Available online at www.sciencedirect.com
SCIENCE DIRECT•
ELSEVIER
Discrete Optimization 2 (2005) 168–184
DISCRETE OPTIMIZATION
www.elsevier.com/locate/disopt

One-way and round-trip center location problems

Arie Tamir*, Nir Halman

School of Mathematical Sciences, Tel Aviv University, Tel-Aviv, Israel

Received 10 April 2003; received in revised form 8 December 2004; accepted 23 December 2004

Abstract

In the classical p-center problem there is a set V of points (customers) in some metric space, and the objective is to locate p centers (servers), minimizing the maximum distance between a customer and his respective nearest server. In this paper we consider an extension, where each customer is associated with a set of existing depots or distribution stations he can use. The service of a customer consists of the travel of a server to some permissible depot, loading of some package (e.g., a spare part) at the depot, and the delivery of the package to the customer. This model is called the customer one-way problem. In the round-trip version of the problem, the service also includes the travel from the customer to the home base of the server. In both problems the customer cost of the service is a linear function of the distance travelled by the server. The objective is to locate p servers, minimizing the maximum customer cost (weighted distance travelled by the respective server).

Since the classical p-center problem is NP-hard, so are the one-way and the round-trip models we study. We present efficient constant factor approximation algorithms for these problems on general networks. Turning to special networks, we prove that the one-way problem is strongly NP-hard even on path networks. We then present polynomial time algorithms for the round-trip problem on general tree networks. We also discuss the single center case, and provide polynomial time algorithms for general networks, tree networks and planar Euclidean and rectilinear metric spaces.

Keywords: Facility location; Center location problems; Tree networks

1. Introduction

Given is an undirected connected graph $G = (V, E)$. Each edge $e \in E$ has a positive edge length, l_e. An edge is an image of a closed interval under a continuous bijective mapping, i.e., a Jordan arc. However, for our purposes an edge $e = (v_r, v_s)$ is identified with an interval of length l_e so that we can refer to its interior points. An interior point is identified by its distances along the edge (interval) from the two nodes v_r and v_s. Let $A(G)$ denote the continuum set of points on the edges of G. We also view $A(G)$ as a connected set which is the union of $|E|$ intervals. The edge lengths induce a distance function on $A(G)$. For any pair of points $x, y \in A(G)$, we let $d(x, y)$ denote the length of a shortest path $P(x, y)$, connecting x and y. For any $Y \subseteq A(G)$ and $x \in A(G)$ we let $d(x, Y) = d(Y, x) = \inf\{d(x, y) : y \in Y\}$. $A(G)$ is a metric space with respect to the above distance function. We refer to $A(G)$ as the network induced by G and the edge lengths $\{l_e\}$ $e \in E$.

$V = \{v_1, \dots, v_n\}$ is viewed as the set of customers. The customers are associated respectively with nonnegative weights $\{w_1, \dots, w_n\}$. There is also a set $X = \{x_1, \dots, x_m\}$ of points in $A(G)$, representing existing depots, or distribution stations.

Each customer v_i is associated with a subset of depots $X^i \subseteq X$, that v_i can potentially select from and use.

* Corresponding author
E-mail address: atamir@post.tau.ac.il (A. Tamir).

1572-5286/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2004.12.004

The goal is to establish p servers, $Y = \{y_1, \dots, y_p\}$ in $A(G)$, optimizing the following objective. Suppose that a customer v_i places a call for service. If he is served by the server at y_j, the service consists of the travel of the server from y_j to some depot $x_k \in X^i$, loading of some package (e.g., a spare part, etc.), at x_k, the travel to v_i, dropping of the package there and traveling back to y_j. (Alternatively, we can reverse the tour direction and talk about loading some waste at v_i and dumping it at some depot x_k.) The cost of the service is measured in terms of the total tour length of the server, i.e., $d(y_j, v_i) + d(v_i, x_k) + d(x_k, y_j)$, and the cost of loading and unloading, denoted by $t_{i,k}$. We assume that $t_{i,k} \geqslant 0$ for $v_i \in V$ and $x_k \in X^i$. The smaller the length the better is the service. Since there are no capacity constraints on the depots and the servers, the best service v_i can get from the servers is measured by

$$S_i(Y) = \min_{y_j \in Y, x_k \in X^i} \{d(y_j, v_i) + d(v_i, x_k) + d(x_k, y_j) + t_{i,k}\}.$$

Equivalently,

$$S_i(Y) = \min_{y_j \in Y} \left\{ d(y_j, v_i) + \min_{x_k \in X^i} \{d(v_i, x_k) + d(x_k, y_j) + t_{i,k}\} \right\}.$$

We assume that the cost of serving v_i is $w_i S_i(Y)$.

We note that in some applications the return portion of the server to its home base after accomplishing the mission, might be irrelevant and costless. (As an example, consider an emergency service where drugs or blood infusion bags are delivered to the homes of patients from central distribution depots.) There are two versions.

In the first one the server goes first to the customer, picks up some waste, travels to a depot and dump it there. In this case the cost of serving v_i is $w_i S_i^2(Y)$, where

$$S_i^2(Y) = \min_{y_j \in Y} \left\{ d(y_j, v_i) + \min_{x_k \in X^i} \{d(v_i, x_k) + t_{i,k}\} \right\}.$$

It is clear that in this variation each customer v_i will use a depot, say $x_{k(i)}$, satisfying $d(v_i, x_{k(i)}) + t_{i,k(i)} = \min_{x_k \in X^i} \{d(v_i, x_k) + t_{i,k}\}$. Thus, we can assume in this case that $X^i = \{x_{k(i)}\}$ is a singleton, and

$$S_i^2(Y) = \min_{y_j \in Y} \{d(y_j, v_i)\} + d(v_i, x_{k(i)}) + t_{i,k(i)} = d(v_i, Y) + d(v_i, x_{k(i)}) + t_{i,k(i)}.$$

In the second variation the server goes first to the depot, picks up a spare part, and then delivers it to the customer. The cost of the service does not depend on the length of the return part from the customer to the home base of the server. Specifically, the cost of serving v_i is $w_i S_i^3(Y)$, where

$$S_i^3(Y) = \min_{y_j \in Y, x_k \in X^i} \{d(y_j, x_k) + d(x_k, v_i) + t_{i,k}\}.$$

In this paper, we consider the MINMAX models corresponding to the three cost terms mentioned above. Specifically, the objective is to find $Y \subseteq A(G)$, $|Y| = p$, minimizing $G_1(Y) = \max_{i=1,\dots,n}\{w_i S_i(Y)\}$, ($G_2(Y) = \max_{i=1,\dots,n}\{w_i S_i^2(Y)\}$, $G_3(Y) \max_{i=1,\dots,n}\{w_i S_i^3(Y)\}$). The MINMAX problem with $G_1(Y)$ is called the *round-trip p-center problem*. The MINMAX problems with $G_2(Y)$ and $G_3(Y)$ are called the *depot one-way p-center problem*, and the *customer one-way p-center problem*, respectively. (In the depot one-way model the tour initiates at the home base of the server and terminates at the depot, while in the customer one-way model termination is at the customer.) Berman et al. [8], introduce the round-trip model where $X^i = X$, for each $i = 1, \dots, n$, and they call it the *collection depots p-center problem*. They consider only the case where $p = 1$, and discuss several applications of their model.

When the set of servers Y is restricted to be a subset of V we will call the respective model *discrete*. Otherwise, it will be referred to as a *continuous* model. We call a problem *unweighted* if $w_i = 1$, $i = 1, \dots, n$.

We let r_p^1, r_p^2 and r_p^3 denote the optimal objective values of the above three problems, respectively.

If for every customer v_i, $v_i \in X^i$, and the respective cost of loading and unloading at v_i is 0, then $S_i(Y) = 2\min_{y_j \in Y}\{d(y_j, v_i)\}$, and the model (mathematically) reduces to the classical p-center. (The classical continuous center problem is also recognized as "the absolute center problem", see [25,37,38].) In the classical models there are practically no depots, and the mission of the server is to travel directly to the customer, and then back to its home base. (We note in passing that the cost function $w_i S_i^2(Y) = w_i d(v_i, Y) + w_i d(v_i, x_{k(i)})$ of the depot one-way model, can be viewed as a variation of the weighted classical p-center problem. The cost of service is the weighted travel time of the server to v_i, $w_i d(v_i, Y)$, plus a fixed service cost, $a_i = w_i d(v_i, x_{k(i)})$.) Another special model which is discussed in the literature (see [10,11,15,29,30]), is the case where each customer v_i can use only one depot, i.e., for each v_i, X^i is a singleton, $X^i = \{x_{k(i)}\}$. We will refer to this special case as the *singleton* variation.

Since the classical problems are NP-hard on general graphs, we conclude that the above three problems are NP-hard.

If for some $i = 1, \ldots, n$, the cost terms $\{t_{i,k}\}$, $x_k \in X^i$, are independent of k, we can assume without loss of generality that the number of depots that v_i can select from, is bounded above by $2|E|$. (For each edge having more than two depots of X^i, it is sufficient to consider only the two which are respectively closest to the nodes of the edge.)

We can transform the above models on general graphs into equivalent models where $t_{i,k} = 0$ for each $i = 1, \ldots, n$, and $x_k \in X^i$. We first augment X to the node set. Consider for example the round-trip model. For each $i = 1, \ldots, n$, and $x_k \in X^i$ with $t_{i,k} > 0$, augment a new node, $x[i, k]$, to the graph G, and connect it with a single edge of length $t_{i,k}/2$ to the depot x_k. $x[i, k]$ will replace x_k as a potential depot of v_i. It is now easy to see that for each point y_j the distances on the augmented network satisfy

$$d(y_j, v_i) + d(v_i, x_k) + d(x_k, y_j) + t_{i,k} = d(y_j, v_i) + d(v_i, x[i, k]) + d(x[i, k], y_j).$$

Similar transformations can be used for the two one-way models. Therefore, throughout the paper we assume that $t_{i,k} = 0$ for all $i = 1, \ldots, n$; $x_k \in X^i$, and $X \subseteq V$.

In Section 2, we provide efficient constant factor approximation algorithms for the three problems, by reducing them to the p-center problem with customer weights and facility setup costs [9,23,24,27,43].

In Section 3 we address the single-facility models, and present polynomial time algorithms. (We also consider here the planar Euclidean and rectilinear cases.)

Section 4 focuses on the p-center problems on tree networks. We show that the depot one-way model reduces to the classical p-center problem, and is therefore solvable in $O(n \log^2 n)$ time. In contrast, the customer one-way problem is shown to be NP-hard even on path networks. Finally, for the discrete and continuous round-trip models on trees we give $O(n^2 \log n)$ time algorithms. More efficient algorithms are presented for the 1-center versions of these models.

In Section 5 we discuss some related extensions and open problems.

