Approximation Algorithms for Distance Constrained Vehicle Routing Problems

Viswanath Nagarajan[*] R. Ravi[†]

Abstract

We study the *distance constrained vehicle routing* problem (DVRP) [20, 21]: given a set of vertices in a metric space, a specified depot, and a distance bound D, find a minimum cardinality set of tours originating at the depot that covers all vertices, such that each tour has length at most D. This problem is NP-complete, even when the underlying metric is induced by a weighted star.

Our main result is a 2-approximation algorithm for DVRP on tree metrics; we also show that no approximation factor better than 1.5 is possible unless P=NP. For the problem on general metrics, we present a $\left(O(\log \frac{1}{\epsilon}),\, 1+\epsilon\right)$-bicriteria approximation algorithm: i.e., for any $\epsilon > 0$, it obtains a solution violating the length bound by a $1+\epsilon$ factor while using at most $O(\log \frac{1}{\epsilon})$ times the optimal number of vehicles.

Keywords: Vehicle Routing, Traveling Salesman Problem, Approximation Algorithms.

1 Introduction

1.1 Motivation

At the core of logistics operations facing modern firms is the problem of routing materials to and from manufacturing or consolidating depots at minimum cost [3, 23]. The most common constraints on such problems involve the capacity of the vehicles and deadlines on

[*]IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA.
[†]Tepper School of Business, Carnegie Mellon University, Pittsburgh PA 15213.

delivery/pickup of materials. Typical cost objectives are the total mileage of all the routes or more coarsely, the number of vehicles deployed to satisfy the demands. In this paper, we study the problem with distance constraints on each route where the objective is to minimize the number of vehicles, called the *Distance Constrained Vehicle Routing Problem (DVRP)* in the literature [3, 20, 21].

A bound on the distance traveled by any vehicle arises commonly, e.g., in scheduling daily routes for courier carriers or milkruns from manufacturing facilities. The distance bound translates to a quality of service guarantee for all customers to be served on the day they are scheduled. Minimizing the number of vehicles over a period of typical demands also allows for better fleet and driver planning and management. However, these problems generalize the classical TSP and are NP-complete.

In this paper, we obtain approximation algorithms for distance constrained vehicle routing problems. We use the well studied notion of approximation guarantees [14, 24] to measure the performance of heuristics. An approximation algorithm for a minimization problem is said to achieve an approximation ratio α (which may be a function of the input instance), if on every instance, the cost of the solution obtained by the algorithm is at most α times the cost of an optimal solution. Such an algorithm is also referred to as an α-approximation algorithm.

1.2 Problem formulation

We model demand locations as vertices in a finite metric space (V, d), with $|V| = n$. The distance function $d : V \times V \to \mathbb{N}$ is symmetric and satisfies the triangle inequality. Throughout this paper we assume that all distances are integral: this can be ensured by a suitable scaling. The input to the *distance constrained vehicle routing problem* (DVRP) is specified by a metric space (V, d), a depot $r \in V$, and a distance constraint D. The objective is to find a *minimum cardinality* set of tours originating from r (corresponding to routes for vehicles), that covers all the vertices in V. Each tour is required to have length at most D (the distance constraint). Tours originating from r are referred to as *r-tours*. The maximum distance of any vertex from the depot is denoted by Δ. We assume that $\Delta \leq \frac{D}{2}$, as otherwise there is no feasible solution.

The *unrooted DVRP* [1] is defined as follows: given a metric space (V, d) and a distance constraint D, find a minimum cardinality set of paths (each of length at most D) that covers all the vertices. Note that in this version, the vehicle routes are paths that are allowed to start and end at any two vertices. The unrooted version can be reduced to DVRP by adding a root vertex that is located at some large distance $L \gg \mathit{diameter}(V)$ from all vertices in V, and setting the distance constraint to $D + 2L$.

