A Capacitated Vehicle Routing Problem on a Tree

02502034 Shin-ya HAMAGUCHI Osaka Prefectural Government
01104684 Naoki KATOH Kyoto University

1 Introduction

In this paper we consider a capacitated vehicle routing problem on a tree-shaped network with a single depot. Let $T = (V, E)$ be a tree, where V is a set of n vertices and E is a set of edges, and $r \in V$ be a designated vertex called *depot*. Nonnegative weight $w(e)$ is associated with each edge $e \in E$, which represents the length of e. Customers are located at vertices of the tree, and a customer at $v \in V$ has a positive demand $D(v)$. Thus, when there is no customer at v, $D(v) = 0$ is assumed. Demands of customers are served by a set of identical vehicles with limited capacity. It is assumed that the capacity of every vehicle is equal to one, and that the demand of a customer is splittable, i.e., it can be served by more than one vehicle. Each vehicle starts at the depot, visits a subset of customers to (partially) serve their demands and returns to the depot without violating the capacity constraint. The problem asks to find a set of tours of vehicles with minimum total lengths to satisfy all the demands of customers. We call this problem TREE-CVRP.

Vehicle routing problems have long been studied by many researchers (see [2, 4] for a survey), and are found in various applications. Recently, AGV and material handling robots are often used in manufacturing systems, but also in offices and hospitals, in order to reduce the material handling efforts. The tree-shaped network can be typically found in buildings with simple structures of corridors and in simple production lines of factories.

Vehicle scheduling problems on tree-shaped networks have recently been studied by several authors [1, 3, 7]. However, TREE-CVRP has not been studied in the literature. We first show that it is strongly NP-complete.

We then turn our attention to approximation algorithms. Since TREE-CVRP is a special class of general CVRP, approximation algorithms for CVRP on general undirected networks can be used. In particular, the iterated tour partitioning (ITP) heuristic proposed by [5] provides $1 + (1 - \frac{1}{k})\alpha$ approximation for such general case when α-approximation algorithm for TSP is available, where the capacity of every vehicle is assumed to be equal to k and the demand of every customer is a positive integer. For tree-shaped networks, TSP can be optimally solved in a straightforward manner, and thus we have a $(2 - \frac{1}{k})$-approximation algorithm.

In this paper, we shall present an improved 1.5-approximation algorithm for TREE-CVRP by exploiting the tree structure of the network. We have implemented our algorithm and carried out computational experiments to see how effective our algorithm is. I was shown that the solutions obtained by our algorithm are very close to optimal [6].

2 Preliminaries

For vertices $u, v \in V$, let $path(u, v)$ be the unique path between u and v. The length of $path(u, v)$ is denoted by $w(path(u, v))$. We often view T as a directed tree rooted at r. For a vertex $v \in V - \{r\}$, let *parent*(v) denote the parent of v, and $C(v)$ the set of children of v. When we write an edge $e = (u, v)$, u is assumed to a parent of v. For any $v \in V$, let T_v denote the subtree rooted at v, and $w(T_v)$ and $D(T_v)$ denote the sum of weights of edges in T_v, and the sum of demands of customers in T_v, respectively. Since customers are located on vertices, customers are often identified with vertices.

Suppose that we are given a set $S \subset V - \{r\}$ with $\sum_{v \in S} D(v) \leq 1$. Then one vehicle is enough to serve all the demands of customers in S, and an optimal tour for S can be trivially obtained by first computing a minimal subtree T' that spans $S \cup \{r\}$ and by performing a depth-first search with r as the starting vertex.

A solution of TREE-CVRP consists of a set of tours. We can represent the tour of the j-th vehicle by

$$\{D_j(v) \mid v \in S_j\}, \tag{1}$$

where S_j is the set of customers for which some positive demands are served in the j-th tour, and $D_j(v)(> 0)$ for $v \in S_j$ is the amount of demand that the j-th vehicle serves at v. The total tour lengths of an optimal solution for TREE-CVRP is often referred to as the *optimal cost*.

For an edge $e = (u, v)$, let

$$LB(e) = 2w(e) \cdot \lceil D(T_v) \rceil. \tag{2}$$

$LB(e)$ represents a lower bound of the cost required for traversing edge e in an optimal solution because, due to the unit capacity of a vehicle, the number of vehicles required for any solution to serve the demands in T_v is at least $\lceil D(T_v) \rceil$ and each such vehicle passes e at least twice (one is in a forward direction and the other is in a backward direction). Thus, we have the following lemma.

Lemma 1 *$\sum_{e \in E} LB(e)$ gives a lower bound of the optimal cost of TREE-CVRP.*

3 Algorithm

A vertex $v \in V$ is called *D-feasible* if $D(T_v) \geq 1$ and is called *D-minimal* if it is *D-feasible* but any of its children is not. The proposed algorithm first finds a *D-minimal* vertex, and determines a routing of one or two vehicles that partially serve demands of vertices in T_v by applying Strategy 1 or 2 depending on the cases as will be described below. We then appropriately update the remaining demands of vertices visited by the routing currently determined. In addition, if the remaining demand of subtree T_v becomes zero, T_v as well as the edge $(parent(v), v)$ is deleted. In this section, we abuse the notations $D(v)$ and $D(T_v)$ to denote the remaining demands of vertex v or subtree T_v, respectively unless confusion occurs. We repeat this process until all the demands of T are served. Notice that, if there is no D-feasible vertex (i.e., $D(T_r) < 1$) any more, we can optimally serve the remaining demands by visiting the relevant vertices in a depth-first manner.

The algorithm consists of a number of applications of Strategy 1 or 2. One application of Strategy 1 or 2 is called a *round*.

