TOP manuscript No.
(will be inserted by the editor)

A Maximum Trip Covering Location Problem with an Alternative Mode of Transportation on Tree Networks and Segments

Mark-Christoph Körner · Juan A. Mesa · Federico Perea · Anita Schöbel · Daniel Scholz

January 31, 2012

Abstract In this paper the following facility location problem in a mixed planar-network space is considered: We assume that traveling along a given network is faster than traveling within the plane according to the Euclidean distance. A pair of points (A_i, A_j) is called covered if the time to access the network from A_i plus the time for traveling along the network plus the time for reaching A_j is lower than, or equal to, a given acceptance level related to the travel time without using the network. The objective is to find facilities (i.e. entry and exit points) on the network that maximize the number of covered pairs. We present a reformulation of the problem using convex covering sets and use this formulation to derive a finite dominating set and an algorithm for locating two facilities on a tree network. Moreover, we adapt a geometric branch & bound approach to the discrete nature of the problem and suggest a procedure for locating more than two facilities on a single line, which is evaluated numerically.

Keywords Location · Covering Problem · Transportation

1 Introduction

Covering problems constitute one of the main branches of Locational Analysis. A maximal covering location problem consists of locating a fixed number of servers or facilities so that the number of given (demand) points for which the facilities are useful is maximized. In this case a point is said to be covered by the facility if the point is located within a given threshold distance or time from the facility. Covering location problems have been studied in continuous and discrete spaces (see [22] and [13] for continuous and discrete settings, respectively).

We generalize covering location problems from two points of view: First, we deal with a mixed planar-network space, and second we do not cover points, but origin-destination pairs (O/D pairs) which are *pairs* of points.

Our first generalization takes into account that the underlying space of the problem is often in practise not homogeneous in terms of spent time, meaning that there are some structures where the speed is higher than in others (ADSL or high-speed railways are examples in the telecommunication and transportation fields, respectively). Secondly, service is not completed only by reaching the facility but also by traveling along the high-speed network and then acceding to another point. We hence assume that the demand to be covered is given by a set of O/D pairs, that is, pairs of points (A_i, A_j) with some traffic t_{ij} between them.

In this paper the goal is to locate several facilities on a given high-speed network that can be used as entry and/or exit points for traveling between the given O/D pairs. We assume three possibilities for

Mark-Christoph Körner, Anita Schöbel and Daniel Scholz
Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Germany

Juan A. Mesa (Corresponding Author)
Departmento de Matemática Aplicada II, Universidad de Sevilla, Spain
Tel.: +34 954486174, Fax: +34 954486165, E-mail: jmesa@us.es

Federico Perea
Departmento de Estadística e Investigación Operativa Aplicadas y Calidad, Universitat Politècnica de València, Spain

traveling: across the plane without using the high-speed network, using the high-speed network only, or by a combination of both transportation systems. Our aim is to maximize the performance of the high-speed network. More precisely, we want to locate the entry and exit points so that it becomes attractive for as much traffic as possible to use the high-speed network instead of the planar direct path.

As an example of this situation in a railway transportation setting, consider Figure 1, which is based on a real section of the Spanish high-speed network. There are two main cities, *Sevilla* and *Córdoba*, linked by a high-speed line. In the geographical area between both cities, there is a number of smaller cities that do not have easy access to this high-speed train because currently there are no intermediate stops. They are linked by a slower network which is dense enough to be roughly approximated by the Euclidean distance. In order to attract more passengers the company operating this high-speed system is thinking about locating two stations in between so that as many potential passengers as possible will use the high-speed line. For instance, in order to travel from *Brenes* to *La Carlota*, assuming the two new stations are located where the big dots are, one could go from *Brenes* to the high-speed train station, take the high-speed line from there to the next station, and finish the journey to *La Carlota* from there by means of the slower network (as depicted in Figure 1). We assume that users will take this trip instead of directly going from one town to the other by the slower network, if the time of this journey is lower than a given threshold (which could be the time of the same journey by the slower network). The idea is therefore to locate two stations so that as many potential passengers as possible will find that traveling by the high-speed line is faster than traveling on the alternative network.

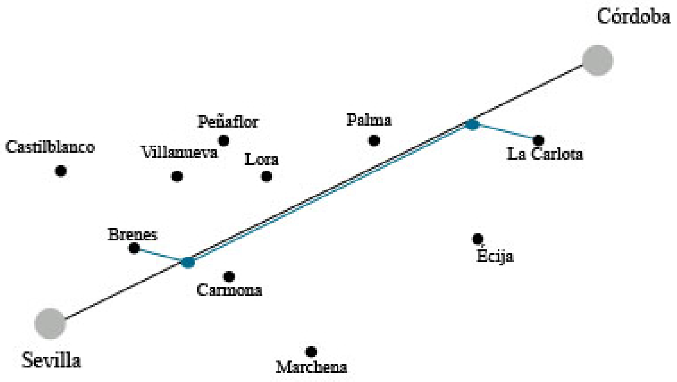

Fig. 1 Illustration of the transportation application..

There is some literature related to our two aspects of generalization: The *maximal trip covering problem* was introduced in [16] regarding the location of a rapid transit alignment and searching for a location which covers as many O/D pairs as possible. In this paper it was noted that in real-world situations high-speed lines compete with other modes of transportation and therefore the real coverage of a high-speed network depends on the performance of its lines in comparison with other modes. In [15] and [17], different models for the railway network design problem in the presence of a competing mode are studied.

There are also some recent papers in which traveling distances are a combination of planar and network distances ([7], [14], [20]). Their goal is to locate one or several points so that the sum of all combined distances from the given points is minimized, i.e. these studies deal with the well-known Weber problem under a new metric. In contrast to these papers our goal is to cover O/D-pairs. Some efforts have recently been made to determine the *City Voronoi diagram* for a set of points in the context of combination of the planar distance l_1 with the network distance ([2], [3], [9]). In these models travelers can enter the network at any point. The same holds for the models considered in [6] for locating a high-speed transportation device and in [1], which deals with the Voronoi diagram for the heavy luggage metric.

The remainder of this paper is structured as follows. We first present a formal description of our model and introduce the notation needed. In Section 3 we develop a solution approach for the case of two facilities both in segments and tree networks. In the case of locating more than two facilities on a single straight line we suggest a geometric branch & bound approach in Section 4. Although this is a methodological paper rather than a computational one, Section 5 is devoted to showing some numerical results. The paper finishes with some conclusions and hints for further research.

2 Problem Formulation and Notation

Our problem makes use of the following input data and notation:

- Let $\mathcal{A} = \{A_i = (a_i, b_i) \in \mathbb{R}^2 \ : \ i = 1, \dots, n\}$ be a set of n points on the plane. In practical applications $\mathcal{A}$ might be a set of users, towns, cities, or centroid points of transportation zones in an urban area.
- We assume that the time for traveling between two points in the plane can be estimated by a metric. In this paper we use the Euclidean metric.
- Let $T = (t_{ij}) \in \mathbb{R}^{n \times n}$ be a matrix in which trip patterns are codified, i.e. t_{ij} is the weight of the ordered pair (A_i, A_j). Note that t_{ij} is a parameter known a priori. In the transportation case, t_{ij} can be seen as the number of trips from an origin i to a destination j. In the telecommunication setting, t_{ij} could represent the amount of data transferred from server i to server j.
- Let $\mathcal{N}(V, E)$ be the network representing the high-speed system. Each vertex $v \in V$ represents a junction or a node, and each edge $e \in E$ has a length l_e and is assumed to be rectifiable. We suppose that $\mathcal{N}(V, E)$ is embedded in the Euclidean plane. Let $\mathcal{N}$ be the continuum set of points on the edges. The edge lengths induce a distance function d on $\mathcal{N}$. For any two points $x, y \in \mathcal{N}$, $d(x, y)$ is the length of any shortest path connecting x and y.
- For any two points $x, y \in \mathcal{N}$ let us define the high-speed distance $d_{\mathcal{N}}(x, y) = \alpha d(x, y), \alpha \in (0, 1)$. It is straightforward that $\mathcal{N}$ is a metric space with this distance. Parameter α is a speed factor: the lower α, the faster is traveling on $\mathcal{N}$ with respect to traveling on the plane.
- Let $D = (d_{ij}) \in \mathbb{R}^{n \times n}$ be a (symmetric) matrix with $0 \le d_{ij} < \|A_i - A_j\|_2$. These values represent *acceptance levels* for using the network, meaning that the O/D pair (i, j) would choose the high-speed network if and only if the traveling time by using it is less than or equal to d_{ij}.

We aim to locate m facilities $\mathcal{X} = \{X_1, \dots, X_m\} \subset \mathcal{N} \subset \mathbb{R}^2$ so that the use of the high-speed network is maximized.

Consider an O/D pair (i, j). If the set of facilities $\mathcal{X} = \{X_1, \dots, X_m\}$ is given, a user of the high-speed network $\mathcal{N}$ has to choose an entry point $X_k \in \mathcal{X}$ and an exit point $X_l \in \mathcal{X}$. For methodological purposes, from now on we will consider $\mathcal{X}$ as an ordered set, therefore denoted by $\mathcal{X} = (X_1, ..., X_m)$ (for convenience also referred to as *m-facility point*). The length of the path (A_i, X_k, X_l, A_j) is denoted by $h_{ij}^{kl}(\mathcal{X})$ and equal to

$$h_{ij}^{kl} := h_{ij}^{kl}(\mathcal{X}) \ := \ \|A_i - X_k\|_2 + d_{\mathcal{N}}(X_k, X_l) + \|X_l - A_j\|_2.$$

We will propose two ways to calculate the traveling time through the network, because a user may prefer to access and exit the high-speed line either via the nearest facilities or via those that minimize the overall distance. Whether a user will utilize the network $\mathcal{N}$ or not may depend on this choice.

- **Minimal Distance to the Facilities.**
  First, we assume that users enter the high-speed network at the closest facility to their origin and exit at the closest facility to their destination. Let

  $$P_{\mathcal{X}}(A_i) \ := \ \left\{ X \in \mathcal{X} \ : \ X = \arg \min_{X \in \mathcal{X}} \{\|A_i - X\|_2\} \right\}$$

  be the set of the closest facilities to the point A_i. (Note that usually $P_{\mathcal{X}}(A_i)$ contains only one facility.) The distance from A_i to A_j is then defined as

  $$f_{ij}^1(\mathcal{X}) \ := \ \min_{\substack{X_k \in P_{\mathcal{X}}(A_i) \\ X_l \in P_{\mathcal{X}}(A_j)}} \{\|A_i - X_k\|_2 + d_{\mathcal{N}}(X_k, X_l) + \|X_l - A_j\|_2\} \ = \ \min_{\substack{X_k \in P_{\mathcal{X}}(A_i) \\ X_l \in P_{\mathcal{X}}(A_j)}} h_{ij}^{kl}. \qquad (1)$$

  Remark 1 In the definition of $P_{\mathcal{X}}(A_i)$ we only take into account the facility or facilities that exactly yield the minimum distance to A_i. Therefore, we do not consider facilities that are further from A_i than the minimum by an ε factor. In a passenger transportation framework this is not realistic, as passengers sometimes use another more distant station if reaching the said takes only, say, 15 seconds more than the closest one. This could be solved by alternatively defining

$$P_{\mathcal{X}}(A_i) \ := \ \left\{ X \in \mathcal{X} \ : \|A_i - X\|_2 \leq (1+\varepsilon)\|A_i - X^*\|_2, \ \varepsilon > 0, \ X^* = \arg\min_{X\in\mathcal{X}}\{\|A_i - X\|_2\} \right\}.$$

In this case the assertion that *usually $P_{\mathcal{X}}(A_i)$ contains only one facility* is not necessarily true.