1.3 Our Results

Our main result is for the DVRP on metrics induced by an underlying edge-weighted tree (Section 2). In this case, we obtain a 2-approximation algorithm. This algorithm can be implemented in a single depth first search of the tree and runs in linear time. We note that even DVRP on star metrics is equivalent to the *bin packing* problem [12], which is NP-complete. Moreover, we show that DVRP on trees is hard to approximate to better than a factor of 1.5 (unless P=NP). We note that DVRP on trees possesses the "scaling property", i.e. inputs with optimal value T can be scaled to inputs of optimal value $k \cdot T$ (for any k). In contrast, the bin packing problem does not have the scaling property, and in fact admits an asymptotic polynomial time approximation scheme [9, 16].

For DVRP on general metrics, we obtain an $(O(\log \frac{1}{\epsilon}), 1 + \epsilon)$ *bicriteria* approximation. That is, for any $\epsilon > 0$, if the vehicles are allowed to exceed the distance constraint by a small multiplicative factor ϵ, then we obtain a solution using at most $O(\log \frac{1}{\epsilon})$ times the optimal number of vehicles (that do not violate the distance constraint). As shown in Jothi and Raghavachari [15] the tour-partitioning algorithm of Li et al. [21] gives an $(O(\frac{1}{\epsilon}), 1 + \epsilon)$ bicriteria approximation for DVRP. We improve upon the approximation ratio on the number of vehicles significantly.

1.4 Related Work

Vehicle routing problems (VRPs) are surveyed in [3, 23]. Practical applications of DVRP can be found in Assad [3] and Laporte et al. [20]. Exact approaches for the objective of minimizing total distance were studied in Laporte et al. [20]. They gave two algorithms using an integer programming formulation: one based on Gomory cuts and the other using branch-and-bound.

Li et al. [21] studied DVRP under the objective functions of total distance and number of vehicles. They showed that the optimal solutions under both objectives are closely related, and any approximation guarantee for one objective implies a guarantee with an additional loss of factor 2, for the other objective. They also studied a tour-partitioning heuristic for this problem, which was shown to achieve a worst case performance guarantee of D.

A closely related problem is *orienteering*: given a metric (V, d), depot $r \in V$ and length bound D, find an r-tour of length at most D that contains the maximum number of vertices. Improving on work by Blum et al. [7] and Bansal et al. [5], Chekuri et al. [8] presented a $2 + \epsilon$ approximation algorithm (for any constant $\epsilon > 0$) for the orienteering problem. Using this as a greedy subroutine within a set-covering framework, it is straightforward to design an $O(\log n)$ approximation for DVRP.

The distance constrained VRP was also studied by Bazgan et al. [6], where the authors gave a constant-factor *differential* approximation algorithm. However, bounds in the differential measure do not imply any bounds in the standard (multiplicative) approximation measure, which we consider in this paper.

Related to the tree metric version we study, Labbe et al. [19] and Karuno et al. [17] discuss some practical situations where tree shaped networks are encountered in VRPs. In the case of capacitated VRP (only capacity constraints), Labbe et al. [19] gave a 2-approximation algorithm on trees when demands are unsplittable. When demands are splittable, Hagamochi and Katoh [13], and Asano et al. [2] gave improved approximation algorithms; the currently best known guarantee is ≈ 1.35. Many other vehicle routing problems on trees have been studied in [11, 4, 17].

2 DVRP on Tree Metrics

In this section, we consider the special case of DVRP when the metric space is induced by a weighted tree $T = (V, d)$. Even in the special case of a star, the problem remains NP-complete (by a reduction from bin-packing). Here we present a 2-approximation for DVRP on trees, and also show that the problem is 1.5-hard to approximate unless P=NP.

We first present the algorithm for DVRP on trees. The main ingredient in this is proving a lower bound on the optimal number of vehicles, which is based on forming clusters of vertices that can not be covered by a single r-tour (Lemma 2). For ease of description, we assume (without loss of generality) that the tree T is binary, and rooted at the depot r. This can be ensured by splitting high degree vertices, and adding zero-length edges. Algorithm *minTVR* for DVRP on trees is as follows.

