JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 89, No. 1, pp. 65-87, APRIL 1996

General Models in Min-Max Planar Location: Checking Optimality Conditions[1,2]

J. B. G. FRENK,[3] J. GROMICHO,[4] AND S. ZHANG[5]

Communicated by S. Schaible

Abstract. This paper studies the problem of deciding whether the present iteration point of some algorithm applied to a planar single-facility min-max location problem, with distances measured by either an l_p-norm or a polyhedral gauge, is optimal or not. It turns out that this problem is equivalent to the decision problem of whether 0 belongs to the convex hull of either a finite number of points in the plane or a finite number of different l_q-circles $\subseteq \mathbb{R}^2$. Although both membership problems are theoretically solvable in polynomial time, the last problem is more difficult to solve in practice than the first one. Moreover, the second problem is solvable only in the weak sense, i.e., up to a predetermined accuracy. Unfortunately, these polynomial-time algorithms are not practical. Although this is a negative result, it is possible to construct an efficient and extremely simple linear-time algorithm to solve the first problem. Moreover, this paper describes an implementable procedure to reduce the second decision problem to the first with any desired precision. Finally, in the last section, some computational results for these algorithms are reported.

Key Words. Optimality conditions, continuous location theory, computational geometry, convex hull, Newton-Raphson method.

---

[1]The work of the second author was supported by JNICT (Portugal), under Contract BD/631/90-RM, during his stay at Erasmus University in Rotterdam.

[2]The authors would like to express their acknowledgments to Jan Brinkhuis for his suggestions and support and to Joe Mitchell for suggesting the simplifications leading to Algorithm 3.1. Also, the anonymous referees are acknowledged for their careful reading of the paper and their useful comments.

[3]Associate Professor of Operations Research, Econometric Institute, Erasmus University, Rotterdam, Netherlands.

[4]Assistant Lecturer, DEIO, Faculdade de Ciências, Universidade de Lisboa, Lisboa, Portugal.

[5]Assistant Professor of Operations Research, Econometric Institute, Erasmus University, Rotterdam, Netherlands.

1. Introduction

In this paper, we discuss two related membership problems in $\mathbb{R}^2$. The first problem is to decide whether 0 belongs to the convex hull of a finite number of points. Moreover, the second problem is to find out whether 0 belongs to the convex hull of a finite number of different l_q-circles. Under some conditions, it is shown in Ref. 1 that these decision problems are equivalent to testing whether a given point is an optimal solution of a min-max single-facility continuous location problem with polyhedral gauges or l_p-norms. To assist the reader, a short introduction to min-max single-facility continuous location models is presented in the next section. Also in this section, it is proved that both membership problems belong to the class of polynomially solvable problems. Since this result is only of theoretical interest, a description of a practical linear-time algorithm to solve the first decision problem together with its proof of correctness is given in Section 3. In Section 4, it is shown that the second decision problem can be reduced to the first. Also in this section, the properties of the associated reduction algorithm are discussed. Finally in Section 5, computational experiments including both algorithms are reported. To conclude this introduction, we remark that this paper is a continuation of Ref. 1, where an algorithm to optimize the underlying min-max location problem is presented.

2. Single-Facility Min-Max Continuous Location

Single-facility continuous location models restricted to the plane include so-called min-max optimization problems (Refs. 2 and 3). As an example of such a problem, we mention the location of an emergency unit. In this case, it is important that each client located at one of the n known different demand points $d_1, d_2, \ldots, d_n$ can be reached as soon as possible, and so the appropriate objective function is clearly of the min-max type. In a general setting, it is assumed that $d_1, d_2, \ldots, d_n$ belong to $\mathbb{R}^s$, $s \geq 2$, and the distance between the location x of a facility and a demand point d_i, $1 \leq i \leq n$, is given by $\gamma_{\mathscr{G}_i}(x)$ with

$$\gamma_{\mathscr{G}_i}(x) := \inf\{t > 0 : x \in t\mathscr{G}_i\}.$$

Observe that the set $\mathscr{G}_i$, $i = 1, \ldots, n$, is a given compact convex set satisfying $0 \in \operatorname{int}(\mathscr{G}_i)$, with $\operatorname{int}(\mathscr{G}_i)$ denoting the interior of $\mathscr{G}_i$. The above distance function is called a gauge and for $\mathscr{G}_i$ symmetric it is called a norm. The motive to consider continuous location problems with arbitrary gauges is discussed in detail in Refs. 3 and 4. To introduce our location model, let the function

$\gamma: \mathbb{R}^s \to \mathbb{R}^n_+$ be given by

$$\gamma(x) := (\gamma_{\mathscr{G}_1}(x-d_1), \ldots, \gamma_{\mathscr{G}_n}(x-d_n)),$$

and assign to each demand point d_i, $1 \leq i \leq n$, a lower semicontinuous function $f_i: \mathbb{R}^n \to \mathbb{R}_+$, $1 \leq i \leq n$, which is nondecreasing on $\mathbb{R}^n_+$. The general single-facility unconstrained min-max location model is then given by

$$\text{(P)} \quad \inf_{x \in \mathbb{R}^s} \max_{1 \leq i \leq n} \varphi_i(x),$$

with $\varphi_i: \mathbb{R}^s \to \mathbb{R}$ defined by $\varphi_i(x) := f_i(\gamma(x))$. It is shown in Ref. 5 that an optimal solution exists, and so we may replace "inf" by "min" in (P). Under various additional assumptions on the functions f_i, different algorithms exist to solve (P) (Refs. 1 and 3). The general framework of each of these algorithms is as follows. The algorithm starts with some initial point x_0. If the algorithm decides that x_0 is not a local or global optimal point, then it constructs a new iteration point x_1 and repeats the optimality check, etc. Since in each step the optimality check has to be executed, it is important that an efficient algorithm is available to evaluate this decision. Under the additional assumptions that the functions f_i are quasiconvex and differentiable on an open convex set $\mathscr{S}$ with $\mathbb{R}^n_+ \subseteq \mathscr{S}$, that its gradient $\nabla f_i(z)$ contains at least one positive component for every $z \in \mathscr{S}$, and that the distance in (P) is given by polyhedral gauges, it is proved in Ref. 1 that a global optimality check is equivalent to the decision whether 0 belongs to $\operatorname{conv}(\{p_1, \ldots, p_k\})$, with $p_1, \ldots, p_k$ some finite number of points in $\mathbb{R}^s$ and $\operatorname{conv}(\{p_1, \ldots, p_k\})$ denoting the convex hull of these points. Moreover, for each demand point d_i, $1 \leq i \leq n$, if the distance in (P) is given by some l_{p_i}-norm $x \mapsto \|x\|_{p_i}$, $1 < p_i < \infty$, then a global optimality check reduces for some $k \leq n$ either to the decision of whether 0 belongs to $\operatorname{conv}(\bigcup_{1 \leq i \leq k} c_i + r_i \mathscr{B}_{q_i})$, with

$$\mathscr{B}_{q_i} := \{x \in \mathbb{R}^2 : \|x\|_{q_i} \leq 1\}$$

the so-called unit l_{q_i}-circle, $c_i \in \mathbb{R}^s$, $r_i > 0$, $1/p_i + 1/q_i = 1$, or to the decision of whether 0 belongs to $\operatorname{conv}(\{p_1, \ldots, p_k\})$. Frequently for the last two problems, k is much smaller than n. We will call the first decision problem (D) and the second decision problem (D′), and observe that these are special cases of the so-called strong membership problem (Ref. 6). Due to this, one can show for problem (D) the following theoretical result. Observe that the inner product of two vectors is denoted by $\langle \cdot, \cdot \rangle$.