2. Approximation algorithms

In view of the NP-hardness of the models we next show how to obtain constant factor approximation polynomial algorithms for the three MINMAX models presented above. We present reductions of these models to instances of the classical p-center problem with customer weights and facility setup costs.

We consider first the discrete case ($Y \subseteq V$, $|Y| = p$). In this discrete case we can assume without loss of generality that we have a complete undirected graph $G = (V, E)$, with edge lengths (weights) satisfying the triangle inequality. In particular, for any pair of nodes $x, y \in V$, the weight of the edge (x, y) is $d(x, y)$, the shortest distance between x and y on G.

For the depot one-way model we augment to the underlying graph G, the set of nodes $U = \{u_1, \ldots, u_n\}$. For $i = 1, \ldots, n$, we connect node u_i to v_i with an edge of length $d(v_i, x_{k(i)})$, where $x_{k(i)}$ is defined above as the closest point to v_i in X^i. Let $G' = (V \cup U, E')$ denote the augmented graph. We then consider the classical p-center problem on G', where U is the set of customers, and servers can be located at V only. For $i = 1, \ldots, n$, the weight of customer u_i is defined to be w_i. It is easy to see that this instance of the classical p-center problem with customer weights and facility setup costs, is equivalent to the depot one-way model.

3-approximation polynomial algorithms for several versions of the above generalized classical p-center problem are described in [9,23,24,27,43]. (The first reference considers only the unweighted case where $w_i = 1$ for $i = 1, \ldots, n$, while the last two provide 3-approximation schemes even for the case where there are setup costs associated with the nodes in $V \cup U$, and an upper bound on the total setup cost of the selected facilities.) We conclude that there is a polynomial 3-approximation algorithm for the depot one-way problem.

In the original version of this paper we used more involved reductions to obtain 12-approximation and 9-approximation polynomial schemes for the round-trip and the customer one-way p-center problems, respectively. Our reductions are obsolete now, since Ageev [4], has recently shown how to obtain 3-approximations by using simple direct reductions to the p-suppliers problem [9,23,24,27], for the round-trip and the customer one-way p-center problems. (The p-suppliers problem with weights and setup costs is actually equivalent to the above generalized classical p-center problem.) For completeness, we briefly describe his reductions. We provide only the respective edge weights, as proposed by Ageev, but skip the proof that they satisfy the triangle inequality.

Let $H = (V \cup U, F)$ be a complete undirected bipartite graph with node sets $V = \{v_1, \ldots, v_n\}$, $U = \{u_1, \ldots, u_t\}$, and edge set F. The edges are associated with positive weights which satisfy the triangle inequality. For each pair of nodes $x, y \in V \cup U$, let $d'(x, y)$ denote the length (total edge weights) of a shortest path connecting x and y. Each node $u_i \in U$ is associated with a nonnegative weight w_i. The p-suppliers problem is to find a subset $S \subseteq V$, $|S| \leqslant p$, minimizing

$$\max_{u_i \in U} \left\{ w_i \min_{v_j \in S} d'(u_i, v_j) \right\}.$$

For the reductions of the customer one-way and the round-trip models on G, define the bipartite graph $H = (V \cup U, F)$ as follows: V is the node set of G and $U = \{u_1, \ldots, u_n\}$ is a copy of V, with u_i corresponding to v_i, $i = 1, \ldots, n$. Each node $u_i \in U$ is associated with the weight w_i that v_i has in G. Next, for each pair $v_i \in V$, $u_j \in U$, define the weight (length) of the edge $(v_i, u_j) \in F$, depending on the particular model, as follows: for the reduction of the customer one-way p-center problem, Ageev defines $d_3(v_i, u_j)$, the weight of (v_i, u_j), by

$$d_3(v_i, u_j) = \min_{x_k \in X^i} \{d(v_i, x_k) + d(x_k, v_j)\},$$

while for the round-trip p-center problem, the weight of (v_i, u_j), $d_1(v_i, u_j)$, is defined by

$$d_1(v_i, u_j) = \min_{x_k \in X^i} \{d(v_i, x_k) + d(x_k, v_j)\} + d(v_j, v_i) = d_3(v_i, u_j) + d(v_j, v_i).$$

We have considered above the weighted discrete model. The same approximation results hold also for the weighted continuous case, since the continuous models can be discretized as in the classical p-center problem. Consider, for example, the weighted round-trip version. From the discussion below (Section 3), on the single-facility model, we conclude that on each edge it is sufficient to consider only a discrete set of cardinality $O(n^2)$, for the location of servers. Specifically, with the notation in Section 3, focusing on a single edge, it is sufficient to consider only the intersection points of pairs of functions in the collection $\{w_i \hat{S}_i(y)\}$, $i = 1, \ldots, n$. Since for each pair i, j the equation $w_i \hat{S}_i(y) = w_j \hat{S}_j(y)$ contributes at most 4 points, we have $O(n^2)$ points in total on each edge. (Berman et al. [8], identify a discrete set for the unweighted problem.)

3. Locating a single facility

In this section, we consider the single-facility one-way and round-trip models.

3.1. The network case

The discrete models can be solved by complete enumeration, i.e., evaluating the respective objectives at each node of G.

Like the classical continuous (absolute) 1-center problem, (see [25]), the continuous single-facility round-trip center problem is solved by finding the best location on each edge of the network space $A(G)$. We note that Berman et al. [8] identify on each edge a finite dominating set (FDS) of polynomial cardinality for the unweighted version of this problem. Hence, the best solution on each edge for this unweighted version can be found in polynomial time. We discuss the weighted version.

Consider an edge $e = (v_s, v_t)$ of $A(G)$, and let y be a real parameter identifying points along this edge. In particular, y is bounded between 0, the value corresponding to the node v_s, and l_e, the length of the edge, which is the value corresponding to the node v_t. For each v_i define

$$\hat{S}_i(y) = \min_{x_k \in X^i} \{d(y, v_i) + d(v_i, x_k) + d(x_k, y)\} = d(y, v_i) + \min_{x_k \in X^i} \{d(v_i, x_k) + d(x_k, y)\}.$$

Our objective is to find a minimum point of the function

$$\hat{G}_1(y) = \max_{i=1,\ldots,n} \{w_i \hat{S}_i(y)\},$$

over the range $0 \leqslant y \leqslant l_e$.

It is easy to see that each function $\hat{S}_i(y)$ is piecewise linear and concave with at most 3 slopes. Moreover, each slope is in the set $\{-2, 0, +2\}$. It takes $O(|X^i|)$ time to construct the (at most) 2 breakpoints of $\hat{S}_i(y)$.

The function $\hat{G}_1(y)$ is the upper envelope of the collection of functions $\{w_i \hat{S}_i(y)\}$. Therefore, $\hat{G}_1(y)$ is a piecewise linear function with at most $O(n)$ breakpoints, (see [41, Lemma 4.2]). One of them is a minimizer of $\hat{G}_1(y)$. Using a standard divide and conquer approach, all the breakpoints of $\hat{G}_1(y)$ and its minimum can be found in $O(n \log n)$ time. We conclude that the round-trip 1-center problem on a network $A(G)$ can be solved in $O(|E|(n \log n + \sum_{i=1}^{n} |X^i|))$ time.

The above results can easily be extended to the two one-way models. For the depot one-way problem the related functions are

$$\hat{S}_i^2(y) = d(y, v_i) + d(v_i, x_{k(i)})$$

and

$$\hat{G}_2(y) = \max_{i=1,\ldots,n} \{w_i \hat{S}_i^2(y)\}.$$

($x_{k(i)}$ is the closest point to v_i in X^i.) Each function $\hat{S}_i^2(y)$ is piecewise linear and concave with at most 2 slopes. (Each slope is either +1 or −1.) $\hat{G}_2(y)$ is a piecewise linear function with at most O(n) breakpoints, (see [25]). As above all the breakpoints of $\hat{G}_2(y)$ and its minimum can be found in O($n \log n$) time. Therefore, the depot one-way 1-center problem on a network $A(G)$ can be solved in O($|E| n \log n$) time.

For the customer one-way problem the related functions are

$$\hat{S}_i^3(y) = \min_{x_k \in X^i} \{d(y, x_k) + d(x_k, v_i)\}$$

and

$$\hat{G}_3(y) = \max_{i=1,\ldots,n} \{w_i \hat{S}_i^3(y)\}.$$

The function $\hat{S}_i^3(y)$ is also piecewise linear and concave with at most 2 slopes: +1 and −1. It takes O($|X^i|$) time to construct the (at most one) breakpoint of $\hat{S}_i^3(y)$. As with $\hat{G}_2(y)$, $\hat{G}_3(y)$ is a piecewise linear function with at most O(n) breakpoints. As in the round-trip model, we conclude that the customer one-way 1-center problem on a network $A(G)$ can be solved in O($|E|(n \log n + \sum_{i=1}^n |X^i|)$) time.

In Section 4, which focuses on tree networks, we provide more efficient algorithms for locating a single facility on such networks.

3.2. The planar Euclidean and rectilinear cases

Although in this paper we focus on network models, it is interesting to consider the single-facility planar geometric versions of the above models. The input consists of the set of n demand points in the plane $V = \{v_1, \ldots, v_n\}$ (customers), and a set of m points in the plane $X = \{x_1, \ldots, x_m\}$ (depots). We consider the continuous versions only, and discuss first the singleton versions.

3.2.1. Planar singleton models

Consider the singleton round-trip problem. Let r be the parameter of the covering problem. (r is an upper bound on the weighted length of the round trip.) For each $i = 1, \ldots, n$, we denote the (planar) demand point (customer) and its unique depot by v_i and $x_{k(i)}$, respectively. To ensure a cover of r, we need to consider all the points y in the planar set $Y_i(r) = \{y | d(y, v_i) + d(y, x_{k(i)}) \leqslant r/w_i - d(v_i, x_{k(i)})\}$.

In the Euclidean case $Y_i(r)$, is an ellipse. The objective is then to find the smallest value of a parameter r, such that $\bigcap_{i=1,\ldots,n} \{Y_i(r)\}$ is nonempty. This problem can be formulated within the framework of the convex algebraic model in [16]. Therefore, we can find r_1^1, the optimal value of r in O(n) time.

Consider next the singleton versions of the one-way problems. (See [10,11].) In the depot one-way version, to ensure a cover of r, for $i = 1, \ldots, n$, we need to consider all the points y in the planar set $Y_i'(r) = \{y | d(y, v_i) \leqslant r/w_i - d(v_i, x_{k(i)})\}$. In the Euclidean case $Y_i'(r)$ is a disk. The objective is to find the smallest value of a parameter r, such that $\bigcap_{i=1,\ldots,n} \{Y_i'(r)\}$ is nonempty. It is shown in [16] how to find the optimal value of r in O(n) time. An almost identical result holds for the customer one-way problem since we also deal with a collection of disks, $\{Y_i''(r)\}$, where, $Y_i''(r) = \{y | d(x_{k(i)}, y) \leqslant r/w_i - d(v_i, x_{k(i)})\}$.