1. Initialize $T' = T$.
2. While $(T' \neq \{r\})$ do
   (a) Pick a deepest vertex $v \in T'$ s.t. the subtree T'_v below v *can not* be covered by just one r-tour, of length at most D. If no such v exists, add an r-tour covering T', and END.
   (b) Let w_1 and w_2 be the two children of v. For $i = 1, 2$, set W_i to be the minimum length r-tour traversing the subtree below w_i.
   (c) Add r-tours W_1 and W_2 to the solution.
   (d) $T' = T' \setminus T'_v$.

Note that the minimum length r-tour covering all the vertices of a subtree is just an Euler tour of the subtree (including the path from r), traversing each edge twice. This property can also be derived as a special case of the "master tour" property of Kalmanson matrices [10]. Thus the condition in step 2a can be checked efficiently.

Theorem 1 *Algorithm minTVR obtains a 2-approximation to DVRP on trees.*

Proof: It is not hard to see that algorithm *minTVR* can be implemented in a single depth-first search of the tree; so the time complexity is linear in the input size, $O(n \log_2 D)$. From the choice of vertex v in step 2a, each r-tour added in step 2c (corresponding to the children of v), has length at most D. So algorithm *minTVR* indeed produces a feasible solution.

A *heavy cluster* is defined to be a set of vertices $C \subseteq V$ such that the subgraph $T[C]$ induced by C on tree T is connected, and the vertices in C can *not* all be covered by a single

r-tour of length at most D. Note that all the subtrees T'_v seen in step 2a of algorithm $minTVR$ are heavy clusters in tree T. Suppose, in its entire execution, the algorithm finds k heavy clusters $C_1, \ldots C_k$ (these vertex sets will be disjoint). Then algorithm $minTVR$ produces a solution using at most $2k+1$ r-tours. The key lemma is the following, which shows that the optimal solution requires at least $k+1$ vehicles, and thus proves Theorem 1.

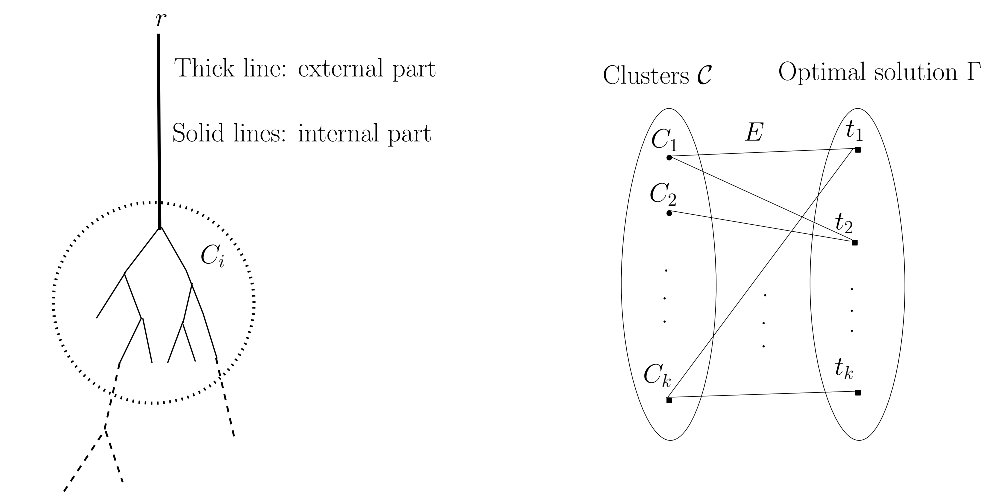
Lemma 2 *If there are k disjoint heavy clusters $C_1, \ldots C_k \subseteq V$ in the tree T, the minimum number of r-tours (of length at most D) required to cover $\bigcup_{i=1}^{k} C_i$ is more than k.*

Proof: The proof of this lemma is by induction on k. For $k = 1$, the lemma is trivially true. Suppose $k > 1$, and assume that the lemma holds for all values up to $k-1$. Suppose, for contradiction, that the minimum number of r-tours required to cover all these clusters, $OPT \leq k$. Note that OPT can not be smaller than k: taking any $k-1$ of these k clusters, we would get a contradiction to the induction hypothesis with $k-1$ clusters. So we may assume $OPT = k$. In the rest of the proof, fix an optimal solution consisting of r-tours $t_1, \ldots, t_k$.

