# Approximate $k$-MSTs and $k$-Steiner Trees via the Primal-Dual Method and Lagrangean Relaxation

Fabián A. Chudak* Tim Roughgarden† David P. Williamson‡

**Abstract**

Garg [10] gives two approximation algorithms for the minimum-cost tree spanning $k$ vertices in an undirected graph. Recently Jain and Vazirani [16] discovered primal-dual approximation algorithms for the metric uncapacitated facility location and $k$-median problems. In this paper we show how Garg's algorithms can be explained simply with ideas introduced by Jain and Vazirani, in particular via a Lagrangean relaxation technique together with the primal-dual method for approximation algorithms. We also derive a constant factor approximation algorithm for the $k$-Steiner tree problem using these ideas, and point out the common features of these problems that allow them to be solved with similar techniques.

## 1 Introduction

Given an undirected graph $G = (V, E)$ with non-negative costs $c_e$ for the edges $e \in E$ and an integer $k$, the $k$-MST problem is that of finding the minimum-cost tree in $G$ that spans at least $k$ vertices. A *rooted* version of the problem has a root vertex $r$ as part of its input, and the tree output must contain $r$. Unless otherwise stated, we will consider the rooted version of the problem. The more natural unrooted version of the problem reduces easily to the rooted one, by trying all $n$ possible roots and returning the cheapest of the $n$ solutions obtained.

The $k$-MST problem is known to be NP-hard [9]; hence, researchers have attempted to find approximation algorithms for the problem. An $\alpha$-approximation algorithm for a minimization problem runs in polynomial time and produces a solution of cost no more than $\alpha$ times that of the optimal solution. The value $\alpha$ is the *performance guarantee* or *approximation ratio* of the algorithm. The first non-trivial approximation algorithm for the $k$-MST problem was given by Ravi et al. [18], who achieved an approximation ratio of $O(\sqrt{k})$. This ratio was subsequently improved to $O(\log^2 k)$ by Awerbuch et al. [4] and $O(\log k)$ by Rajagopalan and Vazirani [17] before a constant-factor approximation algorithm was discovered by Blum et al. [7]. Garg [10] improved upon the constant, giving a simple 5-approximation algorithm and a somewhat more involved 3-approximation algorithm for the problem. Using Garg's algorithm as a black box, Arya and Ramesh [3] gave a 2.5-approximation algorithm for the unrooted version of the problem, and Arora and Karakostas [2] gave a $(2+\epsilon)$-approximation algorithm for any fixed $\epsilon > 0$ for the rooted version. Finally, Garg [11] has announced that a slight modification of his 3-approximation algorithm gives a performance guarantee of 2 for the unrooted version of the problem.

*Address: ETH Zurich, Institut für Operations Research, CLP D 7, Clausiusstrasse 45, 8092 Zürich, Switzerland. Email: `fabian.chudak@ifor.math.ethz.ch`.

†Address: University of California at Berkeley, Computer Science Department, Soda Hall 587, Berkeley, CA, 94720, USA. Email: `timr@cs.berkeley.edu`.

‡Address: IBM Almaden Research Center, 650 Harry Rd. K53/B1, San Jose, CA, 95120, USA. Email: `dpw@almaden.ibm.com`. Web: `www.almaden.ibm.com/cs/people/dpw`.

In addition to the practical motivations given in [18, 4], the $k$-MST problem has been well-studied in recent years in part due to its applications in the context of other approximation algorithms, such as the $k$-TSP problem (the problem of finding the shortest tour visiting at least $k$ vertices) [10, 2] and the minimum latency problem (the problem of finding the tour of $n$ vertices minimizing the average distance from the starting vertex to any other vertex along the tour) [6, 12, 1].