When a D-minimal vertex v is found, we apply the following two strategies and choose the one with cheaper cost. Let $C(v) = \{v_1, v_2, \ldots, v_p\}$. from T. $1 \leq \sum_{v_i \in S} D(T_{v_i}) < 2$. Since $D(T_{v_i}) < 1$ for all $v_i \in C(v)$ from D-minimality of v, such S always exists and can be easily computed as $\cup_{i=1}^{k} T_{v_i}$ satisfying

$$\sum_{i=1}^{k-1} D(T_{v_i}) < 1 \text{ and } \sum_{i=1}^{k} D(T_{v_i}) \geq 1.$$

If $\sum_{i=1}^{k} D(T_{v_i}) = 1$, we simply allocate one vehicle to serve all the demands in $\cup_{i=1}^{k} T_{v_i}$. Thus, we assume otherwise. We can assume $k = 2$ without loss of generality because otherwise we can equivalently modify T_v.

With this assumption, the algorithm considers subtrees T_{v_1} and T_{v_2} satisfying

$$D(T_{v_1}) < 1, D(T_{v_2}) < 1 \text{ and } D(T_{v_1}) + D(T_{v_2}) > 1. \tag{3}$$

The first strategy (Strategy 1) prepares two vehicles to serve all the demands in $T_{v_1} \cup T_{v_2}$, while the second strategy (Strategy 2) prepare one vehicle to partially serve the demands in $T_{v_1} \cup T_{v_2}$ by using its full capacity (thus, the demand of some vertex may possibly be split).

Strategy 1: We prepare one vehicle for T_{v_1} and another vehicle for T_{v_2} to separately serve demands of T_{v_1} and T_{v_2}. The cost to serve these demands is

$$4w(path(r, v)) + 2w(T_{v_1}) + 2w(T_{v_2}) \tag{4}$$

because two vehicles run on the path $path(r, v)$ but each of T_{v_1} and T_{v_2} is traversed by only one vehicle.

Strategy 2: We assume $w(T_{v_1}) \geq w(T_{v_2})$ without loss of generality. We split the demand $D(u)$ of every $u \in T_{v_2}$ into $D'(u)$ and $D''(u)$ so that

$$\sum_{u \in T_{v_1}} D(u) + \sum_{u \in T_{v_2}} D'(u) = 1. \tag{5}$$

We allocate one vehicle to serve the set of demands

$$\{D(u) \mid u \in T_{v_1}\} \text{ and } \{D'(u) \mid u \in T_{v_2}\}.$$

The computation of such $D'(u)$ satisfying (5) is straightforward (the details are omitted). Notice that the demand of at most one vertex is split by this procedure. The cost required for Strategy 2 is at most

$$2w(path(r, v)) + 2w(T_{v_1}) + 2w(T_{v_2}). \tag{6}$$

Demands of some vertices in T_{v_2} remain unserved, and thus T_{v_2} (or its subgraph) will be visited later by other vehicles. Thus, in total the cost to visit T_{v_2} (or its subgraph) will be counted twice or more as (6). For the ease of the analysis of approximation ratio of the proposed algorithm, we amortize the the cost to visit T_{v_2} in the current round so that it is charged to T_{v_1}. Since $w(T_{v_1}) \geq w(T_{v_2})$, the cost of (6) is bounded from above by

$$2w(path(r, v)) + 4w(T_{v_1}). \tag{7}$$

We consider (7) as the cost for Strategy 2.

It should be remarked that our algorithm chooses Strategy 1 or 2 not by directly comparing the costs of (4) and (7), but by the following rule.

Selection rule of Strategy 1 or 2:
We apply Strategy 1 if

$$\frac{4w(path(r, v)) + 2w(T_{v_1}) + 2w(T_{v_2})}{2w(path(r, v)) + 2w(T_{v_1}) + 2w(T_{v_2})} \leq \frac{2w(path(r, v)) + 4w(T_{v_1})}{2w(path(r, v)) + 2w(T_{v_1})}, \tag{8}$$

and apply Strategy 2 otherwise.

The rationale behind this selection rule is as follows: Since the amounts of demands as well as the sets of customers served by Strategies 1 and 2 are different in general, it may not be fair to directly compare (4) and (7), but it is reasonable to compare the costs of (4) and (7) divided by the lower bounds of the costs to optimally execute their corresponding tasks.

Theorem 1 *The approximation of our algorithm for TREE-CVRP is* 1.5. *(* The proof is omitted.*)*

References

[1] I. Averbakh and O. Berman, Sales-delivery man problems on treelike networks, *Networks*, 25 (1995), 45-58.

[2] M.L. Fischer. Vehicle Routing. in *Network Routing*, Handbooks in Operations Research and Management Science, **8**, Ball, M. O., T. L. Magnanti, C. L. Monma and G. L. Nemhauser (Eds.), Elsevier Science, Amsterdam, 1-33, 1995.

[3] G. Frederickson and D. Guan, Preemptive ensemble motion planning on a tree, *SIAM J. Computing*, 21-6 (1992), 1130-1152.

[4] B.L. Golden and A. A. Assad (Eds.). *Vehicle Routing: Methods and Studies*, Studies in Manag. Science and Systems 16, North-Holland Publ., Amsterdam, 1988.

[5] M. Haimovich and A.H.G. Rinnooy Kan. Bounds and Heuristics for capacitated routing problems *Mathematics of Operations Research*, **10**(4), 527-542, 1985.

[6] S. Hamaguchi, A study on Vehicle Routing Problems on Tree-shaped Networks, Master Thesis, Graduate School of Business Administration, Kobe Univ. of Commerce, 1998.

[7] Y. Karuno, H. Nagamochi, T. Ibaraki, Vehicle Scheduling on a tree to Minimize Maximum Lateness, *Journal of the Operations Research Society of Japan*, Vol. 39, No.3 (1996) 345-355.