Rounding via Trees: Deterministic Approximation Algorithms for Group Steiner Trees and k-median

Moses Charikar* Chandra Chekuri† Ashish Goel‡ Sudipto Guha§

Computer Science Department

Stanford University

Abstract

Most optimization problems on an undirected graph reduce in complexity when restricted to instances on a tree. A recent result [3] for probabilistically approximating graph metrics by trees such that no edge stretches (in an expected sense) by more than a factor of $O(\log^2 n)$ has resulted in several approximation algorithms which exploit the ease of solving problems on trees. The tree construction in [3] is inherently randomized and a natural question to ask is whether approximation algorithms which use this construction can be derandomized.

We present a general framework for derandomizing approximation algorithms which use the above tree construction as a primitive. Let Π be a graph optimization problem which can be expressed as an integer program with 0-1 variables $\bar{x}(e)$ for each edge and with an objective function expressible as $\sum_{e \in E} \bar{x}(e) \cdot c(e)$. Given an optimal solution to LP relaxation of the integer program, we show that we can map the solution to a tree approximation of the graph with at most an $O(\log n \log \log n)$ blow-up in the value of the objective function. This can then be coupled with a deterministic LP rounding procedure on the tree to obtain a completely *deterministic* approximation algorithm for Π.

We apply this framework to the group Steiner tree problem, the k-median problem and the minimum communication cost spanning tree problem. We obtain the first *deterministic* approximation algorithms for these problems which match the best known randomized approximations. To do this, we design novel deterministic schemes to round the LP solution for these problems on trees. For the group Steiner tree problem we obtain an approximation ratio of $O(\log^2 n \log k \log \log n)$, and for the k-median problem we obtain a ratio of $O(\log k \log \log k)$. We believe that our rounding procedure on trees for the group Steiner tree problem is of independent interest.

1 Introduction

The powerful result of Bartal [3] on probabilistic approximation of metric spaces has recently led to either the first known or improved approximation algorithms for several NP-Complete graph optimization problems. Some of these problems include the group Steiner tree problem [12], the k-median problem, the buy at bulk network design problem [2], and the minimum communication spanning tree problem [24]. All these problems have a similar structure: given a graph G with a cost metric c defined on the edges, find the minimum cost subset of edges satisfying certain constraints. Bartal's result can be used to approximate the metric space c by a tree such that, for any edge (u, v) in the graph, the *expec-*

*Supported by a Stanford Graduate Fellowship, an ARO MURI Grant DAAH04-96-1-0007 and NSF Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Email: moses@cs.stanford.edu.

†Supported by an IBM Cooperative Fellowship, an ARO MURI Grant DAAH04-96-1-0007 and NSF Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Email: chekuri@cs.stanford.edu.

‡Supported by Center for Telecommunications at Stanford, ARO Grants DAAH04-95-1-0121/ DAAG55-97-1-0221, and NSF Grants CCR9304971/ CCR9307045. Email: agoel@cs.stanford.edu.

§Supported by an ARO MURI Grant DAAH04-96-1-0007 and NSF Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Email: sudipto@cs.stanford.edu.

ted distance between u and v in the tree is at most $O(\log^2 n)$ times its length in the graph. Thus many optimization problems where the objective is to minimize some function of the edge lengths (linear combinations are very common) can be solved now on the tree metric space losing only a poly-logarithmic factor. Due to the simple structure of trees, many NP-Complete problems can be solved exactly or approximated, in polynomial time on them. Buy at bulk network design [2] is an example of a problem where the objective function is non-linear, yet the paradigm can be applied to obtain poly-logarithmic approximations. Recently, Bartal [4] has improved the expected stretch factor in his tree construction to $O(\log n \log \log n)$, thereby improving approximation algorithms that use his result. Since the first step in applying this paradigm consists of constructing a tree probabilistically, all the approximation algorithms which result are randomized. Our focus in this paper is to *derandomize* these results.

One possible approach towards derandomizing Bartal's result would be to obtain a probability distribution on *polynomially* many trees such that the expected stretch of any edge is small. In the following theorem we guarantee the existence of such a distribution.

Theorem 1.1 *Let $G = (V, E)$ be an undirected graph with a weight function c on the edges that satisfies triangle inequality. For any such graph, there exists a $z^* = O(\log n \log \log n)$, $(m+1)$ trees (where m is the number of edges) $T_1, T_2, \ldots, T_{m+1}$, and positive real weights $\alpha_1, \alpha_2, \ldots, \alpha_{m+1}$ such that $\sum_{i=1}^{m+1} \alpha_i = 1$ and*

$$\sum_{i=1}^{m+1} \alpha_i \cdot d_{T_i}(u, v) \leq z^* \cdot c(u, v) \quad \text{for each } (u, v) \in E(G)$$

where $d_{T_i}(u, v)$ is the distance between u and v in T_i.

Our proof of Theorem 1.1 is existential and obtaining a constructive version is an important question [1]. In this paper we take an alternative approach towards derandomizing algorithms obtained in this framework.

The General Paradigm

Let $G = (V, E)$ be an *undirected* graph with a cost function c associated with the edges. Suppose we need to solve a problem which involves finding a minimum cost set of edges (the objective function is the sum of the costs of the chosen edges) satisfying certain connectivity constraints. Many applications in network design and facility location fall in this category of problems. Some examples are Steiner trees, group Steiner trees, capacitated and uncapacitated facility location, and k-median. Since we are interested in connectivity, we can work with the metric induced by the shortest path distances in G. One of the standard techniques for solving such a problem is to round a linear programming relaxation of the problem. Let Π be a graph optimization problem which can be expressed as an integer program with 0-1 variables $\bar{x}(e)$ for each edge and with an objective function expressible as $\sum_{e \in E} \bar{x}(e) \cdot c(e)$. We use the following series of steps to solve Π.

1. Solve the LP relaxation of the problem and let $x(e)$ be the fractional values assigned to the edges. Then $\sum_{e \in E} x(e) \cdot c(e)$ is a lower bound on the optimal.