Lemma 2.1. If the components of the points $p_1, \ldots, p_k$ are rational numbers, then problem (D) is solvable in polynomial time.

Proof. By Theorem 6.4.9 of Ref. 6, the strong separation problem is polynomially equivalent to the strong optimization problem, and strong separation implies immediately strong membership. Hence, we need only to verify the polynomial solvability of the strong optimization problem. Observe that the strong optimization problem connected with the convex set $\text{conv}(\{p_1, \ldots, p_k\})$ is given by $\max\{\langle c, x\rangle : x \in \text{conv}(\{p_1, \ldots, p_k\})\}$ for any vector c with rational components. For each rational vector c, this problem can be solved by evaluating $\langle c, p_i\rangle$, $i = 1, \ldots, k$, and selecting the vector p_j, with

$$\langle c, p_j\rangle = \max\{\langle c, p_i\rangle : i = 1, \ldots, k\}.$$

Clearly, this can be done in polynomial time, and so the result is proved. □

If one of the points p_i has an irrational component, it is impossible to solve (D) in polynomial time, due to the fact that no irrational number can be represented by a finite string of zeros and ones. To discuss in this case polynomial solvability, we need to introduce a rational number $\epsilon > 0$ and restrict ourselves to the associated weak membership problem of (D) (Ref. 6). The same holds for (D′), since the operations involved in calculating l_q-norms for general $1 < q < \infty$ produce in most cases irrational numbers.

Lemma 2.2. For any rational number $\epsilon > 0$, the associated weak membership problems of (D) and (D′) are solvable polynomially.

Proof. By Theorem 4.4.7 of Ref. 6, the weak separation problem is polynomially equivalent to the weak optimization problem, and so we can apply a similar argument as in Lemma 2.1 for the first decision problem. To prove the result for the second problem, we observe that the maximum of any linear function over the convex hull of a finite number of l_q-balls can be obtained by maximizing the same objective function over each of the balls. This problem can be solved analytically as will be shown in the remainder of the proof. Observe that

$$\max\{\langle d, x\rangle : x \in c + r\mathscr{B}_q\} \tag{1}$$

is equivalent to

$$\langle d, c\rangle + r \max\{\langle d, y\rangle : y \in \mathscr{B}_q\}. \tag{2}$$

By the Hölder inequality (Ref. 7), it follows that

$$\max\{\langle d, y\rangle : y \in \mathscr{B}_q\} = \|d\|_p,$$

with

$$1/p + 1/q = 1.$$

Moreover, the solution of (2) is given by the vector

$$y^* := (y_1^*, \ldots, y_j^*, \ldots, y_s^*),$$

with

$$y_j^* := \delta(d_j)|d_j|^{p/q}\|d\|_p^{1-p}, \qquad j=1, \ldots, s,$$

and $\delta(x)$ denoting the sign function of x. Hence, the optimal solution of (1) can be computed analytically; by taking a vector with rational components in some η-neighborhood of this solution with η small enough, we have solved the weak optimization problem. By the first part of this proof and since weak separation implies weak membership, the result is proved. □

Although the polynomial solvability is proved in weak and/or strong sense for both decision problems, it is not possible from a practical point of view to apply these reduction algorithms, since they are based on applying the ellipsoid algorithm a polynomial number of times. Moreover, for our location model, the most important instance is the planar case ($s=2$), and so we will discuss in the next sections some practical algorithms for solving (D) and (D′) in $\mathbb{R}^2$. Although (D′) is solved only approximately, the solution can be found up to any degree of approximation.

3. Solving Problem (D)

Clearly, if $p_i=0$ for any $1 \leq i \leq k$, then 0 belongs to $\mathrm{conv}(\{p_1, p_2, \ldots, p_k\})$, and so we assume that $0 \notin \{p_1, p_2, \ldots, p_k\}$.

A straightforward way to check whether 0 belongs to the convex hull of a set of points $\{p_1, p_2, \ldots, p_k\}$ starts by constructing the convex hull of the enlarged set of points $\{0, p_1, p_2, \ldots, p_k\}$. This can be done by using an efficient algorithm like the Graham scan; see Ref. 8. After executing this $\mathcal{O}(k \log k)$ algorithm, we know the extreme points of the polytope $\mathrm{conv}(\{0, p_1, p_2, \ldots, p_k\})$. It can be verified easily that 0 is an extreme point of $\mathrm{conv}(\{0, p_1, p_2, \ldots, p_k\})$ if and only if 0 does not belong to $\mathrm{conv}(\{p_1, p_2, \ldots, p_k\})$. Although it is not difficult to adapt the Graham scan in such a way that the algorithm will stop with a positive or a negative answer before the complete construction of the polytope $\mathrm{conv}(\{0, p_1, p_2, \ldots, p_k\})$, it still might take $\mathcal{O}(k \log k)$ operations in the worst case. As shown in Section 2, we have to solve this decision problem in each step of an iterative procedure, and so constructing an algorithm with lower complexity might reduce the overall computational effort. The best that we can hope for is a linear-time algorithm.

This is achieved by means of the following construction. First, reformulate the problem into a linear programming problem (Ref. 9) with two variables and k constraints; then, apply to it a known linear-time algorithm (e.g., as in Refs. 10, 11, 12, or 13). However, although being linear in the number of constraints when the number of variables is fixed, the existing procedures exhibit exponential dependence on the space dimension. This dependence is at least of $\mathcal{O}(3^{s^2})$ (Ref. 12), which even for the simpler case $s=2$ of our planar problem yields a solution procedure consisting of about $3^4 k$ operations. Although undoubtedly linear, this is not attractive.

Before presenting a much more efficient algorithm for solving directly our decision problem in the planar case, we need the following well-known definition and results from computational geometry (Ref. 8).

Definition 3.1. A point $p_0 \in \mathbb{R}^2$ is to the right of a directed segment from point p_1 to point p_2 if it is not an element of the line going through p_1 and p_2 and if moving along this line in the direction of p_1 to p_2 the point p_0 is an element of the right-hand side half-plane.

The definition of points to the left of a directed segment is similar. Moreover, given a line segment from p_1 to p_2 and another line segment from p_3 to p_4, a point p_0 is said to be between these two line segments when p_0 is to the right of one and to the left of the other.

In the remainder, a directed line segment from p_i to p_j will be denoted by $[p_i, p_j]$.

The following result can be used in order to determine to which side of a directed line segment a given point belongs.

Lemma 3.1. Let $p_i = (p_{i_1}, p_{i_2})^t$, $0 \leq i \leq 2$. And let

$$\Delta := \det\left(\begin{bmatrix} p_{0_1} & p_{0_2} & 1 \\ p_{1_1} & p_{1_2} & 1 \\ p_{2_1} & p_{2_2} & 1 \end{bmatrix}\right).$$

Then, the following results hold:

(i) p_0 is collinear with $[p_1, p_2]$, if and only if $\Delta = 0$;
(ii) p_0 is to the right of $[p_1, p_2]$, if and only if $\Delta < 0$;
(iii) p_0 is to the left of $[p_1, p_2]$, if and only if $\Delta > 0$.

Proof. A proof of this result can be found in Ref. 8; hence, it is omitted here. $\square$

In order to explain our linear-time algorithm, we need also the following result, which is a trivial consequence of Lemma 3.1.