This paper is an attempt to simplify Garg's two approximation algorithms for the $k$-MST problem. In particular, Jain and Vazirani [16] recently discovered a new approach to the primal-dual method for approximation algorithms, and demonstrated its applicability with constant-factor approximation algorithms for the metric uncapacitated facility location and $k$-median problems. One novel aspect of their approach is the use of their facility location heuristic as a subroutine in their $k$-median approximation algorithm, the latter based on the technique of Lagrangean relaxation. This idea cleverly exploits the similarity of the integer programming formulations of the two problems. We show that Garg's algorithms can be regarded as another application of this approach: that is, as a Lagrangean relaxation algorithm employing a primal-dual approximation algorithm for a closely related problem as a subroutine. We also give a constant-factor approximation algorithm for the $k$-Steiner tree problem, via a similar analysis. We believe that these results will give a clearer and deeper understanding of Garg's algorithms, while simultaneously demonstrating that the techniques of Jain and Vazirani should find application beyond the two problems for which they were originally conceived.

This paper is structured as follows. In Section 2, we give linear programming relaxations for the $k$-MST problem and the closely related prize-collecting Steiner tree problem. In Section 3 we describe and analyze Garg's 5-approximation algorithm for the $k$-MST problem. In Section 4 we discuss extensions to the $k$-Steiner tree problem and outline improvements to the basic 5-approximation algorithm. We conclude in Section 5 with a discussion of the applicability of Jain and Vazirani's technique.

## 2 Two Related LP Relaxations

The rooted $k$-MST problem can be formulated as the following integer program

$$
\begin{array}{llll}
 & \text{Min} & \displaystyle\sum_{e \in E} c_e x_e & \\
 & \text{subject to:} & & \\
(kMST) & & \displaystyle\sum_{e \in \delta(S)} x_e + \sum_{T: T \supseteq S} z_T \geq 1 & \forall S \subseteq V \setminus \{r\} \qquad (1) \\
 & & \displaystyle\sum_{S: S \subseteq V \setminus \{r\}} |S| z_S \leq n - k & \qquad (2) \\
 & & x_e \in \{0,1\} & \forall e \in E \\
 & & z_S \in \{0,1\} & \forall S \subseteq V \setminus \{r\}
\end{array}
$$

where $\delta(S)$ is the set of edges with exactly one endpoint in $S$. The variable $x_e = 1$ indicates that the edge $e$ is included in the solution, and the variable $z_S = 1$ indicates the set of vertices $S$ that are not spanned by the tree. Thus the constraints (1) enforce that for each $S \subseteq V \setminus \{r\}$ either some edge $e$ is selected from the set $\delta(S)$ or that the set $S$ is contained in the set $T$ of unspanned vertices. Collectively, these constraints ensure that all vertices not in any $S$ such that $z_S = 1$ will be connected to the root vertex $r$. The constraint (2) enforces that at most $n - k$ vertices are not spanned. We can relax this integer program to a linear program by replacing the integrality constraints with nonnegativity constraints.

Although the above formulation is not the most natural one, we chose it to highlight the connection of the $k$-MST problem with another problem, the prize-collecting Steiner tree problem. In the prize-collecting Steiner tree problem, we are given an undirected graph $G = (V, E)$ with non-negative costs $c_e$ on edges $e \in E$, a specified root vertex $r$, and non-negative penalties $\pi_i$ on the vertices $i \in V$. The goal is to choose a set $S \subseteq V \setminus \{r\}$ and a tree $F \subseteq E$ spanning the vertices of $V \setminus S$ so as to minimize the cost of $F$ plus the penalties of the vertices in $S$. An integer programming formulation of this problem is

$$
\begin{array}{llll}
 & \text{Min} & \sum_{e \in E} c_e x_e + \sum_{S \subseteq V \setminus \{r\}} \pi(S) z_S & \\
 & \text{subject to:} & & \\
(PCST) & & \sum_{e \in \delta(S)} x_e + \sum_{T : T \supseteq S} z_T \geq 1 & \forall S \subseteq V \setminus \{r\} \\
 & & x_e \in \{0, 1\} & \forall e \in E \\
 & & z_S \in \{0, 1\} & \forall S \subseteq V \setminus \{r\},
\end{array}
$$

where $\pi(S) = \sum_{i \in S} \pi_i$. The interpretation of the variables and the constraints is as above, and again we can relax the integer program to a linear program by replacing the integrality constraints with nonnegativity constraints.