In the rectilinear case, $Y_i(r)$, $Y_i'(r)$ and $Y_i''(r)$ are convex and polyhedral for $i = 1, \ldots, n$. ($Y_i'(r)$ and $Y_i''(r)$ are squares and $Y_i(r)$ is an octagon, possibly degenerate.) Therefore, the optimal value of r can be found in O(n) time by solving a single LP in 3 variables: the 2 components of y and r. (See [33].)

3.2.2. Planar general models

The general single-facility round-trip and customer one-way center problems are more complicated. (Recall that the depot one-way version coincides with the singleton version since each customer uses the closest depot.)

3.2.2.1. Euclidean case In the Euclidean case of the round-trip problem each ellipse $Y_i(r)$ is now replaced by a set, say $Z_i(r)$, which is the union of O($|X^i|$) ellipses. In the covering problem we now need to determine whether the intersection of the collection of the n sets $\{Z_i(r)\}$, $i = 1, \ldots, n$, is nonempty. Due to the convexity of the ellipses, and the fact that each one of the O($|X^i|$) ellipses of $Z_i(r)$ contains v_i, it follows that $Z_i(r)$ is the complement of the infinite single face of the arrangement of these ellipses. From Theorem 5.7 in [40], we conclude that the boundary of $Z_i(r)$ can have at most $\lambda_4(|X^i|)$ vertices and ellipsoidal arcs. (This bound is "almost" linear, i.e., $\lambda_4(|X^i|) = O(|X^i|2^{\alpha(|X^i|)})$, where $\alpha(n)$ is the functional inverse of the Ackermann's function.) We can check whether $\bigcap_{i=1,\dots,n}\{Z_i(r)\}$ is nonempty by constructing the planar arrangement of all the $k = \sum_{i=1}^{n} |X^i| = O(mn)$ ellipses involved, and using a sweep line algorithm on the arrangement. This can be implemented in $O(k^2 \log k)$ time by the algorithms in [17] and Section 6 in [40]. (We suspect that the $O(m^2n^2 \log(mn))$ complexity is "almost" optimal since we have examples of the model where the complexity of the boundary of $\bigcap_{i=1,\dots,n}\{Z_i(r)\}$is $\Theta(m^2n^2)$.)

The optimal value of the single-facility round-trip problem, r_1^1, is the smallest value of the parameter r of the covering problem, for which $\bigcap_{i=1,\dots,n}\{Z_i(r)\}$ is nonempty. To find r_1^1 we use the parametric approach of Megiddo [32], with the parallel implementation of the algorithm from Agarwal et al. [1]. (See [2, Section 4].) The total time to find r_1^1 is then $O(m^2n^2 \log^3(mn))$.

In the customer one-way problem we obtain a collection of n sets, say $\{Z_i''(r)\}$, $i = 1, \dots, n$, each being the union of $O(|X^i|)$ disks. We can use the above $O(m^2n^2 \log(mn))$ approach for the round-trip model to check whether $\bigcap_{i=1,\dots,n}\{Z_i''(r)\}$ is nonempty. However, a further improvement is possible for this case.

$Z_i''(r)$ is not necessarily connected, and its boundary has only $O(|X^i|)$ vertices and circular arcs, see Kedem et al. [26]. For the same reason, for each pair of sets $Z_s''(r)$ and $Z_t''(r)$, the number of vertices and circular arcs of the boundary of $Z_s''(r) \cup Z_t''(r)$, is $O(|X^s| + |X^t|)$. On the other hand, since we deal only with two sets, each vertex of the boundary of $Z_s''(r) \cap Z_t''(r)$, is either a vertex of the boundary of $Z_s''(r) \cup Z_t''(r)$, a vertex of the boundary of $Z_s''(r)$, or a vertex of the boundary of $Z_t''(r)$. Thus, the number of vertices and circular arcs of the boundary of $Z_s''(r) \cap Z_t''(r)$, is also $O(|X^s| + |X^t|)$. Finally, each vertex of the boundary of $\bigcap_{i=1,\dots,n}\{Z_i''(r)\}$ is a vertex of the boundary of $Z_s''(r) \cap Z_t''(r)$, for some pair s, t. Thus, the total number of vertices and circular arcs of the boundary of $\bigcap_{i=1,\dots,n}\{Z_i''(r)\}$ is only $O(mn^2)$. (This result about the complexity of the intersection of n sets, each being the union of at most $O(m)$ disks, is a special case of Theorem 10 in [12].)

We show that by using a divide and conquer algorithm we can determine whether $\bigcap_{i=1,\dots,n}\{Z_i''(r)\}$ is nonempty in $O(mn^2 \log(mn))$ time. Suppose without loss of generality that n is even. Let $I_1 = \{1, \dots, n/2\}$ and $I_2 = \{n/2 + 1, \dots, n\}$. We divide the collection of sets $\{Z_i''(r)\}$ into two subcollections, $\{Z_i''(r); i \in I_1\}$ and $\{Z_i''(r); i \in I_2\}$. Recursively we find $C_1 = \bigcap_{i \in I_1}\{Z_i''(r)\}$ and $C_2 = \bigcap_{i \in I_2}\{Z_i''(r)\}$. (The number of vertices and circular arcs of the boundaries of C_1 and C_2 is $O(mn^2)$). By using a sweep line of the boundaries of C_1 and C_2 we then construct the boundary of $C_1 \cap C_2$ in $O(mn^2 \log(mn))$ time.

Let $T(m, n)$ denote the total time needed to construct the boundary of $\bigcap_{i=1,\dots,n}\{Z_i''(r)\}$. From the above approach we have

$$T(m, n) \leqslant cmn^2 \log(mn) + 2T(m, n/2),$$

for some constant c. Therefore, $T(m, n) \leqslant 2c(mn^2 \log(mn)) = O(mn^2 \log(mn))$.

The optimal value of the single-facility customer one-way problem, r_1^3, is the smallest value of the parameter r of the covering problem, for which $\bigcap_{i=1,\dots,n}\{Z_i''(r)\}$ is nonempty. To find r_1^3 we use the parametric approach of Megiddo [32], with a parallel implementation of the above algorithm, which tests the nonemptiness of $\bigcap_{i=1,\dots,n}\{Z_i''(r)\}$. The details of such a parallel implementation are quite involved, and will be discussed elsewhere. We only note in passing that r_1^3 can be computed by this parametric approach in $O(mn^2 \text{ polylog}(mn))$ time.

3.2.2.2. Rectilinear case Consider the rectilinear case of the round-trip problem. Analogously to the Euclidean case, in the covering problem we need to determine whether the intersection of a collection of n sets, say $\{W_i(r)\}$, $i = 1, \dots, n$, is nonempty.

$W_i(r)$ is the union of $O(|X^i|)$ octagons, say $\{O_t^i\}$, which intersect at v_i. The edges of all these octagons have only four different orientations, (i.e., their slopes are in the set $\{0, +1, -1, +\infty\}$). We decompose the boundary of each octagon O_t^i with respect to the horizontal line passing through v_i. We slightly perturb the (at most) two infinite slopes of the octagon, and view the part of the boundary above this line as a concave piecewise linear function, say f_t^i. Similarly, the lower part, which is convex, is denoted by g_t^i. Consider now the boundary of $W_i(r)$. The part of this boundary which is above the horizontal line containing v_i, can be represented by the function $f^i(x)$, which is the upper envelope (pointwise maximum function) of the $O(|X^i|)$ concave functions $\{f_t^i\}$. Similarly, the lower part of this boundary is represented by $g^i(x)$, which is the lower envelope of the collection $\{g_t^i\}$. $f^i(x)$ and $g^i(x)$ are both piecewise linear, and their slopes are in the set $\{0, +1, -1, +\infty, -\infty\}$. Therefore, the complexity of f^i and g^i is $O(|X^i|)$ due to Lemma 4.2 [41]. The graphs of f^i and g^i can be generated in $O(|X^i| \log |X^i|)$ time by a divide and conquer algorithm [40].

The boundary of $\bigcap_{i=1,\dots,n}\{W_i(r)\}$, which is not necessarily connected, can also be represented by upper and lower graphs, $F(x)$ and $G(x)$, respectively. Let $F(x)$ be the lower envelope of the graphs $\{f^i(x)\}$, $i = 1, \dots, n$, and let $G(x)$ be the upper envelope of the graphs $\{g^i(x)\},\ i = 1, \ldots, n$. We clearly have

$$\bigcap_{i=1,\ldots,n} \{W_i(r)\} = \{(x, z) : G(x) \leqslant z \leqslant F(x)\}.$$

Again, from Lemma 4.2 in [41] it follows that the complexity of $F(x)$ and $G(x)$ is $O(mn)$, and it can be constructed in $O(mn \log(mn))$ time. To conclude, for any nonnegative real r, the complexity of the boundary of $\bigcap_{i=1,\ldots,n}\{W_i(r)\}$ is $O(mn)$, and it can be constructed in $O(mn \log(mn))$ time. In particular, we can determine in $O(mn \log(mn))$ time whether $\bigcap_{i=1,\ldots,n}\{W_i(r)\}$ is empty or not. We also note that since the algorithm used is a divide and conquer scheme, based on merging, it can be implemented by $O(mn)$ processors in $O(\log^2(mn))$ time. Therefore, in the rectilinear case, we can implement the parametric approach of Megiddo [32] to find r_1^1, the optimal value of the single-facility round-trip problem in $O(mn \log^4(mn))$ time. This bound can be further improved to $O(mn \log^3(mn))$ if we use the approach in [14] to implement each one of the merging phases.

In the customer one-way problem we obtain a collection of n sets, say $\{W_i''(r)\},\ i = 1, \ldots, n$, each being the union of (rectilinear) squares. Without loss of generality we rotate the axes by 45 degrees and assume that the edges of all squares are parallel to the axes. The results of Kedem et al. [26] mentioned above in the context of the Euclidean case, apply to this model as well. We conclude that for $i = 1, \ldots, n$, $W_i''(r)$ has only $O(|X^i|)$ vertices and edges. In particular, $W_i''(r)$ can be subdivided into $O(|X^i|)$ axis-parallel rectangles, whose interiors are disjoint. The time needed for this decomposition is $O(|X^i| \log |X^i|)$, (see [12]). Altogether, we will have at most $\sum_{i=1}^n |X^i| = O(mn)$ axis-parallel rectangles.

We can now use the $O(mn \log(mn))$ algorithm of Overmars and Yap [36], as in [12], (or the counting $O(mn \log(mn))$ algorithm of Chew and Kedem [13]), to determine whether $\bigcap_{i=1,\ldots,n}\{W_i''(r)\}$ is nonempty.