Lemma 3.2. The point p_0 is between the two directed line segments $[p_1, p_2]$ and $[p_3, p_4]$ if and only if Δ_1 and Δ_2 have opposite signs, where

$$\Delta_1 := \det\left(\begin{bmatrix} p_{0_1} & p_{0_2} & 1 \\ p_{1_1} & p_{1_2} & 1 \\ p_{2_1} & p_{2_2} & 1 \end{bmatrix}\right),$$

$$\Delta_2 := \det\left(\begin{bmatrix} p_{0_1} & p_{0_2} & 1 \\ p_{3_1} & p_{3_2} & 1 \\ p_{4_1} & p_{4_2} & 1 \end{bmatrix}\right).$$

Observe that, in the sequel of this paper, p_0 will always be 0. Therefore, only a 2×2 determinant need be computed.

Let us denote by $\mathscr{C}(l, r)$ the closed convex cone with vertex at 0 and generated by the line segments $[0, l]$ and $[0, r]$; and let l and r be chosen in such a way that l is to the left of $[0, r]$. The test $x \in \mathscr{C}(l, r)$ can be implemented using Lemma 3.2.

We now present our algorithm to decide if 0 belongs to $\operatorname{conv}(\{p_1, \ldots, p_k\})$.

Algorithm 3.1.

Step 1. If $k=1$ then stop with *Yes* in case $p_1=0$ or with *No* in case $p_1 \neq 0$. Otherwise, proceed to Step 2.

Step 2. Search $\{p_2, \ldots, p_k\}$ for the first point not collinear with $[0, p_1]$. If during the search $0 \in [p_1, p_i]$ for some $i=2, \ldots, k$, then stop with *Yes*. Otherwise, if all the points are collinear with $[0, p_1]$, then stop with *No*. If not all the points are collinear, a first point, say p_i, is found being either to the right or to the left of $[0, p_1]$. In case it is to the right, let $r := p_i$ and $l := p_1$; otherwise, let $r := p_1$ and $l := p_i$. Proceed to Step 3.

Step 3. For each not yet examined point p_l, $i < l \leq k$, check the following four mutually exclusive conditions. If $p_l \in \mathscr{C}(-r, l)$, then update $l := p_l$. If $p_l \in \mathscr{C}(r, -l)$, then update $r := p_l$. Finally, if $p_l \in \mathscr{C}(l, r)$, then leave l and r untouched; and if $p_l \in \mathscr{C}(-l, -r)$, then stop with *Yes*. If the algorithm did not stop, repeat Step 3 considering the next point, or stop with *No* if all the points have been examined.

The present algorithm is a simplification of the algorithm proposed in Ref. 14. First, the trivial case of just one point is eliminated in Step 1. If there are two or more points, then an initialization occurs in Step 2. For the purpose of initialization, p_1 is selected and the algorithm searches for the first point in $\{p_2, \ldots, p_k\}$ noncollinear with $[0, p_1]$. If during the search 0 is found to belong to $[p_1, p_i]$ for any i, the algorithm stops without reaching the main loop and answers *Yes*. However, if all the points are exhausted without finding either a noncollinear point or a point such that $0 \in [p_1, p_i]$, the algorithm also stops before the main loop but answers *No*. If a noncollinear point p_i is found, then l (left) and r (right) are assigned to p_1 and p_i or viceversa, and the main loop begins in Step 3. The main loop is based on the trivial observation that 0 is outside $\text{conv}(\{p_1, \ldots, p_k\})$ if and only if some l and r exist in $\{p_1, \ldots, p_k\}$ defining a closed conex cone $\mathscr{C}(l, r)$ such that $\text{conv}(\{p_1, \ldots, p_k\}) \subseteq \mathscr{C}(l, r)$. The main loop (Step 3) updates $\mathscr{C}(l, r)$, while it remains convex or concludes that $0 \in \text{conv}(\{p_1, \ldots, p_k\})$. In each iteration, the plane is partitioned by the interior-disjoint cones

$$\mathscr{C}(l, r), \mathscr{C}(-r, l), \mathscr{C}(r, -l), \mathscr{C}(-l, -r),$$

and one of the following actions is carried out. In case $p_i \in \mathscr{C}(l, r)$, nothing is done. However, if $p_i \in \mathscr{C}(-r, l)$, then l is replaced by p_i; if $p_i \in \mathscr{C}(r, -l)$, then r is replaced by p_i. Finally if $p_i \in \mathscr{C}(-r, -l)$, then the algorithm stops with *Yes* since 0 belongs to the triangle $\text{conv}(\{l, r, p_i\})$ and this is a subset of $\text{conv}(\{p_1, \ldots, p_k\})$. Algorithm 3.1 is thus a very simple and correct algorithm. Moreover, as the following result states, both the total number of operations and storage requirements are obviously linear functions in the number of points with very small coefficients. Actually, the total number of logical conditions evaluated is below $4k$, while the total number of arithmetic operations is below $7k$, as the proof of the next lemma shows.

Lemma 3.3. Algorithm 3.1 gives the correct answer in all cases. Moreover, both its computational complexity and storage requirements are of $\mathcal{O}(k)$.

Proof. The correctness of the algorithm follows from the previous discussion. The storage requirements amount to storing $p_1, \ldots, p_k$ plus a pair of index pointers to identify l and r. For each point p_i, $1 \leq i \leq k$, one (Step 2 by means of Lemma 3.1) or two (Step 3 by means of Lemma 3.2) 2×2 determinants are computed and at most four logical conditions are evaluated. Each 2×2 determinant requires two multiplications and one subtraction. This yields the complexity as stated in the lemma. □

In the next section, we show how (D′) can be converted into (D) and present a numerical procedure to carry out this reduction.

4. Solving Problem (D′)

In the second decision problem, we want to verify whether 0 belongs to

$$\operatorname{conv}\left(\bigcup_{1\le i\le k} c_i + r_i\mathscr{B}_{q_i}\right).$$

Clearly, by computing $\|c_i\|_{q_i}$, we first check whether $0\in c_i+r_i\mathscr{B}_{q_i}$, i.e., $\|c_i\|_{q_i}\le r_i$, for some $1\le i\le k$. If this holds, then 0 clearly belongs to $\operatorname{conv}(\bigcup_{1\le i\le k} c_i+r_i\mathscr{B}_{q_i})$. However, if this is not true, we have to solve the following nontrivial decision problem:

$$0\notin c_i+r_i\mathscr{B}_{q_i}, \qquad 1\le i\le k,$$

$$\overset{?}{\Rightarrow} 0\in\operatorname{conv}\left(\bigcup_{1\le i\le k} c_i+r_i\mathscr{B}_{q_i}\right).$$

For $0\notin c_i+r_i\mathscr{B}_{q_i}$, there must exist one pair of supporting hyperplanes $\mathscr{H}_i^L$ and $\mathscr{H}_i^R$ of $c_i+r_i\mathscr{B}_{q_i}$ going through 0. Moreover, let t_i^L and t_i^R denote the pair of unique intersection or tangent points of the corresponding hyperplanes and $c_i+r_i\mathscr{B}_{q_i}$. We will refer to $\mathscr{H}_i^L$ as the left tangent hyperplane and to $\mathscr{H}_i^R$ as the right tangent hyperplane. The left hyperplane is defined as the one whose tangent point is to the left of $[0, c_i]$, while the right hyperplane is given by the other one. Keeping in mind the definition of t_i^L and t_i^R, one can now prove the following result for problem (D′).