2. Deterministically construct a tree T such that $\sum_{(u,v) \in E} x(u, v) \cdot d_T(u, v) \leq \alpha \sum_{(u,v) \in E} x(u, v) \cdot c(u, v)$ is satisfied, where $d_T(u, v)$ is the distance between u and v in T. The LP solution carries over to the tree T.

3. Round the LP solution on the tree to an integral solution losing at most a factor of β.

4. Translate solution on T to G obtaining an $\alpha\beta$ approximation.

Observe that we preserve only a *specific* linear combination of edge lengths. This is what lets us create one tree in a deterministic fashion. In Section 2 we show how we can do step 2 of above with $\alpha = O(\log^2 n)$ using ideas from the paper of Garg *et al.* [13]. We can improve this to $O(\log n \log \log n)$ using an idea first developed by Seymour [21] and subsequently applied by Even *et al.* in their spreading metric paper [10], Klein *et al.* [17], and others. This was obtained by us after hearing about Bartal's improved result for probabilistic approximation of metric spaces [4]. We later discovered that the above mentioned tree construction has also been obtained independently by Bartal [4]. Step 3 is problem specific. We now describe three problems to which this paradigm can be applied.

Applications

We consider three important problems: the group Steiner tree problem, the k-median (it is also referred to as the p-median problem in the literature),

[1] This question has been recently settled in [8]. They give a polynomial time procedure that produces a probability distribution on $O(n \log n)$ trees with an $O(\log n \log \log n)$ distortion. Their result subsumes Theorem 1.1 and also provides alternative proofs of the results in [3, 4]

and the minimum communication spanning tree problem and obtain deterministic rounding procedures for all of them.

The group Steiner tree problem is the following. Let G be an undirected graph with edge weights. Let $S = \{t_1, t_2, \ldots, t_k\}$ be a set of k subsets of V. The objective is to find a minimum weight tree which has at least one vertex from each of the groups. This problem is hard to approximate within logarithmic factors even on stars (via reduction from set cover [11]). The problem was introduced by Reich and Widmayer [20] and finds applications in VLSI design. Some related problems are discussed in [14]. Bateman *et al.* [6] gave the first non-trivial approximation algorithm with a ratio of $(1 + \ln k/2)\sqrt{k}$. Charikar *et al.* [7] using a reduction to directed Steiner trees gave an algorithm with a ratio of k^ϵ for any fixed $\epsilon > 0$. They also showed that it is possible to obtain an $O(\log^2 k)$ approximation in quasi-polynomial time. Garg *et al.* [12] gave a randomized approximation algorithm with a ratio of $O(\log^3 n \log k)$ using probabilistic approximation of metric spaces. They use a novel variant of randomized rounding to solve the problem on trees. We obtain a *deterministic* rounding scheme on trees. Plugging this rounding scheme into our paradigm, we obtain a deterministic algorithm with a ratio of $O(\log^2 n \log k \log\log n)$. We believe our deterministic rounding on the tree is of independent interest.

The k-median problem has been studied extensively over the last thirty years. Given a graph G, the objective is to select k centers (vertices) such that the sum over all vertices of the distance from the vertex to its nearest center is minimized. The problem has great practical relevance in diverse areas such as facility location, information retrieval, and data mining. It arises in such varied contexts because it is a good heuristic to cluster points in a metric space. Constant factor approximations are known, provided the number of centers can be relaxed by a small factor. Lin and Vitter [18] provide a $2(1 + \frac{1}{\epsilon})$ approximation using $(1 + \epsilon)k$ centers. Several papers describe polynomial time algorithms to solve the problem on trees (see [15, 22]). Combining this with the paradigm of probabilistic approximations, a randomized $O(\log n \log\log n)$ approximation can be obtained. We show how to obtain a deterministic result by giving a deterministic rounding procedure on the tree. Note that the fact that the problem can be solved exactly on a tree in polynomial time does not imply that there is no integrality gap for the LP on the tree. We show that the fractional solution can be rounded by losing at most a factor of 2 provided the tree satisfies certain properties which are guaranteed by our construction. Thus we obtain a deterministic approximation algorithm for the k-median problem with a ratio of $O(\log n \log\log n)$. We further improve the ratio to $O(\log k \log\log k)$ using ideas from Lin and Vitter [18].

Finally we point out that the problem of minimum communication spanning trees can be derandomized directly by our tree construction procedure. The rest of the paper is organized as follows. Section 2 describes the deterministic construction which preserves a given linear combination of edges to within a poly-logarithmic factor. Sections 3 and 4 describe the rounding procedures for the group Steiner tree problem and the k-median problem respectively.

2 Deterministic construction of k-HSTs

Definition 1 (Bartal [3]) *A k-hierarchically well-separated tree (k-HST) is defined as a rooted weighted tree with the following properties*

1. *The edge weight from any node to each of its children is the same.*
2. *The edge weights along any path from the root to a leaf are decreasing by a factor of at least k.*

Let $G = (V, E)$ be a undirected graph with two positive weight functions c and x defined on the edge set E. In the sequel, distance in the graph is defined with respect to the weight function c. We will assume that for any edge $(u, v) \in E$, that $c(u, v) = d_G(u, v)$. We will further assume that $x(e) > 0$. Let $\Delta(G)$ denote the diameter of G. The following theorem is a modified version of the ball growing theorem in the paper of Garg, Vazirani, and Yannakakis on multicuts [13].

Theorem 2.1 (Garg *et al.* [13]) *Given a graph G with two positive weight functions x and c on the edges, and a value $k > 1$, it is possible to partition G in two induced subgraphs G_1 and G_2, in polynomial time, such that*

1. $\Delta(G_1) \le \Delta(G)/k$.
2. *Let* $E(G_1, G_2) = \{(u, v) | u \in G_1, v \in G_2\}$,

$$\sum_{e \in E(G_1, G_2)} x(e) \le \frac{O(k \log n)}{\Delta(G)} \sum_{e \in E(G_1) \cup E(G_1, G_2)} x(e) \cdot c(e).$$

Define $\mathsf{vol}(G) = \sum_{e \in E(G)} x(e) \cdot c(e)$. The corollary below is a direct consequence of Theorem 2.1.

