvement is valid for the group Steiner problem on planar graphs as well as on a set of points in the 2D-Euclidean case.

1 Introduction.

1.1 Motivation. The group Steiner problem was introduced by Reich and Widmayer [19]. The problem arises in wire routing with multiport terminals in physical VLSI design. The traditional model assuming single ports for each of the terminals to be connected in a net of minimum length is the Steiner tree problem. When the terminal is a collection of different possible ports, so that the net can be connected to any one of them, we have a group Steiner tree problem: each terminal is a collection of ports and we seek a minimum length net containing at least one port from each terminal group The multiple port locations for a single terminal may also model different choices of placing a single port by rotating and/or mirroring the module containing the port in the placement. The choice allows for more interaction between the placement and routing phases of physical VLSI-design, potentially allowing for more overall optimization of the design.

The group Steiner tree problem can be stated formally as follows: we are given a graph $G = (V, E)$ with the cost function $c : E \to \mathbb{R}_+$, and sets of vertices $g_1, g_2, \ldots g_k \subset V$. We call $g_1, \ldots, g_k$ *groups*. The objective is to find the minimum cost subtree T of G that contains at least one vertex from each of the sets g_i. Formally, find $T = (V', E')$ that minimizes $\sum_{e \in E'} c_e$, such that $V' \cap g_i \neq \emptyset$ for all $i \in \{1, \cdots, k\}$. We use n to denote $|V|$ and N to denote the size of the largest group, $N = \max_i |g_i| \leq n$. We may assume that the groups are pairwise disjoint without loss of generality using the following transformation: if a node is in many groups, replace the node by a clique of zero-cost edges of size equal to the number of groups it occurs in; In addition to the neighbors in the clique, every copy of this node has the same neighbors as the original node in the graph.

The group Steiner problem is a generalization of the classical Steiner tree problem [21], and therefore NP-hard. In fact, it is also a direct generalization of the even harder set-covering problem as observed several times earlier [10, 14, 20]. In the set covering problem, we are given a collection of weighted subsets of a given ground set and seek a minimum-weight subcollection whose union is the entire ground set. To reduce this problem to a group Steiner problem, build a star with a leaf for each set and a new center node. Every element in the set-covering problem defines a group of leaves in the star in a natural way, namely, the leaves corresponding to the sets that contain this element. The equivalence is completed by giving the edges the weights of the corresponding sets. (Even if we require the groups to be disjoint, this construction can be realized by expanding each leaf node representing a set by attaching as many

*Max-Planck Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany. Supported by the EU ESPRIT LTR Project No. 20244 (ALCOM-IT). Email: naveen@mpi-sb.mpg.de

†Dept. of Math. Sciences, Carnegie Mellon University, Pittsburgh, PA 15213-3890. Supported by an NSF CAREER grant CCR-9625297. Email: konjevod@andrew.cmu.edu

‡GSIA, Carnegie Mellon University, Pittsburgh, PA 15213-3890. Supported by an NSF CAREER grant CCR-9625297. Email: ravi@cmu.edu

children to it as its size, one for each element, using zero-weight edges). Because of this, it is NP-hard to approximate the group Steiner problem to a factor of $o(\ln k)$ [7, ?, ?].

1.2 Previous Work. The papers of Ihler [9, 10, 11], and Ihler, Reich and Widmayer [12, 13] contain some early work on the group Steiner problem. (In some of the cited papers the group Steiner problem is referred to as the class Steiner problem, or the tree-cover problem.) In particular, in [9] it is proved that the heuristic introduced by Reich and Widmayer [19] has approximation ratio of $k-1$ (k is the number of groups). The related problem of minimum diameter group tree is shown to be polynomially solvable in [12]. The paper [11] gives a polynomial algorithm for a special case of the group Steiner problem where the groups of points are intervals on two parallel lines. [13] shows that the group Steiner tree problem is NP-hard even if the graph is a subgraph of a square grid in the plane, and each group has at most 3 vertices.

Slavík [20] considered the group Steiner problem on trees and gave an algorithm with a performance ratio of $B \cdot H(N) = B \cdot O(\ln N)$, where B is the maximum number of vertices of a group in a subtree of the root, and $H(N)$ is the N-th harmonic number.