Lemma 4.1. If 0 does not belong to $c_i+r_i\mathscr{B}_{q_i}$ for every $1\le i\le k$, then

$$0\in\operatorname{conv}\left(\bigcup_{1\le i\le k} c_i+r_i\mathscr{B}_{q_i}\right),$$

if and only if

$$0\in\operatorname{conv}(\{t_1^L, t_1^R, \dots, t_k^L, t_k^R\}).$$

Proof. Since $t_i^L, t_i^R\in c_i+r_i\mathscr{B}_{q_i}$, for every $1\le i\le k$, it follows that

$$\operatorname{conv}(\{t_1^L, t_1^R, \dots, t_k^L, t_k^R\})\subseteq\operatorname{conv}\left(\bigcup_{1\le i\le k} c_i+r_i\mathscr{B}_{q_i}\right);$$

hence, the second condition implies the first. For the proof of the reverse implication, let

$$0\in\operatorname{conv}\left(\bigcup_{1\le i\le k} c_i+r_i\mathscr{B}_{q_i}\right);$$

i.e., there exist some $\alpha_i \geq 0$, $1 \leq i \leq k$, such that $\sum_{i=1}^k \alpha_i = 1$ and $0 = \sum_{i=1}^k \alpha_i p_i$, with each p_i belonging to $c_i + r_i \mathscr{B}_{q_i}$, $1 \leq i \leq k$. Consider now, for each p_i belonging to $c_i + r_i \mathscr{B}_{q_i}$, the straight line $\mathscr{K}_i$ going through 0 and p_i. Since p_i belongs to the cone generated by the two hyperplanes $\mathscr{H}_i^L$ and $\mathscr{H}_i^R$, it follows that the straight line $\mathscr{K}_i$ intersects the line segment connecting t_i^L and t_i^R in the point q_i. Hence, we can find constants $c_i > 0$ and $0 \leq \tau_i \leq 1$ such that

$$p_i = c_i q_i \quad \text{and} \quad q_i = \tau_i t_i^L + (1 - \tau_i) t_i^R.$$

This implies that

$$p_i = c_i(\tau_i t_i^L + (1 - \tau_i) t_i^R);$$

so, after normalizing, it follows that 0 can be written as a convex combination of the tangent points $t_1^L, t_1^R, \ldots, t_k^L, t_k^R$. □

The result stated by Lemma 4.1 gives us the possibility to reduce every instance of the decision problem (D′) into an instance of the decision problem (D). First, the normal vectors of the tangent hyperplanes need to be computed. After that, we determine the tangent point of each hyperplane and apply Algorithm 3.1 to the set of these points. Let us denote these points by $\{t_1^L, t_1^R, \ldots, t_k^L, t_k^R\}$ as in Lemma 4.1.

In the remainder, we will omit the indices for the sake of notational convenience; so, a tangent hyperplane will be given by $\mathscr{H}$ regardless of being left or right. We will now discuss the computation of each $a \in \mathbb{R}^2$ with a a normal vector of $\mathscr{H}$, i.e.,

$$\mathscr{H} = \{z \in \mathbb{R}^2 : \langle a, z \rangle = 0\}.$$

Let us assume without loss of generality that $\|a\|_p = 1$, with p such that $1/p + 1/q = 1$; also, let us assume that $\langle a, x \rangle \leq 0$ holds for every $x \in c + r\mathscr{B}_q$. This means that a points from 0 to the halfspace not containing $c + r\mathscr{B}_q$.

Since there is a tangent point $t \in \mathscr{H} \cap (c + r\mathscr{B}_q)$, it must follow that $\langle a, t \rangle = 0$; so, by the previous condition, we obtain

$$\max\{\langle a, x \rangle : x \in c + r\mathscr{B}_q\} = 0. \tag{3}$$

Clearly, every element $x \in c + r\mathscr{B}_q$ can be written as $x = c + ry$, where $y \in \mathscr{B}_q$; hence, (3) is equivalent to

$$\langle a, c \rangle + r \max\{\langle a, y \rangle : y \in \mathscr{B}_q\} = 0. \tag{4}$$

Applying the Hölder inequality (Ref. 7) to the second term in (4), it follows that

$$\max\{\langle a, y \rangle : y \in \mathscr{B}_q\} = \|a\|_p. \tag{5}$$

Another way to see (5) is to observe (Ref. 15) that $\|\cdot\|_p$ is the norm of the dual space $\mathscr{A}^*$ of the normed space $\mathscr{A}=(\mathbb{R}^2, \|\cdot\|_q)$. Hence, by (4) and (5), we obtain that the normal vector a of $\mathscr{H}$ must satisfy

$$\|a\|_p=-(1/r)\langle a, c\rangle>0;$$

so, using the normalization $\|a\|_p=1$ and defining $c^* := -(1/r)c$, we have

$$\langle a, c^*\rangle=1.$$

By the above observations, we have verified that the vector a satisfies

$$\langle a, c^*\rangle=1, \qquad \|a\|_p=1. \tag{6}$$

A geometrical argument based on the existence of exactly two supporting hyperplanes guarantees that there are exactly two real-valued vectors solving this system, corresponding to the normal vectors of these hyperplanes. Since it is not yet possible to decide which one is left or right, we index them as one and two.

Unfortunately, in general it is not possible to write down analytical solutions of these nonlinear equations, and so we have to use a numerical procedure to find them. However, for some special cases, this can be done. These cases are listed now. We denote by e_i the ith unit vector; that is,

$$e_1=(1,0)^t \quad \text{and} \quad e_2=(0,1)^t.$$

Case 1. Particular values of p or q.

Case (1a): $p=1$ or $q=\infty$. We have

$$a_j^t=\begin{cases}((1-c_2^*)/(c_1^*-c_2^*),\ (c_1^*-1)/(c_1^*-c_2^*)), & \text{if it belongs to } \mathscr{C}(e_1, e_2),\\ ((1-c_2^*)/(c_1^*+c_2^*),\ (c_1^*+1)/(c_1^*+c_2^*)), & \text{if it belongs to } \mathscr{C}(e_2, -e_1),\\ ((1+c_2^*)/(c_1^*-c_2^*),\ (-1-c_1^*)/(c_1^*-c_2^*)), & \text{if it belongs to } \mathscr{C}(-e_1, -e_2),\\ ((1+c_2^*)/(c_1^*+c_2^*),\ (1-c_1^*)/(c_1^*+c_2^*)), & \text{if it belongs to } \mathscr{C}(-e_2, e_1).\end{cases}$$

Case (1b): $p=2$ or $q=2$. We have

$$a_j^t=((c_1^*\pm c_2^*\sqrt{\Delta^*})/(\Delta^*+1),\ (c_2^*\mp c_1^*\sqrt{\Delta^*})/(\Delta^*+1)),$$

where $\Delta^*=c_1^{*2}+c_2^{*2}-1$.

Case (1c): $p=\infty$ or $q=1$. We have

$$a_j^t=\begin{cases}((1-c_2^*)/c_1^*,1), & \text{if } -1<(1-c_2^*)/c_1^*<1,\\ (-1,(1+c_1^*)/c_2^*), & \text{if } -1<(1+c_1^*)/c_2^*<1,\\ ((1+c_2^*)c_1^*,-1), & \text{if } -1<(1+c_2^*)/c_1^*<1,\\ (1,(1-c_1^*)/c_2^*), & \text{if } -1<(1-c_1^*)/c_2^*<1.\end{cases}$$

Case 2. Particular values of c^*.