- **Minimal Overall Distance.**
  In this case users choose their entry and exit points in such a way that the overall distance of their trip is minimized. Therefore, the distance from A_i to A_j using the network is

$$f^2_{ij}(\mathcal{X}) \ := \min_{X_k, X_l \in \mathcal{X}} \{\|A_i - X_k\|_2 + d_{\mathcal{N}}(X_k, X_l) + \|X_l - A_j\|_2\} \ = \min_{X_k, X_l \in \mathcal{X}} h^{kl}_{ij}. \tag{2}$$

Note that $f^2_{ij}(\mathcal{X}) \leq f^1_{ij}(\mathcal{X})$ for any $\mathcal{X}$. The acceptance levels d_{ij} lead to the following definition.

Definition 1 Given d_{ij} with $0 \leq d_{ij} < \|A_i - A_j\|_2$, the O/D pair (i,j) is *covered* by the facilities $\mathcal{X}$ with respect to function f^r_{ij}, $\ r = 1, 2$, if

$$f^r_{ij}(\mathcal{X}) \ \leq \ d_{ij}.$$

Note that this definition leads to different conditions for the two cases $r = 1$ and $r = 2$, because we can only assure that $f^2_{ij}(\mathcal{X}) \leq f^1_{ij}(\mathcal{X})$. We want to maximize the number of covered pairs given as

$$F^r(\mathcal{X}) \ = \sum_{(i,j): f^r_{ij} \leq d_{ij}} t_{ij}, \quad r \in \{1, 2\}.$$

Summarizing, the problem considered in this paper can now be stated:

OD-pair location problem. Given a high-speed network $\mathcal{N}$, demand points $\mathcal{A}$, function f^r with $r \in \{1, 2\}$, trip patterns T and acceptance levels D, find an m-facility point $\mathcal{X} = (X_1, \ldots, X_m) \in \mathcal{N}^m$ such that the number $F^r(\mathcal{X})$ of covered trips is maximized. In short,

$$\max_{\mathcal{X} \in \mathcal{N}^m} \ F^r(\mathcal{X}). \tag{3}$$

This problem always has an optimal solution, since the feasible region is non-empty and the objective function can only take a finite number of finite values.

Remark 2 If $d_{ij} = d_{ji}$ we can assume $t_{ij} = 0$ for $i \geq j$ since both the Euclidean and the network distances are metrics. The reason is that an O/D pair (i,j) is covered by $\mathcal{X}$ if and only if (j,i) is covered by $\mathcal{X}$. We then use the matrix $T' = (t'_{ij})$ with $t'_{ij} = t_{ij} + t_{ji}$ for $i < j$ and $t'_{ij} = 0$ for $i \geq j$, instead of T.

We conclude this section by providing some special properties for the case where the network $\mathcal{N}$ is a straight line segment. Note that in this case we can assume that $d_{\mathcal{N}}(X_k, X_l) = \alpha \cdot \|X_k - X_l\|_2$ for every pair of facilities $X_k, X_l \in \mathcal{L}$. For this case, given an O/D pair (i,j) and two facilities X_k and X_l we will show that it is possible to predict whether h^{kl}_{ij} or h^{lk}_{ij} applies (see Theorem 1), that is, which will be the entry facility and which will be the exit facility.

In order to build easy-to-read proofs, we assume without loss of generality the line segment $\mathcal{L} = [0, L] \times \{0\}$ (otherwise rotate, translate, and scale the coordinate system). Only for simplicity we sometimes write $\mathcal{L} = [0, L]$ although $\mathcal{L} \subset \mathbb{R}^2$.

For each point $A_i = (a_i, b_i) \in \mathcal{A}$ we assume that $0 \leq a_i \leq L$ and we define its projection on the line segment $\mathcal{L}$ as

$$\mathcal{P}_{\mathcal{L}}(A_i) \ = \ \arg\min_{X \in \mathcal{L}} \|A_i - X\|_2 \ \in \ \mathcal{L}.$$

Note that $\mathcal{P}_{\mathcal{L}}(A_i)$ is a single point since the Euclidean norm is strictly convex and $\mathcal{L}$ is a segment.

In the next theorem we prove that if A_i is left of A_j and the O/D pair (i,j) is covered by $\mathcal{X} = (X_k, X_l)$, then the entry facility X_k must be left of the exit facility X_l.

Theorem 1 *Let $\mathcal{L}$ be a line segment, $\mathcal{X} = (X_1, \ldots, X_m)$ an m-facility point located on $\mathcal{L} = [0, L]$, and $\mathcal{A} = \{A_i = (a_i, b_i) : \ i = 1, \ldots, n\}$ a set of points such that $a_i \leq a_j$ for all $i \leq j$.*

Then, for any O/D pair (i,j) with $i < j$ and any pair of facilities $X_k = (x_k, 0), X_l = (x_l, 0) \in \mathcal{X}$ satisfying $h^{kl}_{ij} \leq d_{ij}$ we have that $\mathcal{P}_{\mathcal{L}}(A_i) \neq \mathcal{P}_{\mathcal{L}}(A_j)$. Besides, we have that $x_k < x_l$.

Proof First, assume that $\mathcal{P}_{\mathcal{L}}(A_i) = \mathcal{P}_{\mathcal{L}}(A_j) = S \in \mathcal{L}$. Then we obtain that

$$\begin{aligned} h_{ij}^{kl} &= \|A_i - X_k\|_2 + \alpha\|X_k - X_l\|_2 + \|X_l - A_j\|_2 \\ &\geq \|A_i - S\|_2 + \|S - A_j\|_2 \\ &\geq \|A_i - A_j\|_2 > d_{ij}. \end{aligned}$$

Hence, $h_{ij}^{kl} \leq d_{ij}$ implies $\mathcal{P}_{\mathcal{L}}(A_i) \neq \mathcal{P}_{\mathcal{L}}(A_j)$.

By contradiction, assume that $x_l \leq x_k$. Since $\mathcal{P}_{\mathcal{L}}(A_i) \neq \mathcal{P}_{\mathcal{L}}(A_j)$ there exists $X = (x, 0) \in \mathcal{L}$ such that $\mathcal{P}_{\mathcal{L}}(A_i)$ lies on the left of X and $\mathcal{P}_{\mathcal{L}}(A_j)$ lies on the right of X. The four possible cases can be distinguished: $x_l \leq x_k < x$, $x < x_l \leq x_k$, $x_l < x < x_k$, and $x_l = x_k$. In the first case $\|X_l - A_j\|_2 \geq \|X_k - A_j\|_2$. Hence

$$\begin{aligned} h_{ij}^{kl} &= \|A_i - X_k\|_2 + \alpha\|X_k - X_l\|_2 + \|X_l - A_j\|_2 \\ &\geq \|A_i - X_k\|_2 + \|X_k - A_j\|_2 \geq \|A_i - A_j\|_2 > d_{ij}, \end{aligned}$$

which contradicts the assumption $h_{ij}^{kl} \leq d_{ij}$. Analogously, in all remaining cases we obtain a contradiction with the assumption $h_{ij}^{kl} \leq d_{ij}$. Hence, neither $x_l < x_k$ nor $x_l = x_k$ can be true. We obtain $x_k < x_l$. □

If we assume that the optimal m-facility point $\mathcal{X} = (X_1, \ldots, X_m) \in \mathcal{L}^m$ with $X_i = (x_i, 0)$ satisfies $x_1 < \ldots < x_m$, we can reformulate Theorem 1 as follows:

$$h_{ij}^{kl}(\mathcal{X}) \leq d_{ij} \text{ and } i < j \Rightarrow k < l.$$

Under this assumption, and for each $(i,j) : i < j$ and $r \in \{1, 2\}$, we can simplify the calculation of f_{ij}^r since only h_{ij}^{kl} or h_{ij}^{lk} has to be considered, namely that with $k < l$, as stated in the following corollary.

Corollary 1 *Assume the points $\mathcal{A} = \{A_i = (a_i, b_i) : i = 1, \ldots, n\}$ such that $a_i \leq a_j$ for all $i \leq j$ and let $\mathcal{X} = (X_1, \ldots, X_m) \in \mathcal{L}^m$ with $X_i = (x_i, 0)$ and $x_1 < \ldots < x_m$. Then*

$$f_{ij}^1(\mathcal{X}) = \min_{\substack{X_k \in P_{\mathcal{X}}(A_i) \\ X_l \in P_{\mathcal{X}}(A_j) \\ k<l}} h_{ij}^{kl}(\mathcal{X}) \qquad \text{and} \qquad f_{ij}^2(\mathcal{X}) = \min_{k<l} h_{ij}^{kl}(\mathcal{X}).$$

Corollary 1 is particularly helpful for the case $m = 2$, which is done next.

3 The Level Set Covering Method for $m = 2$ facilities

In this section we will consider two cases: First, we analyze the case where the two facilities are to be located on the same edge of the network, which is equivalent to locating the facilities on a straight-line segment. We will then extend the approach to locating two facilities on a tree network. Note that this method cannot be directly extended to a general network since the distance between pairs of points on a general network is not a convex function. We start by providing some general results for the case of $m = 2$ new facilities.

If only two facilities X_1 and X_2 are to be located on $\mathcal{N}$, the following lemma shows the analogies between distances f_{ij}^1 and f_{ij}^2, as defined in Equations (1) and (2), respectively.

Lemma 1 *Let $\mathcal{X} = (X_1, X_2) \in \mathcal{N}^2$. Then the O/D pair (i,j) is covered using $f_{ij}^1(\mathcal{X})$ if and only if (i,j) is covered using $f_{ij}^2(\mathcal{X})$. Besides, if (i,j) is covered, then $f_{ij}^1(\mathcal{X}) = f_{ij}^2(\mathcal{X})$.*

Proof The proof is divided into two cases.