To find r_1^3, the optimal value of the single-facility customer one-way problem, we first identify a set containing r_1^3. Recall that r_1^3 is the smallest value of r such that $\bigcap_{i=1,\ldots,n}\{W_i''(r)\}$ is nonempty. Therefore, there exist a pair of points v_i and v_j, and a pair of depots $x_k \in X^i$, and $x_q \in X^j$, such that r_1^3 is the smallest value of the parameter r for which the intersection of the two (rectilinear) squares $\{y | d(x_k, y) \leqslant r/w_i - d(v_i, x_k)\}$ and $\{y | d(x_q, y) \leqslant r/w_j - d(v_j, x_q)\}$ is nonempty. (To simplify, we replace the l_1 norm by the l_∞ norm, i.e., rotate the axes by 45 degrees.)

Let $x_k = (x_k(1), x_k(2))$ and $x_q = (x_q(1), x_q(2))$. Assume without loss of generality that $x_k(1) \geqslant x_q(1)$ and $x_k(2) \geqslant x_q(2)$. Then it is easy to see that

$$r_1^3 = [(x_k(1) + d(v_i, x_k)) - (x_q(1) - d(v_j, x_q))]/[1/w_i + 1/w_j]$$

or

$$r_1^3 = [(x_k(2) + d(v_i, x_k)) - (x_q(2) - d(v_j, x_q))]/[1/w_i + 1/w_j].$$

We are now ready to define a set containing r_1^3.

For $i = 1, \ldots, n$, and $x_k \in X$, define $w_{i,k} = w_i$ if $x_k \in X^i$, and $w_{i,k} = \infty$ otherwise. Also let

$$\alpha_{i,k} = x_k(1) + d(v_i, x_k),$$

$$\beta_{i,k} = x_k(1) - d(v_i, x_k),$$

$$\gamma_{i,k} = x_k(2) + d(v_i, x_k),$$

$$\delta_{i,k} = x_k(2) - d(v_i, x_k).$$

Consider the sets

$$R^*(1) = \{(\alpha_{i,k} - \beta_{j,q})/(1/w_{i,k} + 1/w_{j,q}) : i, j = 1, \ldots, n; k, q = 1, \ldots, m\},$$

$$R^*(2) = \{(\gamma_{i,k} - \delta_{j,q})/(1/w_{i,k} + 1/w_{j,q}) : i, j = 1, \ldots, n; k, q = 1, \ldots, m\},$$

and

$$R^* = R^*(1) \cup R^*(2).$$

An element r in R^* is smaller than r_1^3 if and only if $\bigcap_{i=1,\ldots,n}\{W_i''(r)\}$ is empty. From the above, this decision problem can be solved in $O(mn \log(mn))$ time. The structure of the set R^* enables us to use the search procedure in [34], with the modification in [14], to find r_1^3 in R^*. Since the decision problem is solved in $O(mn \log(mn))$ time, the total time needed to compute r_1^3 in this case is $O(mn \log^2(mn))$.

4. The one-way and the round-trip p-center problems on tree networks

In this section we consider the case when the graph is a tree $T = (V, E)$. The respective tree network metric space is denoted by $A(T)$. A closed and connected subset of $A(T)$ is called a *subtree*. If all the leaves (relative boundary points) of a subtree are nodes of T the subtree is called *discrete*. To simplify the notation we assume that the set of depots $X = \{x_1, \dots, x_m\}$ is a subset of $V = \{v_1, \dots, v_n\}$.

4.1. The one-way p-center problems

As we have noted above in the depot one-way model we can assume that each customer v_i uses the closest depot to v_i in X^i, say $x_{k(i)}$. The cost function of a customer v_i from the set of servers, $w_i S_i^2(Y)$, is a linear function of $d(v_i, Y)$, the distance of v_i from Y.

$$w_i S_i^2(Y) = w_i d(v_i, Y) + w_i d(v_i, x_{k(i)}).$$

Therefore, this model can be solved by the techniques to solve the classical p-center problem on trees. The discrete model can be solved in $O(n \log^2 n)$ time by the algorithm in [35], and the continuous version can be solved with the same complexity by the algorithm in [34], when we implement the modification in [14].

Theorem 4.1. *The discrete and continuous depot one-way p-center problems on a tree network are solvable in* $O(n \log^2 n)$ *time.*

In contrast, we have the following result.

Lemma 4.1. *The customer one-way p-center problem is strongly NP-hard even on path networks, with* $|X^i| = 2$, *for* $i = 1, \dots, n$.

Proof. Consider the vertex cover problem on a general undirected graph $G = (U, E')$, with $U = \{u_1, \dots, u_n\}$. We reduce it to an instance of the customer one-way problem on a path network.

We define the nodes of the path network by a set V of at most $|E'| + n$ points on the real line. For each edge (u_i, u_j) of the graph G define the point $u_{i,j} = (i + j)/2$ on the real line. It represents the location of customer $\{i, j\}$. (Note that a point can represent the location of several customers. There are $|E'|$ customers.) Next, define the set of depots as the set of points on the real line, $X = \{x_1, \dots, x_n\}$, by setting $x_i = i$, $i = 1, \dots, n$. V will consist of the customer points and the depot points. The edges of the path network are then defined by the at most $|E'| + n - 1$ segments on the line connecting consecutive pairs of points in V.

We assume that a customer $\{i, j\}$ can use only two depots, x_i and x_j.

Consider the following customer one-way p-center problem: The cost coefficient $w_{i,j}$ of serving customer $\{i, j\}$, $(u_i, u_j) \in E'$, is $w_{i,j} = 2/|j - i|$. Using the above notation r_p^3 denotes the optimal value of this problem. We claim that $r_p^3 \leqslant 1$ if and only if there is a vertex cover of cardinality p to the graph G. Indeed, if $r_p^3 \leqslant 1$, then a server y can serve customer $\{i, j\}$ if and only if $y = x_i$ or $y = x_j$. Therefore, $r_p^3 \leqslant 1$ if and only if there is a subset X_p of p points in X, such that for each $\{i, j\}$, either $x_i \in X_p$ or $x_j \in X_p$. The latter is clearly equivalent to determining whether $G = (U, E')$ has a vertex cover of cardinality p. □

In spite of the above hardness result, we have learned from Ageev [4], that the customer one-way p-center problem on path networks is polynomially solvable when $X^i = X$, for $i = 1, \dots, n$. This follows from a result by Beresnev [7], for the uncapacitated facility location problem (published in Russian). Specifically, the decision problem corresponding to this customer one-way problem on a path is a minimum cardinality set cover (hitting) problem with a 1-connected constraint matrix. (By definition, a $\{0, 1\}$ matrix (b_{ij}), $i = 1, \dots, k$, $j = 1, \dots, q$, is *1-connected* if for any two rows i_1 and i_2, the sequence $\{b_{i_1 j} - b_{i_2 j}\}$ changes sign at most once when j runs over $1, \dots, q$.) Beresnev [7], proved that even the minimum weight set cover (hitting) problem with an $k \times q$ 1-connected constraint matrix can be solved in $O(kq)$ time. (See also [3,5,6].)

Ageev [4], has also found a direct $O(m + n \log n)$ greedy type algorithm to solve the minimum cardinality set cover problem corresponding to this special instance of the customer one-way problem on a path with $X^i = X$, for $i = 1, \dots, n$.

For completeness we point out that with the above covering algorithm of Ageev, the p-center itself can then be solved in $O((m + n) \log^2(m + n))$ time by using the search procedures in [18,34,36] with the modification in [14].

Given are a path $G = (V, E)$, with $V = \{v_1, \dots, v_n\}$, and $E = \{(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)\}$, and a subset of depots $X = \{x_1, \dots, x_m\} \subseteq V$. Assume without loss of generality that the nodes in V and the depots in X are points on a real, satisfying: $v_1 < \dots < v_n$ and $x_1 < \dots < x_m$.

We start with the discrete case, $Y \subseteq V$. Suppose without loss of generality that $x_1 \leqslant v_1 < v_n \leqslant x_m$. Also, for $i = 1, \ldots, n$, let $x_{k(i)}$ be the largest point in X which is not greater than v_i. In this case, r_p^3, the optimal objective value is of the following form: There exists a server at v_j serving v_i via $x_{k(i)}$ or $x_{k(i)+1}$ with objective value r_p^3. Thus, if $v_j \leqslant x_{k(i)}$, $r_p^3 = w_i(v_i - v_j)$, if $x_{k(i)} \leqslant v_j \leqslant x_{k(i)+1}$, $r_p^3 = w_i(v_j + v_i - 2x_{k(i)})$, or $r_p^3 = w_i(2x_{k(i)+1} - v_j - v_i)$, and if $v_j \geqslant x_{k(i)+1}$, $r_p^3 = w_i(v_j - v_i)$. Fixing v_i and varying v_j, from v_1 to v_n, the potential set of values that r_p^3 can take on, can be represented as the union of 4 sorted subsets, columns. Therefore, the entire set of values that r_p^3 may take on in the discrete version of the problem can be viewed as a matrix with at most $4n$ sorted columns. Using the above $O(m + n \log n)$ algorithm for solving the unweighted covering problem, we can now apply the search procedures in [18,35] to find r_p^3 in this matrix in$O((m + n) \log^2(m + n))$ time.

The representation of a structured compact set containing the optimal value r_p^3 in the continuous case is more complicated. Nevertheless, by a simple local argument, it follows that r_p^3 is attained as the relevant cost between some customer v_i served by a server located at some depot in X, or there exist a pair of customers, v_i and v_k, which are served by some common server, say y_t, and the service costs of the pair of customers are identical. Since we assume that $X \subseteq V$, in the former case, a potential set including r_p^3 is the same as the one obtained above for the discrete model. In the latter case y_t is a "bottleneck point", and r_p^3 can be shown to be an element in a set of the form $\{(a_i - b_j)/(c_i - d_j) | i, j = 1, \ldots, n\}$, where the sequences $\{a_i\}$, $\{b_i\}$, $\{c_i\}$ and $\{d_i\}$ can be computed in $O(n)$ time. (See [34] for a similar argument used to represent the solution to the classical continuous weighted p-center problem on a general tree.)

With the above representation of the optimal value, we can use the search procedures in [34] with the modification in [14], to solve the continuous model in $O((m + n) \log^2(m + n))$ time.

4.2. *The round-trip p-center problem*

In the rest of the section we focus on the round-trip p-center model on a tree network, and present polynomial time algorithms for its solution. In this case if a server located at y serves customer v_i via depot x_k, we have $d(y, v_i) + d(v_i, x_k) + d(x_k, y) = 2(d(v_i, x_k) + d(y, P(v_i, x_k)))$. (Note that the function $f_i(y) = \min_{x_k \in X^i} \{d(y, v_i) + d(v_i, x_k) + d(x_k, y)\}$ is neither convex nor concave, even on the real line, as illustrated by the case where $v_1 = 0$, and $X^1 = \{x_1, x_2\} = \{-1, 2\}$.)

