Dial a Ride from *k*-Forest

ANUPAM GUPTA

Carnegie Mellon University

MOHAMMADTAGHI HAJIAGHAYI

AT&T Research Labs

VISWANATH NAGARAJAN

IBM T.J. Watson Research Center

AND

R. RAVI

Carnegie Mellon University

Abstract. The *k-forest problem* is a common generalization of both the *k-MST* and the *dense-k-subgraph* problems. Formally, given a metric space on n vertices V, with m demand pairs $\subseteq V \times V$ and a "target" $k \leq m$, the goal is to find a minimum cost subgraph that connects *at least* k pairs. In this paper, we give an $O(\min\{\sqrt{n} \cdot \log k, \sqrt{k}\})$-approximation algorithm for k-forest, improving on the previous best ratio of $O(\min\{n^{2/3}, \sqrt{m}\} \log n)$ by Segev and Segev.

We then apply our algorithm for k-forest to obtain approximation algorithms for several *Dial-a-Ride* problems. The basic Dial-a-Ride problem is the following: given an n point metric space with m objects each with its own source and destination, and a vehicle capable of carrying *at most* k objects at any time, find the minimum length tour that uses this vehicle to move each object from its source to destination. We want that the tour be *non-preemptive*: that is, each object, once picked up at its source,

41

A preliminary version of this article appeared in *Proceedings of the 15th European Symposium on Algorithms*, 2007 pp. 241–252.

A. Gupta's research supported in part by an NSF CAREER award CCF-0448095 and by an Alfred P. Sloan Fellowship. V. Nagarajan's research was supported in part by an IBM Ph.D. fellowship. R. Ravi's research was supported in part by NSF grants CCF-0430751 and CCF-0728841.

This work was done while M. T. Hajiaghayi and V. Nagarajan were at Carnegie Mellon University.

Author's addresses: A. Gupta, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, e-mail: anupamg@cs.cmu.edu; M. T. Hajiaghayi, AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932, e-mail: hajiagha@research.att.com; V. Nagarajan IBM T.J. Watson Research Center, Yorktown Heights, NY 10548; e-mail: viswanath@us.ibm.com; R. Ravi, Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, e-mail: ravi@cmu.edu.

DOI 10.1145/1721837.1721857 http://doi.acm.org/10.1145/1721837.1721857

is dropped only at its destination. We prove that an α-approximation algorithm for the k-forest problem implies an $O(\alpha \cdot \log^2 n)$-approximation algorithm for Dial-a-Ride. Using our results for k-forest, we get an $O(\min\{\sqrt{n}, \sqrt{k}\} \cdot \log^2 n)$-approximation algorithm for Dial-a-Ride. The only previous result known for Dial-a-Ride was an $O(\sqrt{k} \log n)$-approximation by Charikar and Raghavachari; our results give a different proof of a similar approximation guarantee—in fact, when the vehicle capacity k is large, we give a slight improvement on their results. The reduction from Dial-a-Ride to the k-forest problem is fairly robust, and allows us to obtain approximation algorithms (with the same guarantee) for some interesting generalizations of Dial-a-Ride.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Approximation algorithms, network design, vehicle routing

ACM Reference Format:

Gupta, A., Hajiaghayi, M., Nagarajan, V., and Ravi, R. 2010. Dial a ride from k-forest. ACM Trans. Algor. 6, 2, Article 41 (March 2010), 21 pages.
DOI = 10.1145/1721837.1721857 http://doi.acm.org/10.1145/1721837.1721857

1. *Introduction*

In the Steiner forest problem, we are given a set of vertex-pairs, and the goal is to find a forest such that each vertex pair lies in the same tree in the forest. This is a generalization of the Steiner tree problem, where all the pairs contain a common vertex called the root; both the tree and forest versions are well-understood fundamental problems in network design, and constant factor approximation algorithms are known [Robins and Zelikovsky 2005; Agrawal et al. 1991; Goemans and Williamson 1992]. An important extension of the Steiner tree problem studied in the late 1990s was the k-MST problem, where one sought the least-cost tree that connected any k of the terminals: several approximation algorithms were given for the problem, culminating in the 2-approximation of Garg [2005]; the k-MST problem proved crucial in many subsequent developments in network design and vehicle routing [Chaudhuri et al. 2003; Fakcharoenphol et al. 2003; Blum et al. 2003; Bansal et al. 2004]. One can analogously define the k-forest problem where one needs to connect *only k of the pairs* in some Steiner forest instance: surprisingly, very little is known about this problem, which was first studied formally only recently [Hajiaghayi and Jain 2006; Segev and Segev 2006]. In this article, we give a simpler and improved approximation algorithm for the k-forest problem.

Moreover, just like the k-MST variant, the k-forest problem seems to be useful in applications to network design and vehicle routing. In the second half of the paper, we show a (somewhat surprising) reduction of a well-studied vehicle routing problem called the Dial-a-Ride problem to the k-forest problem. In the Dial-a-Ride problem, we are given a metric space with objects having sources and destinations, and a vehicle of some capacity k; the goal is to find a route for this vehicle so that each object can be taken from its source to destination without exceeding the capacity of the vehicle at any point, such that the length of the vehicle route is minimized. We show how the results for the k-forest problem slightly improve upon existing results for the Dial-a-Ride problem; in fact, we give the first approximation algorithms for some generalizations of Dial-a-Ride which do not seem amenable to previous techniques.

1.1. THE k-FOREST PROBLEM. Our starting point is the k-forest problem, which generalizes both the k-MST and the dense-k-subgraph problems.

Definition 1.1 (*The k-Forest Problem*). Given an n-vertex metric space (V, d), and *demand pairs* $\{s_i, t_i\}_{i=1}^m \subseteq V \times V$, find the least-cost subgraph that connects at least k pairs.

We note that demand pairs may be repeated; so k and m may be super-polynomial in n. The k-forest problem is a generalization of the (minimization version of the) well-studied dense-k-subgraph problem, for which nothing better than an $O(n^{1/3-\delta})$ approximation is known ($\delta > 0$ is some fixed constant). The k-forest problem was first defined in Hajiaghayi and Jain [2006], and the first nontrivial approximation was given by Segev and Segev [2006], who gave an algorithm with an approximation guarantee of $O(\min\{n^{2/3}, \sqrt{m}\} \log n)$. We obtain the following improved approximation guarantee for k-forest in Section 2.

THEOREM 1.2 (APPROXIMATING k-FOREST). *There is an* $O(\min\{\sqrt{n} \cdot \log k, \sqrt{k}\})$*-approximation algorithm for the k-forest problem.*

Apart from giving an improved approximation guarantee, our algorithm for the k-forest problem is arguably simpler and more direct than that of Segev and Segev [2006] (which is based on Lagrangian relaxations for the problem, and combining solutions to this relaxation). Indeed, we give two algorithms, both reducing the k-forest problem to the k-MST problem in different ways and achieving different approximation guarantees—we then return the better of the two answers. The first algorithm (giving an approximation of $O(\sqrt{k})$) uses the k-MST algorithm to find good solutions on the sources and the sinks independently, and then uses the Erdős-Szekeres theorem on monotone subsequences to find a "good" subset of these sources and sinks to connect cheaply; details are given in Section 2.1. The second algorithm starts off with a single vertex as the initial solution, and uses the k-MST algorithm to repeatedly find a low-cost tree that satisfies a large number of pairs which have one endpoint in the current solution and the other endpoint outside; this tree is then used to greedily augment the current solution and proceed. Choosing the parameters (as described in Section 2.2) gives us an $O(\sqrt{n})$ approximation.

1.2. THE DIAL-A-RIDE PROBLEM. In this article, we use the k-forest problem to give approximation algorithms for the following vehicle routing problem.

Definition 1.3 (*The Dial-a-Ride Problem*). Given an n-vertex metric space (V, d), a starting vertex (or *root*) r, a set of m demand-pairs $\{(s_i, t_i)\}_{i=1}^m$, and a vehicle of capacity k, find a minimum length tour of the vehicle starting (and ending) at r that moves each object i from its source s_i to its destination t_i such that the vehicle carries at most k objects at any point on the tour.

Each demand-pair in the Dial-a-Ride problem corresponds to an object that is to be moved from the specified source to destination. We use the terms: demand-pair, object, and pair interchangeably. In the *preemptive Dial-a-Ride* problem, after picking up an object from its source, it may be left at some intermediate vertices before being delivered to its destination. In this article we will mainly be concerned with the *non-preemptive Dial-a-Ride* problem, where once an object is picked up from its source, it remains in the vehicle until dropped at its destination. A note on the parameters: using triangle inequality, any feasible nonpreemptive tour can be short-cut over vertices that do not participate in any demand-pair, and we can assume that every vertex is an end point of some demand-pair, that is, $n \leq 2m$. Again, we allow multiple demand-pairs between the same pair of vertices; so the number of objects m and the vehicle capacity k may be much larger than the number of vertices n.

The approximability of the Dial-a-Ride problem is not very well understood: the previous best upper bound is an $O(\sqrt{k}\log n)$-approximation algorithm due to Charikar and Raghavachari [1998], whereas the best lower bound that we are aware of is APX-hardness. We establish the following (somewhat surprising) connection between the Dial-a-Ride and k-forest problems in Section 3.

THEOREM 1.4 (REDUCING DIAL-A-RIDE TO k-FOREST). *Given an α-approximation algorithm for k-forest, there is an $O(\alpha \cdot \log^2 n)$-approximation algorithm for the Dial-a-Ride problem.*

In particular, combining Theorems 1.2 and 1.4 gives us an $O(\min\{\sqrt{k}, \sqrt{n}\} \cdot \log^2 n)$-approximation guarantee for Dial-a-Ride. Of course, improving the approximation guarantee for k-forest would improve the result for Dial-a-Ride as well.

