Weber's Problem with Forbidden Regions for Location and Transportation

Pierre HANSEN
GERAD and École des Hautes Études Commerciales, Montréal
pierreh@crt.umontreal.ca

Stéphane KRAU
Globvision, Montréal

Dominique PEETERS
Département de Géographie and CORE
Université de Louvain-la-Neuve

Jacques-François THISSE
CORE, Université de Louvain-la-Neuve
and CERAS, École des Ponts et Chaussées, Paris

June, 2000

Les Cahiers du GERAD

G–2000–26

Abstract

Weber's problem is to locate a facility in the Euclidean plane in order to minimize total transportation costs from that facility to a given set of users with a fixed demand. We consider the case where location in some regions is forbidden as well as transportation through the same or other regions. An exact branch-and-bound algorithm is proposed in two versions to solve this problem. In the first, transportation costs are only assumed to be non-decreasing in the distance traveled from the facility. In the second, they are assumed to be proportional to that distance, which leads to more precise bounds and new tests. Computational experience is reported: problems with up to 1000 users are solved exactly for the first time.

Key Words: Weber's problem, location, transportation, forbidden regions, algorithm.

Résumé

Le problème de Weber consiste à localiser une facilité dans le plan Euclidien de façon à minimiser le coût total de transport de cette facilité à un ensemble donné de clients pour satisfaire une demande fixe. Nous considérons le cas où certaines régions sont interdites pour la localisation ainsi que les mêmes régions ou d'autres pour le transport. Nous proposons une procédure de séparation et évaluation exacte, en deux versions, pour résoudre le problème. Dans la première, les coûts de transport sont seulement supposés non décroissants en la distance parcourue depuis la facilité. Dans la seconde, ils sont suppposés proportionnels à cette distance, ce qui fournit des bornes plus précises et de nouveaux tests. Nous présentons des résultats de calcul : des problèmes avec un millier d'usagers sont résolus exactement pour la première fois.

Mots-clés: Problème de Weber, localisation, transport, régions interdites, algorithme.

Acknowledgments: Work of the first author was supported by NSERC Grant GP0105574, FCAR Grant 95-ER-1048 and a grant from CETAI-HEC.

The classical Weber problem is to locate a facility in the Euclidean plane in order to minimize the sum of distances from that facility to a given set of users. This problem and its many extensions are the most studied of continuous location theory, see e.g., Hansen *et al.* (1987), Love, Morris and Wesolowsky (1988), Wesolowsky (1993) and Plastria (1995) for surveys.

In practice, it appears to be more the rule than the exception that location and / or transportation is subject to further constraints: mountains, parks, lakes and swamps are unfit for location, built regions could make it impossible or too costly, borders render it undesirable, land regulations may forbid it and the like. Similarly, natural obstacles or regions subject to high congestion should be avoided for transportation. Yet only a few papers have been devoted to versions of Weber's problem with forbidden regions for location and / or transportation. Note that a degenerate case of a region forbidden for transportation is a barrier to travel, e.g., a river which may only be crossed at bridges (Larson and Sadiq, 1983). Focusing first on locational constraints, we note that a few papers consider location within or outside of a specified region with a simple geometric form. Hurter, Schaefer and Wendell (1975) consider a feasible closed set. Katz and Cooper (1981) address the case of a disk infeasible for location and transportation. Watson-Gandy (1985) studies the case where maximum distance constraints are imposed.

Hansen, Peeters and Thisse (1982) consider the case where the area feasible for location is the union of a set of simple convex polygons, i.e., triangles and quadrangles. This entails little loss of generality as any allowed region with a more complicated form can easily be approximated by such a set of polygons, to the desired precision. Moreover, the number of polygons does not affect much resolution time. It follows from a result of Schaefer and Hurter (1974) that the optimal solution of this constrained Weber problem is either the solution for the corresponding unconstrained problem (when it belongs to a feasible zone) or visible from that solution, i.e., such that a straight line segment joining the unconstrained solution to the constrained one has no feasible interior point. Moreover, the objective function of Weber's problem on the visible frontier of a convex polygon is quasi-convex, which allows easy obtention of the constrained optimal solution by a few one-dimensional searches. Problems with 100 users and up to 100 feasible triangles or quadrangles are solved in a few seconds of computing time. Aneja and Parlar (1994) consider the case where the area forbidden to location consists of a set of simple polygons. As either the feasible or the infeasible area can be easily decomposed in that way, the algorithms proposed in both cases are equivalent. These algorithms assume transportation costs proportional to distance. This assumption is relaxed in Hansen, Peeters, Richard and Thisse (1985), where transportation costs are only supposed to be non-decreasing in distance. The problem then pertains to global optimization. A branch-and-bound method in continuous variables, called *Big Square Small Square* (BSSS) is proposed. Various extensions of this method are given in Hansen, Peeters and Thisse (1981), Plastria (1992), and Hansen, Jaumard and Krau (1995). Obstacles to transportation have been studied for some time. The shortest path with polygonal obstacles from a fixed origin (e.g. the facility's location) to a fixed destination (e.g. user's location) is obtained by building a visibility graph (e.g. Wangdalh, Pollock and Woodward, 1974, Viegas and Hansen, 1985) with vertices associated to origin(s), destination(s) and extreme points of obstacles and straight-line edges joining mutually visible pairs of vertices (in the sense here that the straight line contains no interior infeasible point). Edges are weighted by their length and e.g. Dijkstra's (1959) algorithm used to find shortest paths and corresponding distances.

Aneja and Parlar (1994) also consider Weber's problem with convex polygonal regions forbidden for both location and transportation. They propose a heuristic of simulated annealing type to solve it, using the visibility graph. Butt (1994) and Butt and Cavalier (1996) propose another heuristic based on an adaptation of Weiszfeld's (1937) iterative scheme. It converges to a local optimum and if started many times from randomly chosen feasible points is likely to find the optimal solution of small to medium size instances, without however proving its optimality. To the best of our knowledge, no exact algorithm has yet been proposed for Weber's problem with forbidden regions for both location and transportation. It is the purpose of this paper to fill in that gap.

The problem is stated in the next section, in two versions according to whether transportation costs are assumed to be nondecreasing in the distance traveled, or proportional to that distance. An algorithm which generalizes the Big Square Small Square algorithm of Hansen, Peeters, Richard and Thisse (1985) is proposed for the second case. Modifications to be brought for it to apply to the first one are also specified. In Section 1, we give the problem formulation. In Section 2, subroutines used in various steps of the algorithm are detailed, and in Section 3 the algorithm itself is presented. Computational experience is reported in Section 4.