In the discrete model where the servers are restricted to be nodes, the optimal value r_p^1 is identified by some triplet $(y, x_k, v_i) = (v_t, v_j, v_i)$. We then have $r_p^1 = 2w_i(d(v_i, v_j) + d(v_s, v_t))$, where $v_s \in P(v_i, v_j)$ is the closest point on $P(v_i, v_j)$ to the server at v_t. Thus, we can explicitly identify in $O(n^3)$ time, a set R' of $O(n^3)$ cardinality containing r_p^1. To avoid the cubic space and time, we consider R^*, a super set of R' of cardinality $O(n^4)$, which has only a compact, quadratic space representation. Moreover, this quadratic representation can be constructed in $O(n^2 \log n)$ time. Knowing that R^* contains r_p^1, we can then search through it with the covering problem algorithm (see below) as a solver of the decision problem.

Let $Q = \{d(v_s, v_t) : v_s, v_t \in V\}$, and let Q^* be the sorted list (vector) of the $O(n^2)$ elements in Q. R^* is represented as a set of $n(n-1)/2$ vectors (columns) of length $|Q|$ each. The (i, j) column, $R_{i,j}$ is given by

$$R_{i,j} = 2w_i d(v_i, v_j) e + 2w_i Q^*.$$

(e is the vector of size $|Q|$, all of whose components are equal to 1.) Since Q^* is a sorted list, $R_{i,j}$ is also a sorted vector or a monotone column. The search procedures of Federickson and Johnson [18] are therefore applicable in searching over R^*. We note in passing that the time needed to obtain this compact representation of R^* is only $O(n^2 \log n)$.

Similarly, we can identify a set of cardinality $O(n^4)$ for the continuous model, but we do not know a compact representation of such a set which requires subcubic time to construct and search over. Nevertheless, we will show a parametric approach to solve the continuous model directly in $O(n^2 \log n)$ time. This approach avoids the construction and search over such a set.

Given a positive real r, the decision problem is to determine whether $r_p^1 \leqslant r$. Equivalently, consider the covering problem where we need to determine whether the minimum number of servers in a set Y, such that $S_i(Y) \leqslant r_i = r/w_i$, for $i = 1, \ldots, n$, is smaller than or equal to p.

Define

$$T_{i,k}(r_i) = \{y | 2(d(v_i, x_k) + d(y, P(v_i, x_k))) \leqslant r_i\} = \{y | d(y, P(v_i, x_k)) \leqslant r_i/2 - d(v_i, x_k)\},$$

$$T_i(r_i) = \bigcup_{x_k \in X^i} \{T_{i,k}(r_i)\}.$$

If $T_{i,k}(r_i)$ is nonempty, i.e., $r_i/2 - d(v_i, x_k) \geqslant 0$, it contains the path $P(v_i, x_k)$, and it is called a *path neighborhood*. If $T_i(r_i)$ is nonempty it is the union of path neighborhoods, where all the paths share a common end point, v_i. Thus, $T_i(r_i)$ is a subtree containing v_i. (In fact, it contains any path connecting v_i with a closest depot in X^i.)

It follows from the above that the covering problem can be solved in polynomial time, since it amounts to covering a family of subtrees with a minimum number of points in $A(T)$. (See [19,20].) We will present an $O(n^2)$ algorithm for this covering problem.

4.2.1. An $O(n^2)$ algorithm for the round-trip covering problem

We assume without loss of generality that the original tree is binary (see [42]), and it is rooted at v_1. We also assume that the set Q of distances between all pairs of nodes is known.

We describe the algorithm for the continuous case. Its modification for the discrete case is straightforward.

The first five steps of the algorithm will generate all the subtrees $\{T_i(r_i)\}$. A subtree $T_i(r_i)$ will be represented by its nodes, its edges and its set of leaves (relative boundary points), which are not nodes of the original tree. (A leaf will be characterized by the edge containing it, and its distances from the two nodes of the edge.) Specifically, we will record $u_i(r_i)$ the closest point to v_1 in $T_i(r_i)$. The importance of the set $\{u_j(r_j)\}$ follows from the fact that the incidence matrix $A=(a_{i,j})$ of the two collections, $\{T_i(r_i)\}$ and $\{u_j(r_j)\}$ is the only input needed to solve the covering problem. In particular, there is an optimal solution to the covering problem, where all the covering points are in the collection $\{u_j(r_j)\}$. (Note that for each pair $i, j = 1, \dots, n,\ i \neq j$, $T_i(r_i) \cap T_j(r_j)$ is nonempty if and only if $u_i(r_i)$ is in $T_j(r_j)$ or $u_j(r_j)$ is in $T_i(r_i)$.)

For each node $v_i \in V$ perform Steps I–V.

Step I: For each $x_j \in X^i$, check whether $T_{i,j}(r_i)$ is nonempty, i.e., check whether $r_i/2 - d(v_i, x_j) \geqslant 0$. Define $X_i(r_i) = \{x_j \in X^i : r_i/2 - d(v_i, x_j) \geqslant 0\}$. (Time spent is $O(n)$.)

Step II: Generate $T_i'(r_i)$ the subtree induced by v_i and $X_i(r_i)$, and rooted at v_i. (Time spent is $O(n)$.)

Step III: For each node v_j of $T_i'(r_i)$ define the surplus radius $s_i^j(r_i)$ by

$$s_i^j(r_i) = \max_{\{x_k \in X_i(r_i)\}} \{r_i/2 - d(v_i, x_k) - d(v_j, P(v_i, x_k))\}.$$

Compute the terms $s_i^j(r_i)$ for all nodes v_j of $T_i'(r_i)$, as follows:

Let

$$\alpha_i^j(r_i) = r_i/2 - s_i^j(r_i) = \min_{\{x_k \in X_i(r_i)\}} \{d(v_i, x_k) + d(v_j, P(v_i, x_k))\},$$

$$\beta_i^j(r_i) = \min_{\{x_k \in X_i(r_i): v_j \in P(x_k, v_i)\}} \{d(v_i, x_k) + d(v_j, P(v_i, x_k))\} = \min_{\{x_k \in X_i(r_i): v_j \in P(x_k, v_i)\}} d(v_i, x_k)$$

and

$$\delta_i^j(r_i) = \min_{\{x_k \in X_i(r_i): v_j \notin P(x_k, v_i)\}} \{d(v_i, x_k) + d(v_j, P(v_i, x_k))\}.$$

We have

$$\alpha_i^j(r_i) = \min(\beta_i^j(r_i), \delta_i^j(r_i)).$$

First use a bottom-up algorithm, starting at the leaves of $T_i'(r_i)$ to compute the terms $\beta_i^j(r_i)$ for all nodes v_j.

Specifically, if v_j is a leaf of $T_i'(r_i)$, then $v_j \in X^i$ and $\beta_i^j(r_i) = d(v_i, v_j)$. Suppose that v_j is not a leaf. If $v_j \in X^i$, $\beta_i^j(r_i) = d(v_i, v_j)$. If $v_j \notin X^i$, and v_j has only one child, say v_s, $\beta_i^j(r_i) = \beta_i^s(r_i)$. If $v_j \notin X^i$, and v_j has two children, say v_s and v_t, $\beta_i^j(r_i) = \min(\beta_i^s(r_i), \beta_i^t(r_i))$.

Next, start at the root of $T_i'(r_i)$, and go down towards its leaves to compute the terms $\delta_i^j(r_i)$ for all nodes v_j.

Set $\delta_i^i(r_i) = \infty$.

Let v_s be the parent of v_j. Suppose first that v_j is the only child of v_s. If $v_s \in X_i(r_i)$, then $\delta_i^j(r_i) = d(v_j, v_s) + \min(\delta_i^s(r_i), d(v_s, v_i))$. If $v_s \notin X_i(r_i)$, then $\delta_i^j(r_i) = d(v_j, v_s) + \delta_i^s(r_i)$.

Suppose next that v_s has two children, v_j and v_t. If $v_s \in X_i(r_i)$, then $\delta_i^j(r_i) = d(v_j, v_s) + \min(\delta_i^s(r_i), d(v_s, v_i), \beta_i^t(r_i))$.

If $v_s \notin X_i(r_i)$, then $\delta_i^j(r_i) = d(v_j, v_s) + \min(\delta_i^s(r_i), \beta_i^t(r_i))$.

(For each $v_i \in V$, the total time spent to generate all the surplus radii $\{s_i^j(r_i)\}$, $v_j \in T_i(r_i)$, is $O(n)$.)

Step IV: For each node v_j of $T_i'(r_i)$ let V_i^j be the set of nodes which connect to $T_i'(r_i)$ via v_j, i.e.,

$$V_i^j = \{v_t : v_t \notin T_i'(r_i), v_j \in P(v_i, v_t)\}.$$

Let $L_i^j = \{d(v_j, v_t) : v_t \in V_i^j\}$.

(For each node $v_i \in V$, it takes O(n) time to generate all the sets $\{L_i^j\}$, $v_j \in T_i'(r_i)$.)

Step V: Using the terms $\{s_i^j(r_i)\}$, $v_j \in T_i'(r_i)$, computed in Step III, and the lists $\{L_i^j\}$, $v_j \in T_i'(r_i)$, computed in Step IV, find in O(n) time all the nodes and the nonnode leaves of $T_i(r_i)$. (Note that a node $v_t \in V_i^j$ is in $T_i(r_i)$ if and only if $d(v_t, v_j) \leqslant s_i^j(r_i)$.) A leaf will be characterized by the edge containing it, and its distances from the two nodes of the edge. ($u_i(r_i)$, the closest point to v_1 in $T_i(r_i)$, is either a node of $T_i(r_i)$ or a nonnode leaf of $T_i(r_i)$.)

(Clearly, for each node $v_i \in V$, it takes O(n) time to generate all the nodes and leaves of $T_i(r_i)$.)

[We now turn to the next step, where we construct the incidence matrix $A = (a_{i,j})$, of the collection of subtrees (rows) $\{T_i(r_i)\}$, and the collection of points (columns) $\{u_j(r_j)\}$, generated above.]

Step VI: Suppose that $u_j(r_j)$ is on the edge (v_s, v_t). $u_j(r_j) \in T_i(r_i)$, i.e., $a_{i,j} = 1$ if and only if one of the following conditions hold:

1. $u_j(r_j)$ is a node in $T_i(r_i)$.
2. $u_j(r_j)$ is not a node, and both v_s and v_t are in $T_i(r_i)$.
3. $u_j(r_j)$ is not a node, v_s is in $T_i(r_i)$, $T_i(r_i)$ has a leaf $z_i(r_i, v_s, v_t)$ on the edge (v_s, v_t), and $u_j(r_j)$ belongs to $P(v_s, z_i(r_i, v_s, v_t))$.
4. $u_j(r_j)$ is not a node, v_t is in $T_i(r_i)$, $T_i(r_i)$ has a leaf $z_i(r_i, v_s, v_t)$ on the edge (v_s, v_t), and $u_j(r_j)$ belongs to $P(v_t, z_i(r_i, v_s, v_t))$.

(It clearly takes O(n^2) time to construct the incidence matrix $A = (a_{i,j})$.)