Note that our results match the results of Charikar and Raghavachari [1998] up to a logarithmic term, and even give a slight improvement when the vehicle capacity $k \gg n$, the number of nodes. Much more interestingly, our algorithm for Dial-a-Ride easily extends to generalizations of the Dial-a-Ride problem. In particular, we consider a substantially more general vehicle routing problem where the vehicle has no *a priori* capacity, and instead the cost of traversing each edge e is an arbitrary non-decreasing function $c_e(l)$ of the number of objects l in the vehicle; setting $c_e(l)$ to the edge-length d_e when $l \leq k$, and $c_e(l) = \infty$ for $l > k$ gives us back the classical Dial-a-Ride setting. In Section 3.2, we show that this general *non-uniform Dial-a-Ride* problem admits an approximation guarantee that matches the best known for the classical Dial-a-Ride problem. Another extension we consider is the *weighted Dial-a-Ride* problem. In this, each object may have a different size, and the total size of the items in the vehicle must be bounded by the vehicle capacity; this is also known as the *pickup and delivery* problem [Savelsbergh and Sol 1995]. We show in Section 3.3 that this problem can be reduced to the (unweighted) Dial-a-Ride problem at the loss of only a constant factor in the approximation guarantee.

As an aside, we also consider the effect of preemptions in the Dial-a-Ride problem (Section 4). It was shown in Charikar and Raghavachari [1998] that the gap between the optimal preemptive and nonpreemptive tours could be as large as $\Omega(n^{1/3})$. We show that the real difference arises between *zero* and *one* preemptions: allowing multiple preemptions does not give us much added power. In particular, we show in Section 4.1 that for any instance of the Dial-a-Ride problem, there is a tour that preempts each object *at most once* and has length at most $O(\log^2 n)$ times an optimal preemptive tour (which may preempt each object an arbitrary number of times). Motivated by obtaining a better guarantee for Dial-a-Ride on the Euclidean plane, we also study the preemption gap in such instances. We show that even in this case, there are instances having a polynomial gap of $\tilde{\Omega}(n^{1/8})$ between optimal preemptive and non-preemptive tours. This preemption gap relies on the connection between the Dial-a-Ride and k-forest problems.

1.3. RELATED WORK.

The k-Forest Problem. The k-forest problem is relatively new: it was defined by Hajiaghayi and Jain [2006]. An $\tilde{O}(k^{2/3})$-approximation algorithm for even the directed k-forest problem can be inferred from [Charikar et al. 1999]. Recently, [Segev and Segev 2006] gave an $O(\min\{n^{2/3}, \sqrt{m}\} \log n)$ approximation algorithm for k-forest. The k-forest problem is a generalization of k-MST, for which a 2-approximation is known [Garg 2005].

Dense k-Subgraph. As shown in Hajiaghayi and Jain [2006], the *k-forest* problem generalizes the dense-k-subgraph problem [Feige et al. 2001]. The best known approximation guarantee for the dense-k-subgraph problem is $O(n^{1/3-\delta})$ where $\delta > 0$ is some constant, due to Feige et al. [2001], and obtaining an improved guarantee has been a long standing open problem. Strictly speaking, Feige et al. [2001] study a potentially harder problem: the *maximization* version of dense-k-subgraph, where one wants to pick k vertices to maximize the number of edges in the induced graph. However, nothing better is known even for the *minimization* version of dense-k-subgraph (where one wants to pick the minimum number of vertices that induce k edges). Moreover, the approximability of these two versions of dense-k-subgraph are polynomially related [Hajiaghayi and Jain 2006]. The minimization dense-k-subgraph problem on graph G reduces to k-forest by considering an unweighted star-metric, with leaves corresponding to vertices of G and pairs corresponding to edges of G.

Dial-a-Ride. While the Dial-a-Ride problem has been studied extensively in the operations research literature, relatively little is known about its approximability. The currently best known approximation ratio for Dial-a-Ride is $O(\sqrt{k} \log n)$ due to Charikar and Raghavachari [1998]. We note that their algorithm assumes instances with unweighted objects. [Krumke et al. 2000] give a 3-approximation algorithm for the Dial-a-Ride problem on a *line metric*; in fact, their algorithm finds a non-preemptive tour that has length at most 3 times the lower bounds for the preemptive version. (Clearly, the cost of an optimal preemptive tour is at most that of an optimal non-preemptive tour.) A 2.5-approximation algorithm for *single source* special case of Dial-a-Ride (also called the "capacitated vehicle routing" problem) was given in Haimovich and Kan [1985]; again, this algorithm outputs a non-preemptive tour with length at most 2.5 times the preemptive lower bounds. The $k = 1$ special case of Dial-a-Ride is also known as the *stacker-crane* problem, for which a 1.8-approximation is known [Frederickson et al. 1978]. For the *preemptive* Dial-a-Ride problem, Charikar and Raghavachari [1998] gave the current-best $O(\log n)$ approximation algorithm, and Gørtz [2006] showed that it is hard to approximate this problem to better than $\Omega(\log^{1/4-\epsilon} n)$. Recall that no super-constant hardness results are known for the non-preemptive Dial-a-Ride problem.

2. *The k-Forest Problem*

In this section, we study the k-forest problem, and give an approximation guarantee of $O(\min\{\sqrt{n}, \sqrt{k}\})$. This result improves upon the previous best $O(n^{2/3} \log n)$-approximation guarantee [Segev and Segev 2006] for this problem. The algorithm in Segev and Segev [2006] is based on a Lagrangian relaxation for this problem, and suitably combining solutions to this relaxation. In contrast, our algorithm uses a more direct approach and is much simpler in description. Our approach is based on approximating the following "density" variant of k-forest.

Definition 2.1 (*Minimum-Ratio k-Forest*). Given an n-vertex metric space (V, d), m pairs of vertices $\{s_i, t_i\}_{i=1}^m$, and a target k, find a tree T that connects *at most* k pairs, and minimizes the ratio of the length of T to the number of pairs connected in T.

Observe that given any forest F connecting some set of pairs, one of the trees in F has ratio (length to number of connected pairs) at most that of F. Hence, even if we relax the above definition to consider any forest, the optimal ratio solution can be assumed to be a tree. Given any feasible solution T to minimum-ratio k-forest, $\mathsf{Ratio}(T)$ denotes the ratio of length of T to the number of pairs connected in T.

We present two different algorithms for *minimum-ratio k-forest*, obtaining approximation guarantees of $O(\sqrt{k})$ (Section 2.1) and $O(\sqrt{n})$ (Section 2.2); these are then combined to give the claimed result for the k-forest problem. Both our algorithms are based on subtle reductions to the k-MST problem, albeit in very different ways. As is usual, when we say that our algorithm *guesses* a parameter in the following discussion, it means that the algorithm is run for each possible value of that parameter, and the best solution found over all the runs is returned. As long as only a constant number of parameters are being guessed and the number of possibilities for each of these parameters is polynomial, the algorithm is repeated only a polynomial number of times.

2.1. An $O(\sqrt{k})$ Approximation Algorithm. In this section, we give an $O(\sqrt{k})$ approximation algorithm for minimum ratio k-forest, which is based on a simple reduction to the k-MST problem. The idea is to look at the optimal solution S to minimum-ratio k-forest and consider an Euler tour of this tree S—a theorem of Erdős and Szekeres on increasing subsequences implies that there must be at least $\sqrt{|S|}$ sources which are visited in the same order as the corresponding sinks. We use this existence result to combine the source-sink pairs to create an instance of $\sqrt{|S|}$-MST from which we can obtain a good solution; the details follow. Below S denotes an optimal ratio tree, that covers q pairs and has length B; let D denote the largest distance between any demand-pair that is covered in S (note $D \leq B$).

The $O(\sqrt{k})$ approximation algorithm proceeds as below. Define a new metric l on the set $\{1, \ldots, m\}$ of pairs as follows. The distance between pairs i and j, $l_{i,j} \doteq d(s_i, s_j) + d(t_i, t_j)$, where (V, d) is the original metric. This metric represents solutions of a special structure: any tree M covering pairs Π can be expressed as $M = M_s \cup M_t \cup \{f\}$ where M_s (respectively, M_t) is a tree connecting all sources (respectively, destinations) in Π and f is any edge connecting a pair in Π. The algorithm guesses the number of pairs q and the largest demand-pair distance D in the optimal tree S (there are at most m choices for each of q and D). The algorithm discards all pairs (s_i, t_i) such that $d(s_i, t_i) > D$ (all the pairs covered in the optimal solution S still remain). Then, the algorithm runs the unrooted k-MST algorithm [Garg 2005] with target $\lfloor\sqrt{q}\rfloor$, in the metric l, to obtain a tree T on the pairs P. From T, we easily obtain trees T_1 (on all sources in P) and T_2 (on all sinks in P) in metric d such that $d(T_1) + d(T_2) = l(T)$. Finally the algorithm outputs the tree $T' = T_1 \cup T_2 \cup \{e\}$, where e is any edge joining a source in T_1 to its corresponding sink in T_2.

Due to the pruning on pairs that have large distance, $d(e) \leq D$ and the length of T', $d(T') \leq l(T) + D \leq l(T) + B$. We now argue that the cost of the solution T found by the k-MST algorithm $l(T) \leq 8B$. Consider the optimal ratio tree S (in metric d) that has q pairs $\{(s_1, t_1), \ldots, (s_q, t_q)\}$, and let τ denote an Euler tour of S. Suppose that in a traversal of τ, the *sources* of pairs in S are seen in the order

$s_1, \ldots, s_q$. Then, in the same traversal, the *sinks* of pairs in S will be seen in the order $t_{\pi(1)}, \ldots, t_{\pi(q)}$, for some permutation π. The following fact is well known (see, e.g., Steele [1995]).