In a recent paper, Bateman, Helvig, Robins and Zelikovsky [5] have given the first algorithm with a sublinear performance guarantee. Their algorithm (with a Java implementation available on the Internet [4]) gives an approximation ratio of $(1+\frac{\ln k}{2})\sqrt{k}$. This ratio comes from approximating the group Steiner tree by a 2-star (tree of depth 2), and then approximating the 2-star within a logarithmic factor.

1.3 Our results. We give a polynomial time algorithm that with high probability (at least $1-c$, for any fixed constant c) finds a group Steiner tree of cost within $O(\log^3 n \log k)$ of the cost of the best group Steiner tree. The main technical result is a randomized algorithm that solves the problem on trees (even this is still as hard as set-covering, by the reduction described above) with a $O(\log k \log N)$ approximation ratio. This is extended to arbitrary graphs by using the result of [3], and the final approximation ratio is then $O(\log^3 n \log k)$ (the size of the largest group, N, is at most the number of nodes n). The results of [15] used in place of Bartal's improve the performance ratio to $O(\log^2 n \log k)$ on graphs that exclude $K_{s,s}$ as a minor for some fixed constant s. An example is planar graphs that exclude $K_{3,3}$. Since planar graph distances approximate distances in the two-dimensional Euclidean plane well [6], the improvement also carries over to group Steiner problems in the plane.

Our approximation algorithm for the case of tree metrics first solves a linear programming relaxation of the group Steiner tree problem. Then an extension of randomized rounding is employed to get the solution subtree. The bound on the cost of the tree follows from the rounding process. On the other hand, to show that the solution tree actually covers all the groups with reasonable probability, we use Janson's inequality.

Our algorithm works with similar performance bounds when applied to the *errand scheduling problem* of [20], and to the service-constrained network design problems of [17, 16].

In the remainder of the paper, we first present our linear programming formulation for the problem on tree metrics and our rounding procedure, and then prove the performance guarantee. Then, we describe the reduction of the general case to the case of tree metrics, and close with applications to related problems.

2 Linear program.

We consider the group Steiner tree problem on a tree $T' = (V, E)$ with nonnegative costs c on its edges. We study the rooted version where a prespecified root node r is required to be in the solution subtree. To solve the unrooted version, we can run through the different nodes in a smallest group as the choice for the root r, and pick the best solution among these runs. Let δS denote the set of edges with exactly one endpoint in S for any subset of nodes $S \subseteq V$. We use the following linear programming relaxation of the (rooted) group Steiner tree problem:

$$
\begin{aligned}
(2.1) \qquad & \min \sum_{e \in E} c_e x_e \\
& \sum_{e \in \delta S} x_e \geq 1, \ \forall S \subseteq V \text{ such that} \\
& \quad r \in S \text{ and } S \cap g_i = \emptyset \text{ for some } i \\
& 0 \leq x_e \leq 1, \quad \forall e \in E
\end{aligned}
$$

where $r \in V$ is the root vertex.

The above linear program can be solved in polynomial time, despite the exponential number of constraints. This follows, for example from [8] and the fact that a separation oracle can be constructed easily, using a minimum cut procedure.

A more direct way to see the polynomial-time solvability of the program is by re-interpreting the constraints using the max-flow min-cut theorem. The constraints require that any cut separating the root from all the nodes of a given group must have capacity at least one. We can think of adding a new source node for this group with edges to all the nodes in it of infinite capacity and interpret the value x_e as capacity of the edge e. Then, by the constraints and the max-flow min-cut theorem, any solution x must support a flow of at least one unit from this source to the root—in other words, the program stipulates that we install sufficient capacity x such that it can support a total flow of value at least one from all the nodes in any group to the root. This can be written as a polynomial-sized set of constraints involving one set of flow variables for each group: the resulting formulation is equivalent to the above.

Let x be the optimal solution of the linear program 2.1, and T' the underlying graph (consisting of all edges e such that $x_e > 0$). We assume that T' is a tree, and give its edges capacities x_e. We denote by z^* the optimal value of the objective function.

3 Random experiment.

In this section we explain our rounding process and prove the main technical results. Our rounding may be seen as an extension of traditional randomized rounding [18] for the set cover problem to our "tree version" of the set-covering problem.