- Suppose (i,j) is covered using f_{ij}^1. Since $f_{ij}^2(\mathcal{X}) \leq f_{ij}^1(\mathcal{X})$, we have that $f_{ij}^2(\mathcal{X}) \leq f_{ij}^1(\mathcal{X}) \leq d_{ij}$. Hence, (i,j) is also covered using f_{ij}^2.
- Suppose (i,j) is covered using f_{ij}^2. Without loss of generality assume that $f_{ij}^2(\mathcal{X}) = h_{ij}^{12}(\mathcal{X})$, otherwise rename the facilities.
  If $X_1 \in P_{\mathcal{X}}(A_i)$ and $X_2 \in P_{\mathcal{X}}(A_j)$, then we have $f_{ij}^1(\mathcal{X}) \leq f_{ij}^2(\mathcal{X})$, hence (i,j) is covered using f_{ij}^1 and $f_{ij}^2(\mathcal{X}) = f_{ij}^1(\mathcal{X})$. We will see that other cases are not possible.

Let us assume that at least one of the inclusions $X_1 \in P_{\mathcal{X}}(A_i)$ or $X_2 \in P_{\mathcal{X}}(A_j)$ is not true, say $X_1 \notin P_{\mathcal{X}}(A_i)$. Then $\{X_2\} = P_{\mathcal{X}}(A_i)$, i.e. $\|A_i - X_2\|_2 < \|A_i - X_1\|_2$, and therefore

$$\begin{aligned} h_{ij}^{12} &= \|A_i - X_1\|_2 + d_{\mathcal{N}}(X_1, X_2) + \|X_2 - A_j\|_2 \\ &> \|A_i - X_2\|_2 + d_{\mathcal{N}}(X_1, X_2) + \|X_2 - A_j\|_2 \\ &\geq \|A_i - X_2\|_2 + \|X_2 - A_j\|_2 \\ &\geq \|A_i - A_j\|_2 \\ &> d_{ij} \;\Rightarrow f_{ij}^2(\mathcal{X}) > d_{ij}, \end{aligned}$$

which means that (i,j) is not covered using f_{ij}^2, in contradiction with the assumption of (i,j) being covered by f_{ij}^2. □

Due to Lemma 1, $F^1 = F^2$ for the case of locating two new facilities, and we therefore make use of the simplified notation

$$F(\mathcal{X}) = F^1(\mathcal{X}) = F^2(\mathcal{X})$$

in the case of $m = 2$. The next definition gives the distance of the best option between the trip using the high-speed network and the direct trip.

Definition 2 For $\mathcal{A} = \{A_i = (a_i, b_i) : \; i = 1, \ldots, n\}$ and $\mathcal{X} = (X_1, X_2) \in \mathcal{N}^2$ we define

$$\begin{aligned} \overline{f_{ij}}(X_1, X_2) := \min\{&\|A_i - X_1\|_2 + d_{\mathcal{N}}(X_1, X_2) + \|X_2 - A_j\|_2, \\ &\|A_i - X_2\|_2 + d_{\mathcal{N}}(X_1, X_2) + \|X_1 - A_j\|_2, \\ &\|A_i - A_j\|_2\} \\ = \min\{&h_{12}^{12}(X_1, X_2), h_{12}^{21}(X_1, X_2), \|A_i - A_j\|_2\}. \end{aligned}$$

The following necessary and sufficient condition for an O/D pair to be covered trivially follows:

Lemma 2 *The O/D pair (i,j) is covered by the 2-facility point $\mathcal{X} = (X_1, X_2) \in \mathcal{N}^2$ using the functions f_{ij}^1 or f_{ij}^2 if and only if $\overline{f_{ij}}(X_1, X_2) \leq d_{ij}$.*

3.1 The case of a straight-line segment

In this section we deal with the problem of locating $m = 2$ new facilities on a straight line. We use the fact that we can order the two new facilities (from left to right along the line), hence the problem is given as follows.

OD-pair location on a line ($m = 2$). Given a straight line $\mathcal{L}$, demand points $\mathcal{A}$, trip patterns T and acceptance levels D, find an $(m = 2)$-facility point $\mathcal{X} = (X_1, X_2) \in \mathcal{L}^2$ such that the number of covered trips $F(\mathcal{X})$ is maximized. In short,

$$\max_{\mathcal{X} \in \mathcal{L}^2} F(\mathcal{X}). \tag{4}$$

As before we may assume $\mathcal{L} = [0, L] \times \{0\}$. Hence the objective is to locate two facilities $X_1 = (x_1, 0)$ and $X_2 = (x_2, 0)$ on $\mathcal{L}$ with $x_1 \leq x_2$. For the sake of readability, we will write $\bar{f}_{ij}(x_1, x_2)$ and $h_{ij}^{kl}(x_1, x_2)$ to denote $\bar{f}_{ij}(X_1, X_2)$ and $h_{ij}^{kl}(X_1, X_2)$, respectively. Therefore both functions depend on two variables when $\mathcal{N}$ is a straight-line segment, and not on four variables as in the general case. Note that for this case, $h_{ij}^{kl}(\mathcal{X})$ is either $h_{ij}^{12}(\mathcal{X})$ or $h_{ij}^{21}(\mathcal{X})$, and according to Theorem 1 we know that $h_{ij}^{12}(\mathcal{X}) \leq h_{ij}^{21}(\mathcal{X})$ if and only if $a_i < a_j$. From now on we will assume $a_i \leq a_j \; \forall \; i < j$. Now two technical results that will help us later on are presented.

Lemma 3 *For any pair $i, j \in \{1, \ldots, n\}$ and any $\eta > 0$ we have that the level set*

$$S_{ij}(\eta) \;:=\; \{(x_1, x_2) \in [0, L] \times [0, L] \;:\; \overline{f_{ij}}(x_1, x_2) \leq \eta\}$$

is convex.

Proof Let $\eta > 0$ and assume $a_i < a_j$. We consider two cases.

- If $\eta < \|A_i - A_j\|_2$, and because Theorem 1 implies $x_1 < x_2$, we have that $\overline{f_{ij}}(x_1, x_2) \leq \eta$ if and only if

$$h_{ij}^{12}(x_1, x_2) = \|A_i - (x_1, 0)\|_2 + \alpha \cdot \|(x_1, 0) - (x_2, 0)\|_2 + \|(x_2, 0) - A_j\|_2 \leq \eta.$$

  Let us define $h_{ij} = \min\{h_{ij}^{12}, h_{ij}^{21}\}$. Hence $S_{ij}(\eta) = \{(x_1, x_2) \in [0, L] \times [0, L] \ : \ h_{ij}(x_1, x_2) \leq \eta\}$. Because in this case $h_{ij} = h_{ij}^{12}$, h_{ij} is a convex function in $[0, L] \times [0, L]$, and $S_{ij}(\eta)$ is a convex set.
- If $\eta \geq \|A_i - A_j\|_2$ we obtain $S_{ij}(\eta) = [0, L] \times [0, L]$ which is convex. □

The convexity of the level sets implies the quasiconvexity of the corresponding function, see [18]

Corollary 2 *The functions $\overline{f_{ij}}(x_1, x_2)$ are quasiconvex for all $i, j \in \{1, \ldots, n\}$, i.e. we have*

$$\overline{f_{ij}}(\lambda\mathcal{X} + (1 - \lambda)\mathcal{Y}) \ \leq \ \max\left\{\overline{f_{ij}}(X), \overline{f_{ij}}(Y)\right\}, \ \forall \ i, j,$$

for all $\mathcal{X} = (x_1, x_2), \mathcal{Y} = (y_1, y_2) \in [0, L] \times [0, L]$ and $\lambda \in [0, 1]$.

In the following we will use

$$S_{ij} := S_{ij}(d_{ij}) \ = \ \{(x_1, x_2) \in [0, L] \times [0, L] \ : \ \overline{f_{ij}}(x_1, x_2) \leq d_{ij}\},$$

$$C_{ij} := \{(x_1, x_2) \in [0, L] \times [0, L] \ : \ \overline{f_{ij}}(x_1, x_2) = d_{ij}\},$$

to denote the set of facilities $(x_1, 0), (x_2, 0) \in \mathcal{L}$ which cover the O/D pair (i, j) and its level curve, respectively. Note that all sets S_{ij} are convex, and that some of them may be empty. If the intersection of two level curves, say C_{ij} and $C_{i'j'}$, is nonempty it consists of (at least) one pair of facilities that cover the demand of both (ordered) pairs (A_i, A_j) and $(A_{i'}, A_{j'})$. This yields the following characterization of optimal solutions.

Theorem 2 *Let*

$$C \ = \bigcup_{\substack{(i,j),(i',j') \\ (i,j) \neq (i',j')}} C_{ij} \cap C_{i'j'}.$$

Then (at least) one of the following three cases occurs:

- *C contains (at least) one optimal solution to Problem (4).*
- *There exists an O/D pair (i, j) such that any point of the level set S_{ij} is optimal.*
- *Any point in $[0, L] \times [0, L]$ is optimal.*

Proof Let $\mathcal{X} = (X_1 = (x_1, 0), X_2 = (x_2, 0))$ be an optimal solution to Problem (4) and define

$$O(\mathcal{X}) := \{(i, j) \ : \ (i, j) \text{ is covered by } \mathcal{X}\},$$

$$G(\mathcal{X}) := \begin{cases} \bigcap_{(i,j) \in O(\mathcal{X})} S_{ij}, & \text{if } O(\mathcal{X}) \neq \emptyset, \\ [0, L] \times [0, L], & \text{if } O(\mathcal{X}) = \emptyset. \end{cases}$$

Note, that $G(\mathcal{X})$ is non-empty. Moreover, all points in $G(\mathcal{X})$ yield the same objective value as $\mathcal{X}$. Hence all points of $G(\mathcal{X})$ are optimal solutions. We distinguish three cases:

1. If $O(\mathcal{X}) = \emptyset$ all points in $[0, L] \times [0, L]$ are optimal, and therefore the objective function value is equal to zero.
2. If $G(\mathcal{X}) = S_{ij}$ for some pair $(i, j) \in O(\mathcal{X})$ all points in S_{ij} are optimal, since all points in $G(\mathcal{X})$ yield the same objective value.
3. If $O(\mathcal{X})$ is non-empty and there is no pair (i, j) such that $G(\mathcal{X}) = S_{ij}$, then $G(\mathcal{X})$ is the intersection of at least two different level sets. Since all S_{ij} are closed convex (see Lemma 3), we have that $G(\mathcal{X})$ is convex. Hence, the boundary of $G(\mathcal{X})$ must contain at least one point $\mathcal{Y} = (Y_1 = (y_1, 0), Y_2 = (y_2, 0)) \in C_{ij} \cap C_{i'j'}$ for some O/D pairs (i, j) and (i', j'). Consequently, $\mathcal{Y} = (Y_1, Y_2) \in C$ is an optimal solution to Problem (4). □

In accordance with Bezout's theorem (see [12]) the equation of the intersection of C_{ij} and $C_{i'j'}$ has at most 4^2 roots, but four of them are at infinity. The remaining twelve solutions could be multiple and/or complex. Therefore, an immediate consequence of Theorem 2 is that there exists a finite dominating set for our problem.