The existing constant approximation algorithms for the $k$-MST problem [7, 10, 2] all use as a subroutine a primal-dual 2-approximation algorithm for the prize-collecting Steiner tree due to Goemans and Williamson [13, 14] (which we will refer to on occasion as "the prize-collecting algorithm"). The integer programming formulations for the two problems are remarkably similar, and recent work on the $k$-median problem by Jain and Vazirani [16] gives a methodology for exploiting such similarities. Jain and Vazirani present an approximation algorithm for the $k$-median problem that applies Lagrangean relaxation to a complicating constraint in a formulation of the problem (namely, that at most $k$ facilities can be chosen). Once relaxed, the problem is an uncapacitated facility location problem for which the Lagrangean variable is the cost of opening a facility. By adjusting this cost and applying an approximation algorithm for the uncapacitated facility location problem, they are able to extract a solution for the $k$-median problem.

One can show that the same dynamic is at work in Garg's algorithms. In particular, if we apply Lagrangean relaxation to the complicating constraint $\sum_{S : S \subseteq V \setminus \{r\}} |S| z_S \leq n - k$ in the relaxation of $(kMST)$, we obtain the following for fixed Lagrangean variable $\lambda \geq 0$:

$$
\begin{array}{llll}
 & \text{Min} & \sum_{e \in E} c_e x_e + \lambda \left( \sum_{S \subseteq V \setminus \{r\}} |S| z_S - (n - k) \right) & \\
 & \text{subject to:} & & \\
(LRk) & & \sum_{e \in \delta(S)} x_e + \sum_{T : T \supseteq S} z_T \geq 1 & \forall S \subseteq V \setminus \{r\} \\
 & & x_e \geq 0 & \forall e \in E \\
 & & z_S \geq 0 & \forall S \subseteq V \setminus \{r\}.
\end{array}
$$

For fixed $\lambda$, this is nearly identical to $(PCST)$ with $\pi_i = \lambda$ for all $i$, except for the constant term of $-(n - k)\lambda$ in the objective function. Observe that any solution feasible for the $(kMST)$ is also feasible for $(LRk)$ with no greater cost, and so the value of $(LRk)$ is a lower bound on the cost of an optimal $k$-MST.

In order to discuss how Garg's algorithms work, we first need to say a little more about the primal-dual approximation algorithm for the prize-collecting Steiner tree. The algorithm constructs a primal-feasible solution $(F, A)$, where $F$ is a tree including the root $r$, and $A$ is the set of vertices not spanned by $F$. The algorithm also constructs a feasible solution $y$ for the dual of $(PCST)$, which is

$$(PCST-D) \qquad \begin{array}{lll} \text{Max} & \displaystyle\sum_{S\subseteq V\setminus\{r\}} y_S & \\ \text{subject to:} & & \\ & \displaystyle\sum_{S:e\in\delta(S)} y_S \le c_e & \forall e \in E \\ & \displaystyle\sum_{T:T\subseteq S} y_T \le \pi(S) & \forall S \subseteq V\setminus\{r\} \\ & y_S \ge 0 & \forall S \subseteq V\setminus\{r\}. \end{array}$$

Then the following is true:

**Theorem 2.1** [Goemans and Williamson [13]] The primal solution $(F, A)$ and the dual solution $y$ produced by the prize-collecting algorithm satisfy

$$\sum_{e\in F} c_e + \left(2 - \frac{1}{n-1}\right)\pi(A) \le \left(2 - \frac{1}{n-1}\right)\sum_{S\subseteq V\setminus\{r\}} y_S.$$

Note that, by weak duality and the feasibility of $y$, $\sum_{S\subseteq V\setminus\{r\}} y_S$ is a lower bound for the cost of any solution to the prize-collecting Steiner tree problem.