1 Problem Formulation

The assumptions of Weber's problem with forbidden regions for location and transportation are the following:

(i) There are n customers (or groups of customers) located at given demand points a^j in the Euclidean plane $\mathbb{R}^2$. Weights w_j for $j \in \{1, 2, \ldots, n\}$ are associated with these points and correspond to the magnitude of the demands.

(ii) There are m_1 disjoint regions P_i forbidden for location, but not for transportation. These regions are assumed to be the interior of (not necessarily convex) polygons. They are described by lists of consecutive endpoints of their sides, denoted by $\ell^{i1}, \ell^{i2}, \ldots, \ell^{i\ell_i}$ for $i \in \{1, 2, \ldots, m_1\}$. Let $P = \sum_{i=1}^{m_1} P_i$ denote the area forbidden for location only.

(iii) There are m_2 disjoint regions Q_i forbidden both for location and for transportation, also assumed to be the interior of (not necessarily convex) polygons. They are described as the polygons P_i by lists of consecutive endpoints of their sides, denoted by $c^{i1}, c^{i2}, \ldots, c^{in_i}$ for $i \in \{1, 2, \ldots, m_2\}$. Let $Q = \sum_{i=1}^{m_2} Q_i$ denote the area forbidden for location and transportation.

(iv) There is a single facility to be located at a feasible point s, i.e., $s \in \mathbb{R}^2 \setminus (P \cup Q)$.

(v) Let $\overline{d}(a^j, s)$ denote the length of the shortest path from a^j to s which avoids all regions Q_i forbidden to transportation. We call $\overline{d}(a^j, s)$ the *feasible distance* from a^j to s. The facility is to be located in order to minimize the total transportation cost $f(s)$ defined by

$$f(s) = \sum_{j=1}^{n} f_j(s) = \sum_{j=1}^{n} w_j g_j\left(\overline{d}(a^j, s)\right) \tag{1}$$

where $g_j\left(\overline{d}(a^j, s)\right)$ is a non decreasing (not necessarily continuous) function of the feasible distance $\overline{d}(a^j, s)$. We assume further that $\overline{d}(a^i, a^k)$ is finite for all $j, k \in \{1, 2, \ldots, n\}$ to ensure existence of a solution.

The nonlinear functions $g_j\left(\overline{d}(a^j, s)\right)$ are introduced to allow modeling of economies of scale in distance traveled, increased costs when the night is spent away from the origin or destination, tariffs for transportation with staircase structure in the distance traveled and the like.

We will mainly consider the particular case when the cost is linear in the feasible distances, i.e.

$$f(s) = \sum_{j=1}^{n} f_j(s) = c \sum_{j=1}^{n} w_j \overline{d}(a^j, s) \tag{2}$$

where c is a constant. Then specific properties lead to a more efficient algorithm.

Without loss of generality, the region in which the facility is located can be restricted to the convex hull of demand points and endpoints of sides of polygons P_i and Q_i. Indeed, a simple argument shows that any point s outside this convex hull is dominated by a point j' on the border of it, i.e. there is such a point s' for which $\overline{d}(a^j, s') \leq \overline{d}(a_j, s)$ for all $j \in \{1, 2, \ldots, n\}$ the inequality being strict for at least one j. It is not true, however, contrary to the case when there are no restrictions on location or transportation, that there always is an optimal solution in the convex hull of the demand points (or in the smallest rectangle containing these points as stated in Step 1 of Algorithm 2 of Aneja and Parlar, 1994). Indeed, consider the example of Figure 1. There are four customers located at $a^1 = (1, 1)$, $a^2 = (1, 2)$, $a^3 = (7, 2)$ and $a^4 = (7, 1)$. A single region Q_1 is forbidden to location and transportation. It is a square with consecutive endpoints of sides $c^1 = (2, 0)$, $c^2 = (2, 4)$, $c^3 = (6, 4)$ and $c^4 = (6, 0)$. All weights w_j are equal to 1, and all functions $f_j(s) = \overline{d}(a^j, s)$, i.e., $c = 1$ in (2). The convex hull of the demand points is the rectangle with consecutive endpoints of sides a^1, a^2, a^3 and a^4. Any point s on the lower side of Q_1, i.e., on the line joining $c^1 = (2, 0)$ to $c^4 = (6, 0)$, has a cost of $8 + 2\sqrt{2} + 2\sqrt{5}$ and is optimal. Indeed, it is easy to see that the optimal solution s^* cannot be above the line passing through a^2 and a^3. Then if s is to the left of Q_1, the shortest paths from a^3 and a^4 to s go through c^4 and c^1 and have lengths $\sqrt{5} + 4 + \overline{d}(c^1, s)$ and $\sqrt{2} + 4 + \overline{d}(c^1, s)$ respectively. The sum of these distances is thus a constant plus twice the distance from c^1 to s. The problem thus reduces to a Weber problem with three customers at a^1, a^2 and

c^1 and weights $w_1 = w_2 = 1$ and $w(c^1) = 2$. From the majority theorem (Witzgall, 1964) an optimal solution s^* is at c^1. A similar argument holds if s is to the right of Q_1 and it follows that all optima are points on the line segment joining c^1 to c^4. This completes the proof.

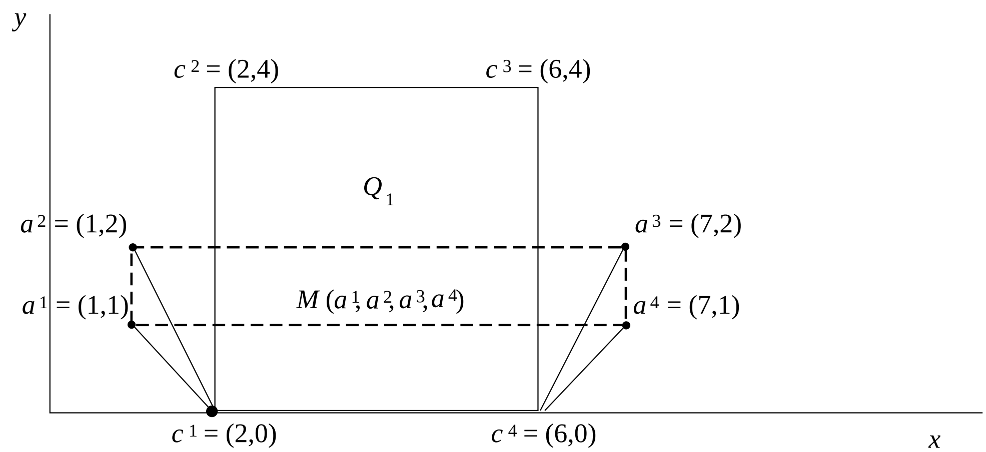


Figure 1: A problem for which the optimal location does not belong to the convex hull of the demand points.