Corollary 3 *Let $B \subset [0,L] \times [0,L]$ be a set containing exactly one point of each level set S_{ij} and let C be the set defined in Theorem 2. Then $CAND := C \cup B \cup \{(0,0)\}$ is a finite dominating set for Problem (4).*

Proof Due to the three cases of Theorem 2, at least one optimal solution to Problem (4) is contained in CAND. From Bezout's theorem we may conclude that the intersection of two level curves C_{ij} and $C_{i'j'}$ contains at most 16 points. Consequently, the set C is finite. Moreover, B is a finite set since we have a finite number of level sets S_{ij}. Finally note that we need the point $(0,0)$ due to the third case of Theorem 2, because in this case both B and C are empty. To sum up, CAND is a finite dominating set for (4). □

If the input size, i.e. the number of demand pairs, is $N = O(n^2)$ the number of intersections is $O(N^2)$. Hence the complexity of the finite dominating set CAND is $O(KN^2)$, where K is an upper bound on the number of finite intersection points between two level sets C_{ij} and $C_{i'j'}$. As mentioned above, an upper bound for K is 16 but in our numerical experience we saw this constant was always lower than or equal to 4. Using Corollary 3, Algorithm 1 (see below) solves Problem (4) in $O(N^3)$ time, since evaluating F in step 5(a) takes $O(N)$ and the inclusion test for each pair can be done in constant time, cf. e.g. [23].

Algorithm 1 (Maximum Pair Covering Algorithm in a Segment)

Input: *A set of points $\mathcal{A} = \{A_i = (a_i, b_i) \ : \ i = 1, \ldots, n\} \subset \mathbb{R}^2$ with $a_i \leq a_j \ \forall \ i < j$, a segment of length L, a demand matrix $T = (t_{ij})$, the high-speed factor of the line $\alpha \in (0,1)$, and the matrix $D = (d_{ij})$ with $0 \leq d_{ij} < \|A_i - A_j\|_2$ for $1 \leq i, j \leq n$.*

```
1. Set $C = \emptyset$.
2. For each O/D pair $(i,j)$ with $i < j$ compute $C_{ij}$, the level curve of $\overline{f_{ij}}$.
3. For each pair of O/D pairs $\{(i,j),(i',j')\}$ compute the intersection between $C_{ij}$ and $C_{i'j'}$. Set $C = C \cup (C_{ij} \cap C_{i'j'})$.
4. If $C_{ij} \cap C_{i'j'} = \emptyset$ and $C_{ij}, C_{i'j'} \neq \emptyset$ apply an inclusion test:
   (a) If $C_{ij}$ is inside $C_{i'j'}$ (or $C_{i'j'}$ inside $C_{ij}$), then take one point of $C_{ij}$ (or $C_{i'j'}$) and add it to $C$, respectively.
   (b) Otherwise take one point of $C_{ij}$ and one of $C_{i'j'}$ and add it to $C$.
5. (a) If $C \neq \emptyset$, then evaluate the objective function $F(x_1,x_2)$ on every point $(x_1,x_2) \in C$ and choose a maximum. Let $(X_1^*, X_2^*)$ be such maximum, with $X_1^* = (x_1^*, 0), X_2^* = (x_2^*, 0)$.
   (b) Otherwise, choose any pair $X_1^*, X_2^* \in \mathcal{L}$.
```

Output: *Two facilities $X_1^*, X_2^* \in \mathcal{L}$ solving problem (4).*

Example 1 Two facilities are to be located on the line segment $\mathcal{L} = [0,5]$ to cover five sites corresponding to the following coordinates:

$$A_1 = (1,1), \quad A_2 = (1,-0.5), \quad A_3 = (3,0.5), \quad A_4 = (4,-0.5), \quad A_5 = (4,-2).$$

Let us assume that the O/D matrix is

$$T = \begin{pmatrix} 0 & 46 & 27 & 90 & 75 \\ 0 & 0 & 70 & 47 & 46 \\ 0 & 0 & 0 & 25 & 74 \\ 0 & 0 & 0 & 0 & 46 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Furthermore, we used the high-speed factor $\alpha = 0.5$ and the acceptance values $d_{ij} = 0.98 \cdot \|A_i - A_j\|_2$, i.e. we assume that the high-speed line will be chosen if at least 2% of traveling time is saved.

Each level set S_{ij} is given by the solution set of the inequality $\overline{f_{ij}} \leq d_{ij}$; that is, each point $(x_1,x_2) \in S_{ij}$ has to satisfy the inequality

$$\sqrt{(a_i - x_1)^2 + b_i^2} + \alpha(x_2 - x_1) + \sqrt{(a_j - x_2)^2 + b_j^2} \ \leq \ d_{ij}.$$

Due to Theorem 1 all level sets are contained in the upper left triangle of $[0,5] \times [0,5]$. The intersection of the level sets S_{14}, S_{15}, S_{23}, and S_{24} is the set of points (x_1,x_2) that maximize the trip coverage, meaning that any (x_1,x_2) in this set covers the pairs $(1,4)$, $(1,5)$, $(2,3)$, $(2,4)$, yielding an optimal objective value equal to 282, see Figure 2.

So, for instance, locating one station at $(1.5,0)$ and the other at $(3,0)$ is an optimal solution (because (1.5,3) belongs to the optimal region depicted as the gray area in Figure 2).

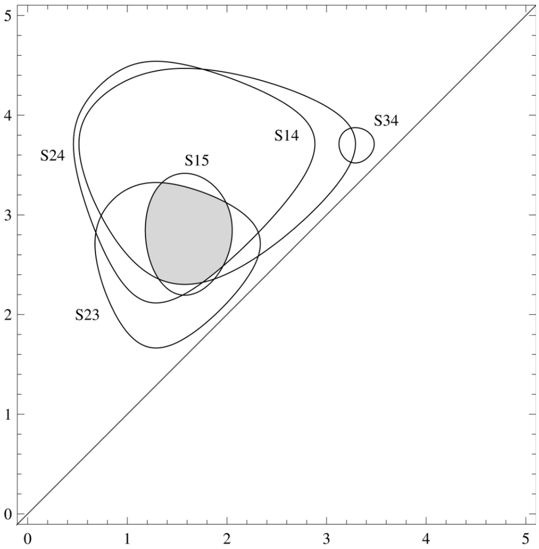

Fig. 2 Non-empty coverage regions for Example 1.

3.2 The case of a tree network

In this section we will deal with the problem of locating two facilities on a given tree network $\mathcal{T}(V, E)$, stated as follows.

OD-pair location on a tree ($m = 2$). Given a high-speed tree network $\mathcal{T}$, demand points $\mathcal{A}$, trip patterns T and acceptance levels D, find an $(m = 2)$-facility point $\mathcal{X} = (X_1, X_2) \in \mathcal{T}^2$ such that the number of covered trips $F(\mathcal{X})$ is maximized. In short,

$$\max_{\mathcal{X} \in \mathcal{T}^2} F(\mathcal{X}). \tag{5}$$

Since the problem of locating two facilities on the same edge coincides with that in the previous subsection, we will now study the case in which the facilities are to be located on different edges. Thus, this restricted problem can be formulated as:

OD-pair location on an edge-pair of a tree ($m = 2$). Given a high-speed tree network $\mathcal{T}$ with two specified edges e_p and e_q ($e_p \neq e_q$), demand points $\mathcal{A}$, trip patterns T and acceptance levels D, find an $(m = 2)$-facility point $\mathcal{X} = (X_1, X_2) \in e_p \times e_q$ such that the number of covered trips $F(\mathcal{X})$ is maximized. In short,

$$\max_{\mathcal{X} \in e_p \times e_q} F(\mathcal{X}). \tag{6}$$

As usual, $F(\mathcal{X}) = \sum_{(i,j) \in G(\mathcal{X})} t_{ij}$ and we do not need to distinguish between F^1 and F^2 since we still consider the case $m = 2$. Our goal is to adapt the Maximum Pair Covering Algorithm of the previous subsection.

In [8], a *path convex combination* of two points Y and Z in a tree is defined as follows:

Definition 3 Let Y, Z be two points on a tree and let $p(Y, Z)$ be the unique path from Y to Z. X is a path convex combination of Y and Z along $p(Y, Z)$ with respect to $\lambda \in [0, 1]$ if

- $X \in p(Y, Z)$, and
- $d_\tau(Y, Z) = \lambda d_\tau(Y, Z)$.

(Note that in this case we have that $d_\tau(X, Z) = (1 - \lambda) d_\tau(Y, Z)$.)

Note as well that this notion of path convex combination is different from the usual convex combination concept in the plane (see Figure 3).

Thus, the segment joining two pairs of points $\mathcal{Y} = (Y_p, Y_q), \mathcal{Z} = (Z_p, Z_q)$ in $e_p \times e_q$ is $\mathcal{PS}(\mathcal{Y}, \mathcal{Z}) = \cup_{\lambda \in [0,1]} \{\mathcal{X} = (X_p, X_q) \in e_p \times e_q : d(Y_i, X_i) + d(X_i, Z_i) = d(Y_i, Z_i); d(X_i, Z_i) = \lambda d(Y_i, Z_i), i \in \{p, q\}\}$,

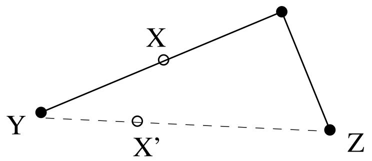

Fig. 3 The point X is a path convex combination of Y and Z with respect to $\lambda = \frac{1}{3}$ while X' is a convex combination of Y and Z with respect to $\lambda = \frac{1}{3}$

where $\mathcal{PS}$ stands for *path segment*. Thus, a function is path convex on $e_p \times e_q$ if $f(\mathcal{X}) \leq \lambda f(\mathcal{Y}) + (1-\lambda) f(\mathcal{Z})$, $\forall\ \mathcal{X} \in \mathcal{PS}(\mathcal{Y}, \mathcal{Z})$, $\forall\ \lambda \in [0,1]$. See [8] for more details, where it is proven that $d_{\mathcal{N}}$ is convex in $\mathcal{N} \times \mathcal{N}$ if and only if $\mathcal{N}$ is a tree. As a consequence $d_{\mathcal{T}}(\cdot,\cdot)$ is a convex function for $(Y,Z) \in e_p \times e_q$.

Normally, the used notion of convexity is clear from the context. Whenever there might be ambiguity, we will specify whether convexity or path convexity applies.