Step VII: With the matrix A computed above, apply the O(n^2) algorithm in [19,39] to solve the minimum covering problem of the collection $\{T_i(r_i)\}$. The output is the minimum number of points (servers) in a set Y, such that $S_i(Y) \leqslant r_i$, for $i = 1, \dots, n$. (Note that the optimal set Y is a subset of the collection $\{u_j(r_j)\}$).

To conclude we observe that the running time of the above algorithm is O(n^2).

4.2.2. *An* O($n^2 \log n$) *algorithm for the discrete model*

To solve the discrete round-trip model we need to use the discrete variation of the above O(n^2) continuous covering problem. The only modifications needed are in Steps V–VI. Specifically, a subtree $T_i(r_i)$ for the discrete case is the discrete subtree induced only by the nodes of the respective continuous subtree, i.e., the nonnode leaves are deleted. In Step VI, $u_j(r_j) \in T_i(r_i)$, i.e., $a_{i,j} = 1$, if and only if $u_j(r_j)$ is a node in $T_i(r_i)$.

We use the O(n^2) algorithm for the discrete covering problem to search for the optimal value over the set of O(n^2) monotone columns $\{R_{i,j}\}$ defined above. In order to determine whether a positive real r is greater than or equal to the optimal value r_p^1, we solve the above covering problem with radii $\{r/w_1, \dots, r/w_n\}$, and compare the minimum number of servers needed, say $p(r)$, with p. $r_p^1 \leqslant r$ if and only if $p(r) \leqslant p$. Following the search procedure in [18], the total time needed to find r_p^1 is O($n^2 \log n$).

4.2.3. *An* O($n^2 \log n$) *algorithm for the continuous model*

We use the above algorithm for the covering problem parametrically, following the general parametric approach of Megiddo [31].

In the parametric version with r as a real parameter, for each v_i we let $r_i = r/w_i$. The optimal value of the continuous problem, r_p^1, is the smallest value of r, for which the solution to the covering problem is at most p. (For convenience, for the parametric problem the entities $X_i(r/w_i)$, $T_i'(r/w_i)$, $T_i(r/w_i)$, etc., will be denoted respectively by $X_i(r)$, $T_i'(r)$, $T_i(r)$, etc.)

Parametric Algorithm

Step I: For each pair i, j, $v_i \in V$, $x_j \in X^i$, let $r^{i,j}$ be the solution to the equation $r/2w_i - d(v_i, x_j) = 0$, i.e., $r^{i,j} = 2w_i d(v_i, x_j)$. Let

$$R_1 = \{r^{i,j} : i = 1, \dots, n; x_j \in X^i\}.$$

Using a binary search, with the (nonparametric) covering algorithm as a solver of the decision problem, find a pair of consecutive elements in the sorted list of elements of R_1, say r_1^- and r_1^+, such that the solution value to the covering problem with $r = r_1^-$ is larger than p and the solution value to the covering problem with $r = r_1^+$ is smaller than or equal to p.

[Note that for all values of r in the interval $[r_1^-, r_1^+)$, the set $X_i(r)$ and the subtree $T_i'(r)$, induced by v_i and $X_i(r_i)$, are fixed and independent of r. The total time spent to identify this fixed collection of subsets $\{X_i(r)\}$ is $O(n^2 \log n)$.]

Step II: For each node v_i, in $O(n)$ time generate the (fixed) subtree $T_i'(r)$.

(The total time spent to identify this fixed collection of subtrees $\{T_i'(r)\}$ is $O(n^2)$.)

Step III: Apply Step III of the nonparametric version of the algorithm to the fixed collection $\{T_i'(r)\}$.

(Note that for each node v_j of a (fixed) subtree $T_i'(r)$, the respective term $\alpha_i^j(r) = r/2w_i - s_i^j(r) = \min_{x_k \in X_i(r_i)}(d(v_i, x_k) + d(v_j, P(v_i, x_k)))$, is also fixed for all values of the parameter r in the interval $[r_1^-, r_1^+)$. In particular, the respective surplus radius $s_i^j(r)$, is a linear function of r in this range, i.e., $s_i^j(r) = r/2w_i - \alpha_i^j(r)$.)

Step IV: For each subtree $T_i'(r)$ and each node v_j of $T_i'(r)$ let V_i^j be the set of nodes which connect to $T_i'(r)$ via v_j, i.e.,

$$V_i^j = \{v_t : v_t \notin T_i'(r), v_j \in P(v_i, v_t)\}.$$

Let $L_i^j = \{d(v_j, v_t) : v_t \in V_i^j\}$.

In $O(n^2)$ total time generate all the sets L_i^j, $v_j \in T_i'(r)$, $i = 1, \dots, n$.

Step V: For $i = 1, \dots, n$, and $v_j \in T_i'(r)$, let $R^{i,j} = \{r | s_i^j(r) = a, a \in L_i^j\}$. (Note that $|\bigcup_{v_j \in T_i'(r)} R^{i,j}| = O(n)$.)

Let $R_2 = \bigcup_{\{i=1,\dots,n; v_j \in T_i'(r)\}} R^{i,j}$.

Using a binary search, with the (nonparametric) covering algorithm as a solver of the decision problem, find a pair of consecutive elements in the sorted list of elements in R_2, say r_2^- and r_2^+, such that the solution value to the covering problem with $r = r_2^-$ is larger than p and the solution value to the covering problem with $r = r_2^+$ is smaller than or equal to p.

[Note that for all values of r in the interval $[r_1^-, r_1^+) \cap [r_2^-, r_2^+)$, and $i = 1, \dots, n$, the topology of the subtree $T_i(r)$ is fixed and independent of r. Specifically, the node set is fixed and each nonnode leaf belongs to its own fixed edge independent of r.]

For each $i = 1, \dots, n$, find all the nodes and the nonnode leaves of $T_i(r)$. A leaf will be characterized by the fixed edge containing it, and its distances from the two nodes of the edge. (These distances are linear functions of r.) One element in the set of all nodes and nonnode leaves of $T_i(r)$ is $u_i(r)$, the closest point to v_1 in $T_i(r)$.

(The total time spent in this step is $O(n^2 \log n)$.)

Step VI: In this step we find a subinterval, $[r_3^-, r_3^+)$, satisfying $r_p^1 \leqslant r_3^+$, such that for each edge of the tree, (v_s, v_t), the ordering of all the leaves of the trees $\{T_i(r)\}$, $i = 1, \dots, n$, in (v_s, v_t) is the same for all values of r in this subinterval. (Each subtree has at most one leaf in (v_s, v_t).)

Let $\{z_i(r, v_s, v_t)\}$, $i = 1, \dots, n$, denote the set of leaves on the edge (v_s, v_t). The leaves can be viewed as linear functions of r. Our goal is to find two consecutive points in the set of intersection points of pairs of functions which bound r_p^1, such that the ordering of all functions is fixed over this interval.

We perform this task simultaneously for all edges (v_s, v_t) of the tree. Specifically, we apply the search technique in [14] to the collection of $O(n^2)$ linear functions, $\{z_i(r, v_s, v_t)\}$, $i = 1, \dots, n$; $(v_s, v_t) \in E$, with the (nonparametric) covering problem algorithm as a solver of the decision problem. Thus, in $O(n^2 \log n)$ time we identify the subinterval $[r_3^-, r_3^+)$, defined above.

To summarize, at the end of this step we have an interval $[r_4^-, r_4^+) = \bigcap_{\{j=1,2,3\}} [r_j^-, r_j^+)$, satisfying $r_p^1 \leqslant r_4^+$, such that for each value of r in the interval, for each $i = 1, \dots, n$, and for each edge (v_s, v_t), the tree $T_i(r)$ is topologically fixed, and the leaves of all trees on (v_s, v_t) have a fixed ordering. Therefore, for each r, satisfying $r_4^- \leqslant r < r_4^+$, the incidence matrix $A = (a_{i,j})$, of the collection of subtrees $\{T_i(r)\}$ and the collection of the closest points to v_1, $\{u_j(r)\}$, is fixed and independent of r. In particular, for each r, satisfying $r_4^- \leqslant r < r_4^+$, the solution value for the respective round-trip covering problem is larger than p.

We conclude that r_p^1, the optimal value of the continuous model, is given by $r_p^1 = r_4^+$. The total time to find r_p^1 with the above algorithm is $O(n^2 \log n)$. Note that for the sake of simplicity we have assumed at the beginning of this section that the set of depots $X = \{x_1, \dots, x_m\}$ is a subset of $V = \{v_1, \dots, v_n\}$. It is easy to see that if we remove this supposition the algorithm can be implemented in $O(m + n^2 \log n)$ time. We summarize with the following theorem.

Theorem 4.2. *The discrete and the continuous round-trip p-center problems on a tree network with n nodes and m depots can be solved in* $O(m + n^2 \log n)$ *time.*

4.3. Locating a single facility on a tree network

In Section 3, we considered the location of a single facility on a general network. We now specialize to the case of tree networks. We use the same notation as in Section 3.

4.3.1. The depot one-way 1-center problem

The 1-center, depot one-way objective, $\hat{G}_2(y)$, is convex on a tree. This property can easily be observed by converting the model to a classical weighted 1-center problem as follows.

At each node $v_i \in V$ augment a new edge to the tree, say (v_i, u_i), of length $d(v_i, x_{k(i)})$. Assign the weight w_i to u_i and a weight of zero to v_i. Now solve the weighted 1-center problem on the augmented tree, using the linear time algorithm of Megiddo [32]. If the unique solution is on an edge of the original tree, it is the optimal solution. Otherwise, due to convexity, if the solution is in some edge (v_i, u_i), the optimal solution to the depot one-way 1-center problem is v_i.

4.3.2. The round-trip 1-center problem

The objective in this case, $\hat{G}_1(y)$, is not convex even on a path (real line), but it is quasi convex (unimodal). (y denotes the location of the center.)

We first show how to solve this problem on a path in linear time. In this case we assume that the nodes are real points, satisfying $v_1 < \cdots < v_n$. For each $i = 1, \ldots, n$, consider the function, defined in Section 3,

$$\hat{S}_i(y) = \min_{x_k \in X^i} \{d(y, v_i) + d(v_i, x_k) + d(x_k, y)\}.$$

Let $x^-_{k(i)}$ be the largest point in X^i which is not larger than v_i, and let $x^+_{k(i)}$ be the smallest point in X^i which is not smaller than v_i. (We assume that the sets $\{x^-_{k(i)}\}$ and $\{x^+_{k(i)}\}$ have already been computed from the sets $\{X^i\}$.) Then clearly,

$$\hat{S}_i(y) = \min\{d(y, v_i) + d(v_i, x^-_{k(i)}) + d(x^-_{k(i)}, y) : d(y, v_i) + d(v_i, x^+_{k(i)}) + d(x^+_{k(i)}, y)\}.$$

It is easy to see that $\hat{S}_i(y)$ is a piecewise linear quasi convex function which has at most 4 breakpoints. Moreover, its slopes are in the set $\{-2, 0, 2\}$, and v_i is one of its global minimum points. Therefore, in constant time $\hat{S}_i(y)$ can be represented as

$$\hat{S}_i(y) = \max\{\hat{S}^-_i(y) : \hat{S}^+_i(y)\},$$

where $\hat{S}^-_i(y)$ and $\hat{S}^+_i(y)$ are piecewise linear with at most 2 breakpoints each. Moreover, $\hat{S}^-_i(y)$ is nonincreasing and $\hat{S}^+_i(y)$ is nondecreasing, and they intersect at v_i. Our objective is to find y^*, a minimizer of the function $\hat{G}_1(y) = \max_{i=1,\ldots,n}\{w_i \hat{S}_i(y)\}$.