From the definition of a heavy cluster, each C_i forms a connected subtree in T. It will be convenient to think of the lengths associated with C_i in the following parts (see Figure 1a): the path from r to the highest vertex in C_i (*external part*); and the induced subgraph $T[C_i]$ (*internal part*). The length of the external part of a cluster C_i is denoted $d(r, C_i)$. We now define a bipartite graph $H = (\Gamma, \mathcal{C}, E)$ where $\Gamma = \{t_1, \ldots, t_k\}$ is the set of r-tours in the optimal solution, and $\mathcal{C} = \{C_1, \ldots, C_k\}$ is the set of the k heavy clusters (see Figure 1b). There is an edge $(t_j, C_i) \in E$ iff r-tour t_j visits some vertex of cluster C_i.

We claim that H must have a perfect matching between $\mathcal{C}$ and Γ. Suppose not - then by Hall's Theorem, we get a set $S \subseteq \mathcal{C}$ such that S has fewer than $|S|$ neighbors in Γ. Note that $S \neq \mathcal{C}$, as $\mathcal{C}$ has $OPT = |\mathcal{C}|$ neighbors. This implies that the clusters in S are visited completely by fewer than $|S|$ r-tours, which contradicts the induction hypothesis with the set of heavy clusters S (as $|S| < k$). Thus H has a perfect matching $\pi : \mathcal{C} \to \Gamma$.

Let $l_1, l_2, \ldots, l_k$ denote the lengths of the r-tours in Γ; clearly each $l_i \leq D$. We assign a *capacity* to each edge $e \in T$: $cap_e = 2\sum_{j=1}^{k} I_e(t_j)$, where $I_e(t_j) = 1$ iff edge e is traversed in r-tour t_j, and 0 otherwise. Note that if an edge is traversed in an r-tour, it is traversed at



(a) Lengths associated with heavy cluster

(b) The graph H

Figure 1: DVRP on trees

least 2 times; so each edge in T has capacity at least 2 (as each vertex is visited). Now, the total weighted capacity over all edges is exactly $\sum_{e\in T} d_e \cdot (2\sum_{j=1}^{k} I_e(t_j)) \leq \sum_{i=1}^{k} l_i \leq kD$.

We will now *charge* each edge an amount at most its capacity, and show that the total weighted charge over all edges is larger than kD, which would be a contradiction. Corresponding to every cluster C_i, charge each edge in its external part (the path from r to C_i) two units against the capacity on that edge attributed to r-tour $\pi(C_i)$; note that tour $\pi(C_i)$ visits C_i and hence traverses all the edges from r to C_i. Since π is a perfect matching, no edge has a charge more than its capacity. The total weighted charge after this step is exactly $2\sum_{i=1}^{k} d(r, C_i)$. Now we will further assign a charge of 2 units to each edge in the internal part of every cluster $C_1, \ldots, C_k$.

Consider any edge e on the internal part of some cluster C_i. Let m denote the number of clusters that appear *below* e in tree T (this does not include C_i). If $m = 0$, this edge has never been charged so far, and thus has at least 2 units of residual capacity. If $0 < m \leq k-1$, then applying induction on the set of m clusters below e, there are at least $m+1$ r-tours that traverse e. So e has a capacity of at least $2m+2$. But we have charged e exactly $2m$ units so far, 2 units corresponding to each cluster below it. So again we have at least 2 units of residual capacity, and we can charge this edge an extra 2 units. The total weighted charge over all edges can now be written as follows:

$$\sum_{i=1}^{k} \left[2d(r, C_i) + 2 \cdot d(\text{internal part of } C_i)\right]$$

The i-th term above corresponds to an r-tour covering C_i. Since each C_i is a heavy cluster, this is more than D. So the total weighted charge is more than $kD \geq$ the total capacity, which is a contradiction. Thus $OPT > k$, and the lemma is proved.■

We now prove the hardness of approximation for DVRP on trees. The scaling property mentioned in the introduction also follows from this proof.