2 Feasible Distance

Before presenting the new algorithm, four preliminary problems must be addressed:

(i) **visibility of a pair of points** s, s': find whether two feasible points s, s' are mutually visible, i.e., such that the straight-line segment joining them contains only feasible points for transportation;

(ii) **feasible distance** from a point s to a square S: find the shortest path from s to a point of S which contains only feasible points for transportation;

(iii) **dominant point**: find if there is a feasible point v_k in a given finite set V such that

$$\overline{d}(v_k, s) \leq \overline{d}(v_\ell, s)$$

for all $v_\ell \in V$, $\ell \neq k$ and all $s \in S$, a given square;

(iv) **visibility of a square** S **from a point** s: find if all points of a square S are visible from a point s such that $s \notin S \cup Q \cup P$.

The difficulty of these problems is increasing. As exact solution of problem *(iii)* would be too time consuming, only a sufficient condition for a point v_k to be dominant will be used. We next present algorithms for these four problems is turn.

Subroutine Visibility

INPUT: Feasible points $s = (s_1, s_2)$, $s' = (s'_1, s'_2)$, polygons Q_i, $i = 1, 2, \ldots, m_2$ defined by their lists of consecutive endpoints of sides.

QUERY: Are s and s' mutually visible?

OUTPUT: YES if s' is visible from s (and conversely), NO otherwise.

Assume without loss of generality $s_1 \leq s'_1$.

For all Q_i, compute abcissæ and ordinates of the extreme points of the smallest rectangle containing Q_i:

$$\underline{c}_1^i = \operatorname*{Min}_{k=1,2,\ldots,n_i} c_1^{ik} \quad , \quad \overline{c}_1^i = \operatorname*{Max}_{k=1,2,\ldots,n_i} c_1^{ik}$$
$$\underline{c}_2^i = \operatorname*{Min}_{k=1,2,\ldots,n_i} c_2^{ik} \quad , \quad \overline{c}_2^i = \operatorname*{Max}_{k=1,2,\ldots,n_i} c_2^{ik}.$$

Consider all Q_i in turn and

(a) if $\underline{c}_1^i > s'_1$ or $\overline{c}_1^i < s_1$ or $\underline{c}_2^i > \max(s_2, s'_2)$ or $\overline{c}_2^i < \min(s_2, s'_2)$, proceed to the next polygon;

(b) if $s_2 > s'_2$ and $(\underline{c}_1^i, \underline{c}_2^i)$ is above the line going through s and s' or $(\overline{c}_1^i, \overline{c}_2^i)$ is below this line, proceed to the next polygon;
if $s_2 < s'_2$ and $(\underline{c}_1^i, \overline{c}_2^i)$ is above the line going through s and s' or $(\overline{c}_1^i, \underline{c}_2^i)$ is below this line, proceed to the next polygon;

(c) consider in turn all sides $[c^{ik}, c^{ik+1}]$ for $k = 1, 2, \ldots, n_i - 1$ and $[c^{in_i}, c^{i1}]$ of Q_i. If a side intersects the line segment $[s, s']$, return with the answer NO; otherwise, proceed to the next polygon;

(d) if steps *(a)*–*(c)* have been applied to all polygons, return with the answer YES.

This subroutine is illustrated on Figure 2; polygons Q_1, Q_2 and Q_3 are shown not to intersect (s, s') by tests *(a)*, *(b)* and *(c)* respectively. Correctness of the subroutine follows from the fact that if $[s, s']$ intersects a polygon Q_i it must intersect a side of that polygon as s and s' are feasible.

Note that as the same polygons Q_i will be considered over and over again the $\underline{c}_1^i, \overline{c}_1^i, \underline{c}_2^i, \overline{c}_2^i$ will be computed once and for all at the outset.

The standard tool to compute feasible distances between pairs of points is the visibility graph $G = (V, E)$. It is defined as follows for the case under study:

(a) the set V contains $n + \sum_{i=1}^{m_2} n_i$ vertices, associated with the n demand points a^j (with indices $1, 2, \ldots, n$) and the $\sum_{i=1}^{m_2} n_i$ endpoints c^{ik} of sides of the m_2 polygon Q_i forbidden for location and transportation (with indices $n+1, n+2, \ldots, n+\sum_{i=1}^{m_2} n_i$). For simplicity of notation we designate vertex v_j and the corresponding point c^{ik} by either symbol according to context;

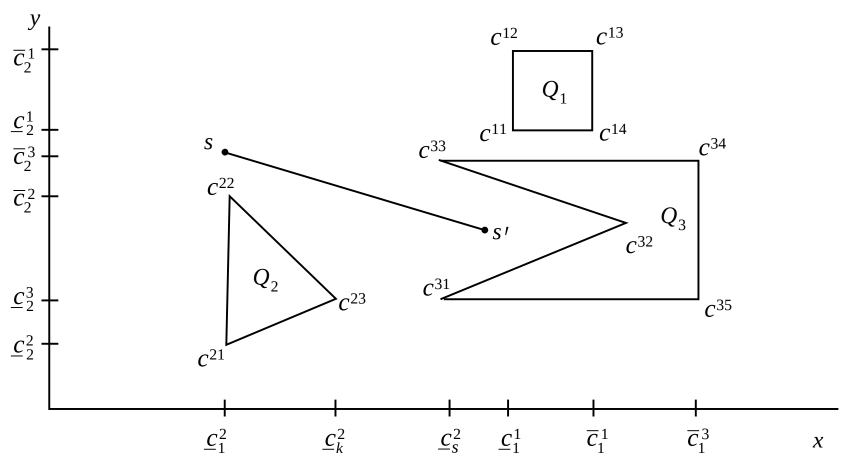


Figure 2: Illustration of the visibility subroutine

(b) the set E contains edges between all pairs of vertices $\{v_k, v_\ell\}$ of V containing at most one vertex associated with a demand point and such that v_k and v_ℓ are mutually visible;

(c) edges $\{v_k, v_\ell\}$ are weighted by the length of the (feasible) straight line segment joining the points associated with v_k and v_ℓ.