Consider the following random experiment. Assume without loss of generality that all group vertices are leaves of T' (internal group vertices can be made leaves by inserting a zero-cost edge). Similarly we may assume without loss of generality that the tree T' is binary by expanding out higher degree internal nodes with zero-cost edges into ternary nodes. For every edge $e \in E(T')$, include e in a forest T with probability x_e/x_f, where f is the edge adjacent to e and closer to r (the parent edge of e). If e is incident on r, then we include it with probability x_e (we think of a fictitious edge above r with unit flow as the parent edge of e denoting that r is always included in T). Then delete all components of T not containing the root r, as well as every edge that is not contained in a path from r to a group vertex. Let T denote the resulting tree.

LEMMA 3.1. *The expected cost of the tree T picked by the random experiment is at most z^*.*

Proof. We show that the probability of including any edge e in T is x_e, and the lemma follows from the linearity of expectation.

An edge e is included in T iff all the edges in the path from r to e, say $e, e_1, \ldots, e_p$ are picked in their respective independent random experiments. The probability of this is $\frac{x_{e_1}}{1} \cdot \frac{x_{e_2}}{x_{e_1}} \cdots \frac{x_e}{x_{e_p}} = x_e$. $\Box$

To analyze this experiment, we use Janson's inequality (see, e.g., [1]), which can be stated as follows: 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 $\mu = \sum_i \Pr[B_i]$, and ϵ be such that $\Pr[B_i] \leq \epsilon$ for all i.

THEOREM 3.1. *(Janson's inequality.) With the notation as above, if $\Delta \geq \mu(1-\epsilon)$, then*

$$\Pr\left[\bigcap_i \overline{B_i}\right] \leq e^{\frac{-\mu^2(1-\epsilon)}{2\Delta}}.$$

In our case, $\Omega = E(T')$, and $p_e = x_e/x_f$. The subsets A_i are edge-sets of paths from r to leaves belonging to a fixed group g, and $\bigcap_i \overline{B_i}$ is the event that we don't reach g in the experiment. In the sequel, we lower bound the success probability of the event of including a group's vertex by using Janson's inequality to upper bound the failure probability of the complementary event.

To prove the main result we need a simple lemma.

LEMMA 3.2. *If T and T' are trees that differ only in the capacity of an edge e, such that $x_T(e) \geq x_{T'}(e)$, then for any group the success probability of including a vertex from the group is no greater in T' than in T.*

Proof. Let e be an edge of T with exactly two child edges f and g (We assumed the trees to be binary without loss of generality). The interesting case is when both the subtrees contain nodes of the group. We have

$$\Pr[\text{fail in } e\text{'s subtree}] = 1 - x_e + x_e\left(1 - \frac{x_f}{x_e}\right)\left(1 - \frac{x_g}{x_e}\right).$$

The $1 - x_e$ in the sum is the chance that the edge e is not included in the subtree containing the root, and the remaining term is the chance, given that e is included, that neither of its two child edges is included. If x_e is decreased by a little, the value of the above expression will decrease as well. But, we can think of the success probability in the subtree of T under the edge e as the probability of another equivalent single edge e' with $x_{e'} = \Pr[\text{succeed in the subtree below } e']$. Now it is clear that if $x_{e'}$ is decreased, then probability of success in T will also decrease. $\Box$

THEOREM 3.2. *If we run the random experiment on a feasible solution to the LP (1), then for every group G, the success probability of including a vertex from G in the randomly chosen tree T is $\Omega(1/\log N)$ where N is the maximum size of a group.*

Proof. Consider the tree spanned by the paths from r to the leaves of a fixed group g. We will transform this tree into one where it will be easier to estimate the success probability. In the process we only decrease the success probability, so that a lower bound carries over to the original tree.

Since this tree comes from a feasible solution x to the LP (1), as argued before, the capacities x support a flow of at least one between r and the nodes of g. Starting from the root and going down the tree, decrease the capacity on the edges so that the flow from r to the nodes of this group is exactly 1 at the end. By lemma 3.2, this only decreases the success probability at each decreasing step.