Define

$$\hat{G}^-_1(y) = \max_{i=1,\ldots,n} \{w_i \hat{S}^-_i(y)\}$$

and

$$\hat{G}^+_1(y) = \max_{i=1,\ldots,n} \{w_i \hat{S}^+_i(y)\}.$$

Then, $\hat{G}^-_1(y)$ is nonincreasing, $\hat{G}^+_1(y)$ is nondecreasing, and $\hat{G}_1(y) = \max\{\hat{G}^-_1(y) : \hat{G}^+_1(y)\}$. y^* is any intersection point of the functions $\hat{G}^-_1(y)$ and $\hat{G}^+_1(y)$.

Using the above, the depot round-trip 1-center problem on the line can now be formulated as:

$$\begin{aligned} &\min \quad z \\ &\text{s.t.} \\ &\qquad z \geqslant \hat{S}^-_i(y), \quad i = 1, \ldots, n, \\ &\qquad z \geqslant \hat{S}^+_i(y), \quad i = 1, \ldots, n. \end{aligned}$$

Each pair of functions in the collection of the $2n$ monotone and piecewise linear functions $\{\hat{S}^-_i(y)\} \cup \{\hat{S}^+_i(y)\}$ intersects at most at 4 points. Moreover, for any real y we can determine in $O(n)$ time whether $y \geqslant y^*$ or not, since the latter holds if and only if $\hat{G}^+_1(y) \geqslant \hat{G}^-_1(y)$.

Finally, we observe that we have all the necessary ingredients to apply the algorithm in Section 2 of Zemel [44], and solve the model in $O(n)$ time.

Lemma 4.2. *The discrete and the continuous round-trip* 1*-center problems on a path can be solved in* $O(n)$ *time.*

We next show that the problem on a tree can be solved in $O((n + \sum_{i=1}^n |X^i|) \log n)$ time. Moreover, if $X^i = X$ for all $i = 1, \ldots, n$, then the total time reduces to $O(n \log n)$.

We first prove several lemmas.

Lemma 4.3. *Let y represent a point in $A(T)$, and consider the function $\hat{S}_i(y)$, $v_i \in V$, defined on $A(T)$. Let $P(v_i, v_k)$ be a path connecting v_i with some leaf of T, v_k. Then $\hat{S}_i(y)$ is monotone nondecreasing on $P(v_i, v_k)$, attaining its minimum at v_i.*

Proof. Consider a pair of points y_1, y_2 on $P(v_i, v_k)$, where y_1 is on $P(v_1, y_2)$. We need to prove that $\hat{S}_i(y_2) \geqslant \hat{S}_i(y_1)$.
Recall that by definition, for each $y \in A(T)$,

$$\hat{S}_i(y) = \min_{x \in X^i} \{2(d(v_i, y) + d(x, P(v_i, y)))\}.$$

Let $x' \in X^i$ be such that $\hat{S}_i(y_2) = d(y_2, x') + d(x', v_i) + d(v_i, y_2)$, and let x'' be the closest point to x' on $P(v_i, y_2)$.
If x'' is on $P(v_1, y_1)$, then

$$\hat{S}_i(y_2) = 2(d(y_2, v_i) + d(x', x'')) \geqslant 2(d(y_1, v_i) + d(x', x'')) \geqslant \hat{S}_i(y_1).$$

Suppose that x'' is on $P(y_1, y_2)$. Therefore, $d(x', P(v_i, y_1)) = d(y_1, x')$. Then

$$\begin{aligned}\hat{S}_i(y_2) &= 2(d(y_2, v_i) + d(x', x'')) \geqslant 2(d(y_1, v_i) + d(y_1, x'') + d(x', x''))\\ &= 2(d(v_i, y_1) + d(y_1, x')) = 2(d(v_i, y_1) + d(x', P(v_i, y_1))) \geqslant \hat{S}_i(y_1). \quad \square\end{aligned}$$

Lemma 4.4. *Suppose that for $y \in A(T)$ $\hat{G}_1(y) = w_i \hat{S}_i(y)$ for some $i = 1, \ldots, n$. Then for any $z \in A(T)$ such that y is on $P(z, v_i)$ we have $\hat{G}_1(y) \leqslant \hat{G}_1(z)$.*

Proof. From the previous lemma, $w_i \hat{S}_i(v_i) \leqslant w_i \hat{S}_i(y) \leqslant w_i \hat{S}_i(z)$. Hence,

$$\hat{G}_1(y) = w_i \hat{S}_i(y) \leqslant w_i \hat{S}_i(z) \leqslant \hat{G}_1(z). \quad \square$$

The above result suggests the following search algorithm to find y^*, an optimal solution to the depot round trip 1-center problem

Step I: Compute a centroid of T, say v_k.

Step II: Evaluate $G_1(v_k)$. Suppose that $\hat{G}_1(v_k) = w_i \hat{S}_i(v_k)$ for some node v_i. Let F_k be the forest obtained from T by deleting v_k, and let $F_k(i)$ be the component of F_k containing v_i. Then $T_k(i)$, the subtree induced by v_k and $F_k(i)$, contains an optimal 1-center.

Step III: If $T_k(i)$ is an edge go to Step IV. Otherwise, continue the search in $T_k(i)$.

The above approach, based on centroid decomposition, will locate an edge containing an optimal 1-center in $O(n + Q(n) \log n)$ time, where $Q(n)$ is the time needed to evaluate the function $\hat{G}_1(y)$ at a point in $A(T)$. (The discrete solution is one of the nodes of this edge.) Finally, to identify an optimal 1-center on an edge we can almost mimic the above approach for solving the problem on a path.

Step IV: Finding a 1-center on an edge.

We have assumed without loss of generality that $X \subseteq V$. Hence, we can assume that the nodes of the edge are v_1 and v_2, and there are no points of X in the interior of the edge.

To express the objective $\hat{G}_1(y)$ in this case, let $V^1 = \{v_t : v_1 \in P(v_t, v_2)\}$ and $V^2 = \{v_t : v_2 \in P(v_t, v_1)\}$. Consider a node $v_i \in V^1$. Let $x^1_{j(i)}$ be the closest node in $X^i \cap V^1$ to $P(v_i, v_1)$, and let, $x^2_{j(i)}$ be the closest node in $X^i \cap V^2$ to v_1.

Then, when y is restricted to the above edge,

$$w_i \hat{S}_1(y) = 2w_i \min\{d(x^1_{j(i)}, P(v_i, v_1)) + d(v_i, v_1) + y; d(v_i, x^2_{j(i)})\}.$$

Thus, $w_i \hat{S}_1(y)$ is a piecewise concave function with slopes in $\{0, 2w_i\}$. A similar expression is obtained if $v_i \in V^2$.

We conclude that $\hat{G}_1(y)$ is the upper envelope of n piecewise linear concave functions, each having at most one breakpoint. Moreover, when the sets $\{x^1_{j(i)}\}$ and $\{x^2_{j(i)}\}$ are available the function $\hat{G}_1(y)$, (and in particular its minimum), can be computed in $O(n \log n)$ time, [22]. (We later explain how to compute these sets of points.)

To determine the complexity of the above algorithm we next show that for any $y \in A(T)$, $\hat{G}_1(y)$ can be evaluated in $O(n + \sum_{i=1}^{n} |X^i|)$ time. Moreover, if $X^i = X$ for all $i = 1, \ldots, n$, the effort reduces to $O(n)$.

Evaluating $\hat{G}_1(y)$: As in Section 4.1, we may assume without loss of generality that T is a binary tree rooted at y. For each node v_i, we let V_i denote the set of descendants of v_i. $v_{p(j)}$ will denote the father of v_j, i.e., the closest node $v_k \neq v_i$ to v_i on the $P(v_i, y)$.

Case I: $X^i \subseteq X$, for $i = 1, \dots, n$.

In this case we need to compute $w_i \hat{S}_i(y) = 2w_i d(v_i, y) + 2w_i \min_{x \in X^i}\{d(x, P(v_i, y))\}$, for all $i = 1, \dots, n$. To show that the total time needed to perform this task is $O(n + \sum_{i=1}^{n} |X^i|)$, we prove that after spending $O(n)$ time on preprocessing, it takes constant time to compute $d(x, P(v_i, y))$ for any given pair x and v_i.

We first note that if $z \in A(T)$ is the nearest common ancestor of x and v_i on the rooted tree, then $d(x, P(v_i, y)) = d(x, z) = d(x, y) - d(z, y)$. The data structures presented in [21] enable us to find a nearest common ancestor in constant time, after spending $O(n)$ time on preprocessing the tree.

Case II: $X^i = X$, for $i = 1, \dots, n$.

In this case we may assume without loss of generality that no edge contains more than 2 points of X. Thus, we assume that $X \subseteq V$. To evaluate $\hat{G}_1(y)$, we need to compute $w_i \hat{S}_i(y) = 2w_i d(v_i, y) + 2w_i \min_{x \in X}\{d(x, P(v_i, y))\}$, for all $i = 1, \dots, n$.

For each node v_i, let $x^+_{j(i)}$ denote a closest point to v_i in $X \cap V_i$ and let and $x^-_{j(i)}$ denote a closest point to $P(v_i, y)$ in $X \cap (V \backslash V_i)$. Then

$$\min_{x \in X}\{d(x, P(v_i, y))\} = \min\{d(x^+_{j(i)}, v_i), d(x^-_{j(i)}, P(v_i, y))\}.$$

Thus, it is sufficient to compute the points $\{x^+_{j(i)}\}$ and $\{x^-_{j(i)}\}$. The latter task can be performed in $O(n)$ time. First use a bottom-up algorithm, initiated at the leaves of T, to compute the set $\{x^+_{j(i)}\}$. Then, using $\{x^+_{j(i)}\}$, start at the root y, and move top-down towards the leaves to compute the set $\{x^-_{j(i)}\}$.

We note in passing that the procedures defined in Cases I and II can also be used to compute the sets of points $\{x^1_{j(i)}\}$ and $\{x^2_{j(i)}\}$, defined in Step IV of the algorithm. Thus, we conclude with the following theorem.