Once the visibility graph has been built, Dijkstra's algorithm (1959) can be applied n times to find feasible distances $\overline{d}(a^j, c^{ik})$ for all a^j and c^{ik}.

Subroutine Feasible Distance

INPUT: Demand point a^j, polygons $Q_i, i = 1, 2, \ldots, m_2$, vector of feasible distances $\overline{d}(a^j, c^{ik})$ between a^j and all endpoints c^{ik} of sides of polygons Q_i, square S given by the clockwise list of its vertices s^1, s^2, s^3, s^4 beginning with the leftmost lowest vertex.

QUERY: Can S be reached from a^j and, if so, give a lower bound on the feasible distance from a^j to S?

OUTPUT: SQUARE WITHIN A POLYGON Q_i; otherwise feasible distance $\overline{d}(a^j, S)$ from a^j to square S, i.e., minimum feasible distance from a^j to a point of S.

We first check whether $S \subset Q_i$ or $S \cap Q_i \neq \emptyset$ for $i = 1...m_2$.
Set SPI = .FALSE. (indicator variable true if S is shown to be partially infeasible).
Consider all Q_i in turn and

(a) if $\underline{c}_1^i > s_1^4$ or $\overline{c}_1^i < s_1^1$ or $\underline{c}_2^i > s_2^2$ or $\overline{c}_2^i < s_2^1$ proceed to the next polygon;

(b) check if a side of Q_i intersects a side of S; if so, set SPI = .TRUE. and go to *(d)*;

(c) check if s^1 is within Q_i. This can be done by drawing a ray from s^1 and counting the number of sides of Q_i which it intersects; if it is odd S is within Q_i, if it is even S is outside Q_i, by a classical result (e.g. Courant and Robbins, 1958, p. 269). If

$s^1 \in Q_i$ return SQUARE WITHIN A POLYGON and stop. Otherwise proceed to the next polygon;

We next compute $\overline{d}(a^j, S)$, which is defined as S is not completely forbidden to travel.

(d) For all $v_k \in V$ with $k > n$ for which $v_k \notin S$ compute the Euclidean distance $d(v_k, c_S)$ where c_S is the center of square S (thus ignoring temporarily constraints on transportation) and $\overline{d}(a^j, v_k) + d(v_k, c_S)$. Rank these quantities (and the corresponding v_k) in order of non-decreasing values.

(e) Set $\overline{d}(a^j, S) = M$ an arbitrarily large value. Consider in turn the vertex associated with a^j and each v_k with $k > n$ and $v_k \notin S$ in the order obtained at step *(d)*.

Determine the point $s(v_k)$ of S closest to v_k ignoring constraints on transportation. The straight lines supporting the sides of S partition $\mathbb{R}^2$ into 9 regions: S itself, 4 side regions and 4 corner regions. The point $s(v_k)$ is the orthogonal projection of v_k on S if v_k belongs to a side region and the corner of S in the corner region containing v_k otherwise. Compute

$$\overline{d}(a^j, v_k) + d(v_k, S) = \overline{d}(a^j, v_k) + d(v_k, s(v_k)) \tag{3}$$

where $d(.,.)$ is the Euclidean distance.

(e1) if

$$\overline{d}(a^j, v_k) + d(v_k, c_S) > \overline{d}(a^j, S) + \frac{\sqrt{2}}{2} d(s^1, s^2) \tag{4}$$

return $\overline{d}(a^j, S)$ and stop (no point of S is at a feasible distance from j smaller than this feasible distance when passing through v_k; moreover, due to ranking, the same holds for the remaining vertices v_k on the list);

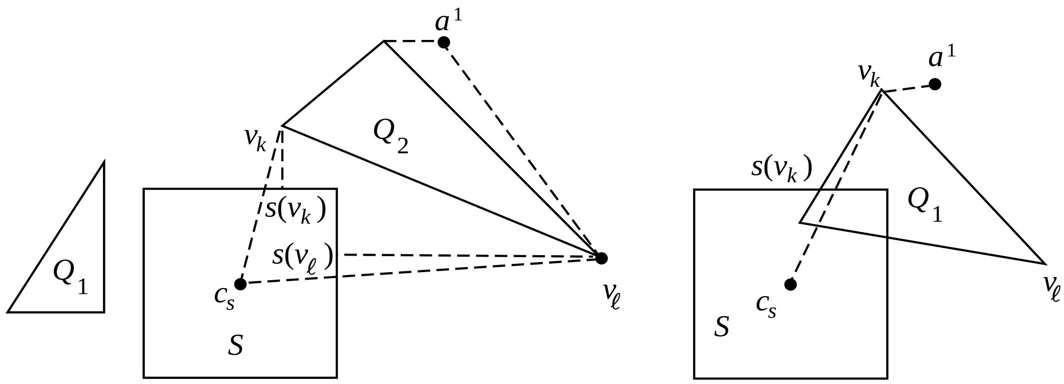
(e2) if $\overline{d}(a^j, v_k) + d(v_k, S) \geq d(a^j, S)$ proceed to the next vertex v_k on the list; otherwise go to the next substep;

(e3) determine if v_k is visible from $s(v_k)$. If so, set $\overline{d}(a^j, S) = \overline{d}(a^j, v_k) + d(v_k, S)$, then proceed to the next vertex v_k on the list; otherwise go to the next substep;

(e4) if SPI = .FALSE. proceed to the next vertex v_k on the list; otherwise consider polygon Q_i for which v_k is an extreme point and the two sides associated with v_k. Check whether these sides intersect S or not. If they do not, proceed to the next vertex v_k on the list; if one or both do, let $s(v_k)$ be the intersection point of one of those sides of Q_i with a side of S closest to v_k. Check if $\overline{d}(a^j, v_k) + d(v_k, s(v_k)) < \overline{d}(a^j, S)$, if not set $\overline{d}(a^j, S) = \overline{d}(a^j, v_k) + d(v_k, s(v_k))$, and proceed to the next v_k on the list.