Next, we will see that Lemma 3 and Corollary 2 can be extended to the case of a tree network as follows.

Lemma 4 *For any pair (i,j) and any $\eta > 0$ we have that the level set*

$$S_{ij}(\eta) \ := \ \{(X_1,X_2) \in e_p \times e_q \ : \ \overline{f_{ij}}(X_1,X_2) \leq \eta\}$$

is the union of two disjoint and convex sets.

Proof According to Definition 2 we have that

$$S_{ij}(\eta) = \begin{cases} S_{ij}^{12}(\eta) \cup S_{ij}^{21}(\eta), & \text{if } \|A_i - A_j\|_2 > \eta, \\ \\ e_p \times e_q, & \text{if } \|A_i - A_j\|_2 \leq \eta, \end{cases}$$

with

$$\begin{aligned} S_{ij}^{12}(\eta) &:= \{(X_1,X_2) \in e_p \times e_q : h_{ij}^{12}(X_1,X_2) \leq \eta\}, \\ S_{ij}^{21}(\eta) &:= \{(X_1,X_2) \in e_p \times e_q : h_{ij}^{21}(X_1,X_2) \leq \eta\}. \end{aligned}$$

Let $\eta > 0$. First note that if $\eta \geq \|A_i - A_j\|_2$ we obtain $S_{ij}(\eta) = e_p \times e_q$ which is convex. We hence analyze the case $\eta < \|A_i - A_j\|_2$, for which we claim:

1. $S_{ij}^{12}(\eta) \cap S_{ij}^{21}(\eta) = \emptyset$.
2. $S_{ij}^{12}(\eta)$ and $S_{ij}^{21}(\eta)$ are convex.

These two assertions are proven as follows.

1. Assume that $(X_1,X_2) \in S_{ij}^{12}(\eta)$. Then we have that

$$h_{ij}^{12} = \|A_i - X_1\|_2 + d_{\mathcal{T}}(X_1,X_2) + \|X_2 - A_j\|_2 \leq \eta < \|A_i - A_j\|_2,$$

therefore,

$$\|A_i - X_1\|_2 + \|X_2 - A_j\|_2 < \|A_i - A_j\|_2.$$

From this we conclude

$$\begin{aligned} \|A_i - A_j\|_2 &\leq \|X_1 - A_i\|_2 + \|X_1 - A_j\|_2 \\ &< \|A_i - A_j\|_2 - \|X_2 - A_j\|_2 + \|X_1 - A_j\|_2 \\ &\leq \|A_i - X_2\|_2 + \|A_j - X_2\|_2 - \|X_2 - A_j\|_2 + \|X_1 - A_j\|_2 \\ &= \|X_2 - A_i\|_2 + \|X_1 - A_j\|_2, \end{aligned}$$

hence,

$$\|A_i - X_2\|_2 + d_{\mathcal{T}}(X_1,X_2) + \|X_1 - A_j\|_2 > \|A_i - A_j\|_2 > \eta,$$

and so $(X_1,X_2) \notin S_{ij}^{21}$.

2. $h_{ij}^{12}(X_1, X_2) \leq \eta$ if and only if

$$h_{ij}^{12}(X_1, X_2) = \|A_i - X_1\|_2 + d_{\mathcal{T}}(X_1, X_2) + \|X_2 - A_j\|_2 \leq \eta,$$

hence $S_{ij}^{12}(\eta) = \{(X_1, X_2) \in e_p \times e_q \ : \ h_{ij}^{12}(X_1, X_2) \leq \eta\}$. Since, $\|A_i - X_1\|_2$ and $\|X_2 - A_j\|_2$ are path convex functions on e_p and e_q, respectively, and $d_{\mathcal{T}}(X_1, X_2)$ is convex on the convex set $e_p \times e_q$ then h_{ij}^{12} is a convex function on $e_p \times e_q$, and therefore $S_{ij}^{12}(\eta)$ is a convex set. The convexity of $S_{ij}^{21}(\eta)$ can be proven analogously. □

Corollary 4 *The functions $h_{ij}^{12}(X_1, X_2)$ and $h_{ij}^{12}(X_1, X_2)$ are quasiconvex on $e_p \times e_q$.*

Remark 3 We remark that the level set in Lemma 4 cannot be taken for any pair $(X_1, X_2) \in \mathcal{T}$ although $d_{\mathcal{T}}(X_1, X_2)$ is still path convex in this case (see [8]). The reason for this is that path convex combinations are not the same as convex combinations for arbitrary X_1, X_2 on a tree (see Figure 3), meaning that the convexity of function h_{ij} would be lost if if X_1 and X_2 vary on a tree.

In order to get a computable representation we parametrize the two edges e_p and e_q such that

$$\begin{aligned} e_p &= \{X = g_p(\lambda) := T_p + \lambda s_p : \lambda \in [0,1]\} \\ e_p &= \{X = g_q(\lambda) := T_q + \lambda s_q : \lambda \in [0,1]\} \end{aligned}$$

with $T_p, T_q, s_p, s_q \in \mathbb{R}^2$ and $g_p(0) = T_p, g_q(0) = T_q$. Note that T_p, s_p, T_q and s_q are the starting points and directions of edges e_p and e_q, respectively. We hence obtain that Problem (6) is a continuous two-dimensional problem formulated as

$$\max_{(\lambda_p, \lambda_q) \in [0,1]^2} F(g_p(\lambda_p), g_q(\lambda_q)) \tag{7}$$

and the level set $S_{ij} := S_{ij}(d_{ij})$ is given as

$$S_{ij} := \{(\lambda_p, \lambda_q) \in [0,1]^2 : \overline{f_{ij}}(g_p(\lambda_p), g_q(\lambda_q)) \leq d_{ij}\}.$$

We now transfer the result of Lemma 4 to this two-dimensional representation.

Corollary 5 *The level set $S_{ij} = S_{ij}^{12} \cup S_{ij}^{21} \subseteq \mathbb{R}^2$ is the union of two disjoint and convex sets, S_{ij}^{12}, S_{ij}^{21}.*

Proof We know that $h_{ij}^{12}, h_{ij}^{21} : e_p \times e_q \to \mathbb{R}$ are quasiconvex according to Corollary 4, and that $g = (g_p, g_q) : [0,1]^2 \to e_p \times e_q$ is linear, hence their compositions $h_{ij}^{12} \circ g, h_{ij}^{21} \circ g$ are quasiconvex, yielding the convexity of

$$\begin{aligned} S_{ij}^{12} &:= \{(\lambda_p, \lambda_q) \in [0,1]^2 : h_{ij}^{12}(g_p(\lambda_p), g_q(\lambda_q)) \leq d_{ij}\} \\ S_{ij}^{21} &:= \{(\lambda_p, \lambda_q) \in [0,1]^2 : h_{ij}^{21}(g_p(\lambda_p), g_q(\lambda_q)) \leq d_{ij}\}. \end{aligned}$$

Since for $\eta < \|A_i - A_j\|$ we have that

$$S_{ij}(\eta) = \{(\lambda_p, \lambda_q) \in [0,1]^2 : h_{ij}^{12}(g_p(\lambda_p), g_q(\lambda_q) \leq \eta\} \cup \{(\lambda_p, \lambda_q) \in [0,1]^2 : h_{ij}^{21}(g_p(\lambda_p), g_q(\lambda_q) \leq \eta\}$$

we obtain that

$$S_{ij} = S_{ij}^{12} \cup S_{ij}^{21}$$

and the result follows. □

For each pair (i, j) let C_{ij}^{12} and C_{ij}^{21} be the boundaries of the sets S_{ij}^{12} and S_{ij}^{21}. This gives us all we need to prove the following statement similar to Theorem 2.

Theorem 3 *Let*

$$C \;=\; \bigcup_{\substack{(i,j),(i',j'),(i,j)\neq(i',j') \\ (k,l),(k',l')\in\{(1,2),(2,1)\}}} C_{ij}^{kl} \cap C_{i'j'}^{k'l'}.$$

Then (at least) one of the following three cases occurs:

- *C contains (at least) one optimal solution to Problem (6).*
- *There exists an O/D pair (i, j) and $(k, l) \in \{(1, 2), (2, 1)\}$ such that any point of the level set S_{ij}^{kl} is optimal.*

– *Any point in* $[0,1] \times [0,1]$ *is optimal.*

Again in accordance with Bezout's theorem (see [12]) the equation of the intersection of C_{ij}^{kl} and $C_{i'j'}^{k'l'}$ has at most 4^2 roots, but four of them are at infinity. The remaining twelve solutions could be multiple and/or complex. Therefore, we again obtain a finite dominating set.

Corollary 6 *Let* $B \subset [0,1] \times [0,1]$ *be a set containing exactly one point of each set* S_{ij}^{12} *and* S_{ij}^{21} *and let* C *be the set defined in Theorem 2. Then* $CAND := C \cup B \cup \{(0,0)\}$ *is a finite dominating set for Problem (4).*

In summary, Algorithm 1 can be extended to the case in which X_1 and X_2 are in different edges.

Algorithm 2 (Maximum Pair Covering Algorithm for two facilities in different edges)

Input: *A set of points* $\mathcal{A} = \{A_i = (a_i, b_i) \ : \ i = 1, \ldots, n\} \subset \mathbb{R}^2$*, a pair of edges* e_p*,* e_q*, a demand matrix* $T = (t_{ij})$*, the high-speed factor of the line* $\alpha \in (0,1)$*, and the matrix* $D = (d_{ij})$ *with* $0 \leq d_{ij} < \|A_i - A_j\|_2$ *for* $1 \leq i, j \leq n$*.*

1. *Set* $C = \emptyset$*.*
2. *For each O/D pair* (i,j)*, and each* $(k,l) \in \{(1,2),(2,1)\}$ *compute the level curves* C_{ij}^{kl}
3. *For each pair of O/D pairs* $\{(i,j),(i',j')\}$*, and* $(k,l),(k',l') \in \{(1,2),(2,1)\}$ *compute the intersections between* C_{ij}^{kl} *and* $C_{i'j'}^{k'l'}$*. Set* $C = C \cup (C_{ij}^{kl} \cap C_{i'j'}^{k'l'})$*.*
4. *If* $C_{ij}^{kl} \cap C_{i'j'}^{k'l'} = \emptyset$ *and* $C_{ij}^{kl}, C_{i'j'}^{k'l'} \neq \emptyset$ *apply an inclusion test:*
   (a) *If* C_{ij}^{kl} *is inside* $C_{i'j'}^{k'l'}$ *(or* $C_{i'j'}^{k'l'}$ *inside* C_{ij}^{kl}*), then take one point of* C_{ij}^{kl} *(or* $C_{i'j'}^{k'l'}$*) and add it to* C*, respectively.*
   (b) *Otherwise take one point of* C_{ij}^{kl} *and one of* $C_{i'j'}^{k'l'}$ *and add it to* C*.*
5. (a) *If* $C \neq \emptyset$*, then evaluate the objective function* $F(g_p(\lambda_p), g_q(\lambda_q))$ *on every point* $(\lambda_p, \lambda_q) \in C$ *and choose a maximum, given by* $(\lambda_p^*, \lambda_q^*)$*.*
   (b) *Otherwise, choose any pair* $\lambda_p^* \in [0,1], \lambda_q^* \in [0,1]$*.*

Output: *Two facilities* $X_1 := g_p(\lambda_p^*), X_2 := g_q(\lambda_q^*) \in e_p \times e_q$ *solving Problem (6).*

The complexity of this algorithm is the same as Algorithm 1, $O(N^3)$.