THEOREM 2.2 (ERDŐS AND SZEKERES). *Every permutation on $\{1, \ldots, q\}$ has either an increasing subsequence of length $\lfloor\sqrt{q}\rfloor$ or a decreasing subsequence of length $\lfloor\sqrt{q}\rfloor$.*

Using Theorem 2.2, we obtain a set M of $p = \lfloor\sqrt{q}\rfloor$ pairs such that (1) the sources in M appear in increasing order in a traversal of the Euler tour τ, and (2) the sinks in M appear in increasing order in a traversal of either τ or τ^R (the reverse traversal of τ). Let $j_0 < j_1 < \cdots < j_{p-1}$ denote the pairs in M in increasing order. From statement (1), $\sum_{i=0}^{p-1} d(s(j_i), s(j_{i+1})) \leq d(\tau)$, where the indices in the summation are modulo p. Similarly, statement (2) implies that $\sum_{i=0}^{p-1} d(t(j_i), t(j_{i+1})) \leq \max\{d(\tau), d(\tau^R)\} = d(\tau)$. Thus we obtain:

$$\sum_{i=0}^{p-1}[d(s(j_i), s(j_{i+1})) + d(t(j_i), t(j_{i+1}))] \leq 2d(\tau) \leq 4B$$

But this sum is precisely the length of the tour $j_0, j_1, \ldots, j_{p-1}, j_0$ in metric l. In other words, there is a tree of length $4B$ in metric l, that contains $\lfloor\sqrt{q}\rfloor$ vertices. So, the cost of the solution T found by the k-MST approximation algorithm is at most $8B$.

Now the final solution T' has length at most $l(T) + B \leq 9B$, and $\mathsf{Ratio}(T') \leq 9\sqrt{q}\frac{B}{q} \leq 9\sqrt{k}\frac{B}{q}$. Thus we have an $O(\sqrt{k})$ approximation algorithm for minimum ratio k-forest.

2.2. AN $O(\sqrt{n})$ APPROXIMATION ALGORITHM. In this section, we show an $O(\sqrt{n})$ approximation algorithm for the minimum ratio k-forest problem. The approach is again to reduce to the k-MST problem; the idea is rather different: either we find a vertex v such that a large number of demand-pairs of the form $(v, *)$ can be satisfied using a small tree (the "high-degree" case), or we use a repeated greedy procedure to cover most vertices without paying too much (since we are in the "low-degree" case, covering most vertices implies covering most pairs too). The details follow.

Let S denote an optimal solution to minimum ratio k-forest, and $q \leq k$ the number of demand pairs covered in S. We define the *degree* Δ of S to be the maximum number of demand-pairs (among those covered in S) that are incident at any vertex in S. The algorithm first guesses the following parameters of the optimal solution S: its length B (within a factor 2), the number of pairs covered q, the degree Δ, and the vertex $w \in S$ that has Δ demand-pairs incident at it. Although, there may be an exponential number of choices for the optimal length, a polynomial number of guesses within a binary-search suffice to get a B such that $B \leq d(S) \leq 2 \cdot B$. The algorithm then returns the better of the two procedures described below.

Procedure 1 (*High-Degree Case*). The algorithm assigns a weight to each vertex u, equal to the number of pairs having an end point at u and the other end point at w (the guessed Δ-degree vertex in S). Then, we run the k-MST algorithm [Garg 2005] with root w and a target weight of Δ, resulting in a solution tree H. Since the degree of vertex w in the optimal solution S is Δ, there is tree rooted at w of length $d(S) \leq 2B$, that contains at least Δ pairs having one end point at w. Hence, the k-MST instance has a feasible solution of length $2B$, and the length of solution H is at most $4B$ (since the algorithm of Garg [2005] is a 2-approximation). Thus, $\mathsf{Ratio}(H) \leq 4B/\Delta = \frac{4q}{\Delta}\frac{B}{q}$.

Procedure 2 (*Low-Degree Case*). Set $t = \frac{q}{2\Delta}$; note that $q \leq \frac{\Delta \cdot n}{2}$ and so $t \leq n/4$. We maintain a current tree T (which is initialized to $T \leftarrow \{w\}$), and iteratively do the following:

(1) Shrink T to a single vertex s in metric (V, d), and run the k-MST algorithm [Garg 2005] with root s and a target of t new vertices. Let T_0 denote the resulting tree.

(2) If T_0 has length at most $4B$, set $T \leftarrow T \cup T_0$ and continue to the next iteration.

(3) If T_0 has length more than $4B$ (or if T already has all vertices), then terminate.

The tree T at the end of these iterations is output as the solution to minimum ratio k-forest. Since t new vertices are added in each iteration, the number of iterations is at most $\frac{n}{t}$; so the length of T is at most $\frac{4n}{t}B$. We now show that T contains at least $\frac{q}{2}$ demand-pairs. Consider the set $S \setminus T$ (recall, S is the optimal solution). It is clear that $|V(S) \setminus V(T)| < t$; otherwise, the k-MST instance in the last iteration (with the current T) would have S as a feasible solution of length at most $2B$ (and hence would find one of length at most $4B$). So the number of pairs covered in S that have at least one end point in $S \setminus T$ is at most $|V(S) \setminus V(T)| \cdot \Delta \leq t \cdot \Delta = q/2$ (as Δ is the degree of solution S). Thus, there are at least $q/2$ pairs *contained* in $S \cap T$, in particular in T. Thus T is a solution with $\mathsf{Ratio}(T) \leq \frac{4n}{t}B \cdot \frac{2}{q} = \frac{8n}{t}\frac{B}{q}$.

The better solution among H and T from the above two procedures has objective value at most $\min\{\frac{4q}{\Delta}, \frac{8n}{t}\} \cdot \frac{B}{q} = \min\{8t, \frac{8n}{t}\} \cdot \frac{B}{q} \leq 8\sqrt{n} \cdot \frac{B}{q} \leq 8\sqrt{n} \cdot \frac{d(S)}{q}$. So this algorithm is an $O(\sqrt{n})$ approximation to the minimum ratio k-forest problem.

2.3. Approximation Algorithm for k-Forest. Given the two algorithms for minimum ratio k-forest, we can use them in a standard greedy fashion (i.e., keep picking approximately minimum-ratio solutions until we obtain a forest connecting at least k pairs); the standard set cover analysis can be used to show an $O(\min\{\sqrt{n}, \sqrt{k}\} \cdot \log k)$-approximation guarantee for k-forest. The $O(\sqrt{k} \cdot \log k)$ part of the bound can be improved slightly to $O(\sqrt{k})$. This uses a tighter analysis of the greedy set-cover algorithm [Charikar et al. 1999]. Lemma 1 from Charikar et al. [1999] implies the following in our context: Suppose there is an $f(k)$ approximation algorithm for minimum ratio k-forest, where $f(x)/x$ is a decreasing function of x. Then, the greedy algorithm for the k-forest problem achieves an approximation guarantee of $\int_0^k f(x)/x \, dx$. Using $f(k) = O(\sqrt{k})$, we obtain an $O(\sqrt{k})$ approximation for k-forest, implying the guarantee in Theorem 1.2.

We note that this greedy approach to solving the k-forest problem may not give an $o(n)$ approximation bound when k is super-polynomial in n. In this case however, our $O(\sqrt{n})$-approximation algorithm for minimum ratio k-forest can be used within the Lagrangian relaxation framework of Segev and Segev [2006] (in place of Theorem 3) to obtain an $O(\sqrt{n} \cdot \log n)$ approximation for k-forest.

3. *Applications to Dial-a-Ride Problems*

In this section, we study applications of k-forest to the Dial-a-Ride problem (Definition 1.3), and some generalizations. A natural solution-structure for Dial-a-Ride involves servicing objects in batches of at most k each, where a batch consisting of a set S of demand-pairs is served as follows: the vehicle starts out being empty, picks up each of the $|S| \leq k$ objects from their sources, then drops off each object at its destination, and is again empty at the end. If we knew that the optimal solution has this structure, we could obtain a greedy framework for Dial-a-Ride by repeatedly finding the best 'batch' of k demand-pairs. However, the optimal solution may involve carrying almost k objects at every point in the tour, in which case it can not be decomposed to be of the above structure. In Theorem 3.1, we show that there is always a near optimal solution having this "pick-drop in batches" structure. Building on Theorem 3.1, we obtain approximation algorithms for the classical Dial-a-Ride problem (Section 3.1), and two interesting extensions: non-uniform Dial-a-Ride (Section 3.2) and weighted Dial-a-Ride (Section 3.3).

THEOREM 3.1 (STRUCTURE THEOREM). *Given any instance of Dial-a-Ride, there exists a feasible tour τ satisfying the following conditions:*

(1) *τ can be split into a set of segments $\{S_1, \ldots, S_t\}$ (i.e., $\tau = S_1 \cdot S_2 \cdots S_t$) where each segment S_i services a set O_i of at most k objects such that S_i is a path that first picks up each object in O_i and then drops each of them.*
(2) *The length of τ is at most $O(\log m)$ times the length of an optimal tour.*

PROOF. Consider an optimal non-preemptive tour σ: let $d(\sigma)$ denote its length, and $|\sigma|$ denote the number of edge traversals in σ. Note that if in some visit to a vertex v in σ there is no pick-up or drop-off, then the tour can be short-cut over vertex v, and it still remains feasible. Further, due to triangle inequality, the length $d(\sigma)$ does not increase by this operation. So we may assume that each vertex visit in σ involves a pick-up or drop-off of some object. Since there is exactly one pick-up and drop-off for each object, we have $|\sigma| \leq 2m + 1$. Define the *stretch* of a demand-pair i to be the number of edge traversals in σ between the pick-up and drop-off of object i. The demand-pairs are partitioned as follows: for each $j = 1, \ldots, \lceil \log(2m) \rceil$, group G_j consists of all pairs having stretch between 2^{j-1} and 2^j. We consider each group G_j separately.