Case (2a): $c_1^*=0 \Rightarrow a_j^t=(\pm(|c_2^*|^p-1)^{1/p}/|c_2^*|,\ 1/c_2^*)$.

Case (2b): $c_2^*=0 \Rightarrow a_j^t=(1/c_1^*,\ \pm(|c_1^*|^p-1)^{1/p}/|c_1^*|)$.

Case (2c): Incomplete solutions for particular values of c^*,

(i) $c_1^*=1 \Rightarrow a_1^t=(1,0)$,
(ii) $c_1^*=-1 \Rightarrow a_1^t=(-1,0)$,
(iii) $c_2^*=1 \Rightarrow a_1^t=(0,1)$,
(iv) $c_2^*=-1 \Rightarrow a_1^t=(0,-1)$.

For Case (2c) only one solution was found analytically. Cases (1a) and (1c) are listed only for completeness, since $1<p<\infty$ is assumed. In fact, the cases $p=1$ and $p=\infty$ correspond to polyhedral gauges. Moreover, the solution given in Case (1c) will be needed later.

In order to find a solution for the other cases, we need to use a nonlinear procedure. Without loss of generality, assume now that $c_1^*, c_2^*\notin\{-1,0,1\}$. Observe that the solution of (6) is given by the two points where the line

$$\mathscr{K}:=\{x\in\mathbb{R}^2: \langle x,c^*\rangle=1\}$$

intersects the unit l_p-circle $\mathscr{B}_p$.

In Fig. 1, the three cases, modulo possible rotations, are shown. For the sake of abstraction, Euclidean l_2-circles are used in Fig. 1 to represent general l_p-circles. Clearly, every point $x\in\mathscr{K}$ can be represented by $x=b+\mu u$, where b is some point on $\mathscr{K}$ and u is a vector parallel to $\mathscr{K}$.

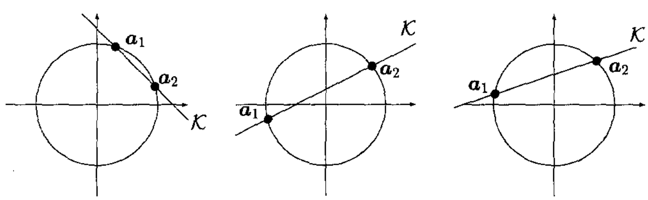

Fig. 1. Intersections of $\mathscr{K}$ and the unit l_p-circle.

The problem now reduces to finding the roots of the equation $\psi(\mu) = 1$, with $\psi: \mathbb{R} \to \mathbb{R}$ given by

$$\psi(\mu) := \|b + \mu u\|_p.$$

We first give some properties of the function ψ. These properties are independent of the choice of b and u.

Lemma 4.2. The real function $\psi: \mu \mapsto \|b + \mu u\|_p$ is a differentiable convex function. Moreover, it is Lipschitz continuous with Lipschitz constant $\|u\|_p$.

Proof. The convexity of the function ψ is an easy consequence of the convexity of the mapping $v \mapsto \|v\|_p$. Also, since this mapping is Fréchet differentiable in $\mathbb{R}^2 \setminus \{0\}$ and 0 does not belong to the line $\mathscr{K}$, the differentiability of ψ follows. Hence, it remains only to verify the Lipschitz continuity of ψ. Clearly, for every μ_1 and μ_2, we obtain

$$\begin{aligned} |\psi(\mu_1) - \psi(\mu_2)| &= | \|b + \mu_1 u\|_p - \|b + \mu_2 u\|_p | \\ &\leq \|(b + \mu_1 u) - (b + \mu_2 u)\|_p \\ &= \|(\mu_1 - \mu_2) u\|_p = |\mu_1 - \mu_2| \, \|u\|_p, \end{aligned}$$

and this concludes the proof. □

For the computation of the two roots μ_1^* and μ_2^*, corresponding to the normal vectors

$$a_1 := b + \mu_1^* u \quad \text{and} \quad a_2 := b + \mu_2^* u,$$

we first derive an interval containing both roots.

Recall the well-known inequality (Ref. 7)

$$1 \leq p_1 < p_2 \leq \infty$$

$$\Rightarrow \|x\| p_1 \geq \|x\| p_2.$$

Moreover, if x does not belong to any of the coordinate axis, it is easy to show that the inequality sign between the l_p-norms can be replaced by a strict inequality sign.

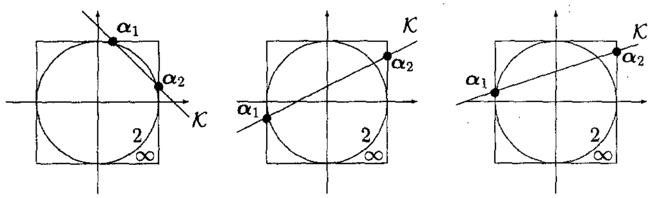

Fig. 2. Starting points.

From this inequality, it follows that the unit l_p-circle is contained in the unit l_∞-circle. Hence (see also Fig. 2), the two intersection points α_1 and α_2 of the line $\mathcal{K}$ and the unit l_∞-circle are outside the unit l_p-circle, and so it must follow that both solutions a_1 and a_2 of (6) belong to the open line segment (α_1, α_2).

Since by Case (1c), α_1 and α_2 can be computed analytically, we take $b := \alpha_1$ and $u := \alpha_2 - \alpha_1$ (cf. Fig. 3). This implies that the pair of different roots μ_1^* and μ_2^* belongs to (0, 1). Also, since

$$\psi(\mu_1^*) = \psi(\mu_2^*) = 1 \quad \text{and} \quad \psi(0) > 1,\ \psi(1) > 1,$$

it follows by the convexity of ψ that a minimum point belongs to (μ_1^*, μ_2^*); so, again by the convexity of ψ, we obtain (see Fig. 4) that

$$\psi'(\mu) < 0, \text{if } \mu \le \mu_1^*, \qquad \psi'(\mu) > 0, \text{if } \mu \ge \mu_2^*.$$

These observations allow us to state the following well-known result for the Newton–Raphson method (Ref. 16). We list an alternative short proof for completeness.

Lemma 4.3. If the Newton–Raphson method is applied with starting point 0 to solve the equation $\psi(\mu) - 1 = 0$, it produces an increasing sequence of iterates converging from below to μ_1^*.

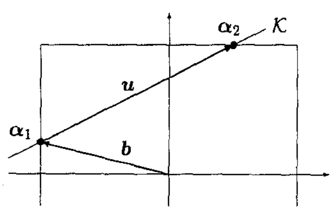

Fig. 3. Construction of ψ.

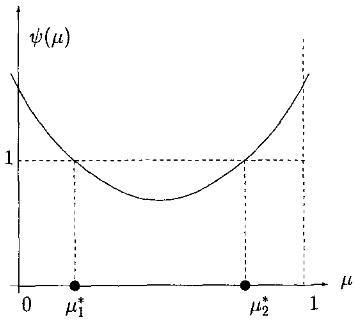

Fig. 4. Sample graph of ψ.