Note that Algorithm 2 gives a pair of points $X_1 \in e_p, X_2 \in e_q$ maximizing F. From algorithms 1 and 2, a procedure that calculates a pair of facilities maximizing F on a tree network $\mathcal{T}(\mathcal{V}, \mathcal{E})$ is given by the following algorithm. The idea is to iteratively apply the two previous algorithms to locate a pair of facilities on the same edge, or to locate a pair of facilities on different edges, respectively, for any possible combination of edges on the tree, to then pick the best solution. These ideas are summarized in Algorithm 3.

Algorithm 3 (Maximum Pair Covering Algorithm for a tree network)

Input: *A set of points* $\mathcal{A} = \{A_i = (a_i, b_i) \ : \ i = 1, \ldots, n\} \subset \mathbb{R}^2$*, a tree network* $\mathcal{T}(V, E)$*, a demand matrix* $T = (t_{ij})$*, the high-speed factor of the line* $\alpha \in (0,1)$*, and the matrix* $D = (d_{ij})$ *with* $0 \leq d_{ij} < \|A_i - A_j\|_2$ *for* $1 \leq i, j \leq n$*.*

1. *For each edge* e_p*, apply Algorithm 1. Let* (X_1^p, X_2^p) *be the solution obtained.*
2. *For each pair of distinct edges* e_p*,* e_q *with* $p < q$*, apply Algorithm 2. Let* (X_1^{pq}, X_2^{pq}) *be the solution obtained.*
3. *Determine the best solution* (X_1, X_2) *of all* $(X_1^p, X_2^p), (X_1^{pq}, X_2^{pq})$*.*
   Output: *Two facilities* $X_1, X_2 \in \mathcal{T}$ *solving problem (5).*

The complexity of this algorithm is $O(MN^3 + M^2N^3) = O(M^2N^3)$, where $M = |E|$.

4 The Big Cube Small Cube Method for $m \geq 2$ facilities on a line segment

In this section we present an algorithm for locating m facilities along a line, i.e., we consider the following problem.

OD-pair location on a line. Given a straight line $\mathcal{L}$, demand points $\mathcal{A}$, trip patterns T, function f^r with $r = 1$ or 2, and acceptance levels D, find an m-facility point $\mathcal{X} = (X_1, ..., X_m) \in \mathcal{L}^m$ such that the number of covered trips $F(\mathcal{X})$ is maximized. In short,

$$\max_{\mathcal{X} \in \mathcal{L}^m} F(\mathcal{X}). \tag{8}$$

Our approach is a modification of the big cube small cube method, see [24], which is a generalization of the big square small square (BSSS for short) algorithm, see [10] and [21].

We want to locate m facilities $\mathcal{X} = (X_1, \ldots, X_m)$ on the line segment $\mathcal{L} = [0, L] \times \{0\}$, using the overall minimum distance functions f^2_{ij}. Therefore, identifying X_i with its first coordinate x_i, our objective function

$$F^2(\mathcal{X}) \;=\; F^2(x_1, \ldots, x_m) \;=\; \sum_{(i,j): f^2_{ij} \le d_{ij}} t_{ij}$$

deals with m variables.

The idea we use is based on a geometric branch-and-bound: The solution space is divided into a list of boxes. In every step, one box is selected and split into several smaller boxes. For each smaller box, an upper bound is computed and the objective function is evaluated at the center of the smaller boxes. The greatest value of the evaluated objective values is stored as incumbent. If the upper bounds of some boxes are smaller than the incumbent, these boxes can be discarded. Techniques for obtaining these bounds exist for continuous functions, in particular for d.c-functions.

However, we cannot directly apply these techniques to our problem since for counting the number of covered pairs we have to introduce a discrete variable for any O/D-pair. In the following we will demonstrate how we can nevertheless make use of well-known techniques for solving such a (partly discrete) problem.

We first show how we can make use of our knowledge for calculating bounds for the continuous functions f^2_{ij} to derive bounds on the function F^2.

Theorem 4 *Assume that for each box $Z \subset \mathcal{L}^m$ and each pair (i, j) we have a lower bound, i.e.,*

$$w_{ij}(Z) \le f^2_{ij}(\mathcal{X}) \text{ for all } \mathcal{X} \in Z,$$

and let $P(Z) \in Z$. Then we have:

1. $$LB(Z) := F^2(P(Z)) \le \max_{\mathcal{X} \in Z} F^2(\mathcal{X}),$$
   i.e. $F^2(P(Z))$ is a lower bound on the objective value on Z.
2. $$UB(Z) := \sum_{(i,j): w_{ij}(Z) \le d_{ij}} t_{ij} \ge F^2(\mathcal{X}) \text{ for all } \mathcal{X} \in Z,$$
   i.e., $UB(Z)$ is an upper bound on the objective value on Z.
3. *Assume that every O/D-pair (i, j) satisfies one of the following three conditions,*
   (i) $f^2_{ij}(P(Z)) = w_{ij}(Z)$, or,
   (ii) $d_{ij} > f^2_{ij}(P(Z))$, or,
   (iii) $d_{ij} < w_{ij}(Z)$.
   Then $UB(Z) = LB(Z)$, i.e., $F^2(P(Z))$ is the optimal value on Z.

Note that $w_{ij}(Z)$ together with P form a *bounding operation* as defined in [26], i.e., the theorem holds true for any bounding operation on Z.

Proof Consider one box Z and let $w_{ij} := w_{ij}(Z)$ and $P := P(Z)$.

1. $P \in Z$, i.e., the objective value on Z is better than $F^2(P)$.
2. Since $w_{ij} \le f^2_{ij}(\mathcal{X})$ for all $\mathcal{X} \in Z$ we obtain
   $$UB(Z) = \sum_{(i,j): w_{ij} \le d_{ij}} t_{ij} \ge \sum_{(i,j): f^2_{ij}(\mathcal{X}) \le d_{ij}} t_{ij} = F(\mathcal{X}).$$

3.
$$UB(Z) - F^2(P) = \sum_{(i,j):w_{ij} \le d_{ij}} t_{ij} - \sum_{(i,j):f^2_{ij}(P) \le d_{ij}} t_{ij} = \sum_{(i,j):w_{ij} \le d_{ij} < f^2_{ij}(P)} t_{ij}.$$

This expression becomes zero if and only if for all (i, j) we have that

$$d_{ij} \notin [w_{ij}, f^2_{ij}(P)),$$

which is exactly the case if one of the conditions (i),(ii),(iii) applies. □

In order to derive a bounding operation for the functions f^2_{ij} the whole set of possibilities used for geometric branch & bound methods can be applied, see e.g., [5,11], or [25,26] for an overview and a discussion of various methods. Having such a bounding operation at hand, the following algorithm calculates an approximate optimal solution.

Algorithm 4 (Geometric branch & bound)

Input: *A set of points* $\mathcal{A} = \{A_i = (a_i, b_i) \ : \ i = 1, \ldots, n\} \subset \mathbb{R}^2$*, a segment of length* L*, a demand matrix* $T = (t_{ij})$*, the high-speed factor of the line* $\alpha \in (0, 1)$*, the matrix* $D = (d_{ij})$ *with* $0 \le d_{ij} < \|A_i - A_j\|_2$ *for* $1 \le i, j \le n$*.*

1. *Set* $\mathcal{Z} = \{Z = [0, L]^m\}$, $P = (\frac{L}{2}, \ldots, \frac{L}{2}) \in Z$*.*
2. *Calculate* $LB := F^2(P)$ *and* $UB_{\max} := UB(Z)$*.*
3. *Division rule: Select a box* $Z \in \mathcal{Z}$ *with largest upper bound* $UB_{\max} = UB(Z)$ *and split it into some smaller boxes.*
4. *Insert the smaller boxes in* $\mathcal{Z}$ *and delete the original box from* $\mathcal{Z}$*.*
5. *Evaluate the objective function at the center of each smaller box. If at least one of these values is* $\ge LB$*, update* LB *to the largest value and update* P *to the center of the associated box.*
6. *Calculate* $UB(Z)$ *for all smaller boxes and update* $UB_{\max} := \max_{Z \in \mathcal{Z}} UB(Z)$*.*
7. *Discarding test: Discard all boxes* Z
$$Z \ = \ [\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_m, \overline{x}_m]$$
with $UB(Z) < LB$ *from* $\mathcal{Z}$*. If* LB *has not changed we have to check only the smaller boxes. Furthermore, discard all smaller boxes with* $\underline{x}_k > \overline{x}_l$ *for some* $k < l$ *from* $\mathcal{Z}$*.*
8. *Termination rule: If* $UB_{max} = LB$*, stop. Otherwise return to Step 3.*

Output: *An optimal solution* $\mathcal{X} = (x_1, \ldots, x_m) := P$ *containing a set of* m *optimal facilities* $(x_1, 0), \ldots, (x_m, 0)$ *solving Problem (4). (If required,* $\mathcal{Z}$ *is the set of boxes that contains* $\mathcal{X}$ *and might contain (other) optimal solutions.)*

Note that the discarding test in Step 7 of the algorithm uses the result of Theorem 4.

We remark that no bounding operation for f^2_{ij} yields a consistent bounding operation for the function F^2, since it may happen that splitting a box into subboxes does not affect the value of $F(Z)$. Nevertheless, due to the third point of Theorem 4, many boxes can be discarded throughout the algorithm such that it will finally stop.

We now use another feature of our problem to derive the set of all optimal solutions for (4). Most algorithms only calculate one individual solution. But sometimes the set of all optimal solutions is required. For Problem (4) we can use Lemma 3 and propose Algorithm 5, which computes a subset of all optimal solutions for the case of locating two facilities if the optimal objective value OPT of Problem (4) is known.