Suppose we set $\pi_i = \lambda \ge 0$ for all $i \in V$ and run the prize-collecting algorithm. The theorem statement implies that we obtain $(F, A)$ and $y$ such that

$$\sum_{e\in F} c_e + 2|A|\lambda \le 2\sum_{S\subseteq V\setminus\{r\}} y_S. \tag{3}$$

We wish to reinterpret the tree $F$ as a feasible solution for the $k$-MST instance, and extract a lower bound on the cost of an optimal $k$-MST from $y$. Toward this end, we consider the dual of the $(LRk)$ LP, as follows (recall that $\lambda$ is a fixed constant):

$$(LRk-D) \qquad \begin{array}{lll} \text{Max} & \displaystyle\sum_{S\subseteq V\setminus\{r\}} y_S - (n-k)\lambda & \\ \text{subject to:} & & \\ & \displaystyle\sum_{S:e\in\delta(S)} y_S \le c_e & \forall e \in E \\ & \displaystyle\sum_{T:T\subseteq S} y_T \le |S|\lambda & \forall S \subseteq V\setminus\{r\} \\ & y_S \ge 0 & \forall S \subseteq V\setminus\{r\}. \end{array}$$

The dual solution $y$ created by the prize-collecting algorithm is feasible for $(LRk - D)$ when all prizes $\pi_i = \lambda$. Furthermore, its value will be no greater than the cost of an optimal $k$-MST. After subtracting $2(n-k)\lambda$ from both sides of (3), by weak duality we obtain the following:

$$\begin{aligned} \sum_{e\in F} c_e + 2\lambda(|A| - (n-k)) &\le 2\left(\sum_{S\subseteq V\setminus\{r\}} y_S - (n-k)\lambda\right) &(4)\\ &\le 2\cdot OPT_k, &(5) \end{aligned}$$

where $OPT_k$ is the optimal solution to the $k$-MST problem. In the lucky event that $|A| = n - k$, $F$ is a feasible solution having cost no more than twice optimal. Otherwise, our solution will either not be feasible (if $|A| > n - k$) or the relations (4) and (5) will not give a useful upper bound on the cost of the solution (if $|A| < n - k$). However, in the next section we combine these ideas with a Lagrangean relaxation approach to derive an algorithm that always produces a near-optimal feasible solution (though with a somewhat inferior performance guarantee).

Observe that it is crucial for the analysis given above that there is no loss in performance guarantee for the cost of the primal associated with the Lagrangean variable; in this case, the cost of vertices not spanned by the tree is bounded above by the dual objective function for $(PCST-D)$. This condition seems necessary for this technique to be applied to approximation algorithms. We discuss this observation a bit further in Section 5; see also Section 3.6 of Jain and Vazirani [16], and the discussion of the "Lagrangean Multiplier Preserving" property in Jain, Mahdian, and Saberi [15].

# 3 Garg's 5-approximation algorithm

We begin with three assumptions, each without loss of generality. First, by standard techniques [18], one can show that it is no loss of generality to assume that the edge costs satisfy the triangle inequality. Second, we assume that the distance between any vertex $v$ and the root vertex $r$ is at most $OPT_k$; this is accomplished by "guessing" the distance $D$ of the farthest vertex from $r$ in the optimal solution (there are but $n - 1$ "guesses" to enumerate) and deleting all nodes of distance more than $D$ from $r$. Note that $D \leq OPT_k$. The cheapest feasible solution of these $n - 1$ subproblems is the final output of the algorithm. Third, we assume that $OPT_k \geq c_{\min}$, where $c_{\min}$ denotes the smallest non-zero edge cost. If this is not true, then $OPT_k = 0$ and the optimal solution is a connected component containing $r$ of at least $k$ nodes in the graph of zero-cost edges. We can easily check whether such a solution exists before we run Garg's algorithm.

Garg's algorithm is essentially a sequence of calls to the prize-collecting algorithm, each with a different value for the Lagrangean variable $\lambda$. First, the behavior of the algorithm is such that for $\lambda$ sufficiently small (e.g., $\lambda = 0$), the prize-collecting algorithm will return $(\emptyset, V \setminus \{r\})$ as a solution (that is, the degenerate solution of the empty tree trivially spanning $r$) and for $\lambda$ sufficiently large (e.g., $\lambda = \sum_{e \in E} c_e$) the prize-collecting algorithm will return a tree spanning all $n$ vertices. Second, if any call to the prize-collecting algorithm returns a tree $T$ spanning precisely $k$ vertices, then by the analysis in the previous section, $T$ is within a factor 2 of optimal, and the $k$-MST algorithm can halt with $T$ as its output.