Corollary 1 *For any $k > 1$, we can partition $V(G)$ in to $V_1, V_2, \ldots, V_p$ such that*

1. *For $1 \le i \le p$, $\Delta(G_i) \le \Delta(G)/k$ where G_i is the induced subgraph on V_i.*
2. $\displaystyle\sum_{e \in F} x(e) \le \frac{O(k \log n)}{\Delta(G)} \mathsf{vol}(G)$,
   where $F = \{(u, v) \in E | u \in V_i \Rightarrow v \notin V_i\}$.

We use the above theorem to construct a k-HST for G which preserves the weighted sum of the edge lengths to within a $O(\log^2 n)$ factor.

Theorem 2.2 *For any $k > 1$ we can construct in polynomial time a k-HST, $T = (V', E_T)$ where every vertex of G occurs as a leaf of T and the following is true.*

- *For a node i in T let T_i be a subtree rooted at i and let G_i be the subgraph of G induced by the leaves of T_i. For every i, the following holds*
  1. *$d_{T_i}(u, v) \ge c(u, v)$ where $d_{T_i}(u, v)$ is the distance between u and v in T_i.*
  2. *The edges from i to its children have a length of $\Delta(G_i)/2$.*
  3. *The distance from i to any leaf is at most $\frac{k}{2(k-1)}\Delta(G_i)$.*
- $$\sum_{(u,v) \in E} x(u, v) \cdot d_T(u, v) \le O(\frac{k^2}{k-1} \log_k n \log n) \sum_{(u,v) \in E} x(u, v) \cdot c(u, v).$$

Proof: We construct the tree as follows. We first contract all edges e such that $c(e) \le \frac{\Delta(G)}{2kn}$. Let G' be the resulting graph. Using Corollary 1, we partition G' into $G'_1, \ldots, G'_p$ with a parameter $2k$ such that $\Delta(G'_i) \le \Delta(G)/2k$. For $1 \le i \le p$ let G_i be the graph obtained by expanding the contracted edges of G'_i. It is easy to see that $\Delta(G_i) \le \Delta(G'_i) + n \cdot \frac{\Delta(G)}{2kn} \le \Delta(G)/2k + \Delta(G)/2k \le \Delta(G)/k$. We recursively build the trees for each G_i. Let $T_1, \ldots, T_p$ be the respective trees. We create a tree T for G by adding a new root r and connecting it to the roots of $T_1, \ldots, T_p$ by edges of length $\Delta(G)/2$. Inductively the length of the longest path from the root to any leaf in each of T_i is at most $\frac{k}{2(k-1)}\Delta(G_i)$. Therefore the length of the longest path from the root in T is bounded by

$$\begin{aligned} \frac{1}{2}\Delta(G) + \frac{k\Delta(G_i)}{2(k-1)} &\le \frac{1}{2}\Delta(G) + \frac{\Delta(G)}{2(k-1)} \\ &\le \frac{k}{2(k-1)}\Delta(G) \end{aligned}$$

It is also easy to see that the distance in T between any two leaves u and v such that $u \in T_i$ and $v \in T_j$, $i \ne j$, is at least $\Delta(G)$ which is at least their distance in G. Since the diameter of the graphs falls by a factor of k at each level, the construction also ensures that the resulting tree is a k-HST. The only property left to verify is the bound on the sum of the distances in the tree T. Let $\mathsf{cost}(T) = \sum_{(u,v) \in E} x(u, v) \cdot d_T(u, v)$. Notice $V(G) \subseteq V(T)$. Let $F = \{(u, v) \in E | u \in V_i \Rightarrow v \notin V_i\}$. Then it is clear that

$$\begin{aligned} \mathsf{cost}(T) &= \sum_{i=1}^{p} \mathsf{cost}(T_i) + \sum_{(u,v) \in F} x(u, v) \cdot d_T(u, v) \\ &\le \sum_{i=1}^{p} \mathsf{cost}(T_i) + \frac{k \cdot \Delta(G)}{k-1} \sum_{(u,v) \in F} x(u, v) \\ &\le \sum_{i=1}^{p} \mathsf{cost}(T_i) + \frac{k}{k-1} O(k \log n) \mathsf{vol}(G'). \end{aligned}$$

Observe that we have G' instead of G in the above equation. From the above equation it follows that $\mathsf{cost}(T)$ is bounded by $O(\frac{k^2}{k-1} \log n)$ times the sum of the volumes of all graphs used for partitioning. Instead of summing up over the volumes of graphs, we can sum up over all edges e, the quantity $q(e) \cdot x(e) \cdot c(e)$ where $q(e)$ is the number of levels an edge participates in before it is cut. We claim that $q(e) = O(\log_k n)$ for all e. This immediately gives the desired bound on the cost of the tree T. To prove the claim observe that an edge stays contracted as long as the current diameter of the graph being partitioned is greater than $2kn \cdot c(e)$. Since the diameter of the graphs decreases geometrically by a factor of k, it follows that an edges contributes to only $O(\log_k n)$ levels. □

Remark 1 *The tree constructed in Theorem 2.2 can be modified to obtain a different tree T' whose vertex set is identical to that of G such that $d_{T'}(u, v) \ge c(u, v)$ for all $(u, v) \in E(G)$, and $\sum_{(u,v) \in E(G)} x(u, v) \cdot d_{T'}(u, v) \le O(\log^2 n) \sum_{(u,v) \in E} x(u, v) \cdot c(u, v)$.*

After hearing about Bartal's [4] improvement to probabilistic approximations, we obtained the following theorem for preserving a fixed linear combination of edges of a graph to within an $O(\log n \log \log n)$ factor on a tree. We later discovered that this theorem has also been obtained independently by Bartal [4]. The tree construction is very similar to the one presented above and the details of the construction can be found in [4]. The idea for improving the analysis to obtain an improved ratio was introduced by Seymour [21] and has been subsequently applied by Even *et al.* [10] for spreading metrics, Klein *et al.* [17], and others.