Algorithm 5 (Approximation of optimal solution set for $m = 2$)

Input: *A set of points* $\mathcal{A} = \{A_i = (a_i, b_i) \ : \ i = 1, \ldots, n\} \subset \mathbb{R}^2$*, a segment of length* L*, a demand matrix* $T = (t_{ij})$*, the high-speed factor of the line* $\alpha \in (0, 1)$*, the matrix* $D = (d_{ij})$ *with* $0 \le d_{ij} < \|A_i - A_j\|_2$ *for* $1 \le i, j \le n$*.*

1. *Solve problem (4) using Algorithm 1 or Algorithm 4*

2. Set $\mathcal{Z} := \{[0, L]^2\}$ in case of Algorithm 1; or take the output $\mathcal{Z}$ of Algorithm 4 if this algorithm has been used in Step 1. Set $\mathcal{D} := \emptyset$.
3. Split all boxes $Z \in \mathcal{Z}$ into some smaller boxes.
4. Add all smaller boxes to $\mathcal{Z}$ and delete the original boxes from $\mathcal{Z}$.
5. For all boxes $Z \in \mathcal{Z}$ do: Evaluate the functions $\overline{f_{ij}}(V_k)$ for all $i < j$ at the four vertexes V_1, V_2, V_3, V_4 of Z. Define

$$r_{ij} := \begin{cases} t_{ij}, & \text{if} \quad \overline{f_{ij}}(V_k) < d_{ij} \text{ for } k = 1, 2, 3, 4, \\ 0, & \text{otherwise,} \end{cases}$$

and let

$$R := \sum_{i<j} r_{ij}.$$

If $R = OPT$, add the box Z to $\mathcal{D}$ and delete Z from $\mathcal{Z}$.
6. If all boxes $Z \in \mathcal{Z}$ are small enough, stop. Otherwise return to Step 3.

***Output**: A set of optimal solutions $\mathcal{D}^* := \bigcup_{\mathcal{D} \in \mathcal{Z}} \mathcal{D}$ to problem (4).*

The correctness of this algorithm is justified as follows.

Lemma 5 *Let $Z \in \mathcal{D}^*$ and $\mathcal{X}^* = (X_1^*, X_2^*) \in Z$. Then $\mathcal{X}^*$ is an optimal solution to Problem (4).*

Proof We have to show that all $\mathcal{X} \in Z$ are optimal solutions. Let V_1, V_2, V_3, V_4 be the four vertices of Z. From Step 5 of the algorithm we know that $\overline{f_{ij}}(V_k) < d_{ij}$ for $k = 1, 2, 3, 4$, in particular, $V_k \in S_{ij}$ for $k = 1, \ldots, 4,\ i < j$. Since S_{ij} is convex according to Lemma 3, the complete box Z is in S_{ij} and the result follows. □

5 Case Study and Experimental Results

We first use Algorithm 4 to analyze the example presented in the introduction for the case $2 \leq m \leq 4$.

Example 2 The $n = 12$ demand points are given by the data in Table 1 (note that all coordinates have been rotated and translated so that Sevilla (point A_1) coincides with the origin and Córdoba (point A_{12}) is on the x-axis). Furthermore, we used the Euclidean distance d_2, the high-speed factor $\alpha = 0.5$, the line segment $\mathcal{L} = [0, 112]$, and the acceptance values

A_i	a_i	b_i	A_i	a_i	b_i	A_i	a_i	b_i
A_1	0.000	0.000	A_5	59.686	3.392	A_9	65.326	2.004
A_2	6.426	19.707	A_6	29.096	−4.160	A_{10}	78.302	−14.491
A_3	11.689	7.352	A_7	43.495	5.109	A_{11}	94.086	−9.701
A_4	27.491	10.234	A_8	45.980	−19.119	A_{12}	111.181	0.000

Table 1 The demand points for Example 2.

$$d_{ij} = 0.9 \cdot d_2(A_i, A_j).$$

The trip patterns are estimated by $T = (t_{ij})$ using the gravitation rule, see [19], as follows:

$$t_{ij} = \tau \cdot \frac{p_i \cdot p_j}{d_2^2(A_i, A_j)}, \text{ and } \tau = 0{,}00016029$$

for $i < j$ and $t_{ij} = 0$ otherwise. Here, p_i is the population of city A_i, see Table 2. The value of τ has been estimated from traffic data between Sevilla and Córdoba. Note that t_{ij} need not be integer.

Table 3 shows optimal solutions for the cases $m = 2$ to $m = 4$. The last column presents the run time of the algorithm. The solutions are depicted in Figure 4. Note as well that the gravitational rule penalizes the number of trips between distant cities.

Example 3 In this example we consider the same input data as in Example 2 but we now ignore all traffic flow between Sevilla and any other city as well as between Córdoba and any other city. This is a realistic assumption since there already exist high-speed train stations at Sevilla and Córdoba. Our result for $m = 2$ new stations can be found in Figure 5. Here, one optimal solution is $(x_1, x_2) = (42.65625, 66.28125)$ with an objective value of 460.05994, which is some 15.30% of the potential demand.

p_1	704198	p_5	3814	p_9	21605
p_2	5156	p_6	28576	p_{10}	40534
p_3	12580	p_7	19393	p_{11}	13469
p_4	5260	p_8	19861	p_{12}	328534

Table 2 Population of the cities A_1 to A_{12} for Example 2.

m	x_1	x_2	x_3	x_4	$F(x_1, \dots, x_m)$	%	run time (sec)
2	0.109375	9.078125			13011.657	46.86%	0.01
3	0.109375	9.078125	50.203125		18603.935	67.00%	0.13
4	0.109375	9.078125	95.703125	111.015625	21813.235	78.56%	0.43

Table 3 Optimal solutions for Example 2 obtained by Algorithm 4, objective function values, percentages of covered potential travelers, and computational times.

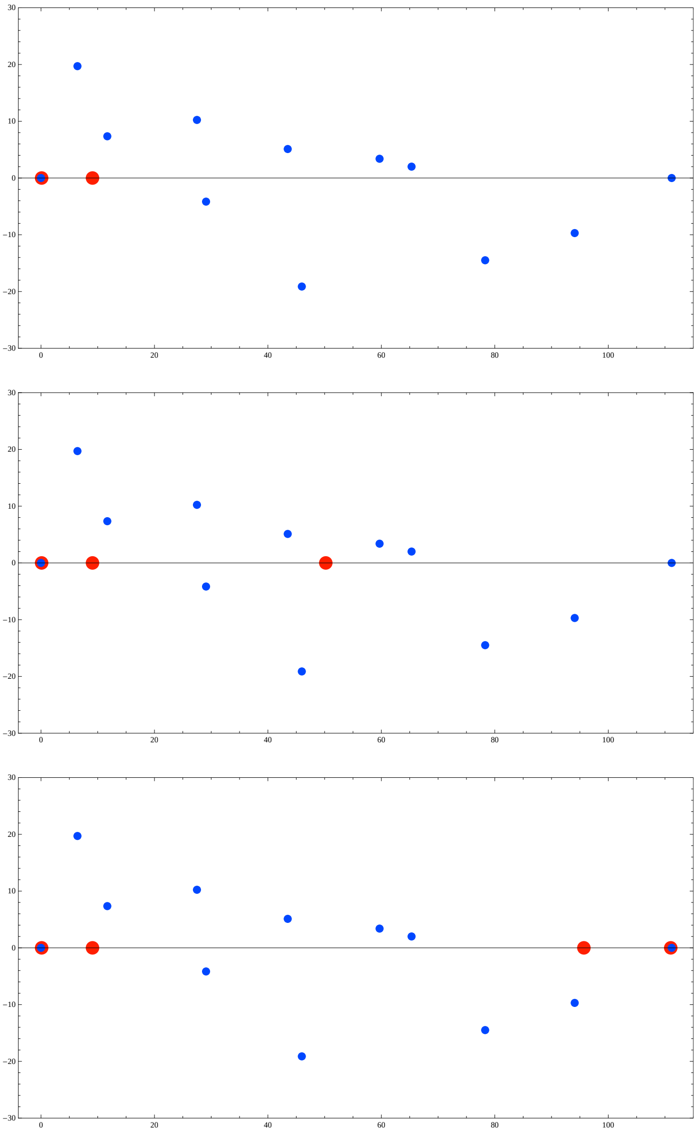

Fig. 4 Optimal solutions for Example 2 and $m = 2$ to $m = 4$.

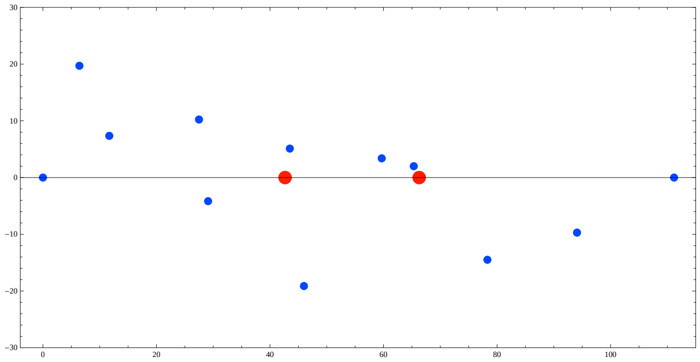

Fig. 5 Optimal solutions for Example 3 and $m = 2$.

We now illustrate the efficiency of the geometric branch & bound method, Algorithm 4, when we used the d.c bounding operations, see [11]. Our algorithm was implemented in JAVA, compiled by JAVA 2 SDK 1.4, using double precision arithmetic. All tests were run on a 2.4 GHz computer with 1,024 MB of memory.

We randomly generated $10 \leq n \leq 200$ existing facilities $A_1, \ldots, A_n$ in $[0, 10] \times [-2.5, 2.5]$ ordered by their first coordinates. The line segment was $\mathcal{L} = [0, 10]$, the high-speed factor was $\alpha = 0.5$, and the acceptance values were given by $d_{ij} \;=\; 0.9 \cdot \|A_i - A_j\|_2$. The traffic was also randomly generated as follows: For $i < j$ we randomly set $t_{ij} \in \{0, 1, \ldots, 8\}$ in such a way that the probability for $t_{ij} = 0$ was $1/3$ and for $k = 1, \ldots, 8$ the probability for $t_{ij} = k$ was $1/12$, respectively.

Ten problems instances were run for various values of n and m. Algorithm 4 stopped when one optimal solution was found. Our results are reported in Table 4 and Figure 6. Since all results were quite similar for various values of α, we only reported run times for $\alpha = 0.5$.

		Run time (sec.)		
n	m	Ave.	Min	Max
50	2	0.22	0.13	0.32
100	2	2.02	1.22	2.79
150	2	6.71	3.62	10.11
200	2	15.01	9.60	22.28
20	3	0.76	0.41	1.76
40	3	6.01	2.90	15.57
60	3	17.30	9.01	28.31
80	3	63.70	29.24	113.19
10	4	3.10	0.32	11.88
15	4	9.75	1.93	18.67
20	4	25.44	6.79	85.36
25	4	67.09	8.83	216.59

Table 4 Results using the big cube small cube solution method.