CLAIM 3.2. *For each $j = 1, \ldots, \lceil \log(2m) \rceil$, there is a tour τ_j that serves all the pairs in group G_j, satisfies condition 1 of Theorem* 3.1*, and has length at most* $6 \cdot d(\sigma)$.

PROOF. Consider tour σ as a line $\mathcal{L}$, with every edge traversal in σ represented by a distinct edge in $\mathcal{L}$. Number the vertices in $\mathcal{L}$ from 0 to h, where $h = |\sigma|$ is the number of edge traversals in σ. Note that each vertex in V may be represented multiple times in $\mathcal{L}$. Each object is associated with the numbers of the vertices (in $\mathcal{L}$) where it is picked up and dropped off.

Let $r = 2^{j-1}$, and partition G_j as follows: for $l = 1, \ldots, \lceil \frac{h}{r} \rceil$, set $O_{l,j}$ consists of all objects in G_j that are picked up at a vertex numbered between $(l-1)r$ and $lr - 1$. Since every object in G_j has stretch in the interval $[r, 2r]$, every object in $O_{l,j}$ is dropped off at a vertex numbered between lr and $(l+2)r - 1$. Note that $|O_{l,j}|$ equals the number of objects in G_j carried over edge $(lr - 1, lr)$ by tour σ, which is at most k. We define segment $S_{l,j}$ to start at vertex number $(l-1)r$ and traverse all edges in $\mathcal{L}$ until vertex number $(l+2)r-1$ (servicing all demand-pairs in $O_{l,j}$ by first picking up each object between vertices $(l-1)r$ and $lr-1$; then dropping off each object between vertices lr and $(l+2)r-1$), and then return (with the vehicle being empty) to vertex lr. Clearly, the number of objects carried over any edge in $S_{l,j}$ is at most the number carried over the corresponding edge traversal in σ. Also, each edge in $\mathcal{L}$ participates in at most 3 segments $\{S_{l,j} \mid 1 \le l \le \lceil h/r \rceil\}$, and each edge is traversed at most twice in any segment. So the total length of all segments $\{S_{l,j}\}$ is at most $6 \cdot d(\sigma)$. We define tour τ_j to be the concatenation $S_{1,j} \cdots S_{\lceil h/r \rceil, j}$. It is clear that this tour satisfies condition 1 of Theorem 3.1. □

Applying this claim to each group G_j, and concatenating the resulting tours, we obtain the tour τ satisfying condition 1 and having length at most $6\log(2m)\cdot d(\sigma) = O(\log m)\cdot d(\sigma)$.

Remark. The ratio $O(\log m)$ in Theorem 3.1 is almost best possible. As mentioned in Gørtz [2005], there are instances of Dial-a-Ride on an unweighted line, where every solution satisfying condition 1 of Theorem 3.1 has length at least $\Omega(\max\{\frac{\log m}{\log\log m}, \frac{k}{\log k}\})$ times the optimal nonpreemptive tour. These instances consist of $n = 2^k + 1$ equally spaced vertices on a line, numbered 1 through $n+1$ from left to right, with demand-pairs $\{(j \cdot 2^i, (j+1)2^i) \mid 0 \le i \le k,\ 0 \le j \le 2^{k-i} - 1\}$. It can be seen that the optimal non-preemptive tour has length $O(n)$, whereas any tour satisfying condition 1 of Theorem 3.1 has length at least $\Omega(n \cdot \frac{\log n}{\log\log n})$. So, if we only use solutions of this "pick-drop" structure, then it is not possible to obtain an approximation factor (just in terms of capacity k) for Dial-a-Ride that is better than $\Omega(k/\log k)$. The solutions found by the algorithm for Dial-a-Ride in Charikar and Raghavachari [1998] also satisfy condition 1 of Theorem 3.1. It is interesting to note that when the underlying metric is a hierarchically well-separated tree (HST), [Charikar and Raghavachari 1998] obtain a solution of such structure having length $O(\sqrt{k})$ times the optimum, whereas there is a lower bound of $\Omega(\frac{k}{\log k})$ even for the simple case of an unweighted line (which is not an HST).

3.1. CLASSICAL DIAL-A-RIDE. Theorem 3.1 suggests a greedy strategy for Dial-a-Ride, based on repeatedly finding the best batch of k objects to service. This greedy subproblem turns out to be the minimum ratio k-forest problem (Definition 2.1), for which we already have an approximation algorithm. The next theorem sets up this reduction.

THEOREM 3.3 (REDUCING DIAL-A-RIDE TO MINIMUM RATIO k-FOREST). *A ρ-approximation algorithm for minimum ratio k-forest implies an $O(\rho \log^2 m)$-approximation algorithm for Dial-a-Ride.*

PROOF. The algorithm for Dial-a-Ride is as follows.

(1) $\mathcal{C} = \phi$.
(2) Until there are no uncovered demand-pairs, do:
    (a) Solve the minimum ratio k-forest problem, to obtain a tree C covering $k_C \le k$ new pairs.
    (b) Set $\mathcal{C} \leftarrow \mathcal{C} \cup C$.
(3) For each tree $C \in \mathcal{C}$, obtain an Euler tour on C to locally service all demand-pairs (pick up all k_C objects in the first traversal, and drop them all in the second traversal). Then use a 1.5-approximate TSP tour on the sources, to connect all the local tours, and obtain a feasible non-preemptive tour.

Consider the tour τ and its segments as in Theorem 3.1. If the number of uncovered pairs in some iteration is m', one of the segments in τ is a solution to the minimum ratio k-forest problem of value at most $\frac{d(\tau)}{m'}$. Since we have a ρ-approximation algorithm for this problem, we would find a segment of ratio at most $O(\rho) \cdot \frac{d(\tau)}{m'}$. Now a standard set cover type argument shows that the total length of trees in $\mathcal{C}$ is at most $O(\rho \log m) \cdot d(\tau) \leq O(\rho \log^2 m) \cdot \mathsf{OPT}$, where OPT is the optimal value of the Dial-a-Ride instance. Further, the TSP tour on all sources is a lower bound on OPT, and we use a 1.5-approximate solution [Christofides 1977]. So the final non-preemptive tour output in step 5 above has length at most $O(\rho \log^2 m) \cdot \mathsf{OPT}$. □

This theorem is in fact stronger than Theorem 1.4 claimed earlier: any approximation algorithm for k-forest implies an algorithm with the same guarantee for minimum ratio k-forest. Note that, m and k may be super-polynomial in n. However, we show in Section 3.3 that with the loss of a constant factor, even the weighted Dial-a-Ride problem can be reduced to classical Dial-a-Ride where the number of objects $m \leq n^4$. Based on this and Theorem 3.3, a ρ approximation algorithm for minimum ratio k-forest actually implies an $O(\rho \log^2 n)$ approximation algorithm for Dial-a-Ride. Using the approximation algorithm for minimum ratio k-forest (Section 2), we obtain an $O(\min\{\sqrt{n}, \sqrt{k}\} \cdot \log^2 n)$ approximation algorithm for the Dial-a-Ride problem.

Remark. If we use the $O(\sqrt{k})$ approximation for k-forest, the resulting non-preemptive tour is in fact feasible even for a $\sqrt{k}$ capacity vehicle! As noted in Charikar and Raghavachari [1998], this property is also true of their algorithm, which is based on an entirely different approach.

3.2. NONUNIFORM DIAL-A-RIDE. The greedy framework for Dial-a-Ride described above is actually more generally applicable than to just the classical Dial-a-Ride problem. In this section, we consider the Dial-a-Ride problem under a substantially more general class of cost functions, and show how the k-forest problem can be used to obtain an approximation algorithm for this generalization as well. In fact, the approximation guarantee we obtain by this approach matches (up to logarithmic factors) the best known for the classical Dial-a-Ride problem. Our framework for Dial-a-Ride is well suited for such a generalization since it is based on directly approximating a near-optimal solution; this approach is not too sensitive to the cost function. On the other hand, the algorithm in Charikar and Raghavachari [1998] is based on obtaining a good lower bound, which depends heavily on the cost function. Thus, it is unclear whether their techniques can be extended to handle such a generalization.

Definition 3.4 (*Nonuniform Dial-a-Ride*). Given an n vertex undirected graph $G = (V, E)$, a root vertex r, a set of m demand-pairs $\{(s_i, t_i)\}_{i=1}^m$, and a nondecreasing cost function $c_e : \{0, 1, \ldots, m\} \rightarrow \mathbb{R}^+$ on each edge $e \in E$ (where $c_e(l)$ is the cost incurred by the vehicle in traversing edge e while carrying l objects), find a nonpreemptive tour (starting and ending at r) of minimum *total cost* that moves each object i from s_i to t_i.

Note that the classical Dial-a-Ride problem is a special case when the edge costs are given by: $c_e(l) = d_e$ if $l \leq k$ and $c_e(l) = \infty$ otherwise, where d_e is the edge length in the underlying metric. We may assume (without loss in generality) that for any *fixed* value $l \in [0, m]$, the edge costs $c_e(l)$ induce a metric on V. Similar to Theorem 3.1, we have a near optimal solution with a 'batch' structure for the non-uniform Dial-a-Ride problem as well, which implies the algorithm in Theorem 3.6.