By a straightforward binary search procedure consisting of polynomially many subroutine calls to the prize-collecting algorithm, Garg's algorithm either finds a tree spanning precisely $k$ vertices (via a lucky choice of $\lambda$) or two values $\lambda_1 < \lambda_2$ such that the following two conditions hold:

(i) $\lambda_2 - \lambda_1 \leq \frac{c_{\min}}{2n(2n+1)}$, where (as above) $c_{\min}$ denotes the smallest non-zero edge cost and

(ii) for $i = 1, 2$, running the prize-collecting algorithm with $\lambda$ set to $\lambda_i$ yields a primal solution $(F_i, A_i)$ spanning $k_i$ vertices and a dual solution $y^{(i)}$, with $k_1 < k < k_2$.

To be more precise, we maintain an interval $[\lambda_1, \lambda_2]$ such that running the prize-collecting algorithm with $\lambda$ set to $\lambda_i$ yields a primal solution spanning $k_i$ vertices, with $k_1 < k < k_2$. By the discussion above, the interval can initially be $[0, \sum_e c_e]$. We then run the prize-collecting algorithm using $\lambda = \frac{1}{2}(\lambda_1 + \lambda_2)$. If a tree is returned with $k$ vertices, we are done. If it has more than $k$ vertices, we update $\lambda_2$ to be $\frac{1}{2}(\lambda_1 + \lambda_2)$; otherwise it has less than $k$ vertices and we update $\lambda_1$ to this value.

After $O(\log \frac{n^2 \sum_e c_e}{c_{\min}})$ calls to the prize-collecting algorithm, we have $\lambda_1, \lambda_2$ with the two desired properties above.

Henceforth we assume the algorithm failed to find a value of $\lambda$ resulting in a tree spanning exactly $k$ vertices. Then, the final step of the algorithm combines the two primal solutions, $(F_1, A_1)$ and $(F_2, A_2)$, into a single tree spanning precisely $k$ vertices. For the analysis, the two dual solutions will also be combined.

From Theorem 2.1, we have the following inequalities:

$$\sum_{e \in F_1} c_e \leq \left(2 - \frac{1}{n}\right)\left(\sum_{S \subseteq V \setminus \{r\}} y_S^{(1)} - |A_1|\lambda_1\right) \tag{6}$$

$$\sum_{e \in F_2} c_e \leq \left(2 - \frac{1}{n}\right)\left(\sum_{S \subseteq V \setminus \{r\}} y_S^{(2)} - |A_2|\lambda_2\right) \tag{7}$$

We would like to take a convex combination of these two inequalities so as to get a bound on the cost of $F_1$ and $F_2$ in terms of $OPT_k$. Let $\alpha_1, \alpha_2 \geq 0$ satisfy $\alpha_1|A_1| + \alpha_2|A_2| = n - k$ and $\alpha_1 + \alpha_2 = 1$, and for all $S \subseteq V \setminus \{r\}$, let $y_S = \alpha_1 y_S^{(1)} + \alpha_2 y_S^{(2)}$. Note that

$$\alpha_1 = \frac{n - k - |A_2|}{|A_1| - |A_2|} \qquad \text{and} \qquad \alpha_2 = \frac{|A_1| - (n - k)}{|A_1| - |A_2|}.$$