Theorem 3 *Unless P=NP, there is no* 1.5*-approximation algorithm for DVRP on trees.*

Proof: We reduce from the *subset sum* problem [12]. Given a collection $\{a_1, \ldots, a_m\}$ of m non-negative integers with $\sum_{i=1}^{m} a_i$ even and $B := \frac{1}{2}\sum_{i=1}^{m} a_i$, the goal is to determine whether there exists a subset $S \subseteq [m]$ such that $\sum_{i \in S} a_i = B$. This problem is known to be NP-complete; note that the input size is at least $m + \log_2 B$. Let $\mathcal{I}$ denote an arbitrary instance of the subset sum problem, as above.

Fix a parameter k that is polynomial in the size of $\mathcal{I}$. We construct an instance of DVRP on trees as follows. The root r has k children $\{c_1, \ldots, c_k\}$ and each of the edges $\{(r, c_i)\}_{i=1}^{k}$ has length $2B$. Each vertex c_i (for $i \in [k]$), has m children $\{l_{i,j}\}_{j=1}^{m}$ where for each $j \in [m]$, edge (c_i, l_{ij}) has length a_j. The length bound $D := 6B$. Note that the size of the DVRP instance is polynomial in the size of $\mathcal{I}$, and the construction runs in polynomial time.

Suppose $\mathcal{I}$ is a yes-instance, i.e. there is some subset $S \subseteq [m]$ with $\sum_{j \in S} a_j = B$. Consider the following solution for the DVRP instance: for each $i \in [k]$, there are two r-tours, visiting vertices $\{l_{ij} \mid j \in S\}$ and $\{l_{ij} \mid j \in [m] \setminus S\}$ respectively. It is clear that each vertex is covered in some tour. Note that the length of the tour covering $\{l_{ij} \mid j \in S\}$ (for any $i \in [k]$) equals $2 \cdot \left(2B + \sum_{j \in S} a_j\right) = 6B$. Similarly, the length of the tour covering $\{l_{ij} \mid j \in S\}$ (for each $i \in [k]$) equals $6B$. Thus the above solution satisfies the distance constraint, and the optimal value of the DVRP instance is at most $2k$.

Suppose $\mathcal{I}$ is a no-instance, then we claim that the optimal value of the DVRP instance is at least $3k$. Observe that any tour of length at most $6B$ can visit at most one of the subtrees rooted at $\{c_i\}_{i=1}^k$. Thus any feasible solution to the DVRP instance is a disjoint union $\bigcup_{i=1}^k \mathcal{C}_i$, where for each $i \in [k]$, $\mathcal{C}_i$ is a collection of r-tours (each of length $\leq 6B$) that covers $\{l_{i,j}\}_{j=1}^m$. We claim that $|\mathcal{C}_i| \geq 3$ for any $i \in [k]$: for otherwise, the two tours covering $\{l_{i,j}\}_{j=1}^m$ would yield a subset $S \subseteq [m]$ with $\sum_{j \in S} a_j = B$, which is impossible since $\mathcal{I}$ is a no-instance.

Since the subset sum problem is NP-complete, it follows that it is NP-hard to approximate DVRP on trees to better than a factor of 1.5. ■

Since this reduction is from the subset-sum problem, it does not rule out better pseudo-polynomial time approximation algorithms for DVRP on trees, i.e. where the running time is polynomial in n and D (rather than polynomial in n and $\log_2 D$).

3 Bicriteria Approximation for DVRP on General Metrics

In this section, we study DVRP on general metrics, and present a bicriteria approximation algorithm. Our algorithm uses as a subroutine, the *unrooted* DVRP (Section 1.2), and the 3-approximation algorithm for this problem from Arkin et al. [1]. The basic idea of the algorithm for DVRP is the following: if an r-tour visits some vertices a "large" distance from the root, it resembles an unrooted path (with smaller length) when restricted to just those vertices. So we partition the vertices of the graph into parts, according to their distance from the root, and solve the unrooted DVRP (with appropriate distance bounds) in each part. Algorithm *minVR* for DVRP on general metrics is described below. The algorithm also takes as input a parameter $\epsilon \in (0,1)$ that denotes the allowed violation of the distance constraint.