(e5) If all vertices v_k on the list have been considered return $\overline{d}(a^j, S)$.

This subroutine is illustrated on Figure 6a for the case SPI = .FALSE. and Figure 6b for the case SPI = .TRUE. In Figure 3a, $S \cap Q_1 = \emptyset$ is shown in step *(a)*, $S \cap Q_2 = \emptyset$ in step *(b)*. Justification follows from easy geometric arguments.



(a) SPI = .FALSE. (b) SPI = .TRUE.

Figure 3: Illustration of feasible distance subroutine.

Subroutine Dominant Point

INPUT: Demand point a^j, polygons $Q_i, i = 1, 2, \ldots, m_2$, vector of feasible distances $\overline{d}(a^j, c^{ik})$ between a^j and all endpoints c^{ik} of sides of polygons Q_i, square S given by the clockwise list of its vertices s^1, s^2, s^3, s^4 beginning with the leftmost lowest vertex.

QUERY: is there a point $v_k \in V$ satisfying a sufficient condition (see OUTPUT below) for all feasible shortest paths from a^j to a point $s \in S$ to go through v_k?

OUTPUT: YES if there is a point v_k of V, called the *dominant point* of a^j on S, such that

$$\overline{d}(a^j, v_k) + \max_{s' \in S} d(v_k, s') < \overline{d}(a^j, v_\ell) + \overline{d}(v_\ell, s)$$

for all $v_\ell \in V$, $\ell \neq k$ and all $s \in S$. NO, otherwise.

Consider all Q_i in turn and

(a) For all $v_k \in V$ with $k > n$ for which $v_k \notin S$ compute the Euclidean distance $d(v_k, c_S)$ where c_S is the center of square S (thus ignoring temporarily constraints on transportation) and $\overline{d}(a^j, v_k) + d(v_k, c_S)$. Rank these quantities (and the corresponding v_k) in order of non-decreasing values.

(b) Set $\overline{d}(a^j, S) = M$, $DIST = M$, arbitrarily large values. Consider in turn each a^j and each v_k with $k > n$ and $v_k \notin S$ in the order obtained at step *(a)*. Determine the point $s(v_k)$ of S closest to v_k ignoring constraints on transportation. Compute:

$$\overline{d}(a^j, v_k) + d(v_k, S) = \overline{d}(a^j, v_k) + d(v_k, s(v_k)) \quad (5)$$

where $d(.,.)$ is the Euclidean distance. Set VIS = .FALSE. (Indicator variable true if a vertex v_k such that S is visible from v_k has been found.)

(b1) if VIS = .FALSE.:
if v_k is visible from $s(v_k)$:

- $*$ if v_k is visible from all points in S (subroutine visibility on a square, see below) then set VIS = .TRUE. and compute:

$$DIST = \overline{d}(a^j, v_k) + \operatorname*{Max}_{s \in S} d(v_k, s)$$

Set $v_k^* = v_k$, then proceed to the next vertex v_k on the list;

- $*$ else return NO.

Otherwise proceed to the next vertex v_k on the list;

(b2) if VIS = .TRUE.:

- $*$ if:

$$\overline{d}(a^j, v_k) + d(v_k, c_S) - \frac{\sqrt{2}}{2} d(s^1, s^2) > DIST$$

then return YES and v_k^*.

- $*$ if v_k is visible from $s(v_k)$ and $\overline{d}(a^j, s(v_k)) < DIST$ then return NO;
- $*$ otherwise proceed to the next vertex v_k on the list;

(c) return YES and v_k^*.

Remark. The conditions used in the definition of the output as well as in step (b2) are not the strongest possible for the query to hold. They are used for ease of computation, e.g. exploitation of the ranking at the v_k^j of L.

Subroutine constrained Weber problem on a square

INPUT: a set of dominant points v_k^j in list (L), the feasible distances $\overline{d}(a^j, v_k^j)$ from the corresponding demand points a^j, a square S with $S \cap (P \cup Q) = \emptyset$.

QUERY: solve the constrained Weber problem on S with demand points whose corresponding dominant point is in L.

OUTPUT: $f^L(s_L^*)$ the optimal value of the constrained Weber problem reduced to the set L, in S.

The objective function of the reduced Weber's problem is convex (the sum of weighted Euclidean distances is convex). Details of the algorithm are as follows:

(1) compute gradients g^1, g^2, g^3, g^4 of the objective function at the corners s^1, s^2, s^3, s^4 of the square S (See Figure 4);

(2) compute the minimum point of f^L on each side of S :
As the objective function is convex in any direction, the minimum point ($[s^1, s^2]$ and $[s^3, s^4]$) follows the following rules:
if $g_x^1 * g_x^2 \geq 0$: the minimum point of $[s^1, s^2]$ is s^1 or s^2;
if $g_x^3 * g_x^4 \geq 0$: the minimum point of $[s^3, s^4]$ is s^4 or s^3;
if $g_x^1 * g_x^2 < 0$: the minimum point of $[s^1, s^2]$ is between s^1 and s^2 and an unidimensional search is used to find it;

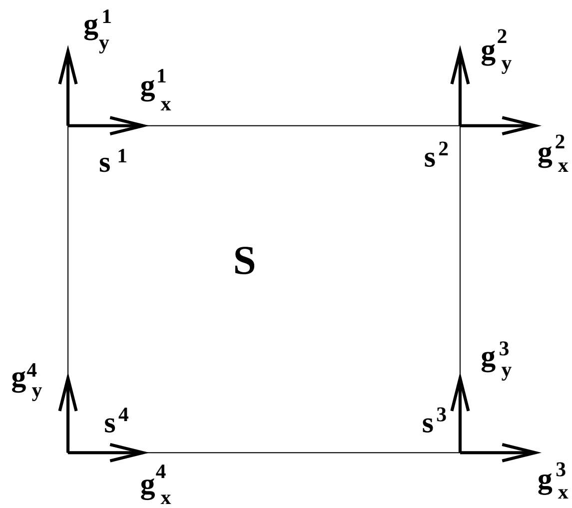


Figure 4: Gradients at the corners of S

if $g_x^3 * g_x^4 < 0$: the minimum point of $[s^3, s^4]$ is between s^3 and s^4 and an unidimensional search is used to find it.
A similar approach may be followed for sides $[s^2, s^3]$ and $[s^4, s^1]$;

(3) determine the minimum point s_L^s on the sides of S. (Comparing the different minimum points on each side of S.);

(4) perform one iteration of Weiszfeld's algorithm (gradient method with predefined step) starting from s^s. If this point is outside S_i then $s_L^* = s_L^s$. Otherwise resume application of Weiszfeld's algorithm; s_L^* is the optimal point obtained, which is necessarily in S (because of convexity and the decreasing values of successive points given by the Weiszfeld algorithm).