Theorem 2.3 *A k-HST satisfying properties* 1 *to* 3 *of Theorem 2.2 can be constructed in polynomial time such that*

$$\sum_{u,v\in V(G)} x(u,v)\cdot d_T(u,v) = O\left(\frac{k^2}{k-1}\log n\log\log n\right) \sum_{u,v\in V(G)} x(u,v)\cdot c(u,v)$$

For planar graphs Bartal *et al.* [5] give an $O(\log n)$ probabilistic approximation by tree metrics, as well as a procedure to approximate a fixed linear combination of edges within an $O(\log n)$ factor on a tree. Their result is based on a stronger partitioning procedure than that of [13] for graphs with shallow excluded minors presented by Klein *et al.* [16]. We can use the improved partitioning procedure in our tree construction. Thus our approximations for the group Steiner tree problem and the k-median problem improve by a factor of $\log\log n$ on planar graphs.

Lower Bounds: Consider the metric of the $n\times n$ grid, denoted by $G(n,2)$.

Definition 2 *A tree T is said to be a* super-spanning *tree of the grid $G = G(n,2)$, if* **(1)** *All vertices of G are leaves of T, and* **(2)** *$d_T(x,y)\geq d_G(x,y)$ for all $x,y\in V(G)$.*

Konjevod, Ravi and Salman [5] extended the results of Alon, Karp, Peleg and West [1] to prove the following theorem, which gives a lower bound of $\Omega(\log n)$ for the deterministic tree construction in Theorem 2.3, even for planar graphs.

Theorem 2.4 ([1, 5]) *If T is a super-spanning tree of $G = G(n,2)$, and an edge is chosen uniformly at random from $E(G)$, then $E[d_T(e)]\geq\Omega(\log n)$.*

3 The group Steiner tree problem

The undirected group Steiner tree problem $G = (V,E,c,r,S)$ consists of a vertex set V, an edge set E, a cost metric c defined on E, and a set S of groups, where each group is a subset of V. Also, r is a distinguished vertex (the root) in G. A solution to the problem is a subtree T of G such that $T\cap t\neq\emptyset$ for all $t\in S$ and T contains r. The cost of the solution is $\sum_{e\in T} c(e)$ and the objective is to find a minimum cost solution. In this section we give a *deterministic* $O(\log^2 n\log k\log\log n)$ approximation algorithm for this problem, where $k = |S|$. Without loss of generality we can assume that G is a complete graph, and that c satisfies the triangle inequality.

We solve the same LP relaxation as [12], but we solve it on the original graph. We then invoke Theorem 2.3 to translate this LP to the tree (Section 3.1). Finally, we use the method of conditional probabilities (Section 3.2) to give a deterministic rounding procedure which produces a solution that is within a factor of $O(\log n\log k)$ of the LP relaxation.

Bartal's [4] recent improvement can be plugged into the result of Garg *et al.* [12] to obtain the same approximation ratio as we obtain here. However our result is deterministic, and we believe that our rounding technique is of independent interest.

3.1 Obtaining a tree LP for group Steiner trees

The group Steiner tree Problem can be modeled as the following 0-1 integer program. It will help to think of f as flows and x as capacities. The idea behind the formulation is that there is a flow of value 1 from the root to each group (f_t denotes flow for group t), in the graph with edge capacities defined by x. The flows are directed even though the graph is not. We relax the integer program to obtain the following LP.

$$\min \textstyle\sum_{(u,v)\in E} c(u,v)\cdot x(u,v)$$

$$\begin{array}{rcll}
\sum_{u\neq v} f_t(u,v) - \sum_{v\neq w} f_t(v,w) & = & 0 & \\
 & & & t\in S, v\in V-\{r,t\} \\
f_t(t,v) & = & 0 & t\in S, v\in V \\
\sum_{u\in t}\sum_{v\in V} f_t(v,u) & = & 1 & t\in S \\
f_t(u,v)+f_t(v,u) & \leq & x(u,v) & u,v\in V \\
x(u,v) & = & x(v,u) & \\
f_t(u,v) & \geq & 0 &
\end{array}$$

Using Theorem 2.3, we obtain a spanning tree T of G such that

$$\sum_{(u,v)\in E(G)} d_T(u,v)x(u,v) \leq O(\log n\log\log n)\cdot \sum_{(u,v)\in E(G)} c(u,v)x(u,v)$$

where $d_T(u,v)$ is the length of the path $P_T(u,v)$ between vertices u and v in the tree T. For each group $t \in S$ and any edge $e \in T$, define $f_t^T(e) = \sum_{e \in P_T(u,v)} f_t(u,v)$. Short circuit the flows f_t^T by eliminating cycles – this can only decrease $f_t^T(e)$ on any edge $e \in T$. Define $x_T(e) = \max_{t \in S} f_t^T(e)$. Since T is a spanning tree of G the flows f_t^T can also be looked upon as flows in G.

Lemma 3.1 *The flows f_t^T define a feasible solution to the LP for group Steiner trees. Further,*

$$\sum_{e \in T} x_T(e)c(e) \leq O(\log n \log\log n) \sum_{(u,v)\in E} c(u,v)x(u,v).$$

We define $f_t^\tau = \sum_{u: u \in t \cap \tau} \sum_{v:(v,u)\in T} f_t^T(v,u)$. Informally, f_t^τ is the amount of flow on tree τ for group t. Due to technical reasons we will need the following two properties of the LP: that for all $e \in T$, $t \in S$, the flow $f_t^T(e)$ is a multiple of $1/n$, and that $f_t^T = 1$ for all $t \in S$. The first property is ensured by doubling all flows and then rounding them down to multiples of $1/n$. This doubles the cost of the LP solution by at most a factor 2. The second property can be obtained by reducing flows if necessary. Let z^* represent the cost of the above LP.