1. Define vertex sets $V_0, V_1, \ldots, V_t$ as follows (where $t = \lceil \log_2(1/\epsilon) \rceil$):

$$V_j = \begin{cases} \{v : (1-\epsilon) \cdot \frac{D}{2} < d(r,v) \leq \frac{D}{2}\} & \text{if } j = 0 \\ \{v : (1-2^j\,\epsilon) \cdot \frac{D}{2} < d(r,v) \leq (1-2^{j-1}\,\epsilon) \cdot \frac{D}{2}\} & \text{if } 1 \leq j \leq t-1 \\ \{v : 0 < d(r,v) \leq \left(1-2^{t-1}\,\epsilon\right) \cdot \frac{D}{2}\} & \text{if } j = t \end{cases}$$

2. For $j = 0, \ldots, t$ do:

   (a) Run the algorithm for unrooted DVRP [1], for the vertex set V_j, with distance constraint $2^{j-1}\,\epsilon\cdot D$. Let Π_j denote the set of paths obtained.

(b) For every path in Π_j, append both its end points with edges from the depot r, to obtain the r-tours $\{r\cdot\pi\cdot r \mid \ \pi\in\Pi_j\}$.

3. Return all r-tours obtained above.

Theorem 4 *For every $0<\epsilon<1$, algorithm minVR is an $(O(\log\frac{1}{\epsilon}), 1+\epsilon)$ bicriteria approximation algorithm for DVRP.*

Proof: We first show that each r-tour produced by algorithm minVR has length at most $(1+\epsilon)\,D$. For $j=0$, each r-tour added in step 2 consists of two direct edges from r to V_0 and a path of length at most $\frac{\epsilon}{2}\,D$; so such a tour has length at most $2\cdot\frac{D}{2}+\epsilon\,\frac{D}{2}\le(1+\epsilon)\,D$. Now consider the r-tours corresponding to vertex sets V_j $(1\le j\le t)$. Each path $\pi\in\Pi_j$ has length at most $2^{j-1}\,\epsilon\,D$, and every vertex of V_j (and hence the end points of π) is at distance at most $(1-2^{j-1}\,\epsilon)\cdot\frac{D}{2}$ from r. So each r-tour $(r\cdot\pi\cdot r)$ added in this step has length at most $2^{j-1}\,\epsilon\,D+(1-2^{j-1}\,\epsilon)\cdot D=D$.

We now prove the performance guarantee of this algorithm. Below OPT denotes the optimal number of r-tours (each of length at most D) for the DVRP instance.

Claim 5 *For each $j=0,\ldots,t$, the optimal value of the unrooted DVRP instance defined in step 2a is at most $2\cdot OPT$.*

Proof: Fix any $j\in\{0,\ldots,t\}$. Let Γ denote an optimal solution to the original DVRP instance. Consider any r-tour $\sigma\in\Gamma$, and let σ_j denote the *path* induced by σ on the vertices in V_j. The length of σ_j is at most $D-2\cdot\frac{D}{2}\,(1-2^j\,\epsilon)=2^j\epsilon\,D$. This is because every vertex in V_j (hence the end points of σ_j) is located at distance at least $(1-2^j\,\epsilon)\,\frac{D}{2}$ from r. So the path σ_j can be split into two (unrooted) paths, each of length at most $2^{j-1}\epsilon\,D$. Splitting each tour in Γ in this manner gives us a set Θ of at most $2|\Gamma|=2\cdot OPT$ unrooted paths over V_j, that together cover all vertices of V_j. So Θ is a feasible solution to the unrooted DVRP instance on V_j with length bound $2^{j-1}\epsilon\,D$. Thus we have the claim. ■

Using Claim 5 and the 3-approximation to unrooted DVRP [1], we get $|\Pi_j|\le 6\cdot OPT$, for all $j=0,\ldots,t$. Thus the total number of r-tours in the solution is at most $6(t+1)\cdot OPT$, giving the theorem. ■

We note that the above bicriteria approximation was obtained independently in the preliminary version of this paper [22] and in Khuller et al. [18]. It remains an interesting open question to obtain a constant factor approximation for DVRP on general metrics.