Subroutine visibility on a square

INPUT: A point s in the plane, polygons Q_i with $i = 1...m_2$ and a square S with its four corners s^1, s^2, s^3, s^4 such that $s \notin S$.

QUERY: is S visible from s?

OUTPUT: YES, if all points in S are visible from s; NO, otherwise.

We assume that s is in a side region of S, (a similar approach may be applied if s belongs to a corner region). Let d^1, d^2, d^3 and d^4 denote the four corners of the rectangle R obtained by intersecting the side region of S containing s with the half plane containing S obtained by drawing a parallel to the closest side of S through s. Let d^3 and d^4 be the two corners of R belonging to S.

Consider all Q_i in turn and

(a) if $\overline{c}_1^i < d_1^1$ or $\underline{c}_2^i > d_2^2$ or $\underline{c}_1^i > d_1^3$ or $\overline{c}_2^i < d_2^1$, proceed to the next polygon;

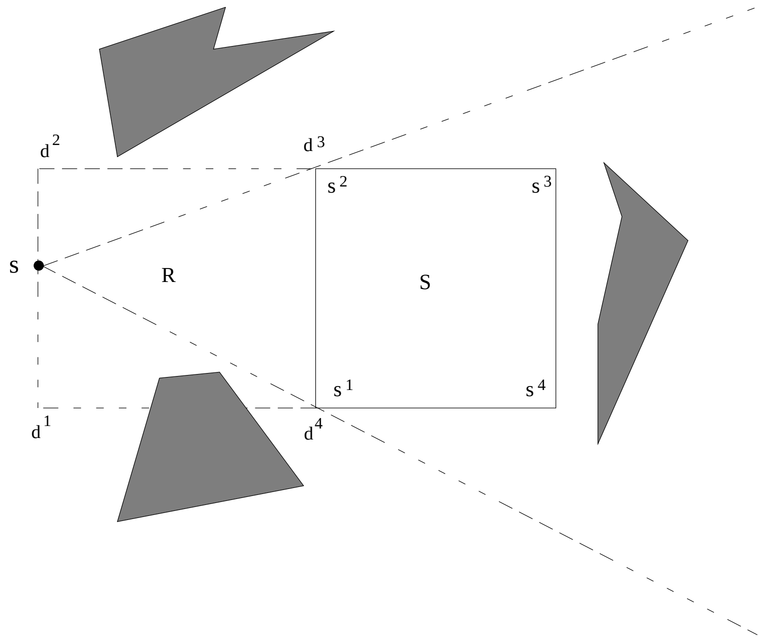


Figure 5: Example of visibility on S

(b) if Q_i has a vertex in the triangle with vertices s, d^3 and d^4 or a side which intersects line (s, d^3) or line (s, d^4) return NO, S is not visible from s. Otherwise, proceed to the next polygon; if all Q_i have been considered, return YES.

Remark. We assume implicitly $S \cap Q = \emptyset$ in this subroutine; however, if this condition is not satisfied one can consider a relaxation of the given problem in which it holds when computing a lower bound on the total distance traveled.

3 Algorithm

As mentioned above, this algorithm is an extension of the constrained case of 'BSSS' developed by Hansen, Peeters, Richard and Thisse (1985). It is a branch and bound method in continuous space and proceeds by:

(1) partitioning the plane into squares with sides parallel to the coordinate axes, starting with the smallest square containing all demand points and polygons $(P \cup Q)$;

(2) deleting squares inside polygons of $(P \cup Q)$;

(3) computing lower bounds $\underline{f}^i$ on $\underline{f}$ on the remaining squares S_i and deleting those squares for which the lower bound is greater than or equal to the value f_{opt} of the best solution s_{opt} known so far;

(4) computing the value of a feasible point in each remaining square S_i and updating f_{opt} and s_{opt} if a point with a smaller value than that of the incumbent is found;

(5) choosing the remaining square S_i with smallest lower bound $\underline{f}^i$ and partitioning it into four new squares;

(6) iterating the tests on the new squares S_i so obtained until the relative error $(f_{opt} - \underline{f}^i)/f_{opt}$ is smaller than a given tolerance ϵ.

'BSSS' uses a best first strategy. Assuming $\underline{f}^i$ is precise in the limit, i.e., $f(s) - \underline{f}^i$ goes to 0 for all $s \in S_i$ when the diagonal of S_i goes to 0, its convergence follows from general conditions; see e.g. in Horst and Tuy (1993).

The detailed rules of the new algorithm proposed are as follows:

(a) Initialization $S_1 \leftarrow S$; (the smallest square containing demand points and polygons $(P \cup Q)$);
$I \leftarrow \{1\}$ (I is the index set of unsolved subproblems); $I_{new} \leftarrow \{1\}$ (I_{new} is the index set of subproblems for which a lower bound has not yet been computed);
$s_{opt} \leftarrow$ randomly generated point in S (if one can be found, else $s_{opt} \leftarrow$ m.v., i.e., a missing value);
$f_{opt} \leftarrow f(s_{opt})$ if a point in S has been found, else $f_{opt} \leftarrow \infty$;

(b) Feasibility test For all $i \in I_{new}$, compute $S_i \cap (Q \cup P)$ and delete i from I_{new} if $S_i \cap (Q \cup P) = \emptyset$;

(c) Optimality test
For all $i \in I_{new}$, compute a lower bound $\underline{f}^i$ on $f(s)$ for $s \in S_i$ (using rules of step d below) and delete i from I_{new} if $\underline{f}^i \geq f_{opt}$;

(d) Lower bound computation
(This lower bound is computed in two steps: feasible distances to S_i are used for demand points a^j without dominating points; then dominating points, listed in L, are used to solve exactly the Weber problem constrained to S_i, but neglecting other constraints, for all other demand points.)
Set $L = \emptyset$ and $\underline{f}^i = 0$.
Check for all demand points a^j if they have a dominant point v_k^j ($v_k^j \in V$) on S_i: if so, set $L \leftarrow L \cup v_k^j$ (dominant point subroutine); otherwise compute the weighted feasible distance $(\overline{d}(a^j, S_i) = \overline{d}(a^j, v_k) + d(v_k, S_i))$ from a^j to S_i (feasible distance subroutine) and add it to the lower bound:
$\underline{f}^i = \underline{f}^i + w_j g_j(\overline{d}(a^j, S_i))$.
Compute on S_i the optimal solution s_L^* and value $f^L(s_L^*)$ of the constrained Weber problem reduced to the set L (constrained Weber problem on a square subroutine) and add this value to the lower bound:
$\underline{f}^i = \underline{f}^i + f^L(s_L^*)$.