3.2 Rounding the LP on the tree

Our rounding scheme is essentially a derandomization of the rounding scheme give by Garg *et al.* [12]. We use the method of conditional probabilities in conjunction with a pessimistic estimator [19].

Let X be any subtree of T, such that $r \in X$. Define the *density* of the tree X rooted at r to be its cost divided by the number of groups satisfied by X. We will maintain a pessimistic estimator $\tilde{D}$ for the density that we hope to obtain. We will look at edges one by one, and either pick the edge or discard it, making sure that $\tilde{D}$ does not increase. After examining all the edges, we obtain a tree X with density no worse than the pessimistic estimator that we had at the very beginning. We show that our pessimistic estimator, and therefore the density of X, is at most $z^* \log(2n)/k$. Of course not all groups might get satisfied by X, so this procedure will be repeated till all groups get satisfied. Repeating the procedure will cost us an additional factor of $\log k$ in the approximation ratio. We will now focus on the construction of X.

The algorithm only examines edges that are currently incident on the root r. If an edge e gets chosen, it gets contracted into the root and all flows in the subtree attached to this edge get multiplied by $1/y_e$ where y_e is the *current* capacity of the edge (since the flows on edges change, so do the capacities. We use x and f for the initial values, and y

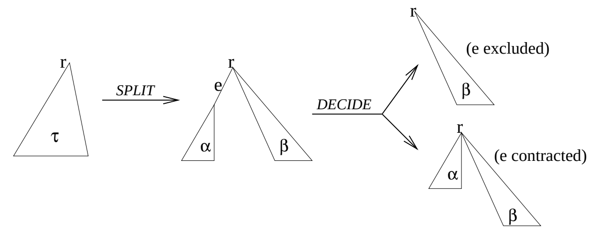

Figure 1: One iteration of ROUND.

and g for the current ones.). The algorithm maintains a set of disjoint trees $\tau_1 \ldots \tau_J$ hanging from the root. Splitting a tree into two is allowed, but merging two trees is not. Initially there is just one tree T hanging from r. Let f_t^τ and g_t^τ represent the initial and current flows, respectively, on tree τ for group t. To obtain an estimator $\tilde{D}$ for density, we define estimators for the cost and the profit.

Costs: Define $\tilde{C}^\tau = \sum_{e\in\tau} y(e)c(e)$, and let C^r be the cost of all edges that have been contracted into the root. $\tilde{C} = C^r + \sum_{i=1}^J \tilde{C}^\tau$ be the estimator for cost.

Profits: Define $\tilde{P}_t^\tau = g_t^\tau / \log_2(2nf_t^\tau)$. Lemma 3.2 guarantees that $0 \leq g_t^\tau \leq 1$, and therefore $\tilde{P}_t^\tau$ can be looked upon as probabilities (recall that all flows are multiples of $1/n$). We are now going to pretend that $\tilde{P}_t^\tau$ indeed represents the probability of group t being satisfied in tree τ. Let $\mathbf{Prob}(\{a_1 \ldots a_J\})$ be the probability of at least one of J independent events happening where a_i is the probability of the i-th event, *ie.* $\mathbf{Prob}(\{a_1 \ldots a_J\}) = 1 - \prod_{i=1}^J (1-a_i)$. Define $\tilde{P}_t = \mathbf{Prob}(\{\tilde{P}_t^{\tau_i} : 1 \leq i \leq J\})$. Let $\tilde{P} = \sum_{t \in S} \tilde{P}_t$ be the estimator for profit.

The intuition behind the above definition of profit is provided by the analysis of Garg *et al.* of their randomized rounding procedure [12]. Having defined estimators for cost and profit, we define $\tilde{D} = \tilde{C}/\tilde{P}$. The algorithm ROUND repeatedly picks any subtree τ out of $\{\tau_1 \ldots \tau_J\}$, and picks any edge e in τ incident on r. It then does the following two steps (See Figure 1):

Step 1–SPLIT: Partition the tree τ into α which is the tree hanging from r via edge e, and β, the rest of the tree. The quantity $\tilde{C}$ remains unchanged but the quantity $\tilde{P}$ needs to be recomputed, since the event corresponding to P_t^τ has now been split into two independent events.

Step 2–DECIDE: Calculate $\tilde{C}^X$ and $\tilde{P}^X$, the estimators for cost and profit if e (and hence the entire subtree α) were to be excluded. Also calculate the estimators $\tilde{C}^I$ and $\tilde{P}^I$ if the edge e were to be chosen. Choosing e involves contracting it into the root r and multiplying the flows (and hence the capacities) on all the other edges in α by $1/y(e)$. Let $\tilde{D}^X = \tilde{C}^X/\tilde{P}^X$ and $\tilde{D}^I = \tilde{C}^I/\tilde{P}^I$. Pick the smaller of $\tilde{D}^X$ and $\tilde{D}^I$ and perform the corresponding action. In case of a tie use $\tilde{D}^I$.

During each iteration at least one edge gets either contracted or removed. So the above algorithm terminates after n iterations. The cost and profit estimators can be computed in time $O(nk)$. After all the edges have been considered, the set of all contracted edges forms the tree X.