Acknowledgements

We thank Shuchi Chawla for helpful discussions. We also thank an anonymous referee for raising a question regarding the scaling property for DVRP on trees.

This research was supported in part by NSF grants CCF-0430751, CCF-0728841 and ITR grant CCR-0122581 (The ALADDIN project).

References

[1] E. M. Arkin, R. Hassin, and A. Levin. Approximations for Minimum and Min-max Vehicle Routing Problems. *Journal of Algorithms*, 2005.

[2] T. Asano, N. Katoh, and K. Kawashima. A New Approximation Algorithm for the Capacitated Vehicle Routing Problem on a Tree. *Journal of Combinatorial Optimization*, 5(2):213 – 231, 2001.

[3] A.A. Assad. Modeling and Implementation Issues in Vehicle Routing. *Vehicle Routing: Methods and Studies*, pages 7–45, 1988.

[4] I. Averbakh and O. Berman. Sales-delivery man problems on treelike networks. *Networks*, 25:45–58, 1995.

[5] N. Bansal, A. Blum, S. Chawla, and A. Meyerson. Approximation Algorithms for Deadline-TSP and Vehicle Routing with Time Windows. In *Symposium on Theory of Computing*, pages 166–174, 2004.

[6] C. Bazgan, R. Hassin, and J. Monnot. Approximation Algorithms for Some Vehicle Routing Problems. *Discrete Applied Mathematics*, 146:27–42, 2005.

[7] A. Blum, S. Chawla, D.R. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation Algorithms for Orienteering and Discounted-Reward TSP. *SIAM Journal on Computing*, 37(2):653–670, 2007.

[8] C. Chekuri, N. Korula, and M. Pal. Improved Algorithms for Orienteering and Related Problems. In *Symposium on Discrete Algorithms*, pages 661–670, 2008.

[9] W. Fernandez de la Vega and G. Lueker. Bin Packing Can Be Solved within $1 + \epsilon$ in Linear Time. *Combinatorica*, 1(4):349–355, 1981.

[10] V.G. Deineko, R. Rudolf, and G.J. Woeginger. Sometimes Travelling is Easy: The Master Tour Problem. *SIAM J. Discrete Math.*, 11(1):81–93, 1998.

[11] G.N. Frederickson and D.J. Guan. Preemptive ensemble motion planning on a tree. *SIAM J. Comput.*, 21(6):1130–1152, 1992.

[12] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. 1979.

[13] S. Hamaguchi and N. Katoh. A Capacitated Vehicle Routing Problem on a Tree. In *International Symposium on Algorithms and Computation*, pages 397–406, 1998.

[14] D.S. Hochbaum, editor. *Approximation algorithms for NP-hard problems*. 1997.

[15] R. Jothi and B. Raghavachari. Approximating the k-Traveling Repairman Problem with Repairtimes. *Journal of Discrete Algorithms*, 5(2):293–303, 2007.

[16] N. Karmarkar and R.M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Foundations of Computer Science*, pages 312–320, 1982.

[17] Y. Karuno, H. Nagamochi, and T. Ibaraki. Vehicle scheduling on a tree with release and handling times . *Annals of Operations Research*, 69(1):193–207, 1997.

[18] S. Khuller, A. Malekian, and J. Mestre. To Fill or not to Fill: The Gas Station Problem. In *European Symposium on Algorithms*, pages 534–545, 2007.

[19] M. Labbe, G. Laporte, and H. Mercure. Capacitated Vehicle Routing on Trees. *Operations Research*, 39(4):616–622, 1991.

[20] G. Laporte, M. Desrochers, and Y. Nobert. Two Exact Algorithms for the Distance Constrained Vehicle Routing Problem. *Networks*, 14:47–61, 1984.

[21] C. Li, D. Simchi-Levi, and M. Desrochers. On the distance constrained vehicle routing problem. *Operations Research*, 40:790–799, 1992.

[22] V. Nagarajan and R. Ravi. Minimum Vehicle Routing with a Common Deadline. In *Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 212–223, 2006.

[23] P. Toth and D. Vigo. *The vehicle routing problem.* 2001.

[24] V.V. Vazirani. *Approximation Algorithms* . 2001.