COROLLARY 3.5 (NONUNIFORM STRUCTURE THEOREM). *Given any instance of nonuniform Dial-a-Ride, there exists a feasible tour τ satisfying the following conditions:*

(1) *τ can be split into a set of segments $\{S_1, \ldots, S_t\}$ (i.e., $\tau = S_1 \cdot S_2 \cdots S_t$) where each segment S_i services a set O_i of demand-pairs such that S_i is a path that first picks up each object in O_i and then drops each of them.*

(2) *The cost of τ is at most $O(\log m)$ times the cost of an optimal tour.*

PROOF. We only give a proof sketch highlighting how the proof of Theorem 3.1 carries over to this case. We may again assume that the number of edge traversals in the an optimal tour σ is at most $2m$: this uses triangle inequality in the edge-costs $c_e(l)$ for any fixed $l \in [m]$. The definitions of groups $\{G_j \mid j = 1, \ldots, \lceil \log(2m) \rceil\}$, and $\{O_{l.j}\}$ for each $1 \leq j \leq \lceil \log(2m) \rceil$ are identical to those in Theorem 3.1. The traversal $S_{l,j}$ serving any group $O_{l.j}$ has the property that the number of objects carried over any edge in $S_{l,j}$ is at most that carried over the same edge in σ: this implies that the cost of $S_{l,j}$ is at most that of σ between vertex numbers $(l-1)r$ and $(l+2)r - 1$. Finally, concatenating all the local tours $S_{l,j}$, we obtain the desired property. $\square$

THEOREM 3.6 (APPROXIMATING NONUNIFORM DIAL-A-RIDE). *A ρ-approximation algorithm for minimum ratio k-forest (for all values of k) implies an $O(\rho \log^2 m)$-approximation algorithm for non-uniform Dial-a-Ride. In particular, there is an $O(\sqrt{n} \log^2 m)$-approximation algorithm.*

PROOF. Corollary 3.5 again suggests a greedy algorithm for non-uniform Dial-a-Ride based on the following *greedy subproblem*: find a set T of uncovered pairs and a path τ_0 that first picks up each object in T and then drops off each of them, such that the ratio of the cost of τ_0 to $|T|$ is minimized. However, unlike in the classical Dial-a-Ride problem, in this case the cost of path τ_0 does not come from a single metric. Nevertheless, the minimum ratio k-forest problem can be used to solve this subproblem as follows.

(1) For every $k = 1, \ldots, m$:
   (a) Define length function $d_e^{(k)} = c_e(k)$ on the edges.
   (b) Solve the minimum ratio k-forest problem on metric $(V, d^{(k)})$ with bound k, to obtain tree T'_k covering $n_k \leq k$ pairs.
   (c) Obtain an Euler tour T_k of T'_k that services these n_k objects, by picking up all objects in one traversal and then dropping them all in a second traversal.

(2) Return the tour T_k having the smallest ratio $\frac{c(T_k)}{n_k}$ (over all $1 \leq k \leq m$).

Assuming a ρ-approximation algorithm for minimum ratio k-forest (for all values of k), we now show that the above algorithm obtains a 16ρ-approximate solution to the greedy subproblem. The cost of tour T_k in step 1 is $c(T_k) \leq 4 \cdot d^{(k)}(T'_k)$, since T_k involves traversing a tour on tree T'_k twice and the vehicle carries at most $n_k \leq k$ objects at every point in T_k. So tour T_k has $\frac{c(T_k)}{n_k} \leq 4\frac{d^{(k)}(T'_k)}{n_k} = 4 \cdot \mathsf{Ratio}(T'_k)$ (recall that $\mathsf{Ratio}(F)$ for any solution to minimum ratio k-forest is the ratio of length of F to the number of pairs connected by F). Let τ denote the optimal path for the greedy subproblem, T the set of objects that it services, and $t = |T|$. Let T_1 denote the last $\frac{3}{4}t$ objects that are picked up, and T_2 denote the first $\frac{3}{4}t$ objects that are dropped off. It is clear that $T_1 \cap T_2$ has at least $t/2$ objects; let $T' \subset T_1 \cap T_2$ be any subset with $|T'| = t/4$. Let τ' denote the portion of τ between the $\frac{t}{4}$-th pick up and the $\frac{3t}{4}$-th drop off. Note that when path τ is traversed, there are at least $\frac{t}{4}$ objects in the vehicle while traversing each edge in τ'. So the cost of τ, $c(\tau) \geq \sum_{e\in\tau'} c_e(t/4)$. Also τ' contains the end points of all objects in $T' \supseteq T_1 \cap T_2$. Hence, τ' corresponds to a feasible solution F' (covering pairs T') to minimum ratio k-forest with bound $k = t/4$ in metric $d^{(t/4)}$. Solution F' has $\mathsf{Ratio}(F') = (\sum_{e\in\tau'} c_e(t/4))/\frac{t}{4} \leq \frac{4c(\tau)}{t}$. Thus the ρ-approximate solution $T'_{t/4}$ has $\mathsf{Ratio}(T'_{t/4}) \leq 4\rho\frac{c(\tau)}{t}$. So the tour $T_{t/4}$ has ratio $\frac{c(T_k)}{n_k} \leq 4 \cdot \mathsf{Ratio}(T'_k) \leq 16\rho\frac{c(\tau)}{t}$. Thus, we have a 16ρ-approximation algorithm for the greedy subproblem.

Based on Corollary 3.5, it can now be shown (as in Theorem 3.3) that a ρ'-approximation algorithm for the greedy subproblem implies an $O(\rho' \cdot \log^2 m)$-approximation algorithm for non-uniform Dial-a-Ride. Using the above 16ρ-approximation for the greedy subproblem, we have the theorem. □

3.3. WEIGHTED DIAL-A-RIDE. So far, we worked with the unweighted version of Dial-a-Ride, where each object has the same weight. In this section, we extend our greedy framework for Dial-a-Ride to the case when objects have different sizes, and the total size of objects in the vehicle must be bounded by the vehicle capacity. Here we only extend the classical Dial-a-Ride problem and not the generalization of Section 3.2. The problem studied in this section is also known as the *pickup and delivery* problem [Savelsbergh and Sol 1995].

Definition 3.7 (*Weighted Dial-a-Ride*). Given a vehicle of capacity $Q \in \mathbb{N}$, an n-vertex metric space (V, d), a root vertex r, and a set of m objects $\{(s_i, t_i, w_i)\}_{i=1}^m$ (with object i having source s_i, destination t_i and an integer size $1 \leq w_i \leq Q$), find a minimum length (nonpreemptive) tour of the vehicle starting (and ending) at r that moves each object i from its source to its destination such that the total size of objects carried by the vehicle is at most Q at any point on the tour.

The classical Dial-a-Ride problem is a special case when $w_i = 1$ for all objects and the vehicle capacity $Q = k$. The following are two lower bounds for weighted Dial-a-Ride: a TSP tour on the set of all sources and destinations (Steiner lower bound); and $\sum_{i=1}^m \frac{w_i \cdot d(s_i,t_i)}{Q}$ (flow lower bound). In fact, as can be seen easily, these two lower bounds are valid even for the preemptive version of weighted Dial-a-Ride; so they are termed *preemptive lower bounds*.

The main result of this section (Theorem 3.9) reduces weighted Dial-a-Ride to the classical Dial-a-Ride problem with the additional property that the number m of objects is small (polynomial in the number of vertices n). This shows that in order to approximate weighted Dial-a-Ride, it suffices to consider instances of the classical Dial-a-Ride problem with a small number of objects. The next lemma shows that even if the vehicle is allowed to split each object over multiple deliveries, the resulting tour is *not* much shorter than the tour where each object is required to be served in a single delivery (as is the case in weighted Dial-a-Ride). This lemma is the main ingredient in the proof of Theorem 3.9. In the following, for any instance of weighted Dial-a-Ride, we define the *unweighted instance* corresponding to it as a classical Dial-a-Ride instance with vehicle capacity Q, having w_i (unweighted) objects with source s_i and destination t_i (for each $1 \le i \le m$).

LEMMA 3.8. *Given any instance $\mathcal{I}$ of weighted Dial-a-Ride, and a solution τ to the unweighted instance corresponding to $\mathcal{I}$, there is a polynomial-time computable solution to $\mathcal{I}$ having length at most $O(1) \cdot d(\tau)$.*

PROOF. Let $\mathcal{J}$ denote the unweighted instance corresponding to $\mathcal{I}$. Define line $\mathscr{L}$ as in the proof of Theorem 3.1 constructed by traversing τ from r: for every edge traversal in τ, add a new edge of the same length at the end of $\mathscr{L}$. Note that there is a 1-1 correspondence between edges in $\mathscr{L}$ and edge-traversals in τ. For each unweighted object in $\mathcal{J}$ corresponding to object i in $\mathcal{I}$, there is a segment in τ (correspondingly in $\mathscr{L}$) where it is moved from s_i to t_i. So each object $i \in \mathcal{I}$ corresponds to w_i segments in τ (each being a path from s_i to t_i). For each object i in $\mathcal{I}$, we assign i to one of its w_i segments picked uniformly at random: call this segment l_i. For an edge $e \in \mathscr{L}$, let $N_e = \sum_{i:e \in l_i} w_i$ denote the random variable which equals the *total weight* of objects whose assigned segments contain e. Note that the expected value of N_e is exactly the number of unweighted objects carried by τ when traversing the edge corresponding to e. Since τ is a feasible tour for $\mathcal{J}$, $E[N_e] \le Q$ for all $e \in \mathscr{L}$.