We now have a tree with flow of 1 between the root and g. Round down all the capacities to next powers of 2. This in the worst case halves the flow from r to g. (Note that flow constraints need not be satisfied anymore.) Let $N_g = |g|$ be the number of leaves in this tree, and let $d = \lceil \log N_g \rceil$. Delete all edges of capacity less than $1/2^{d+2}$. This reduces the flow again, but since there were only N_g leaves to begin with, the total flow we lose now is at most $N_g 2^{-(d+2)} \leq 1/4$. Assume that the flow is now exactly $1/4$: otherwise delete some leaves until this is true. Finally, shrink every edge (except the pendant edges) that is preceded (on the path from the root) by another edge of the same weight. This doesn't change the success probability, and reduces the depth of the tree to d. We abuse notation slightly and continue to denote the resulting rounded edge capacity values carrying a flow of value exactly $1/4$ to group g by x.

We will show that in this tree (henceforth referred to as T^{rnd}), $\Delta = O(\log N_g)$. Consider a pendant edge e, and let

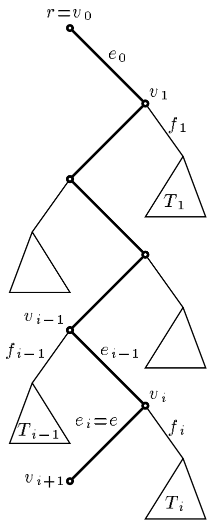
$$\Delta_e = \sum_{f \sim e} \frac{x_e x_f}{x_g},$$

where g is the least common ancestor of e and f. Recall that $f \sim e$ implies that f is another pendant edge to a node in this group whose path to r shares at least one edge with the path from e to r. Thus Δ_e is the contribution to Δ of the edge e, and $\Delta = \sum_e \Delta_e$.

Suppose edge e goes from level i to level $i+1$ of T^{rnd}, and denote by $e_j = v_j v_{j+1}$ the edges on the path from the root to e ($j = 0, \ldots, i$, $v_0 = r$ and $e_i = e$). Further, let T_j be the subtree of T^{rnd} whose root is v_j, and which does not include e_j (See figure). Let f_j be the total flow from subtree T_j to the root. Then we have

$$\Delta_e = \sum_{j=1}^{i} \frac{x_e f_j}{x_{e_{j-1}}},$$

where we again assume $x_{e_{-1}} = 1$.



Since the capacities on these edges are a result of rounding down to powers of 2, it follows that $f_j \leq 2x_{e_{j-1}}$. (Indeed, assume $f_j > 2x_{e_{j-1}}$. In the rounding, $x_{e_{j-1}}$ was at most halved, and so before the rounding it must have been true that $f_j > x_{e_{j-1}}$. But this would contradict the flow constraints that were satisfied before the rounding.) So,

$$\Delta_e \leq x_e \sum_{j=0}^{i} 2 = 2(i+1)x_e.$$

Therefore,

$$\Delta = \sum_e \Delta_e \leq \sum_e 2(d+2)x_e = \frac{1}{2}(d+2) \leq \log N_g.$$

Now we can apply Janson's equality with $\mu = \frac{1}{4}$, $\Delta = \log N_g$ and $\epsilon = \frac{1}{2}$. We get

$$\Pr[\text{fail to reach } g] \leq e^{-\frac{1}{64\Delta}} = e^{-\frac{1}{64 \log N_g}} = 1 - \frac{1}{64 \log N_g} + \frac{1}{4096 \log^2 N_g} - \cdots,$$

and we see that we will reach group g with probability of about $\frac{1}{64 \log N_g}$. Since $N_g \leq N$, the maximum size of any group, the theorem follows. $\square$

Our analysis in the above theorem is essentially tight as can be seen by considering T^{rnd} to be a complete binary tree where all capacities in a level are equal and this decreases by a factor of two as we go down the tree. The success probability for this tree can be shown (via a recurrence relation) to be $O(\frac{1}{\log n})$ where n is the number of leaves.

4 Building the Steiner tree.

Now we show how to use the result of the previous section and amplify the probability of success, while keeping the final cost low.

When we pick a single tree randomly, the probability that it covers g is at least $\frac{1}{64 \log N}$ for any group g. If we pick $64 \log N$ trees in the same way, we will cover any group with a constant probability of about $1 - 1/e$. If we pick about $64 \log N \ln 2k$ trees, the probability of missing a given group is at most $1/2k$, and by subadditivity, the probability of missing any group is at most $1/2$. So, if we have picked $A \log N \ln 2k$ trees, where, for example $A = 128$, the union of these trees will cover every group with probability at least $3/4$.