**Lemma 3.1**

$$\alpha_1 \sum_{e \in F_1} c_e + \alpha_2 \sum_{e \in F_2} c_e < 2OPT_k.$$

*Proof.* From inequality (6) we have

$$\begin{aligned}
\sum_{e \in F_1} c_e &\leq \left(2 - \frac{1}{n}\right)\left(\sum_{S \subseteq V \setminus \{r\}} y_S^{(1)} - |A_1|(\lambda_1 + \lambda_2 - \lambda_2)\right) \\
&\leq \left(2 - \frac{1}{n}\right)\left(\sum_{S \subseteq V \setminus \{r\}} y_S^{(1)} - |A_1|\lambda_2\right) + \left(2 - \frac{1}{n}\right)\frac{c_{\min}|A_1|}{2n(2n+1)} \\
&< \left(2 - \frac{1}{n}\right)\left(\sum_{S \subseteq V \setminus \{r\}} y_S^{(1)} - |A_1|\lambda_2\right) + \frac{c_{\min}}{2n+1}.
\end{aligned}$$

By a convex combination of this inequality and inequality (7), it follows that

$$\begin{aligned}
\alpha_1 \sum_{e \in F_1} c_e + \alpha_2 \sum_{e \in F_2} c_e &< \left(2 - \frac{1}{n}\right)\left(\sum_{S \subseteq V \setminus \{r\}} y_S - \lambda_2(\alpha_1|A_1| + \alpha_2|A_2|)\right) + \frac{\alpha_1 c_{\min}}{2n+1} \\
&= \left(2 - \frac{1}{n}\right)\left(\sum_{S \subseteq V \setminus \{r\}} y_S - \lambda_2(n-k)\right) + \frac{\alpha_1 c_{\min}}{2n+1} && (8) \\
&\leq \left(2 - \frac{1}{n}\right) OPT_k + \frac{\alpha_1 c_{\min}}{2n+1} && (9) \\
&\leq \left(2 - \frac{1}{n}\right) OPT_k + \frac{1}{2n+1} OPT_k && (10) \\
&\leq 2OPT_k.
\end{aligned}$$

Equality (8) follows by our choice of $\alpha_1, \alpha_2$. Inequality (9) follows since $y$ is feasible for $(LRk-D)$ with the Lagrangean variable set to $\lambda_2$ by the convexity of the feasible region, and the fact that $\lambda_2 > \lambda_1$. Inequality (10) follows since $\alpha_1 \leq 1$ and $OPT_k \geq c_{\min}$. ■

Garg considers two different solutions to obtain a 5-approximation algorithm. First, if $\alpha_2 \geq \frac{1}{2}$, then $F_2$ is already a good solution; since $|A_2| < n-k$, it spans more than $k$ vertices, and

$$\sum_{e \in F_2} c_e \leq 2\alpha_2 \sum_{e \in F_2} c_e \leq 4 \cdot OPT_k$$

by Lemma 3.1. Now suppose $\alpha_2 < \frac{1}{2}$. In this case the tree $F_1$ is supplemented by vertices from $F_2$. Let $\ell \geq k_2 - k_1$ be the number of vertices spanned by $F_2$ but not $F_1$. Then by doubling the tree $F_2$, shortcutting the resulting tour down to a simple tour of the $\ell$ vertices spanned solely by $F_2$, and choosing the cheapest path of $k-k_1$ vertices from this tour, we obtain a tree (in fact, a path) on $k-k_1$ vertices of cost at most

$$2\frac{k-k_1}{k_2-k_1} \sum_{e \in F_2} c_e.$$

This set of vertices can be connected to $F_1$ by adding an edge from the root to the set, which will have cost no more than $OPT_k$ (due to the second assumption made at the beginning of this section). Since

$$\frac{k-k_1}{k_2-k_1} = \frac{n-k_1-(n-k)}{n-k_1-(n-k_2)} = \frac{|A_1|-(n-k)}{|A_1|-|A_2|} = \alpha_2,$$

the total cost of this solution is

$$\begin{aligned} \sum_{e \in F_1} c_e + 2\alpha_2 \sum_{e \in F_2} c_e + OPT_k \quad &\leq \quad 2\left(\alpha_1 \sum_{e \in F_1} c_e + \alpha_2 \sum_{e \in F_2} c_e\right) + OPT_k \\ &\leq \quad 4OPT_k + OPT_k, \end{aligned}$$

since $\alpha_2 < \frac{1}{2}$ implies $\alpha_1 > \frac{1}{2}$, and by Lemma 3.1.