Consider a random instance $\mathcal{R}$ of Dial-a-Ride on line $\mathscr{L}$ with vehicle capacity Q and objects as follows: for each object i in $\mathcal{I}$, an object of weight w_i is to be moved along segment l_i (chosen randomly as above). Clearly, any feasible tour for $\mathcal{R}$ corresponds to a feasible tour for $\mathcal{I}$ of the same length. Note that the flow lower bound for instance $\mathcal{R}$ is $F = \sum_{e \in \mathscr{L}} d_e \frac{N_e}{Q}$, and the Steiner lower bound is $\sum_{e \in \mathscr{L}} d_e = d(\tau)$. Using linearity of expectation, $E[F] \le \sum_{e \in \mathscr{L}} d_e \frac{E[N_e]}{Q} \le \sum_{e \in \mathscr{L}} d_e = d(\tau)$. Let R^* denote the Dial-a-Ride instance on line $\mathscr{L}$ obtained by assigning each object i in $\mathcal{I}$ to the segment corresponding to it (among its w_i segments) that has the smallest number of edges. Clearly, this assignment minimizes the flow lower bound (over all assignments of objects to segments). So R^* has flow bound $\le E[F] \le d(\tau)$, and Steiner lower bound $d(\tau)$.

Finally, we note that the 3-approximation algorithm for Dial-a-Ride on a line [Krumke et al. 2000] extends to a constant factor approximation algorithm for the case with weighted objects as well (this can be seen directly from Krumke et al. [2000]). Additionally, this approximation guarantee is relative to the preemptive lower bounds. Thus, using this algorithm on R^*, we obtain a feasible solution to $\mathcal{I}$ of length at most $O(1) \cdot d(\tau)$. □

THEOREM 3.9 (WEIGHTED DIAL-A-RIDE TO UNWEIGHTED). *Suppose there is a ρ-approximation algorithm for instances of classical Dial-a-Ride with at most $O(n^4)$ objects. Then there is an $O(\rho)$-approximation algorithm for weighted Dial-a-Ride (with any number of objects). In particular, there is an $O(\sqrt{n}\log^2 n)$ approximation for weighted Dial-a-Ride.*

PROOF. Let $\mathcal{I}$ denote an instance of weighted Dial-a-Ride with objects $\{(w_i, s_i, t_i): 1 \le i \le m\}$, and τ^* an optimal tour for $\mathcal{I}$. Let $\mathcal{P} = \{(s_1, t_1), \ldots, (s_l, t_l)\}$ be the distinct pairs of vertices that have some demand-pair between them, and let T_i denote the total size of *all* objects having source s_i and destination t_i. Note that $l \le n(n-1)$. Let $\mathcal{P}_{high} = \{i \in \mathcal{P} : T_i \ge \frac{Q}{2}\}$, $\mathcal{P}_{low} = \{i \in \mathcal{P} : T_i \le \frac{Q}{l}\}$, and $\mathcal{P}' = \mathcal{P} \setminus (\mathcal{P}_{high} \cup \mathcal{P}_{low})$. We now show how to separately service objects in $\mathcal{P}_{low}$, $\mathcal{P}_{high}$ and $\mathcal{P}'$.

Servicing $\mathcal{P}_{low}$. The total size in $\mathcal{P}_{low}$ is at most Q; so we can service all these pairs by using a 1.5-approximate tour [Christofides 1977] on sources and destinations, and traversing it twice: once to pick up all objects and once to drop them. Note that the length of this tour is at most three times the Steiner lower bound, hence at most $3 \cdot d(\tau^*)$.

Servicing $\mathcal{P}_{high}$. Let C be a 1.5-approximate minimum tour on all the sources. The pairs in $\mathcal{P}_{high}$ are serviced by a tour τ_1 as follows. Traverse along C, and when a source s_i in $\mathcal{P}_{high}$ is visited, traverse the direct edge to the corresponding destination t_i and back, as few times as possible so as to move all the objects between s_i and t_i, as described next. Note that every object to be moved between s_i and t_i has size (the original w_i size) at most Q, and the total size of such objects $T_i \geq Q/2$. So these objects can be partitioned such that the size of each part (except possibly the last) is in the interval $[\frac{Q}{2}, Q]$. So the number of times edge (s_i, t_i) is traversed to service the demand-pairs between them is at most $2\lceil\frac{2T_i}{Q}\rceil \leq 2(\frac{2T_i}{Q}+1) \leq 8\frac{T_i}{Q}$. Now, the length of tour τ_1 is at most $d(C) + \sum_{i \in \mathcal{P}_{high}} 8d(s_i, t_i)\frac{T_i}{Q} \leq d(C) + \frac{8}{Q}\sum_{i=1}^{m} w_i \cdot d(s_i, t_i)$. Note that $d(C)$ is at most 1.5 times the minimum tour on all sources (Steiner lower bound), and the second term above is the flow lower bound. So tour τ_1 has length at most $O(1)$ times the preemptive lower bounds for $\mathcal{I}$, which is at most $O(1) \cdot d(\tau^*)$.

Servicing $\mathcal{P}'$. We know that the total size T_i of each pair i in $\mathcal{P}'$ lies in the interval $(Q/l, Q/2)$. Let $\mathcal{I}'$ denote the instance of weighted Dial-a-Ride with objects $\{(s_i, t_i, T_i) : i \in \mathcal{P}'\}$ and vehicle capacity Q; note that the number of objects in $\mathcal{I}'$ is at most l. The tour τ^* restricted to the objects corresponding to pairs in $\mathcal{P}'$ is a feasible solution to the *unweighted instance* corresponding to $\mathcal{I}'$ (but it may not be feasible for $\mathcal{I}'$ itself). However, Lemma 3.8 implies that the optimal value of $\mathcal{I}'$, $\mathsf{OPT}(\mathcal{I}') \leq O(1) \cdot d(\tau^*)$.

Next, we reduce instance $\mathcal{I}'$ to an instance $\mathcal{J}$ of weighted Dial-a-Ride satisfying the following conditions: (i) $\mathcal{J}$ has at most l objects, (ii) each object in $\mathcal{J}$ has size at most $2l$, (iii) any feasible solution to $\mathcal{J}$ is feasible for $\mathcal{I}'$, and **(iv)** the optimal value $\mathsf{OPT}(\mathcal{J}) \leq O(1) \cdot \mathsf{OPT}(\mathcal{I}')$. If $Q \leq 2l$, $\mathcal{J} = \mathcal{I}'$ itself satisfies the required conditions. Suppose $Q \geq 2l$, then define $p = \lfloor\frac{Q}{l}\rfloor$; note that $Q \geq l \cdot p \geq Q - l \geq \frac{Q}{2}$. Round up each size T_i to the smallest integral multiple T_i' of p, and round down the capacity Q to $Q' = l \cdot p$. Since each size $T_i \in (\frac{Q}{l}, \frac{Q}{2})$, all sizes $T_i' \in \{p, 2p, \ldots, lp\}$. Now let $\mathcal{I}''$ denote the weighted Dial-a-Ride instance with objects $\{(s_i, t_i, T_i') : i \in \mathcal{P}'\}$ and vehicle capacity $Q' = lp$.

One can obtain a feasible solution for $\mathcal{I}''$ from any feasible solution σ for $\mathcal{I}'$ by traversing σ a constant number of times as follows. Consider simulating a traversal of a capacity Q vehicle α along σ by 16 capacity Q' vehicles $\{\beta_g\}_{g=1}^{16}$, each running in parallel along σ. The objects $\{i \mid T_i' \leq \frac{Q}{4}\}$ are served by vehicles $\{\beta_g\}_{g=1}^{8}$, and the rest by vehicles $\{\beta_g\}_{g=9}^{16}$. Whenever vehicle α picks-up an object i, one of the vehicles $\{\beta_g\}_{g=1}^{16}$ picks up i: if $T_i' \leq \frac{Q}{4}$, any vehicle $\{\beta_g\}_{g=1}^{8}$ that has free capacity picks up i; if $T_i' > \frac{Q}{4}$, any vehicle $\{T_g'\}_{g=9}^{16}$ that is empty picks up i. It can be seen that if some object is not picked by any vehicle $\{\beta_g\}_{g=1}^{16}$, then there must be a capacity violation in α (since $Q' \geq \frac{Q}{2}$ and $T_i' \leq \max\{2T_i, Q'\}$).

So the optimal value of $\mathcal{I}''$ is at most $O(1) \cdot \mathsf{OPT}(\mathcal{I}')$. Now note that all sizes and the vehicle capacity in $\mathcal{I}''$ are multiples of p; scaling down each of these quantities by p, we get an instance $\mathcal{J}$ *equivalent* to $\mathcal{I}''$ where the vehicle capacity is l (and every object size is at most l). This instance $\mathcal{J}$ satisfies all the four conditions claimed above.

Since $\mathcal{J}$ has at most l objects (each of size $\leq 2l$), the unweighted instance corresponding to $\mathcal{J}$ has at most $2l^2 \leq 2n^4$ objects. Thus, this unweighted instance can be solved using the ρ-approximation algorithm for such instances, assumed in the theorem. Then using the algorithm in Lemma 3.8, we obtain a solution to $\mathcal{J}$, of length at most $O(\rho) \cdot \mathsf{OPT}(\mathcal{J}) \leq O(\rho) \cdot \mathsf{OPT}(\mathcal{I}') \leq O(\rho) \cdot d(\tau^*)$. Since any feasible solution to $\mathcal{J}$ corresponds to one for $\mathcal{I}'$, we have a tour servicing $\mathcal{P}'$ of length at most $O(\rho) \cdot d(\tau^*)$.