For the case $m = 2$, all problems with up to $n = 200$ points and therefore with up to $n(n-1)/2 = 19,900$ O/D pairs could be efficiently solved in less than 23 seconds of computer time. But if we want to locate more than two facilities, the run time increases as typical for geometric branch & bound methods. For $m = 4$ and $n = 25$ the average run time was approximately 67 seconds.

6 Conclusions and Extensions

In this paper we have dealt with the problem of locating facilities along a given tree-like or segment-like high speed network with respect to the Euclidean distance.

- A facility location problem over a high speed network has been modeled, so that the competition with another (slower) existing alternative means is taken into account.
- Thanks to some convexity properties, a method for locating two facilities has been proposed when the high speed network is a straight line segment or a tree network.

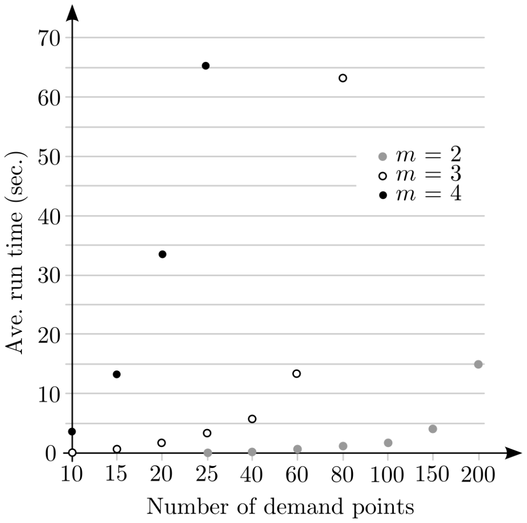

Fig. 6 Plot of average run time versus number of points for two, three, and four facilities.

- Geometric branch and bound methods have been extended to deal with the combinatorial part of the problem and can be used for the case of locating $m \geq 2$ facilities on a straight-line segment.
- Since the final goal was based on realistic applications, the proposed methodology demanded a deep theoretical analysis.
- The problem could be extended so that more constraints could be included. This way a better approximation to real situations could be made just by slightly modifying the methods and algorithms proposed here.

As a first extension, the problem can be generalized to other distances. For example, an application for the problem with rectangular distance is the location of bus stops in a line in Manhattan: The speed along the line is based on the rectangular distance (multiplied by a speed factor) as well as the speed for accessing the bus line. Some of our results can be transferred to arbitrary norms and also the big cube small cube approach can still be used when the norm used is differentiable. Moreover, we plan to extend our work to other types of metrics and to gauges. In this paper the allocation of trips was made on the basis of a binary variable. In order to extend these reasonings, the type of function that substitutes the 0-1 model should be discussed (see [4]), which is beyond the scope of this work and will be addressed in a future paper.

In this work we have treated edges as straight-line segments. Nevertheless, assuming these edges are sufficiently regular curves, one could apply homeomorphisms and their inverses so our reasoning can be extended to this sufficiently-regular-curve case.

Another extension is to allow different speed factors due to different speeds that might be possible in different parts of the network. To be more precise, in reality the trip between two facilities is often modeled in three phases: acceleration to maximum speed (Phase 1), journey at maximum speed (Phase 2), and braking in order to stop (Phase 3). If distances between facilities do not fall below a critical level, then the quota of Phase 1 and Phase 3 is small and our model assuming constant speed is appropriate. Therefore and for other reasons (e.g. acceptance by travelers) we are interested in computing solutions such that facilities do not fall below a minimum inter-facility space. This extension is to avoid situations in which two stations are located too close to each other, as one may say happens in Figure 4. To this end let $2\delta \geq 0$ be the minimum inter-facility space we would accept. Then an extension of Problem (3) can be formulated as

$$\max_{\mathcal{X} \subset \mathcal{N}} F^r(\mathcal{X}) \quad \text{s.t.} \quad \mathcal{N}(X_1, \delta) \cap \mathcal{N}(X_2, \delta) \ = \ \emptyset \quad \text{for all} \quad X_1 \neq X_2 \in \mathcal{X}, \tag{9}$$

where $\mathcal{N}(X, \delta)$ describes a neighborhood around X with radius δ. In the case where the given network is a line segment, the neighborhood of a facility becomes a smaller line segment, i.e. $\mathcal{N}(X, \delta) = [\max\{0, x - \delta\}, \min\{x + \delta, L\}] \times \{0\}$.

The Maximum Pair Covering Algorithm can be adapted in order to solve Problem (9). Roughly speaking, we need to exclude solutions from the finite dominating set CAND (see Corollary 3) that do not satisfy the inter-facility space condition. This can be done by deleting all solutions from CAND that lie in the halfspace $d(x, y) \leq \delta$ (note that if $(x, y) \in$ CAND then $x \leq y$). After deleting these points from CAND it could be possible that no optimal solution is included in CAND. Therefore the finite number of intersections between level sets C_{ij} and the straight line $d(x, y) = \delta$ must be added to CAND.

Beyond inter-facility space, there could be other reasons why it is not possible to locate facilities in certain sections of a high-speed network. For example it is possible that a line segment of the network passes under a river or crosses a natural park. Hence, we also extend Problem (3) to more general forbidden regions:

$$\max_{\mathcal{X} \subset \mathcal{N} \setminus \mathcal{R}} F^r(\mathcal{X}), \tag{10}$$

where $\mathcal{R}$ denotes the forbidden regions where the network cannot pass by. Note that formulation (10) also includes formulation (9) ($\mathcal{R} = \{(x, y) : \ d(x, y) \leq \delta\}$) and formulation (3) ($\mathcal{R} = \emptyset$). Just like for Problem (9), it is possible to adapt the Maximum Pair Covering Algorithm in order to deal with forbidden regions. To this end we remove points that lie in forbidden regions from the finite dominating set CAND and add finite intersections between $\mathcal{R}$ and the union of all level sets C_{ij}.

We have obtained optimal extreme points. In a more general formulation, one could explore a multi-objective version of the problem. As a second objective, one could consider the minimization of the distances between covered pairs (MinSum model, see [14]), or the maximization of the differences between the acceptance values and the distances h_{ij}^{kl} (MaxMin model). This new formulation will yield inner optimal points, which are more realistic than the points that are on the borders of the acceptance levels.

The fact that all mentioned extensions can be easily included in our models reinforces the validity of our reasoning from the applicability point of view.

Acknowledgments

This work was partially supported by the Future and Emerging Technologies Unit of EC (IST priority - 6th FP), under contract no. FP6-021235-2 (project ARRIVAL), by Ministerio de Educación, Ciencia e Innovación (Spain)/FEDER under project MTM2009-14243 and by Junta de Andalucía (Spain)/FEDER under excellence projects P09-TEP-5022 and FQM-5849.

References

1. Abellanas, M., Hurtado, F., Palop, B.: The heavy luggage metric. International Journal of Computational Geometry & Applications **18**, 295–306 (2008)
2. Abellanas, M., Hurtado, F., Sacristán, V., Icking, C., Ma, L., Klein, R., Langetep, E., Palop, B.: Voronoi diagram for services neighboring a highway. Information Processing Letters **86**, 283–288 (2003)
3. Bae, S.W., Kim, J.H., Chwa, K.Y.: Optimal construction of the city voronoi diagram. International Journal of Computational Geometry **19**, 95–117 (2009)
4. Berman, O., Krass, D., Drezner, Z.: The gradual covering decay location problem on a network. European Journal of Operational Research **151**, 474–480 (2003)
5. Blanquero, R., Carrizosa, E.: Continuous location problems and big triangle small triangle: Constructing better bounds. Journal of Global Optimization **45**, 389–402 (2009)
6. Cardinal, J., Collette, S., Hurtado, F., Langerman, S., Palop, B.: Optimal location of transportation devices. Computational Geometry **41**, 319–329 (2008)
7. Carrizosa, E., Rodríguez-Chía, A.: Weber problems with alternative transportation systems. European Journal of Operational Research **97**, 87–93 (1997)
8. Dearing, P., Francis, R.L., Lowe, T.J.: Convex location problems on tree networks. Operations Research **24**, 628–642 (1976)
9. Görke, R., Shin, C.S., Wolff, A.: Constructing the city voronoi diagram faster. International Journal of Computational Geometry & Applications **18**, 275–294 (2008)
10. Hansen, P., Peeters, D., Richard, D., Thisse, J.F.: The minisum and minimax location problems revisited. Operations Research **33**, 1251–1265 (1985)
11. Horst, R., Thoai, N.: DC programming: Overview. Journal of Optimization Theory and Applications **103**, 1–43 (1999)
12. Kirwan, F.: Complex Algebraic Curves. Cambridge University Press (1992)
13. Koolen, A., Tamir, A.: Covering problems. In: P. Mirchandani, R. Francis (eds.) Discrete Location Theory. Wiley-Interscience, New York-Chichester (1990)
14. Körner, M.C., Schöbel, A.: Weber problems with high–speed lines. TOP **18**, 223–241 (2010)
15. Laporte, G., Marín, A., Mesa, J.A., Perea, F.: Designing robust rapid transit networks with alternative routes. Journal of Advanced Transportation **45**, 54–65 (2011)

16. Laporte, G., Mesa, J.A., Ortega, F.A., Sevillano, I.: Maximizing trip coverage in the location of a single rapid transit alignment. Annals of Operations Research **136**, 49–63 (2005)
17. Laporte, G., Mesa, J.A., Perea, F.: A game theoretic framework for the robust railway transit network design problem. Transportation Research Part B **44**, 447–459 (2010)
18. Márquez-Diez-Canedo, J.: Fundamentos de Teoría de Optimización. Limusa, México (1987)
19. Ortúzar, J.D., Willumsen, L.G.: Modelling Transport. Wiley (2001)
20. Pfeiffer, B., Klamroth, K.: Unified model for weber problems with continuous and network distances. Computers & Operations Research **35**, 312–326 (2008)
21. Plastria, F.: GBSSS: The generalized big square small square method for planar single-facility location. European Journal of Operational Research **62**, 163–174 (1992)
22. Plastria, F.: Continuous covering location problems. In: Z.Drezner, H. Hamacher (eds.) Facility Location: Applications and Theory. Springer, Berlin-Heidelberg (2002)
23. Preparata, F., Shamos, M.: Computational Geometry, an introduction. Springer-Verlag (1985)
24. Schöbel, A., Scholz, D.: The Big Cube Small Cube Solution Method for Multidimensional Facility Location Problems. Computers & Operations Research **37**, 115–122 (2010)
25. Scholz, D.: Geometric branch & bound methods in global optimization: Theory and applications to facility location problems. Ph.D. thesis, Universität Göttingen (2010). To appear at Springer
26. Scholz, D., Schöbel, A.: The theoretical and empirical rate of convergence for geometric branch-and-bound methods. Journal of Global Optimization **48**(3), 473–495 (2010)