The total cost of those trees is at most the sum of their costs. Denote this by $c(\mathcal{T})$. Then by Markov's inequality,

$$\Pr[c(\mathcal{T}) \geq 4A \log N \log 2k \; z^*] \leq 1/4.$$

Thus the tree $\mathcal{T}$ has low cost with probability at least $3/4$.

Since the two "good" events each occupy at least $3/4$ of the probability space, they must overlap in at least $1/2$, and so with probability at least $1/2$, we cover all groups with a tree of cost $O(\log N \log k \; z^*)$. Since z^* is a lower bound on the cost of an optimal group Steiner tree, we obtain the following theorem.

THEOREM 4.1. *There is a randomized polynomial time algorithm that, with probability at least $1/2$, finds a group Steiner tree on an underlying graph which is a tree, of cost no more than $O(\log N \log k)$ times the minimum, where N is the maximum size of a group and k is the number of groups.*

Note that we have also proved the following:

COROLLARY 4.1. *The integrality gap of the LP 2.1 is at most $O(\log n \log k)$ when the underlying graph is a tree.*

5 General graphs.

DEFINITION 5.1. *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 [3] proved

THEOREM 5.1. *Every weighted n-node connected graph G can be α-probabilistically approximated by a set of weighted trees, where $\alpha = O(\log^2 n)$. 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 group Steiner tree problem on a general graph G, first find a set of trees and the distribution on them that $O(\log^2 n)$-approximates G. Then pick a tree from the distribution and solve the group 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 at most $O(\log^3 n \log k) z^*$. By Markov's inequality we can say that with high probability (at least $1 - \epsilon$, for any constant ϵ) the cost of our tree is at most $O(\log^3 n \log k)$ times the cost of the optimal tree.

THEOREM 5.2. *The algorithm described above with high probability finds a group Steiner tree of cost $O(\log^3 n \log k)$ times the cost of the optimal tree.*

6 Other formulations and applications.

In this section, we first sketch the improvement in the case of graphs that exclude small minors. Then we give two more applications of our results. One is to a bicriteria network design problem that involves location-theoretic constraints, and the other to the errand scheduling problem which generalizes the traveling salesman problem.

6.1 Improved metric approximations.

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

THEOREM 6.1. *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 $O(\log^2 n \log k)$ for such graphs.

Since distances in the Euclidean plane can be approximated to within a factor of 2 by a planar graph [6], 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 group Steiner tree only changes by a factor of $1 + \epsilon$ for any constant ϵ as in [2].

6.2 Service constrained network design problems. Marathe et al. [17, 16] study the following problem: given an undirected graph $G = (V, E)$ with two different cost functions c (modeling the service cost) and d (modeling the construction or communication cost) for each edge $e \in E$, and a bound $\mathcal{S}_v$ (on the service distance for each vertex v). The goal is to find a minimum d-cost tree such that every node v in the graph is *serviced* by some node in the tree, i.e. every node v is within distance $\mathcal{S}_v$ (under the c-costs) of some node in the tree.

An (α, β)-bicriteria approximation for such a problem is an algorithm which finds a solution whose cost under d is within a β factor of the optimal one that satisfies the budget constraints, and whose budget constraints are not exceeded by more than a factor of α. [17, 16] give a $(1, 2\Delta)$-approximation algorithm, where Δ is the maximum service degree of any node, the maximum number of nodes that can service any given node.

We observe that if the first approximation factor α is fixed at 1, this problem is equivalent to the group Steiner tree problem.

First we reduce their problem to a group Steiner tree problem. Define n groups g_v, one for each vertex v of G. Let g_v consist of all vertices w that are within the budget (c-)distance of v,

$$g_v = \{w \in V \mid c(vw) \leq \mathcal{S}_v\},$$

where $\mathcal{S}_v$ denotes the value of the service constraint for v. Now any group Steiner tree will satisfy the service constraints, and conversely, any tree that services all vertices within the budget will be a group Steiner tree.

Note that our algorithm improves the approximation guarantee of [16] to $(1, O(\log^3 n \log k))$, where $n = |V|$ and k is the maximum service degree of any vertex (in particular, $k \leq n$).