# 4 Extensions

The $k$-Steiner tree problem is defined as follows: given an undirected graph $G=(V,E)$ with non-negative costs $c_e$ for the edges $e \in E$, a set $R \subseteq V$ of *required vertices* (also called *terminals*), and an integer $k$, find the minimum-cost tree in $G$ that spans at least $k$ of the required vertices. Of course, the problem is only feasible when $k \leq |R|$. The $k$-Steiner tree problem includes the classical Steiner tree problem (set $k=|R|$) and is thus both NP-hard and MAX SNP-hard [5]. The problem was studied by Ravi et al. [18], who gave a simple reduction showing that an $\alpha$-approximation algorithm for the $k$-MST problem yields a $2\alpha$-approximation algorithm for the $k$-Steiner tree problem. Thus, the result of the previous section implies the existence of a 10-approximation algorithm for the problem. However, we can show that a modification of Garg's 5-approximation algorithm achieves a performance guarantee of 5 for this problem as well. Consider the following LP relaxation for the $k$-Steiner tree problem

$$\begin{array}{llll} & \text{Min} & \sum_{e \in E} c_e x_e & \\ & \text{subject to:} & & \\ (kST) & & \sum_{e \in \delta(S)} x_e + \sum_{T: T \supseteq S} z_T \geq 1 & \forall S \subseteq V \setminus \{r\} \end{array}$$

$$\sum_{S:S\subseteq V\setminus\{r\}} |S\cap R| z_S \leq |R| - k$$

$$x_e \geq 0 \qquad \forall e \in E$$

$$z_S \geq 0 \qquad \forall S \subseteq V \setminus \{r\}.$$

We modify Garg's algorithm at the point where the prize-collecting algorithm is called as a subroutine with a fixed value of $\lambda$ inside the main Lagrangean relaxation loop. To reflect that we are only interested in how many required vertices are spanned by a solution, we assign required vertices a penalty of $\lambda$ and Steiner (non-required) vertices a penalty of 0. In the notation of the linear program ($PCST$), we put $\pi_i = \lambda$ for $i \in R$ and $\pi_i = 0$ for $i \notin R$. Then an analog of Lemma 3.1 can be shown, leading as in Section 3 to a 5-approximation algorithm for the $k$-Steiner tree problem.

We now discuss improving the approximation ratio of the two algorithms. Using ideas of Arora and Karakostas [2], the $k$-MST and $k$-Steiner tree algorithms can be refined to achieve performance guarantees of $(4+\epsilon)$, for an arbitrarily small constant $\epsilon$. Roughly speaking, their idea is as follows. Garg's algorithm essentially "guesses" one vertex that appears in the optimal solution, namely the root $r$. Instead, one can "guess" $O(\frac{1}{\epsilon})$ vertices and edges in the optimal solution (for fixed $\epsilon$, there are but polynomially many guesses to enumerate) such that any other vertex in the optimal solution has distance at most $O(\epsilon OPT)$ from the guessed subgraph $H$. After $H$ is guessed, all vertices of distance more than $O(\epsilon OPT)$ from $H$ can then be deleted. It is not difficult to modify the prize-collecting algorithm to handle the additional guessed vertices. Then, when creating a feasible solution from two subsolutions as at the end of Section 3, the final edge connecting the two subtrees costs no more that $\epsilon OPT$, leading to a final upper bound of $(4+\epsilon)OPT$. The reader is referred to [2] for the details of this refinement. Note, however, that the running time of the algorithm becomes $\Omega(n^{O(1/\epsilon)})$ in order to enumerate all possible guesses of $O(1/\epsilon)$ vertices and edges that appear in the solution.