Finally, combining the tours servicing $\mathcal{P}_{low}$, $\mathcal{P}_{high}$ and $\mathcal{P}'$, we obtain a feasible tour for $\mathcal{I}$ having length $O(\rho) \cdot d(\tau^*)$, which gives us the desired approximation algorithm. □

Theorem 3.9 also justifies the assumption $\log m = O(\log n)$ made at the end of Section 3. This is important because in general m may be super-polynomial in n.

4. *The Effect of Preemptions*

In this section, we study the effect of the number of preemptions in the Dial-a-Ride problem. We mentioned two versions of the Dial-a-Ride problem (Definition 1.3): in the preemptive version, an object may be preempted any number of times, and in the nonpreemptive version objects are not allowed to be preempted even once. Clearly the preemptive version is least restrictive and the non-preemptive version is most restrictive. One may consider other versions of the Dial-a-Ride problem, where there is a specified upper bound P on the number of times an object can be preempted. Note that the case $P = 0$ is the nonpreemptive version, and the case $P = n$ is the preemptive version. In Theorem 4.1, we show that for any instance of the Dial-a-Ride problem, there is a tour that preempts each object at most once (i.e., $P = 1$) and has length at most $O(\log^2 n)$ times an optimal preemptive tour (i.e., $P = n$). This implies that the real gap between preemptive and non-preemptive tours is between zero and one preemption per object. A tour that preempts each object at most once is called a *1-preemptive tour*.

THEOREM 4.1 (MANY PREEMPTIONS TO ONE PREEMPTION). *Given any instance of the Dial-a-Ride problem, there is a* 1*-preemptive tour of length at most* $O(\log^2 n)$ *times the length of an optimal preemptive tour. Such a tour can be found in randomized polynomial time.*

PROOF. We first show how general Dial-a-Ride instances can be reduced to instances where the metric is a *hierarchically well-separated* tree T having $O(\log n)$ levels. This uses the results on probabilistic tree embedding [Fakcharoenphol et al. 2003], and only increases the expected length of the optimal preemptive tour by a factor of $O(\log n)$. Then we show how to obtain a 1-preemptive tour on such tree-instances losing only an additional $O(\log n)$ factor in the tour length. The resulting

1-preemptive tour has the property that each object is moved nonpreemptively in two phases: first from its source to the least-common-ancestor (lca) of its source and destination, and then from the lca to its destination.

Applying the probabilistic tree embedding of Fakcharoenphol et al. [2003] to the given metric (V, d), we obtain a tree metric T such that the optimal preemptive tour in T has length $O(\log n)$ times that in (V, d), and any feasible solution in T corresponds to one in (V, d) of the same length. Additionally, the tree T has $O(\log \frac{d_{max}}{d_{min}})$ levels, where d_{max} and d_{min} denote the maximum and minimum (non-zero) distances in the original metric. Let OPT_{pmt} denote the optimal value of the original preemptive Dial-a-Ride instance. We first observe that using standard scaling arguments, it suffices to assume that for metric (V, d), $\frac{d_{max}}{d_{min}}$ is polynomial in n. Without loss of generality, any preemptive tour involves at most $2m \cdot n$ edge traversals: each object is picked or dropped at most $2n$ times (once at each vertex), and every visit to a vertex involves picking or dropping at least one object (otherwise the tour can be shortcut over this vertex-visit at no increase in length). By retaining only vertices within distance $\mathsf{OPT}_{pmt}/2$ from the root r, we preserve the optimal preemptive tour and ensure that $d_{max} \leq \mathsf{OPT}_{pmt}$. Now consider modifying the metric by setting all edges of length smaller than $\mathsf{OPT}_{pmt}/2mn^3$ to length 0; the new distances are shortest paths under the modified edge lengths. So any pairwise distance decreases by at most $\frac{\mathsf{OPT}_{pmt}}{2mn^2}$. Clearly, the length of the optimal preemptive tour only decreases under this modification. Since there are at most $2mn$ edge traversals in any preemptive tour, the increase in tour length in going from the new metric to the original metric is at most $2mn \cdot \frac{\mathsf{OPT}_{pmt}}{2mn^2} \leq \mathsf{OPT}_{pmt}/n$. Thus, at the loss of a constant factor, we may assume that $d_{max}/d_{min} \leq 2mn^3$. Furthermore, Theorem 3.8 also holds for preemptive Dial-a-Ride; so we may assume (at the loss of an additional constant factor) that the number of objects $m \leq O(n^4)$. So we have $d_{max}/d_{min} \leq O(n^7)$ and hence tree T has $O(\log n)$ levels.

The tree T resulting from Fakcharoenphol et al. [2003] has several Steiner vertices that are not present in the original metric; so the tour that we find on T may actually preempt objects at Steiner vertices, in which case it is not feasible in the original metric. However, as shown in Gupta [2001], these Steiner vertices can be simulated by vertices in the original metric (at the loss of a constant factor). Based on the preceding observations, we assume that the underlying metric is a tree T on the original vertex set having $l = O(\log n)$ levels, such that the expected length of the optimal preemptive tour is $\tilde{\mathsf{OPT}} = O(\log n) \cdot \mathsf{OPT}_{pmt}$.

We now partition the demand-pairs in T into l sets with D_i (for $i = 1, \ldots, l$) consisting of all pairs having their least common ancestor (lca) in level i. We service each D_i separately using a tour of length $O(\tilde{\mathsf{OPT}})$. Then, concatenating the tours for each level i, we obtain the theorem.

Servicing D_i. For each vertex v at level i in T, let L_v denote the pairs in D_i that have v as their lca. Consider an optimal *preemptive* tour that services the objects D_i. Since the subtrees under any two different level i vertices are disjoint and there is no pair in D_i across such subtrees, we may assume that this optimal tour is a concatenation of disjoint preemptive tours servicing each L_v separately. If $\tilde{\mathsf{OPT}}(v)$ denotes the length of an optimal preemptive tour servicing L_v with v as the starting vertex, then $\sum_v \mathsf{OPT}_{pmt}(v) \leq \mathsf{OPT}_{pmt}$.

Now consider an optimal preemptive tour τ_v servicing L_v. Since the $s_j - t_j$ path of each pair $j \in L_v$ crosses vertex v, at some point in tour τ_v the vehicle is at v with object j in it. Consider the tour σ_v obtained by modifying τ_v so that it drops each object j at v when the vehicle is at v with object j in it. Clearly $d(\sigma_v) = d(\tau_v) = \tilde{\mathsf{OPT}}(v)$. Note that σ_v is a feasible preemptive tour for the *single source Dial-a-Ride* problem with sink v and all sources in L_v. Thus the algorithm of Haimovich and Kan [1985] gives a *nonpreemptive* tour σ_v' that moves all objects in L_v from their sources to v, having length at most $2.5d(\sigma_v) = 2.5\tilde{\mathsf{OPT}}(v)$. Similarly, we can obtain a nonpreemptive tour σ_v'' that moves all objects in L_v from v to their destinations, having length at most $2.5\tilde{\mathsf{OPT}}(v)$. Now $\sigma_v' \cdot \sigma_v''$ is a 1-preemptive tour servicing L_v of length at most $5 \cdot \tilde{\mathsf{OPT}}(v)$.

We now run a depth-first-search on tree T to visit all vertices in level i, and use the algorithm described above for servicing objects L_v when v is visited in the DFS. This results in a tour servicing D_i, having length at most $2 \cdot d(T) + 5 \sum_v \tilde{\mathsf{OPT}}(v)$. Here $2 \cdot d(T)$ is the Steiner lower bound, and $\sum_v \tilde{\mathsf{OPT}}(v) \leq \tilde{\mathsf{OPT}}$. Thus, the tour servicing D_i has length at most $6 \cdot \tilde{\mathsf{OPT}}$.

Finally concatenating the tours for each level $i = 1, \ldots, l$, we obtain a 1-preemptive tour on tree-instance T of length $O(\log n) \cdot \tilde{\mathsf{OPT}}$, which translates to a 1-preemptive tour on the original metric having length $O(\log^2 n) \cdot \mathsf{OPT}_{pmt}$. □

Motivated by obtaining an improved approximation for Dial-a-Ride on the Euclidean plane, we next consider the worst case gap between an optimal nonpreemptive tour and the preemptive lower bounds. As mentioned earlier, Charikar and Raghavachari [1998] showed that there are instances of Dial-a-Ride where the ratio of the optimal nonpreemptive tour to the optimal preemptive tour is $\Omega(n^{1/3})$. However, the metric involved in this example was the uniform metric on n points, which can not be embedded in the Euclidean plane. The following theorem shows that even in this special case, there can be a polynomial gap between nonpreemptive and preemptive tours, and hence preemptive lower bounds do not suffice to obtain a poly-logarithmic approximation guarantee for nonpreemptive Dial-a-Ride.

THEOREM 4.2 (PREEMPTION GAP IN EUCLIDEAN PLANE). *There are instances of Dial-a-Ride on the Euclidean plane where the optimal nonpreemptive tour has length $\Omega(\frac{n^{1/8}}{\log^3 n})$ times the optimal preemptive tour.*

PROOF. Consider a square of side 1 in the Euclidean plane, in which a set of n demand-pairs $\{s_i, t_i\}_{i=1}^n$ are distributed uniformly at random (each demand point is generated independently and is uniformly from the square). The vehicle capacity is set to $k = \sqrt{n}$. Let $\mathcal{R}$ denote a random instance of Dial-a-Ride obtained as above. We show that in this case, the optimal nonpreemptive tour has length $\tilde{\Omega}(n^{1/8})$ with high probability. We first show the following claim.