Next we reduce the group Steiner problem to a version of their network design problem. Assume without loss of generality that the groups are disjoint. Let the weights in the given graph represent the d-costs. Define the c-costs as follows: between a pair of nodes in the same group, the c-cost is zero, and between all other pairs, the c-cost is unit. The service radius $\mathcal{S}_v$ is set to zero for every node in every group and to n for all other nodes. Any solution output by an (α, β)-approximation algorithm for this service-constrained network design problem must include at least one node from every group, and is therefore a group Steiner tree of cost at most α times the minimum.

6.3 Errand scheduling. This problem was formulated by Slavík [20] as a generalization of the traveling salesman problem. Let U be a set of tasks, and G an edge weighted graph. A set of tasks S_v is associated with each vertex v of G. The goal is to find a shortest closed walk in G such that each element of U is contained in at least one visited vertex.

We can interpret an instance of the errand scheduling problem as an instance of the group Steiner problem, and then it is easy to see that the best group Steiner tree and the best errand tour differ in cost by at most a factor of 2. Therefore, our algorithm also gives a $O(\log^3 n \log k)$ approximation for the errand scheduling problem.

7 Conclusion.

We have presented the first algorithm with a polylogarithmic approximation ratio for the group Steiner problem.

The only known lower bounds for these problems are the ones that arise from the hardness of the set-covering problem. It is a natural open problem then, to reduce the approximation ratio to $O(\log n)$. Another interesting question is whether better (say, constant ratio) results are possible for the Euclidean case.

References

[1] N. Alon and J. Spencer. *The Probabilistic Method.* Wiley Interscience, New York, 1992.

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

[3] 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.

[4] C. D. Bateman, C. S. Helvig, G. Robins, and A. Zelikovsky, 1997. Java program, available at `http ://www.cs.virginia.edu/~robins/groupSteiner/`.

[5] C. D. Bateman, C. S. Helvig, G. Robins, and A. Zelikovsky. Provably good routing tree construction with multi-port terminals. In *Proceedings of ACM/SIGDA International Symposium on Physical Design*, Apr. 1997.

[6] 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.

[7] U. Feige. A threshold of $\ln n$ for approximating set cover. In *Proceedings of the 28thAnnual ACM Symposium on Theory of Computing*, pages 314–318, May 1996.

[8] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer, 1988.

[9] E. Ihler. Bounds on the quality of approximate solutions to the group Steiner problem. In *Graph-Theoretic Concepts in Computer ScienceWG90*, volume 484 of

Lecture Notes in Computer Science, pages 109–118. Springer, 1991.

[10] E. Ihler. The complexity of approximating the class Steiner tree problem. In *Graph-Theoretic Concepts in Computer ScienceWG91*, volume 570 of *Lecture Notes in Computer Science*, pages 85–96. Springer, 1992.

[11] E. Ihler. The rectilinear class Steiner tree problem for intervals on two parallel lines. *Math. Programming, Ser. B*, 63:281–296, 1994.

[12] E. Ihler, G. Reich, and P. Widmayer. On shortest networks for classes of points in the plane. In *Computational Geometry—Methods, Algorithms and ApplicationsCG'91*, volume 553 of *Lecture Notes in Computer Science*, pages 103–111. Springer, 1992.

[13] E. Ihler, G. Reich, and P. Widmayer. Class Steiner trees and VLSI design. Draft Paper, `http://www.gmd.de/People/Edmund.Ihler/`, Oct. 1996.

[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 submitted to SODA 98, Jul. 1997.

[16] M. V. Marathe, R. Ravi, and R. Sundaram. Improved results on service-constrained network design problems. to appear in *Proc. of the DIMACS workshop on network design.*

[17] M. V. Marathe, R. Ravi, and R. Sundaram. Service-constrained network design problems. *Nordic J. Comput.*, 3:367–387, 1996.

[18] P. Raghavan and C. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[19] G. Reich and P. Widmayer. Beyond Steiner's problem: A VLSI oriented generalization. In *Graph-Theoretic Concepts in Computer ScienceWG89*, volume 411 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 1990.

[20] P. Slavík. The errand scheduling problem. manuscript, available at `http://www.cs.buffalo.edu/~slavik/papers/`, 1996.

[21] P. Winter. Steiner problem in networks: A survey. *Networks*, 17:129–167, 1987.