In addition to the 5-approximation algorithm discussed in Section 3, Garg [10] gave a more sophisticated 3-approximation algorithm for the $k$-MST problem. Unfortunately, the analysis seems to require a careful discussion of the inner workings of the prize-collecting algorithm, a task we will not undertake here. (Similarly, improving Jain and Vazirani's 6-approximation algorithm for the $k$-median problem to a 4-approximation algorithm required a detailed analysis of the primal-dual facility location subroutine; see the paper of Charikar and Guha [8].) However, we believe that Garg's 3-approximation algorithm can also be recast in the language of Jain and Vazirani and of this paper, and that it will extend to a 3-approximation algorithm for the $k$-Steiner tree problem as well. The same ideas that led from a 5-approximation algorithm to one with performance guarantee $(4+\epsilon)$ should then yield $(2+\epsilon)$-approximation algorithms for the $k$-MST problem (as in [2]) and the $k$-Steiner tree problem.

# 5 Conclusion

We have shown that the techniques of Jain and Vazirani [16], invented for a constant-factor approximation algorithm for the $k$-median problem, also give constant-factor approximation algorithms for the $k$-MST problem (essentially reinventing older algorithms of Garg [10]) and the $k$-Steiner tree problem. A natural direction for future research is the investigation of the applicability and limitations of this Lagrangean relaxation approach. The three problems solved in this framework so far share several characteristics. First, each problem admits an LP relaxation with an obvious "complicating" constraint. Moreover, once the complicating constraint is lifted into the objective function, the new linear program corresponds to the relaxation of a problem known to be well-approximable (in our cases by a primal-dual approximation algorithm). Lastly, and perhaps most importantly, the subroutine for the relaxed problem produces a pair of primal and dual solutions such that the portion of the primal cost corresponding to the constraint of the original problem (e.g., the $\sum_S \pi(S) z_S$ term in the prize-collecting Steiner tree objective function) is bounded above by the value of the dual. Note that this is a stronger condition than merely ensuring that the primal solution has cost no more than some constant times the dual solution value. For example, in Theorem 2.1, the total primal cost is upper-bounded by twice the value of the dual solution, $2\sum_S y_S$, and in addition the second term of the primal cost is bounded above by the dual solution, $\sum_S y_S$. (Note such a statement does not hold in general for the first primal cost term of Theorem 2.1.) This last property seems necessary for extracting lower bounds for the problem of interest (via the dual LP) from the dual solutions returned by the subroutine, and may turn out to be the primary factor limiting the applicability of the Lagrangean relaxation approach. It would be of great interest to find further problems that can be approximately solved in this framework, and to devise more general variants of the framework that apply to a broader class of problems.

**Acknowledgements**

The first author was supported by an IBM Postdoctoral Research Fellowship. The second author was supported by ONR grant N00014-98-1-0589, an NSF Fellowship, and a Cornell University Fellowship. His research was carried out while visiting IBM.

# References

[1] S. Arora and G. Karakostas. Approximation schemes for minimum latency problems. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing*, pages 688–693, 1999.

[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*, pages 754–759, 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] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. Improved approximation guarantees for minimum-weight $k$-trees and prize-collecting salesmen. *SIAM Journal on Computing*, 28(1):254–262, 1999.

[5] M. Bern and P. Plassmann. The Steiner problem with edge lengths 1 and 2. *Information Processing Letters*, 32:171–176, 1989.

[6] A. Blum, P. Chalasani, D. Coppersmith, W. Pulleyblank, P. Raghavan, and M. Sudan. The minimum latency problem. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, pages 163–171, 1994.

[7] A. Blum, R. Ravi, and S. Vempala. A constant-factor approximation algorithm for the $k$-MST problem. *Journal of Computer and System Sciences*, 58(1):101–108, 1999.

[8] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and $k$-median problems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 378–388, 1999.

[9] M. Fischetti, H. 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 Symposium on Foundations of Computer Science*, pages 302–309, 1996.

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

[12] M. Goemans and J. Kleinberg. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, 82:111–124, 1998.

[13] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.

[14] M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, chapter 4, pages 144–191. PWS Publishing Company, 1997.

[15] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing*, pages 731–740, 2002.

[16] K. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and $k$-median problems. *Journal of the ACM*, 48:274–296, 2001.

[17] S. Rajagopalan and Vijay V. Vazirani. Logarithmic approximation of minimum weight $k$ trees. Unpublished manuscript, 1995.

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