(e) Improved solution test
For all $i \in I_{new}$ compute a feasible solution $f(s_i)$ with $s_i \in S_i \setminus (Q \cup P)$ and if $f(s_i) < f_{opt}$ set $s_{opt} \leftarrow s_i$, $f_{opt} \leftarrow f(s_i)$;

(f) Branching and stopping conditions
If $I = \emptyset$, stop: the problem is infeasible; else select S_i such that $\underline{f}^i = \min_{j \in I} \underline{f}^j$.

If $(f_{opt} - \underline{f}^i)/\underline{f}^i \leq \epsilon$; stop, an ϵ-optimal solution s_{opt} of value f_{opt} has been found, else partition S_i into four new equal squares $S_{|I|+j}$, $j = 1, 2, 3, 4$;
Remove i from I and set I_{new} equal to the set of indices of the new squares;
Return to (b); □

Remark 1 To make step (f) efficient, the lower bound $\underline{f}^i$ of subproblems in the current list, i.e., those with $i \in I$ are kept in a heap. In step (c) subproblems such that $\underline{f}^i \geq f_{opt}$ can be deleted in order to save memory space; if this is done, a double-ended priority queue is used to store the $\underline{f}^i$; see Atkinson *et al.* (1986) for the definition of this data structure.

Remark 2. A more precise lower bound could be obtained by taking into account the constraints that s must belong to $S_i \setminus (P \cup Q)$ when considering the Weber problem reduced to L. One should then use repeatedly the algorithm of Hansen, Peeters and Thisse (1982), which might be time-consuming.

Remark 3. In step (e) the center of the square is first tested for feasibility. If it is not feasible one can draw points at random in S_i and check their feasibility, or consider intersection points of sides of S_i and some polygon of P and Q. The algorithm still works if no feasible point is found for some or even all S_i at some iteration(s).

Remark 4. If transportation costs are nonlinear functions increasing in the distance traveled, instead of linear functions of that distance, the proposed algorithm still works with a simple modification: one does not seek dominant points. The resulting bound being less precise, only instances with a moderate number of users would be solvable in reasonable time, as is the case with BSSS, unless particular properties of the cost functions are exploited.

4 Numerical results

In this section we assume that transportation costs are proportional to distance. All problems are solved on a SPARC SUN 20 workstation and the program is written in C. We first consider a test problem from the literature (Aneja and Parlar, 1994, represented on Figure 6). This problem has 18 demand points and 12 regions forbidden both for location and transportation. The tolerance $\epsilon = (f_{opt} - \underline{f}^i)/\underline{f}^i$ is equal to 10^{-5}.

Results are reported in Table 1, which gives, in successive columns the number of forbidden regions (as numbered on Figure 6), coordinates of the facility, objective function value, time to reach the optimum solution and prove its optimality, number of iterations and number of squares for which an optimal solution was found (other squares were eliminated through bounding). Optimal solutions are the same as those obtained by Aneja and Parlar (1994) except for the cases of 10 and 12 forbidden regions.

Table 2 gives results on the efficiency of the algorithm, for problems derived from 12 forbidden regions example but with up to 1000 demand points. Five problems are solved for each series and averages are given.

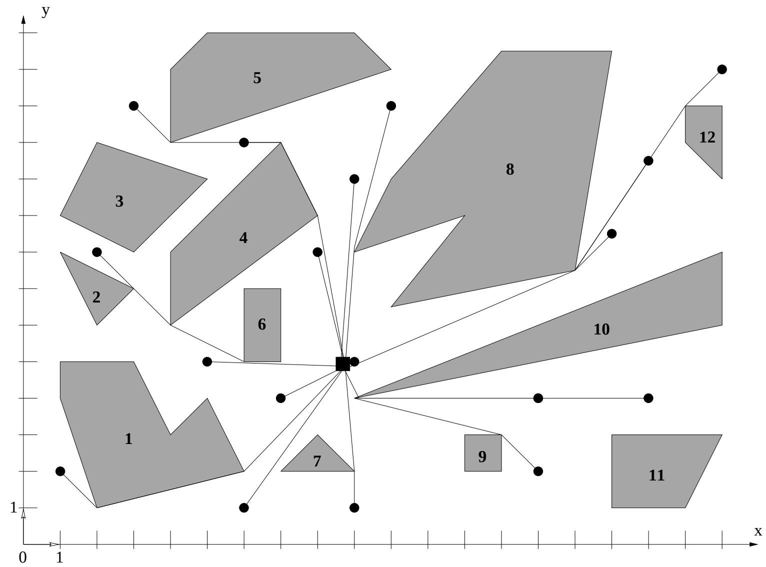


Figure 6: Aneja and Parlar example.

Table 1: Results of Aneja and Parlar example.

nb. of forbidden regions	x^*	y^*	F^*	cpu (sec.)	it	opt
1-12	8.767	4.981	119.139	2.55	28	20
1-10	8.767	4.981	119.105	2.27	28	20
1-8	9.188	5.486	116.398	2.14	27	32
1-6	9.266	6.253	114.561	1.01	25	20
1-4	9.217	6.153	113.766	0.69	27	21
1-2	9.037	6.115	111.689	0.17	13	9
$\emptyset$	8.913	6.356	110.007	.01	1	4

Table 2: Results for $n = 100 \dots 1000$, $m_2 = 12$

n	m_2	iterations		time		F^*		squares opt.	
		μ	σ	μ	σ	μ	σ	μ	σ
100	12	47.6	3.	23.7	1.7	776.1	26.7	20.	4.9
200	12	52.4	9.9	55.5	11.6	1533.2	55.7	11.8	2.9
300	12	50.2	2.1	83.8	3.7	2319.3	54.4	10.4	2.6
400	12	52.2	2.6	126.1	7.6	3103.3	71.2	6.	1.6
500	12	52.2	2.5	166.7	9.4	3901.9	76.8	4.6	1.2
600	12	55.4	7.4	230.7	42.1	4680.6	62.5	4.2	2.4
700	12	56.2	6.	287.4	41.6	5463.2	63.4	3.2	1.7
800	12	59.8	5.1	376.9	45.8	6271.2	53.9	2.	1.6
900	12	56	4.2	415.3	48.5	7047.8	52.	2.4	1.6
1000	12	60.4	8.3	544.2	102.9	7838.4	39.4	1.2	1.1

Computing time increases linearly and smoothly with the number of demand points whereas the number of iterations remains roughly the same.