Proof. Let us denote by μ^l, $l=1, 2, \ldots$, the sequence of iterates produced by the Newton–Raphson method, i.e.,

$$\mu^{l+1} := \mu^l + [1 - \psi(\mu^l)]/\psi'(\mu^l). \tag{7}$$

By the subgradient inequality (Ref. 17), it follows that, for every μ^l,

$$1 = \psi(\mu_1^*) \geq \psi(\mu^l) + \psi'(\mu^l)(\mu_1^* - \mu^l). \tag{8}$$

Take now $\mu^0 := 0$. Since $\psi'(\mu^0) < 0$ and $\psi(\mu^0) > 1$, we obtain

$$[1 - \psi(\mu^0)]/\psi'(\mu^0) > 0. \tag{9}$$

Using (7) and (8) and taking $l=0$ leads to

$$\mu^1 := \mu^0 + [1 - \psi(\mu^0)]/\psi'(\mu^0) \leq \mu_1^*. \tag{10}$$

Now from (7), (9), and (10), we conclude that $\mu^0 < \mu^1 \leq \mu_1^*$; by induction, it is easily proved that μ^l is increasing and bounded from above by μ_1^*. Due to $\mu^l \leq \mu_1^*$ for every $l \geq 0$, it follows that the sequence converges and satisfies

$$\mu^\infty := \lim_{l \uparrow \infty} \mu^l \leq \mu_1^*.$$

After observing that the derivative ψ' is continuous and negative on $[0, \mu_1^*]$, we obtain from (7) that this limit satisfies

$$\mu^\infty = \mu^\infty + [1 - \psi(\mu^\infty)]/\psi'(\mu^\infty),$$

and hence $\psi(\mu^\infty) = 1$. By the strict monotonicity of ψ on $[0, \mu_1^*]$, this finally yields $\mu^\infty = \mu_1^*$. □

In order to guarantee finite termination, it is now important to derive a stopping rule for this procedure meeting any prespecified error bound.

First, we show that a minimum point of ψ, or equivalently a point μ^* such that $\psi'(\mu^*)=0$, is analytically computable. We recall that ψ is differentiable everywhere. Moreover, its derivative ψ' has the following expression:

$$\psi'(\mu)=\psi(\mu)^{1-p}\sum_{i=1}^{2}|b_i+\mu u_i|^{p-2}(b_i+\mu u_i)u_i.$$

Hence, μ^* satisfies the equation

$$\psi(\mu^*)^{1-p}\sum_{i=1}^{2}|b_i+\mu^* u_i|^{p-2}(b_i+\mu^* u_i)u_i=0.$$

Since $0\notin\mathscr{K}$, it follows that $\psi(\mu)>0$, for every μ, and so μ^* must satisfy

$$|b_1+\mu^* u_1|^{p-2}(b_1+\mu^* u_1)u_1=-|b_2+\mu^* u_2|^{p-2}(b_2+\mu^* u_2)u_2. \tag{11}$$

Obviously $u\neq 0$, and so only the following two cases hold.

Case 1. Either $u_1=0$ or $u_2=0$ but not both. Without loss of generality, we can assume that $u_1=0$. In this case, (11) reduces to $b_2+\mu^* u_2=0$ or $\mu^*=-b_2/u_2$ and $\psi(\mu^*)=|b_1|$.

Case 2. Both $u_1\neq 0$ and $u_2\neq 0$. After taking absolute values in (11), we obtain

$$|b_1+\mu^* u_1|^{p-1}|u_1|=|b_2+\mu^* u_2|^{p-1}|u_2|.$$

Raising now both members to the power $1/(p-1)$, it follows that

$$(b_1+\mu^* u_1)=\pm(b_2+\mu^* u_2)|u_2/u_1|^{1/(p-1)},$$

and so we get

$$\mu^*=-[b_1\pm|u_2/u_1|^{1/(p-1)}b_2]/[u_1\pm|u_2/u_1|^{1/(p-1)}u_2].$$

Substituting the above expression for μ^* in (11), one can check that the undetermined sign $\pm$ should be a $+$ if u_2 and u_1 have the same sign and $-$ otherwise. This leads to the final expression, where $\delta(x)$ stands for the sign function of x,

$$\mu^*=-[b_1+\delta(u_2/u_1)|u_2/u_1|^{1/(p-1)}b_2]/[u_1+\delta(u_2/u_1)|u_2/u_1|^{1/(p-1)}u_2].$$

For this μ^*, it follows that $\psi(\mu^*)=|u_1b_2-u_2b_1|/\|u\|_q$.

The following lemma provides a stopping rule for the Newton–Raphson procedure.

Lemma 4.4. If the Newton–Raphson method is applied with starting point 0 to solve the equation $\psi(\mu)-1=0$, it follows that, for every $l\geq 1$,

$$\mu^l\leq\mu_1^*\leq\sigma^l,$$

where

$$\sigma^l:=\mu^*+(\mu^l-\mu^*)[1-\psi(\mu^*)]/[\psi(\mu^l)-\psi(\mu^*)]. \tag{12}$$

Moreover, the sequence σ^l is decreasing and converges from above to μ_1^*.

Proof. Observe first that

$$\sigma^l=\lambda\mu^l+(1-\lambda)\mu^*, \tag{13}$$

with

$$0<\lambda:=[1-\psi(\mu^*)]/[\psi(\mu^l)-\psi(\mu^*)]<1.$$

Therefore, since ψ is convex by Lemma 4.2, we obtain that

$$\psi(\sigma^l)\leq\lambda\psi(\mu^l)+(1-\lambda)\psi(\mu^*).$$

It is now easy to check that

$$\lambda\psi(\mu^l)+(1-\lambda)\psi(\mu^*)=1,$$

and so,

$$\psi(\sigma^l)\leq 1=\psi(\mu_1^*).$$

Moreover, since $\mu^l<\mu^*$, it follows that $\sigma^l<\mu^*$; since ψ is decreasing in the interval $(-\infty,\mu^*]$ and $\psi(\sigma^l)\leq\psi(\mu_1^*)$, this yields $\sigma^l\geq\mu_1^*$. Observe now that

$$\begin{aligned}\sigma^{l+1}\geq\sigma^l &\Leftrightarrow \{[1-\psi(\mu^*)]/[\psi(\mu^{l+1})-\psi(\mu^*)]\}(\mu^{l+1}-\mu^*)\\ &\qquad\geq\{[1-\psi(\mu^*)]/[\psi(\mu^l)-\psi(\mu^*)]\}(\mu^l-\mu^*)\\ &\Leftrightarrow [\mu^{l+1}-\mu^*]/[\psi(\mu^{l+1})-\psi(\mu^*)]\geq[\mu^l-\mu^*]/[\psi(\mu^l)-\psi(\mu^*)],\end{aligned}$$

and this is immediately clear from the convexity of ψ. The remainder of the proof follows easily from the continuity of ψ by computing the limit in (12) after observing that $\psi(\mu_1^*)-\psi(\mu^*)>0$. □

Clearly, Lemma 4.4 yields the following stopping rule:

$$\sigma^l-\mu^l\leq\epsilon\Rightarrow 0\leq\mu_1^*-\mu^l\leq\epsilon.$$

Obviously, to find μ_2^*, it is enough to take $b:=\alpha_2$, $u:=\alpha_1-\alpha_2$, and then apply exactly the same steps as for computing μ_1^*.

As a final remark, we note that, for $1<p<2$, the unit l_2-circle plays the same role as the unit l_∞-circle; i.e., there are two intersection points of the line $\mathscr{K}$ and the unit l_2-circle (let us denote them by β_j, $j=1,2$), and they satisfy

$$[a_1, a_2] \subset [\beta_1, \beta_2] \subset [\alpha_1, \alpha_2].$$

Hence for $1<p<2$, the points β_j, also analytically computable by Case (1b), can be used instead of the points α_j, and they provide a better first approximation with the same properties.