Lemma 3.2 *$0 \le g_t^\tau \le 1$ for all trees τ attached at the root and all groups t.*

Lemma 3.3 (a) *$\tilde{D}$ does not increase during the* SPLIT *step.* (b) *$\tilde{D}$ does not increase during the* DECIDE *step.*

Proof: *Part (a)* $\tilde{C}$ does not change during this step. To show that $\tilde{P}$ does not decrease, it suffices to show that

$$\tilde{P}_t^\tau \le \tilde{P}_t^\alpha + \tilde{P}_t^\beta - \tilde{P}_t^\alpha \cdot \tilde{P}_t^\beta. \tag{1}$$

because $\tilde{P}_t$ is monotonically increasing in each $\tilde{P}_t^\tau$. Let a be the larger of f_t^α and f_t^β and b be the smaller. Since the trees α and β were part of a single tree τ before the split, the flows on α and β have been increased by the same factor, say λ. Therefore $g_t^\alpha = \lambda f_t^\alpha$ and $g_t^\beta = \lambda f_t^\beta$. Our proof is by contradiction: assume that equation 1 gets violated. If $a = 0$ then the contradiction is quite straight forward. So assume $a > 0$. By Lemma 3.2, $a\lambda$ and $b\lambda$ are at most 1.

$$\begin{aligned}
\tfrac{\lambda(a+b)}{\log(2n(a+b))} &> \tfrac{\lambda a}{\log(2na)} + \tfrac{\lambda b}{\log(2nb)}\left(1 - \tfrac{\lambda a}{\log(2na)}\right) \\
&\ge \tfrac{\lambda a}{\log(2na)} + \tfrac{\lambda b}{\log(2n(a+b))}\left(1 - \tfrac{\lambda a}{\log(2na)}\right)
\end{aligned}$$

From this we obtain,

$$\begin{aligned}
\tfrac{a}{\log(2n(a+b))} &> \tfrac{a}{\log(2na)} - \lambda \cdot \tfrac{a}{\log(2na)} \cdot \tfrac{b}{\log(2n(a+b))} \\
\Rightarrow \log \tfrac{a+b}{a} &< \lambda b,
\end{aligned}$$

But $a \le b$ which implies that $(a+b)/a \ge 2$, or $\log_2((a+b)/a) \ge 1$. Also $\lambda b \le 1$, which gives a contradiction.

Part (b) We define $A \triangleq \tilde{C}^\alpha + y_e c_e$, and

$$B \triangleq \sum_t \left(1 - \mathbf{Prob}(\{\tilde{P}_t^{\tau_i} : \tau_i \ne \tau\} \cup \{P_t^\beta\})\right) \tilde{P}_t^\alpha.$$

Now

$$\begin{aligned}
\tilde{D} &= (\tilde{C}^X + A)/(\tilde{P}^X + B), \\
\tilde{D}^I &= (\tilde{C}^X + A/y(e))/(\tilde{P}^X + B/y(e)).
\end{aligned}$$

If $A/B > \tilde{C}^X/\tilde{P}^X$ then $\tilde{D} > \tilde{D}^X$. On the other hand, if $A/B \le \tilde{C}^X/\tilde{P}^X$ then since $y(e) \le 1$, $\tilde{D} \ge \tilde{D}^I$. Since the algorithm chooses the smaller of $\tilde{D}^X$ and $\tilde{D}^I$, the estimator for density does not increase. $\square$

Lemma 3.4 *Let D be the density of the tree X, and $\tilde{D}$ the density estimator at the end of algorithm* ROUND. *Then $D \le \tilde{D}$.*

Lemma 3.5 *The initial estimator for the density is $z^* \log(2n)/k$.*

Theorem 3.1 *The cost of the final solution is at most* $\log(2n)\log(2k) \cdot z^*$.

Theorem 3.2 *Combining Theorems 2.3 and 3.1 gives an* $O(\log^2 n \log k \log\log n)$ deterministic *approximation algorithm for the group Steiner tree problem.*

4 The k-median problem

The k-median problem is the following: Given a graph $G = (V, E)$ with a length function on edges and an integer k, we have to select a subset $C \subseteq V$ of k vertices such that $\sum_{u \in V} d(u, C)$ is minimized. Here $d(u, C)$ denotes the distance of u from the closest vertex in C. The vertices in C are termed *centers*.

We can formulate the problem as an integer program. The LP relaxation of this is as follows:

$$\begin{aligned}
\min \sum_{u \in V} \sum_{v \in V} x_{uv} \cdot c(u,v) & \\
\sum_{v \in V} y_v = k & \\
x_{uv} \le y_v & \quad \forall (u,v) \in E \\
\sum_{v \in V} x_{uv} = 1 & \quad \forall u \in V \\
x_{uv}, y_v \ge 0 &
\end{aligned}$$

We take the optimal solution to this LP and produce a tree T, which is a 2-HST approximation of the graph and preserves $\sum_{u \in V} \sum_{v \in V} x_{uv} \cdot c(u,v)$. The leaves of T are real vertices in G and the rest of the vertices of T are *virtual* vertices. The tree has a corresponding LP solution with the same y_v values. If the value of the original optimal LP solution is

OPT, the value of the LP solution on the tree is at most $O(\log n \log\log n) \cdot OPT$.

For a vertex $u \in T$ (possibly a virtual vertex), let T_u denote the subtree of T rooted at u and let l_u denote the length of the edge connecting u to its parent in T. For a subtree X, let $n(X)$ be the number of vertices of the original graph in X. Let $r(X) = \min_{v \in X} \sum_{u \in X} d_T(u, v)$; let $\mathsf{center}(X)$ be the vertex $v \in X$ that minimizes this sum. Let $y(X) = \sum_{v \in X} y_v$. We call subtree X *saturated* if $y(X) \geq 1$ and *unsaturated* otherwise.

T_u is said to be *maximal* unsaturated (abbreviated max unsat) if T_u is unsaturated and T_v is saturated where v is the parent of u. T_v is said to be *minimal* saturated (abbreviated min sat) if T_v is saturated and T_u is unsaturated for all children u of v.

We now present the algorithm to determine the locations of the k centers. As described below, the algorithm selects k maximal unsaturated subtrees. For each selected maximal unsaturated subtree X, we place a center at $\mathsf{center}(X)$.

1. For each minimal saturated tree T_u, look at the subtrees T_v for every child v of u. Each subtree T_v is a maximal unsaturated tree. Select the subtree T_v with the largest value of $2l_v \cdot n(T_v) - r(T_v)$. Note that there can be at most k minimal saturated trees, hence at most k centers are picked up in this step.

2. Suppose r subtrees were picked in the previous step. Order all the remaining maximal unsaturated subtrees T_u in decreasing order of $2l_u \cdot n(T_u) - r(T_u)$. Pick the first $k - r$ subtrees in this order.