The last table concerns sensitivity of the algorithm to the number of constraints. We present results obtained for problems with 100 demand points and 50 or 100 regions forbidden to travel (randomly distributed in size and in the plane). Each series is composed of 5 test problems. Computing times augment relative to the 12 constraints case by a factor of about 5 to 20, but remain reasonable.

Table 3: Results for $n = 100$, $m_2 = 50, 100$

n	m_2	iterations		time		F^*		squares opt.	
		μ	σ	μ	σ	μ	σ	μ	σ
50	100	83.8	43.4	127.1	66.4	629.4	315.8	25.2	14.1
100	100	105	64.1	414.6	246.9	643.4	323.7	67.2	61.2

In conclusion, the proposed algorithm solves exactly and for the first time Weber's problem with regions forbidden to location and to travel. Computing times are moderate even for problems of large size. Sensitivity to the number of constraints is larger than to the number of users.

References

[1] ANEJA, Y.P., and PARLAR, M. (1994). Algorithms for Weber Facility Location in the Presence of Forbidden Regions and (or) Barriers to Travel. *Transportation Science* **28**, 70–76.

[2] ATKINSON, M.D., SACK, J.-R., SANTORO, N., and STROTHOTTE, T. (1986). Min-Max Heaps and Generalized Priority Queues. *Communications of the ACM* **29**, 996–1000.

[3] BUTT, E.S. (1994). Facility Location in the Presence of Forbidden Regions and Congested Regions. Ph.D. Thesis, Pennsylvania State University.

[4] BUTT, E.S. and CAVALIER, T.M. (1996). An Efficient Algorithm for Facility Location in the Presence of Forbidden Regions. *European Journal of Operational Research* **90**, 56–70.

[5] COURANT, R., and ROBBINS, H. (1958). *What is Mathematics*. Oxford University Press, London (Ninth printing).

[6] DIJKSTRA, E.W. (1959). A Note on Two Problems in Connection with Graphs. *Numerische Mathematiek* **1**, 269–271.

[7] HAMACHER, N.W., and NICKEL, S. (1994). Combinatorial Algorithms for some 1-Facility Median Problems in the Plane. *European Journal of Operational Research* **79**, 340–351.

[8] HANSEN, P., JAUMARD, B., and KRAU, S. (1995). An Algorithm for Weber's Problem on the Sphere. *Location Science* **3**(4), 217–237.

[9] HANSEN, P., PEETERS, D., and THISSE, J.F. (1981). Constrained Location and the Weber-Rawls Problem. *Annals of Operations Research* **11**, 147–166.

[10] HANSEN, P., PEETERS, D., and THISSE, J.F. (1981). The Generalized Weber-Rawls Problem, in J.-P. BRANS (editor) *Operational Research 81*, Amsterdam:North-Holland, 569–577.

[11] HANSEN, P., PEETERS, D., and THISSE, J.F. (1982). An Algorithm for the Constrained Weber Problem. *Management Science* **28**, 1285–1295.

[12] HANSEN, P., LABBÉ, M., PEETERS, D., and THISSE, J.F. (1987). Facility Location Analysis. *Fundamentals of Pure and Applied Economics* **22**, 1–67.

[13] HANSEN, P., PEETERS, D., RICHARD, D., and THISSE, J.F. (1985). The Minisum and Minimax Location Problems Revisited. *Operations Research* **35**, 1251–1265.

[14] HORST, R., and TUY, H. (1993). *Global Optimization: Deterministic Methods*. Berlin: Springer.

[15] HURTER, A.P., SCHAEFER, M.K., and WENDELL, R.E. (1975). Solutions of Constrained Location Problems. *Management Science* **22**, 51–56.

[16] KATZ, I.N. and COOPER, L. (1981). Facility Location in the Presence of Forbidden Regions. I. Formulation and the Case of Euclidean Distance with one Forbidden Circle. *European Journal of Operational Research* **6**, 166–173.

[17] LARSON, R.C., and SADIQ, G. (1983). Facility Location with the Manhattan Metric in the Presence of Barriers to Travel. *Operations Research* **31**, 652–669.

[18] LOVE, R.F., MORRIS, J.G., and WESOLOWSKY, G.O. (1988). *Facility location: Models and Methods*. New-York: North-Holland.

[19] PLASTRIA, F. (1992). The Generalized Big Square Small Square Method. *European Journal of Operational Research* **62**, 163–174.

[20] PLASTRIA, F. (1995). Continuous Location Problems, in Z. DREZNER (editor) *Facility Location. A Survey of Applications and Methods*. New-York:Springer, 225–262.

[21] SCHAEFER, M.K., and HURTER, A.P. (1974). An Algorithm for the Solution of a Location Problem with Metric Constraints. *Naval Research Logistic Quarterly* **21**, 625–636.

[22] VIEGAS, J., and HANSEN, P. (1985). Finding Shortest Paths in the Plane in the Presence of Barriers to Travel (for any lp-norm). *European Journal of Operational Research* **20**, 373–381.

[23] WANGDALH, G.E., POLLOCK, S.M., and WOODWARD, J.B. (1974). Minimizing Trajectory Pipe Routing. *Journal of Ship Research* **18**, 46–49.

[24] WATSON-GANDY, C.D.T. (1985). The Solution of Distance Constrained Mini-Sum Location Problem. *Operations Research* **33**, 784–802.

[25] WEISZFELD, E. (1937). Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tôhoku Mathematical Journal* **43**, 335–386.

[26] WESOLOWSKY, G.O. (1993). The Weber Problem: History and Perspectives. *Location Science* **1**, 5–23.