Theorem 4.3. *The discrete and the continuous round-trip* 1*-center problems on a tree network with n nodes can be solved in* $O((n + \sum_{i=1}^{n} |X^i|) \log n)$ *time. Moreover, if* $X^i = X$ *for all* $i = 1, \dots, n$, *and* $|X| = m$, *the complexity reduces to* $O(m + n \log n)$.

4.3.3. The customer one-way 1-center problem

Unlike the round-trip 1-center problem, the customer one-way version objective does not satisfy any quasi-convexity property even on the real line. Thus, the best solution approach we can offer now is based on optimizing independently on each edge of the given tree. To solve the problem on an edge, we can basically follow the approach shown in Step IV of the algorithm given in the previous section. For the sake of brevity, we omit the details, and state the results only.

Theorem 4.4. *The discrete and the continuous customer one-way* 1*-center problems on a tree network with n nodes can be solved in* $O(n\sum_{i=1}^{n} |X^i|)$ *and* $O(n\sum_{i=1}^{n} |X^i| + n^2 \log n)$ *times, respectively. Moreover, if* $X^i = X$ *for all* $i = 1, \dots, n$, *and* $|X| = m$, *the complexities reduce to* $O(m + n^2)$ *and* $O(m + n^2 \log n)$, *respectively.*

5. Questions, comments and concluding remarks:

General networks: In Section 2 we have noted that 3-approximation polynomial algorithms for the one-way and round-trip p-center problems in general networks, can be obtained by reducing them to the p-suppliers problem. It is known [23,24], that the constant 3 is best possible even for the unweighted general p-suppliers problem. Thus, the question is whether the constant 3 is also best possible for our special instances of the p-suppliers model. Achieving a constant which is strictly smaller than 2 is certainly NP-hard for our models, since the constant 2 is best possible for the classical unweighted p-center problem on general graphs [23,24], and even for planar rectilinear instances [28]. 2-approximation algorithms for the classical weighted, discrete and continuous p-center problem on general graphs are given in [37,38].

As in the classical p-center problem, we can consider a generalization of the discrete version of our one-way and round-trip problems, where there are setup costs for the facilities depending on the location of the servers. (A special case of this generalization corresponds to the case where servers can be established only at some proper subset V' of V. This specialization is already strongly NP-hard for the round-trip problem on star trees [30].) The constant factor approximation schemes of Section 2 can be extended to the case with setup costs. We only need to use the approximation algorithms for the p-suppliers problems with setup costs in [9,27,43].

Extensions to R^d: An interesting topic to study is the extension of the results in Section 3.2 on the planar case to R^d, for $d > 2$.

Tree networks: There are several interesting questions related to the results in Section 4 on tree networks. The first one is the possible existence of a subquadratic algorithm for the round-trip covering problem on trees.

We have shown that the customer one-way problem on a path network is NP-hard even for a collection $\{X^i\}$, satisfying $|X^i| = 2$, for all $v_i \in V$, and polynomially solvable in the case when $X^i = X$ for all $i = 1, \dots, n$. It is not known whether the customer one-way problem on general tree networks is polynomially solvable if $X^i = X$ for all $i = 1, \dots, n$.

Finally, we note that the results in Sections 3 and 4 can be generalized to the following doubly weighted model. Suppose that each pair (v_i, x_k), where $v_i \in V$ and $x_k \in X$ is associated with a nonnegative weight $w_{i,k}$. The objective of the round-trip p-center problem is to find a subset Y of the metric space of cardinality p, minimizing $G_1(Y) = \max_{i=1,\dots,n}\{S_i(Y)\}$, where

$$S_i(Y) = \min_{y \in Y, x_k \in X}\{w_{i,k}(d(y, v_i) + d(v_i, x_k) + d(x_k, y))\}.$$

The respective one-way models are similarly defined.

References

[1] P.K. Agarwal, B. Aronov, M. Sharir, S. Suri, Selecting distances in the plane, Algorithmica 9 (1993) 495–514.
[2] P.K. Agarwal, M. Sharir, S. Toledo, Applications of parametric searching in geometric optimization, J. Algorithms 17 (1994) 292–318.
[3] A.A. Ageev, Graphs, matrices and an elementary location problem, Upravlyaemye Sistemy 29 (1989) 3–10 [83. 90B80 (90C10), MR1110574] (in Russian).
[4] A.A. Ageev, Private Communication, 2004.
[5] A.A. Ageev, V.L. Beresnev, Minimization algorithms for some classes of polynomials in Boolean variables, Trudy Inst. Mat. 10 (1988) 5–17 (in Russian).
[6] A.A. Ageev, V.L. Beresnev, Polynomially solvable cases of the simple plant location problem, in: Proceedings of the First Conference on Integer Programming and Combinatorial Optimization (IPCO), Waterloo, Canada, 1990, pp. 1–6.
[7] V.L. Beresnev, Algorithms for the minimization of polynomials from Boolean variables, Problemy Kibernet. 36 (1979) 225–246, 280 (Reviewer: C. Ilioi) [65K10 (90C09) MR0568455 (83k:65053)] (in Russian).
[8] O. Berman, Z. Drezner, G.O. Wesolowsky, The collection depots location problem on networks, Naval Res. Logist. 49 (2002) 15–24.
[9] R. Bhatia, S. Guha, S. Khuller, Y.J. Sussmann, Facility location with dynamic distance functions, J. Combin. Optim. 2 (1998) 199–217.
[10] A. Chan, R.L. Francis, A round-trip location problem on a tree graph, Transportation Sci. 10 (1976) 35–51.
[11] A. Chan, D.W. Hearn, A rectilinear distance round-trip location problem, Transportation Sci. 11 (1977) 107–123.
[12] I.P. Chew, D. Dor, A. Efrat, K. Kedem, Geometric pattern matching in d-dimensional space, Discrete Comput. Geom. 21 (1999) 257–274.
[13] I.P. Chew, K. Kedem, Improvements on geometric pattern matching problems, in: O. Nurmi, E. Ukkonen (Eds.), Scandinavian Workshop on Algorithm Theory-SWAT 1992, Lecture Notes in Computer Science, vol. 621, Springer, Berlin, July 1992, pp. 318–325.
[14] R. Cole, Slowing down sorting networks to obtain faster sorting algorithms, J. Assoc. Comput. Mach. 34 (1987) 200–208.
[15] Z. Drezner, G. Steiner, G.O. Wesolowsky, One-facility location with rectilinear tour distances, Naval Res. Logist. Quart. 32 (1985) 391–405.
[16] M. Dyer, A class of convex programs with applications to computational geometry, Technical Report, University of Leeds, 1991.
[17] H. Edelsbrunner, L. Guibas, J. Pach, R. Pollack, R. Seidel, M. Sharir, Arrangements of curves in the plane-topology, combinatorics, and algorithms, Theoret. Comput. Sci. 92 (1992) 319–336.
[18] G.N. Frederickson, D.B. Johnson, The complexity of selection and ranking in $X + Y$ and matrices with sorted columns, J. Comput. System Sci. 24 (1982) 197–208.
[19] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph, SIAM J. Comput. 1 (1971) 180–187.
[20] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[21] D. Harel, R.E. Tarjan, Fast algorithms for finding nearest common ancestors, SIAM J. Comput. 13 (1984) 338–355.
[22] J. Hershberger, Finding the upper envelope of n line segments in $O(n \log n)$ time, Inform. Process. Lett. 33 (1989) 169–174.
[23] D.S. Hochbaum, D.B. Shmoys, A best possible approximation algorithm for the K-center problem, Math. Oper. Res. 10 (1985) 180–184.
[24] D.S. Hochbaum, D.B. Shmoys, A unified approach to approximation algorithms for bottleneck problems, J. Assoc. Comput. Mach. 33 (1986) 533–550.
[25] O. Kariv, S.L. Hakimi, An algorithmic approach to network location problems, Part 1: the p-centers, SIAM J. Appl. Math. 37 (1979) 539–560.
[26] K. Kedem, R. Livne, J. Pach, M. Sharir, On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles, Discrete Comput. Geom. 1 (1986) 59–71.
[27] S. Khuller, R. Pless, Y.J. Sussmann, Fault tolerant K-center problems, Theoret. Comput. Sci. 242 (2000) 237–245.
[28] M.T. Ko, R.C. Lee, J.S. Chang, An optimal approximation algorithm for the rectilinear m-center problem, Algorithmica 5 (1990) 341–352.
[29] A. Kolen, The round trip p-center and covering problem on a tree, Transportation Sci. 19 (1985) 222–234.
[30] A. Kolen, A. Tamir, Covering problems, in: P.B. Mirchandani, R.L. Francis (Eds.), Discrete Location Theory, Wiley, New York, 1990(Chapter 6).
[31] N. Megiddo, Combinatorial optimization with rational objective functions, Math. Oper. Res. 4 (1979) 414–424.
[32] N. Megiddo, Applying parallel computation algorithms in the design of serial algorithms, J. Assoc. Comput. Mach. 30 (1983) 852–865.
[33] N. Megiddo, Linear programming in linear time when the dimension is fixed, J. Assoc. Comput. Mach. 31 (1984) 114–127.
[34] N. Megiddo, A. Tamir, New results on p-center problems, SIAM J. Comput. 12 (1983) 751–758.

[35] N. Megiddo, A. Tamir, E. Zemel, R. Chandrasekarn, An $O(n \log^2 n)$ algorithm for the kth longest path in a tree with applications to location problems, SIAM J. Comput. 10 (1981) 328–337.
[36] M.H. Overmars, C.-K. Yap, New upper bounds in Klee's measure problem, SIAM J. Comput. 20 (1991) 1034–1045.
[37] J. Plesnik, A heuristic for the p-center problem in graphs, Discrete Appl. Math. 17 (1987) 263–268.
[38] J. Plesnik, Two heuristics for the absolute p-center problem in graphs, Math. Slovaca 38 (1988) 227–233.
[39] D.J. Rose, R.E. Tarjan, G.S. Lueker, Algorithmic aspects of vertex elimination in graphs, SIAM J. Comput. 5 (1976) 266–281.
[40] M. Sharir, P.K. Agarwal, Davenport–Schinzel Sequences and their Geometric Applications, Cambridge University Press, Cambridge, 1995.
[41] A. Tamir, Improved complexity bounds for center location problems on networks by using dynamic data structures, SIAM J. Discrete Math. 1 (1988) 377–396.
[42] A. Tamir, An $O(pn^2)$ algorithm for the p-median and related problems on tree graphs, Oper. Res. Lett. 19 (1996) 59–64.
[43] Q. Wang, K.H. Cheng, A heuristic for the k-center problem with costs and usage weights, in: Proceedings of the 28th Annual Allerton Conference on Communication, Control, and Computing, 1990, pp. 324–333.
[44] E. Zemel, An $O(n)$ algorithm for the linear multiple choice problem and related problems, Inform. Process. Lett. 18 (1984) 123–128.