We use the LP solution on T to prove a bound on the cost of the solution produced by the algorithm.

Lemma 4.1 *The value of the LP solution on T is at least*

$$(2) \sum_{T_u \text{ max unsat}} [2l_u n(T_u) \cdot (1 - y(T_u)) + y(T_u) r(T_u)]$$

Lemma 4.2 *At the end of Step 1, for every vertex v in a maximal unsaturated subtree T_u, there is a center within a distance $4l_u$ from v.*

Lemma 4.3 *The algorithm produces a solution within a factor 2 of the LP solution.*

Proof: We will transform the LP solution to the solution obtained by the algorithm and prove that the final solution has value at most 2 times the lower bound (2) on the LP solution. The intermediate steps in this transformation are fractional solutions, defined as follows: A fractional solution is an assignment of weights $w(T_u)$ to each maximal unsaturated subtree T_u such that $0 \leq w(T_u) \leq 1$ and $\sum_{T_u \text{ max unsat}} w(T_u) = k$. With each fractional solution, we associate the following lower bound, of the same form as (2).

$$(3) \sum_{T_u \text{ max unsat}} [2l_u n(T_u) \cdot (1 - w(T_u)) + w(T_u) r(T_u)]$$

Initially $w(T_u) = y(T_u)$ for every maximal unsaturated subtree, and the lower bound (3) is the same as the bound (2) on the LP solution. We will transform the fractional solution into one where $w(T_u)$ is 1 for all the maximal unsaturated subtrees T_u selected by the algorithm, and 0 for the rest. In doing this, we ensure that the lower bound (3) does not increase.

Consider a minimal saturated subtree T_u of T and the subtrees T_v for every child v of u. The subtree that maximizes $2l_v \cdot n(T_v) - r(T_v)$ is selected by the algorithm. Suppose this subtree is T_x. We increase $w(T_x)$ and decrease $w(T_v)$ for some $v \neq x$, till $w(T_x) = 1$. By the choice of T_x, the lower bound (3) does not increase. We repeat this for every minimal saturated T_u. At the end of this, we have $w(T_x) = 1$ for all r subtrees T_x selected by the algorithm in Step 1.

Order the remaining maximal unsaturated subtrees T_u in decreasing order of $2l_u \cdot n(T_u) - r(T_u)$. We increase $w(T_u)$ for the first $k - r$ subtrees in this order and decrease $w(T_u)$ for the rest, till the weights of each of the first $k - r$ trees is 1. Again, the lower bound (3) does not increase. This gives us the fractional solution with the desired properties.

For each maximal unsaturated subtree T_u with $w(T_u) = 1$, the algorithm places a center at $\mathsf{center}(T_u)$. We bound the value of this solution in terms of the lower bound (3) associated with the final fractional solution.

For each subtree T_u with $w(T_u) = 1$, the sum of the contributions of vertices u in T_u is simply $r(T_u)$. Consider a maximal unsaturated subtree T_u with $w(T_u) = 0$. By Lemma 4.2, every vertex in T_u has some center within a distance $4l_u$ of it. Hence the total contribution from the vertices of T_u is at most $4l_u \cdot n(T_u)$. Thus the value of the solution is at most 2 times the value of the lower bound associated with the final fractional solution, which in turn is at most 2 times the lower bound on the original LP solution. □

Since we lost a factor of $O(\log n \log\log n)$ in constructing the tree T, this gives a polynomial time deterministic $O(\log n \log\log n)$ approximation algorithm. We can obtain an $O(\log k \log\log k)$ approximation as follows. Consider the optimal k-median LP solution on G. We use the rounding procedure of Lin and Vitter [18] to produce a solution with $2k$ centers with value within a constant factor of the LP solution. Now consider the graph G' induced on the $2k$ centers. Suppose center u serves n_u vertices. We replace vertex u in G' with a clique of n_u nodes at a distance 0 from each other. We now solve the k-median problem on G'. Since the graph G' effectively has $2k$ nodes, it follows that we can obtain a $O(\log k \log\log k)$ approximation for the k-median problem on G'.

Lemma 4.4 *Any β approximate solution to the k-median problem on G' is an $O(\beta)$ approximate solution to the k-median problem on G.*

Theorem 4.1 *There is a polynomial time deterministic $O(\log k \log\log k)$ approximation algorithm for the k-median problem.*

Polynomial-time exact algorithms are known for solving the k-median problem on trees [15, 22]. We can use an exact algorithm to solve the problem on the tree obtained from the deterministic tree construction procedure. However the above analysis is necessary to prove that the optimal integer solution on the tree we obtain is within a constant factor of the value of the LP solution. De Vries *et al.* [9] have recently demonstrated matching upper and lower bounds of 2 on the gap between the optimal integral and fractional solutions of the k-median problem on a tree. Their upper bound proof is a simpler version of the proof given by Ward *et al.* [23] in an unpublished manuscript. Our upper bound was obtained independently of the work of Ward *et al.* .

5 The minimum communication cost spanning tree problem

The communication cost spanning tree problem is a minimization problem defined on an undirected weighted graph $G = (V, E)$. With each pair $(u, v) \in V \times V$ there is an associated postive weight w_{uv}. The objective is to find a spanning tree T that minimizes $\sum_{u,v} w_{uv} \cdot d_T(u, v)$ where $d_T(u, v)$ is the distance in the tree between u and v. We restrict ourselves to the case when the graph G is induced by an n point metric space (a *complete* graph on n vertices with the edge weights satisfying triangle inequality).

This problem has applications in routing with delays, and computational biology. For details see [24] where it is observed that an $O(\log^2 n)$ ratio is achievable for the metric case using probabilistic approximation of metric spaces (the ratio now improves to $O(\log n \log\log n)$ [4]). We can formulate the problem of finding a spanning tree as an integer program and relax it to obtain an LP with fractional values $x(e)$ on the edges. Since the graph is a metric, we observe that an alternative lower bound to the integer optimal is given by the trivial solution of setting $x(e) = 1$ for all edges (u, v). It is easy to see that the objective function is a linear combination of edge lengths and the tree we construct according to Theorem 2.3 is within a $O(\log n \log\log n)$ factor of the lower bound. From Remark 1 we obtain a tree whose vertex set is the same as that of G. We then use the fact that G is a metric to obtain a spanning tree.