So, we can find both values of μ_j^*, $j=1,2$, and consequently both values of the normal vectors a_j, by solving a pair of independent nonlinear equations of the form

$$\|b+\mu u\|_p = 1,$$

with given b and u, using the Newton-Raphson method. Let us assume now that the two values of a_j are known. We still need to find the pair of tangent points t^L and t^R in order to apply Lemma 4.1.

Recall that each a is orthogonal to the corresponding $\mathscr{H}$ and is pointing to the half-plane not containing $c+r\mathscr{B}_p$. From a similar argument as used in the proof of Lemma 2.1, by using the Hölder inequality (Ref. 7) we can compute that

$$t_j := \operatorname{argmax}\{\langle a_j, t\rangle : t \in c + r\mathscr{B}_q\},$$

for $j=1,2$. Now, it is trivial to label them as left and right.

Concerning each approximation μ^l of μ_1^*, we can derive an approximation τ of t. The question that arises naturally concerns now the safety of such an approximation regarding the optimality of the underlying location problem. Observe that the decision upon optimality is taken by Algorithm 3.1 when applied to check whether 0 belongs to the convex hull of the set of approximated points $\{\tau_i^L, \tau_i^R : 1 \le i \le l\}$.

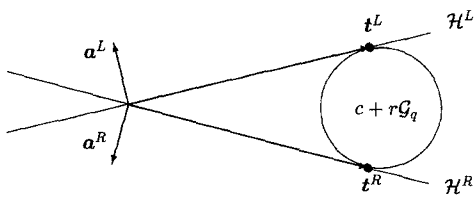

Fig. 5. Assigning left and right.

Consider $a^l := b + \mu^l u$. Since a^l converge to a from outside the unit l_p-circle, it follows that $\|a^l\|_p \geq 1$ and $a^l \to a$, i.e., $\|a^l\|_p \downarrow 1$. So, with respect to system (6), a^l satisfies the following system:

$$\langle a^l, c^* \rangle = 1 \quad \text{and} \quad \|a^l\|_p > 1,$$

and so,

$$\max\{\langle a^l, y\rangle : y \in \mathscr{B}_q\} = \|a^l\|_p > 1.$$

This implies that

$$\langle a^l, c\rangle + r \max\{\langle a^l, y\rangle : y \in \mathscr{B}_q\} = -r\langle a^l, c^*\rangle + r\|a^l\|_p > 0,$$

and hence there exists one $x \in c + r\mathscr{B}_q$ such that $\langle a^l, x\rangle > 0$. This yields that the hyperplane

$$\mathscr{H}^l := \{z \in \mathbb{R}^2 : \langle a^l, z\rangle = 0\}$$

is secant to $c + r\mathscr{B}_q$ and that τ is a lower estimate of t; i.e., it belongs to the cone generated by $\mathscr{H}^L$ and $\mathscr{H}^R$ (see the proof of Lemma 4.1).

This guarantees that, if the output of Algorithm 3.1 regarding the set of approximations $\{\tau_i^L, \tau_i^R : 1 \leq i \leq l\}$ is *Yes*, then the true answer is also *Yes*; therefore, regarding the underlying location problem mentioned in Section 2, no false optimality is detected, making this decision a safe one. Of course, a point may be optimal and, due to the approximation used may be identified as nonoptimal. This is in general safer than the opposite situation.

However, a safe *No* can be produced by the same reasoning, if the roles of μ^l and σ^l are reversed. This time *Yes* would be unsafe, but in early stages of the underlying optimization procedure, it may be interesting to first expect a sequence of negative answers, and only after the first *Yes* is reported, switch to the safe *Yes* form.

5. Computational Results

In order to test the algorithms, they were coded completely in Turbo Pascal version 7.0 and executed on an AST Bravo 4/33, a PC/AT compatible with an Intel 80486 CPU with built-in numerical processor and clock speed of 33 MHz. The numerical precision used is the Turbo Pascal specific extended precision, a non-IEEE 80-bit numerical format superior in precision to the IEEE 64-bit double precision format. The computational experience was carried out over 360 uncorrelated instances of problem (D′). Those instances were generated randomly in the following way.

The number m of l_p-circles belongs to $\{10, 25, 50, 100, 250, 500\}$. For the l_p-norms being used, we take $p \in \{1.1, 1.5, 1.9, 2.1, 3.0\}$. Finally, the

Table 1. Results of the decision algorithm in the "easy" case.

Problem		Newton–Raphson			Algorithm 3.1		CPU Time
n	p	It	$\max(It)$	$\%T$	$\%T$	$\%Yes$	(sec)
10	1.1	2.4	5	97.5	2.5	20.0	0.017
10	1.5	3.9	6	98.7	1.3	20.0	0.025
10	1.9	4.1	5	98.7	1.3	30.0	0.026
10	2.1	3.8	4	98.8	1.2	50.0	0.024
10	3.0	4.1	5	98.7	1.3	20.0	0.026
10	mixed	3.7	6	98.6	1.4	40.0	0.024
25	1.1	2.5	5	98.5	1.5	50.0	0.044
25	1.5	3.9	6	99.0	1.0	60.0	0.063
25	1.9	4.2	6	99.0	1.0	50.0	0.064
25	2.1	3.7	4	98.9	1.1	50.0	0.059
25	3.0	4.2	5	99.2	0.8	100.0	0.067
25	mixed	3.6	6	98.7	1.3	40.0	0.059
50	1.1	2.5	6	98.7	1.3	70.0	0.089
50	1.5	3.9	6	99.1	0.9	80.0	0.124
50	1.9	4.1	6	99.3	0.7	100.0	0.127
50	2.1	3.8	4	99.2	0.8	90.0	0.118
50	3.0	4.1	5	99.3	0.7	100.0	0.132
50	mixed	3.7	6	99.2	0.8	100.0	0.118
100	1.1	2.6	6	99.5	0.5	100.0	0.175
100	1.5	3.9	6	99.4	0.6	100.0	0.248
100	1.9	4.1	6	99.2	0.8	100.0	0.256
100	2.1	3.7	4	99.2	0.8	100.0	0.235
100	3.0	4.1	5	99.6	0.4	100.0	0.264
100	mixed	3.7	6	99.5	0.5	100.0	0.236
250	1.1	2.6	6	99.8	0.2	100.0	0.438
250	1.5	3.9	6	99.9	0.1	100.0	0.616
250	1.9	4.1	6	99.8	0.2	100.0	0.637
250	2.1	3.7	4	99.7	0.3	100.0	0.586
250	3.0	4.1	5	99.8	0.2	100.0	0.653
250	mixed	3.7	6	99.8	0.2	100.0	0.587
500	1.1	2.6	6	99.9	0.1	100.0	0.878
500	1.5	3.8	6	99.9	0.1	100.0	1.231
500	1.9	4.1	6	99.9	0.1	100.0	1.270
500	2.1	3.7	4	99.9	0.1	100.0	1.171
500	3.0	4.1	5	99.9	0.1	100.0	1.306
500	mixed	3.7	6	99.9	0.1	100.0	1.173

tolerance parameter used in the stopping rule is given by $\epsilon := 5 \times 10^{-16}$. This unrealistic precision, only possibly by means of the extended precision, was deliberately adopted to test the Newton–Raphson method to the limit.

Now, we describe the procedure to generate the circles. All the centers are generated uniformly within the square $[-10, 90] \times [-10, 90]$.

Table 2. Results of the decision algorithm in the "difficult" case.