CLAIM 4.3. *With high probability, the minimum length of a tree connecting k pairs in $\mathcal{R}$ is $\Omega(\frac{n^{1/8}}{\log n})$.*

PROOF. Take any set S of $k = \sqrt{n}$ demand-pairs, say $\{s_i, t_i\}_{i=1}^k$. Note that the number of such sets S is $\binom{n}{k}$. Set S has $2k$ vertices, each generated uniformly at random. It is well known that there are p^{p-2} different labeled trees on p vertices (see, e.g., van Lint and Wilson [1992, Ch. 2]). The term *labeled* emphasizes that we are not identifying isomorphic graphs, that is, two trees are counted as the same if and only if exactly the same pairs of vertices are adjacent. Thus, there are at most $(2k)^{2k-2}$ such trees on set S. Consider any labeled tree T on vertices S, and root it at the source vertex with minimum label (here s_1). We assume that T has been generated using the "Principle of Deferred Decisions", that is, vertices will be generated one by one according to some breadth-first ordering of T. We say that an edge is *short* if its length is at most $\frac{c}{\alpha k}$ (c and $\alpha \in (0, \frac{1}{2})$ will be fixed later).

If T has length at most c, it is clear that at most an α fraction of its edges are *not* short. So $Pr[\text{length of } T \le c] \le \sum_H Pr[\text{edges in } H \text{ are short}]$, where H in the summation ranges over all edge-subsets of T with $|H| \ge (1-\alpha)2k$. For a fixed H, we bound $Pr[\text{edges in } H \text{ are short}]$ as follows. For any edge $(v, \text{parent}(v))$ (note parent(v) is well-defined since T is rooted), assuming that parent(v) is fixed, the probability that this edge is short is $p = \pi(\frac{c}{\alpha k})^2$. So we can upper bound the probability that edges H are short by $p^{|H|} \le p^{(1-\alpha)2k}$. So we have $Pr[\text{length of } T \le c] \le 2^{2k} \cdot p^{(1-\alpha)2k}$, as the number of different edge sets H is at most 2^{2k}.

By a union bound over all such labeled trees T, the probability that the length of the minimum spanning tree on S is less than c is at most $(2k)^{2k} \cdot 2^{2k} \cdot p^{(1-\alpha)2k}$. Now taking a union bound over all k-sets S, the probability that the minimum length of a tree containing *some* k pairs is less than c is at most $\binom{n}{k}(2k)^{2k}2^{2k}p^{(1-\alpha)2k}$. Since $k = \sqrt{n}$, this term can be upper bounded as follows:

$$(ek)^k(4k)^{2k}\pi^{(1-\alpha)2k}\left(\frac{c}{\alpha k}\right)^{(1-\alpha)4k} \le 500^k k^{3k}\left(\frac{c}{\alpha k}\right)^{(1-\alpha)4k}$$
$$= \left[500 \cdot \left(\frac{c}{\alpha}\right)^{4-4\alpha}\left(\frac{1}{k}\right)^{1-4\alpha}\right]^k \le 2^{-k}$$

The last inequality above holds when $c \le \frac{\alpha}{1000} \cdot k^{1/4-3\alpha/(1-4\alpha)}$. Setting $\alpha = \frac{1}{\log k}$,

$$Pr\left[\exists\ \frac{k^{1/4}}{8000 \cdot \log k} \text{ length tree connecting some } k \text{ pairs in } \mathcal{R}\right] \le 2^{-k}$$

So, with probability at least $1 - 2^{-\sqrt{n}}$, the minimum length of a tree containing k pairs in $\mathcal{R}$ is at least $\Omega(\frac{n^{1/8}}{\log n})$. $\square$

From Theorem 3.1, we obtain that there is a near optimal nonpreemptive tour servicing all the objects in segments, where each segment (except possibly the last) involves servicing a set of $\frac{k}{2} \le t \le k$ objects. Although the lower bound of $k/2$ is not stated in Theorem 3.1, it is easy to extend the statement to include it. This implies that any solution of this structure has at least $\frac{n}{k} = k$ segments. Since each segment covers at least $k/2$ pairs, Claim 4.3 implies that each of these segments has length $\Omega(n^{1/8}/\log n)$. So the best solution of the structure given in Theorem 3.1 has length $\Omega(\frac{n^{1/8}}{\log n}k)$. But since there is a near-optimal solution of this structure, the optimal nonpreemptive tour on $\mathcal{R}$ has length $\Omega(\frac{n^{1/8}}{\log^2 n}k)$.

On the other hand, the flow lower bound for $\mathcal{R}$ is at most $\frac{n}{k} = k$, and the Steiner lower bound is at most $O(\sqrt{n}) = O(k)$ (an $O(\sqrt{n})$ length tree on the $2n$ points can be constructed using a $\sqrt{2n} \times \sqrt{2n}$ gridding). So the preemptive lower bounds are both $O(k)$; now using the algorithm of Charikar and Raghavachari [1998], we see that the optimal preemptive tour has length $O(k \log n)$. Combined with the lower bound for nonpreemptive tours, we obtain the Theorem.

ACKNOWLEDGMENT. We thank Alan Frieze for his help in proving Theorem 4.2.

REFERENCES

AGRAWAL, A., KLEIN, P., AND RAVI, R. 1991. When trees collide: an approximation algorithm for the generalized steiner problem on networks. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*. ACM, New York, 134–144.

BANSAL, N., BLUM, A., CHAWLA, S., AND MEYERSON, A. 2004. Approximation algorithms for deadline-TSP and vehicle routing with time windows. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*. ACM, New York, 166–174.

BLUM, A., CHAWLA, S., KARGER, D. R., LANE, T., MEYERSON, A., AND MINKOFF, M. 2003. Approximation algorithms for orienteering and discounted-reward TSP. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 46–55.

CHARIKAR, M., CHEKURI, C., YAT CHEUNG, T., DAI, Z., GOEL, A., GUHA, S., AND LI, M. 1999. Approximation algorithms for directed Steiner problems. *J. Algorithms 33*, 73–91.

CHARIKAR, M., AND RAGHAVACHARI, B. 1998. The finite capacity dial-a-ride problem. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 458–467.

CHAUDHURI, K., GODFREY, B., RAO, S., AND TALWAR, K. 2003. Paths, trees, and minimum latency tours. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 36–45.

CHRISTOFIDES, N. 1977. Worst-case analysis of a new heuristic for the travelling salesman problem. *GSIA, CMU-Report 388*.

FAKCHAROENPHOL, J., HARRELSON, C., AND RAO, S. 2003. The k-traveling repairman problem. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 655–664.

FAKCHAROENPHOL, J., RAO, S., AND TALWAR, K. 2003. A tight bound on approximating arbitrary metrics by tree metrics. In *STOC '03: Proceedings of the 35th Annual ACM Symposium on Theory of Computing*. ACM, New York, 448–455.

FEIGE, U., PELEG, D., AND KORTSARZ, G. 2001. The Dense k-Subgraph Problem. *Algorithmica 29,* 3, 410–421.

FREDERICKSON, G. N., HECHT, M. S., AND KIM, C. E. 1978. Approximation algorithms for some routing problems. *SIAM J. Computing 7,* 2, 178–193.

GARG, N. 2005. Saving an epsilon: A 2-approximation for the k-MST problem in graphs. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. ACM, New York, 396–402.

GØRTZ, I. L. 2005. Topics in algorithms: data structures on trees and approximation algorithms on graphs. Ph.D. dissertation, IT University of Copenhagen. Copenhagen, Denmark.

GØRTZ, I. L. 2006. Hardness of preemptive finite capacity dial-a-ride. In *Proceedings of the 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*. 200–211.

GOEMANS, M. X., AND WILLIAMSON, D. P. 1992. A general approximation technique for constrained forest problems. In *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 307–316.

GUPTA, A. 2001. Steiner points in tree metrics don't (really) help. In *SODA '01: Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 220–227.

HAIMOVICH, M., AND KAN, A. H. G. R. 1985. Bounds and heuristics for capacitated routing problems. *Math. Oper. Res. 10*, 527–542.

HAJIAGHAYI, M. T., AND JAIN, K. 2006. The prize-collecting generalized steiner tree problem via a new approach of primal-dual schema. In *SODA '06: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm*. ACM, New York, 631–640.

KRUMKE, S., RAMBAU, J., AND WEIDER, S. 2000. An approximation algorithm for the nonpreemptive capacitated dial-a-ride problem. *Preprint 00-53, Konrad-Zuse-Zentrum fr Informationstechnik Berlin*.

ROBINS, G., AND ZELIKOVSKY, A. 2005. Tighter bounds for graph steiner tree approximation. *SIAM J. Discr. Math. 19*, 122–134.

SAVELSBERGH, M., AND SOL, M. 1995. The general pickup and delivery problem. *Transport. Sci. 29*, 17–29.

SEGEV, D., AND SEGEV, G. 2006. Approximate k-Steiner forests via the lagrangian relaxation technique with internal preprocessing. In *Proceedings of the 14th Annual European Symposium on Algorithms*. 600–611.

STEELE, J. M. 1995. Variations on the monotone subsequence theme of Erdős and Szekeres. In *Discrete Probability and Algorithms*. IMA Vol. Math. Appl., vol. 72. Springer-Verlag, New York, 111–131.

VAN LINT, J. H., AND WILSON, R. M. 1992. *A Course in Combinatorics*. Cambridge University Press, Cambridge, MA.

RECEIVED JULY 2007; REVISED NOVEMBER 2008; ACCEPTED JANUARY 2009