Wu *et al.* [24] show that the general graph case can be reduced to the metric case if $w_{u,v} = 1$ for all pairs (u, v). Such a result does not appear to be true if w_{uv} are different and no nontrivial approximation algorithm is known for arbitrary graphs.

6 Conclusions

In this paper we presented a general paradigm to derandomize the use of probabilistic approximation of metric spaces in approximation algorithms. This paradigm is applied to group Steiner trees and k-median by presenting rounding procedures for them on trees and obtaining *deterministic* algorithms with approximation ratios matching the best known randomized algorithms. Improving the current bound of $O(\log^2 n \log k \log\log n)$ for group Steiner trees is an obvious open problem. The k-median problem in general graphs can be shown to be hard to approximate within a factor of $(1 + 1/e) \simeq 1.36$ via a reduction from dominating sets. The upper bound we guarantee is $O(\log k \log\log k)$. Improving either the hardness or the approximation ratio is an interesting open problem. Wu *et al.* [24] give a PTAS for a special case of the minimum communication cost spanning tree problem which they refer to as the minimum routing cost spanning tree problem. For the general case not even Max-SNP hardness is known. Finally, as mentioned in the introduction, a polynomial time constructive proof of Theorem 1.1 has been recently obtained [8], and this completely derandomizes the use of probabilistic approximation of metric spaces for approximation algorithms.

Acknowledgments

We thank Yair Bartal for information about his result [4], and Rajeev Motwani and Serge Plotkin for

useful discussions.

References

[1] N. Alon, R.M. Karp, D. Peleg, and D. West. "A graph-theoretic game and its application to the k-server problem", *SIAM J. Comput.*, 24:1, 78–100 (1995).

[2] B. Awerbuch, and Y. Azar. "Buy-at-bulk network design", *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pp. 542–547 (1997).

[3] Y. Bartal. "Probabilistic approximation of metric spaces and its algorithmic applications", *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, pp. 184-93 (1996).

[4] Y. Bartal. "On approximating arbitrary metrics by tree metrics", *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, (1998).

[5] G.Konjevod, R.Ravi, and F.S. Salman. "On approximating planar metrics by tree metrics", *manuscript* (1997).

[6] C.D. Bateman, C.S. Helvig, G. Robins, and A. Zelikovsky. "Provably good routing tree construction with multi-port terminals", *Proceedings of the 28th ACM/SIGDA International Symposium on Physical Design* (Apr 1997).

[7] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. "Approximation Algorithms for Directed Steiner Problems", *Proceeding of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (1998).

[8] M. Charikar, C. Chekuri, A. Goel, S. Guha and S. Plotkin. "Approximating an arbitrary metric by $O(n \log n)$ trees", *manuscript*, (1998).

[9] S. de Vries, M. Posner, and R. Vohra. *Personal Communication* (Nov 1997).

[10] G. Even, J. Naor, S. Rao, and B. Schieber. "Divide-and-conquer approximation algorithms via spreading metrics", *Proceedings of IEEE 36th Symposium on Foundations of Computer Science*, 62–71 (1995).

[11] U. Feige, "A threshold of $\ln n$ for approximating set-cover", *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, 314–318 (1996).

[12] N. Garg, G. Konjevod, and R. Ravi. "A polylogarithmic approximation algorithm for the group Steiner tree problem", *Proceeding of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (1998).

[13] N. Garg, V. Vaziarni, and M. Yannakakis. "Approximate max-flow min-(multi)cut theorems and their applications", *Proceedings of 25th Annual ACM Symposium on the Theory of Computing*, 698–707 (1993). Also in *SIAM J. on Computing*, 25:2, 235–251 (1996).

[14] S. Guha, and S. Khuller, "Approximation algorithms for Connected Dominating Sets", To appear in Algorithmica. A primary version appeared in *Proceedings of 4th Annual European Symposium on Algorithms* (1996).

[15] O. Kariv, and S.L. Hakimi. "An algorithmic approach to network location problems, Part II: p-medians", *SIAM J. Appl. Math.*, vol 37, 539–560 (1979).

[16] P. Klein, S. Plotkin, and S. Rao. "Excluded minors, network decomposition and multicommodity flows", *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, 682–690 (1993).

[17] P. Klein, S. Plotkin, S. Rao, and E. Tardos. "Approximation Algorithms for Steiner and Directed Multicuts", *Journal of Algorithms*, 22:2 241–269 (1997).

[18] J. Lin, and J. Vitter, "ϵ-approximations with minimum packing constraint violation", *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing*, 771–82 (1992).

[19] P. Raghavan. "Probabilistic construction of deterministic algorithms: Approximating packing integer problems", *Journal of Comp. Sys. Sci.*, 37:130–143 (1988).

[20] G. Reich, and P. Widmayer. "Beyond Steiner problem: a VLSI oriented generalization", *Lecture Notes in Computer Science*, vol. 411, 196–210 (Springer, 1990)

[21] P.D. Seymour, "Packing directed circuits fractionally", *Combinatorica*, 15:2 281–288 (1995).

[22] A. Tamir. "An $O(pn^2)$ algorithm for the p-median and related problems on tree graphs", *Oper. Res. Letters*, 19(1996) 59-94.

[23] J. Ward, R. T. Wong, P. Lemke and A. Oudjit. 1994. "Properties of the Tree K-median Linear Programming Relaxation", *Manuscript* (1994).

[24] B.Y. Wu, G. Lancia, V. Bafna, K. Chao, R. Ravi, and C.Y. Tang. "A Polynomial Time Approximation Scheme for Minimum Routing Cost Spanning Trees", *Proceeding of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (1998).