Problem		Newton-Raphson			Algorithm 3.1		CPU Time
n	p	It	max(It)	$\%T$	$\%T$	$\%Yes$	(sec)
10	1.1	2.5	6	97.6	2.4	0.0	0.018
10	1.5	3.8	6	98.5	1.5	0.0	0.025
10	1.9	4.2	5	98.6	1.4	0.0	0.026
10	2.1	3.8	4	98.5	1.5	0.0	0.024
10	3.0	4.0	5	98.6	1.4	0.0	0.026
10	mixed	3.7	5	98.5	1.5	0.0	0.024
25	1.1	2.5	6	97.9	2.1	0.0	0.044
25	1.5	3.9	6	98.5	1.5	0.0	0.063
25	1.9	4.1	6	98.5	1.5	0.0	0.065
25	2.1	3.7	4	98.6	1.4	10.0	0.065
25	3.0	4.2	5	98.6	1.4	20.0	0.067
25	mixed	3.6	5	98.4	1.6	0.0	0.059
50	1.1	2.5	6	97.8	2.2	0.0	0.089
50	1.5	3.8	6	98.4	1.6	0.0	0.125
50	1.9	4.1	6	98.5	1.5	10.0	0.129
50	2.1	3.8	4	98.4	1.6	0.0	0.119
50	3.0	4.1	5	98.5	1.5	0.0	0.133
50	mixed	3.7	6	98.3	1.7	0.0	0.119
100	1.1	2.5	6	97.7	2.3	0.0	0.176
100	1.5	3.8	6	98.4	1.6	0.0	0.250
100	1.9	4.1	6	98.4	1.6	0.0	0.258
100	2.1	3.7	4	98.3	1.7	10.0	0.237
100	3.0	4.1	5	98.9	1.1	80.0	0.265
100	mixed	3.7	6	98.4	1.6	10.0	0.238
250	1.1	2.6	6	97.9	2.1	10.0	0.447
250	1.5	3.8	6	98.5	1.5	20.0	0.623
250	1.9	4.1	6	98.5	1.5	20.0	0.645
250	2.1	3.7	4	98.5	1.5	20.0	0.594
250	3.0	4.1	5	99.0	1.0	100.0	0.657
250	mixed	3.7	6	98.5	1.5	30.0	0.593
500	1.1	2.6	6	97.8	2.2	10.0	0.895
500	1.5	3.8	6	98.5	1.5	20.0	1.247
500	1.9	4.1	6	98.5	1.5	20.0	1.286
500	2.1	3.7	4	98.9	1.1	70.0	1.183
500	3.0	4.1	5	99.1	0.9	100.0	1.312
500	mixed	3.7	6	99.0	1.0	90.0	1.182

Subsequently, we generate randomly one radius for each circle in the interval $(0, (3/4)\|c\|_q)$. Like this, each circle is guaranteed not to include 0, and we try to avoid having 0 almost always inside the convex hull.

The results obtained are included in Table 1. Each line of the table corresponds to averages of 10 uncorrelated examples. The first three columns of Table 1 describe the problem characteristics. The last row in each group, mixed norms, is generated by selecting randomly $p_i \in \{1.1, 1.5, 1.9, 2.1, 3.0\}$ for each circle $1 \leq i \leq n$. The following two columns describe the behavior of the Newton–Raphson algorithm to find the two tangent hyperplanes, i.e., *It* contains the average number of iterations per execution of the Newton–Raphson algorithm, max(*It*) contains the maximum number of iterations taken by an execution of the same algorithm, and %*T* contains the percentage of the total average computation time spent on finding the hyperplanes.

The following two columns describe the behavior of Algorithm 3.1; i.e. %*T* is the percentage of the total average time taken by Algorithm 3.1, and %*Yes* is the percentage of problems where 0 was found to be in the convex hull of the corresponding tangent points.

Finally, the last column includes the total average execution times in seconds of AST Bravo.

Since in Table 1 we generated instances with a high percentage of *Yes* answers, we also applied the algorithm to the probably more difficult instances with a high percentage of *No* answers. This is achieved by changing in the procedure that the centers are drawn uniformly from $[-1, 99] \times [-1, 99]$. These results are summarized in Table 2.

Some interesting conclusions can be drawn from these results. First of all, the number of Newton–Raphson iterations required to compute each hyperplane with the given precision is always very low. Secondly, Algorithm 3.1 proves to be extremely efficient in practice (recall that the number of points of its input is twice the number of circles). Finally, when the percentage of time taken by each stage of the algorithm is considered, the joint effort exhibits a very strong regularity.

References

1. FRENK, J. B. G., GROMICHO, J., and ZHANG, S., *General Models in Min-Max Continuous Location: Theory and Solution Techniques*, Journal of Optimization Theory and Applications, Vol. 89, pp. 39–63, 1996.
2. LOVE, R. F., MORRIS, J. G., and WESOLOWSKY, G. O., *Facility Location: Models and Methods*, Elsevier Science Publisher B. V., North-Holland, Amsterdam, Holland, 1988.
3. PLASTRIA, F., *Continuous Location Anno 1992: A Progress Report*, Studies in Locational Analysis, ISOLDE VI Survey Papers, Vol. 5, pp. 85–127, 1993.
4. PLASTRIA, F., *On Destination Optimality in Asymmetric Distance Fermat–Weber Problems*, Annals of Operations Research, Vol. 40, pp. 355–369, 1992.
5. PLASTRIA, F., *Localization in Single Facility Location*, European Journal of Operational Research, Vol. 18, pp. 215–219, 1984.

6. GRÖTSCHEL, M., LOVÁSZ, L., and SCHRIJVER, A., *Geometric Algorithms and Combinatorial Optimization*, Springer Verlag, Berlin, Germany, 1988.
7. VARBERG, D. E., and ROBERTS, A. W., *Convex Functions*, Academic Press, New York, New York, 1973.
8. PREPARATA, F. P., and SHAMOS, M. I., *Computational Geometry: An Introduction*, Springer Verlag, New York, New York, 1985.
9. KALLAY, M., *Convex Hulls Made Easy*, Information Processing Letters, Vol. 22, p. 161, 1986.
10. MEGIDDO, N., *Linear Programming in Linear Time When the Dimension Is Fixed*, Journal of the Association for Computing Machinery, Vol. 31, pp. 114–127, 1984.
11. DYER, M. E., *Linear Time Algorithms for Two and Three-Variable Linear Programs*, SIAM Journal on Computing, Vol. 13, pp. 31–45, 1984.
12. SEIDEL, R., *Small-Dimensional Linear Programming and Convex Hulls Made Easy*, Discrete and Computational Geometry, Vol. 6, pp. 423–434, 1991.
13. SCHRIJVER, A., *Theory of Linear and Integer Programming*, Wiley, Chichester, England, 1986.
14. FRENK, J. B. G., GROMICHO, J., FIDRICH, M., and ZHANG, S., *An Efficient Algorithm to Check Whether* 0 *Belongs to the Convex Hull of a Finite Number of* l_p*-Circles*, Technical Report 9204/A, Econometric Institute, Erasmus University, Rotterdam, Netherlands, 1992.
15. KREYSZIG, E., *Introduction to Functional Analysis with Applications*, Wiley, New York, New York, 1978.
16. DAHLQUIST, G., BJÖRCK, Å, and ANDERSON, N., *Numerical Methods*, Prentice-Hall, Englewoods Cliffs, New Jersey, 1974.
17. BAZARAA, M. S., and SHETTY, C. M., *Nonlinear Programming: Theory and Algorithms*, Wiley, New York, New York, 1979.