ACTA MATHEMATICA VIETNAMICA
Volume 28, Number 2, 2003, pp. 185-199

A DISCRETE LOCATION PROBLEM

HOANG TUY, NGUYEN DUC NGHIA AND LE SI VINH

Abstract. An efficient algorithm based on discrete monotonic optimization is proposed for finding the largest ball centered in a given finite $S \subset \mathbb{R}^n$ and disjoint from any of m given balls in $\mathbb{R}^n$.

1. Introduction

We are concerned with a discrete version of the following computational geometry problem:

(DC) *Given m balls in $\mathbb{R}^n$, of centres $a^j \in \mathbb{R}^n_{++}$ and radii $\theta_j \geq 0$, and a bounded set $S \subset \mathbb{R}^n_+$, find the largest ball that has centre in S and is disjoint from any of these m balls.*

In engineering design this problem appears as a variant of the "design centring problem" [4]. An important special case of it when $\theta_j = 0$, $j = 1, \ldots, m$, is the "largest empty ball problem" formulated as follows:

(LB) *Given m points $a^1, \ldots, a^m$ in $\mathbb{R}^n_{++}$ and a bounded set $S \subset \mathbb{R}^n_+$ find the largest ball that has centre in S and contains none of these m given points.*

In location theory (DC) is interpreted as a "maximin location problem". Then (LB) is the problem that arises when we want to construct a residence in a given area S so that it lies outside the polluted sphere of each of m obnoxious facilities and as far as possible from the nearest of these obnoxious facilities.

A global optimization method was first proposed in [9] for solving the general design centring problem (see also [3], [4]). Since analytically (DC) can be reformulated as a nonconvex quadratic optimization problem it can, theoretically, be approached by different nonconvex quadratic programming methods developed in recent years (see e.g. [12], [2]). Let us also mention a d.c. optimization method, combining outer, inner approximation and branch and bound, proposed in [8] for the problem (LB).

However, to date very little experience has been available on the practical computational implementation of any of the above methods in dimension $n > 2$. Although the largest empty ball problem in dimension 2 is known to be solvable in time $O(m \log m)$, traditional methods for this problem, such as Voronoi diagram,

Received May 25, 2002; in revised form September 5, 2002.
Key words and phrases. Maximin location problem, design centring, largest empty ball, discrete monotonic optimization.

which have proved to be very efficient in dimension 2, cannot be easily extended to higher dimension.

In [15] it was shown that Problem (DC) or (LB) can be reduced to solving a number of successive feasibility problems of the following form:

(FP) *Given m balls in $\mathbb{R}^n$, of centres $a^j \in \mathbb{R}^n_{++}$ and radii $\alpha_j \geq 0$, and a bounded set $S \subset \mathbb{R}^n_+$, find a point $x \in S$ lying outside all these m balls, if there is one.*

In the case when $S \subset \mathbb{R}^n$ is a polytope, an efficient d.c. optimization algorithm was developed in [15] to solve the feasibility problem (FP) and thereby, to serve as the main subroutine for solving problems (DC) and (LB).

When *S is a finite set* problems (DC), (LB), (FP) turned out to be quite complicated if S is very large. In particular, the discrete problem (FP), which is basic for the study of discrete problems (DC) and (LB), becomes a combinatorial optimization problem belonging to a class of notoriously difficult problems. To our knowledge no serious computational result for the discrete problem (FP) in dimension $n > 2$ is available as yet.

The aim of the present paper is to discuss a practical approach to the discrete problem (FP) based on ideas of discrete monotonic optimization developed in [5].

The paper consists of several sections. In Section 2 we formulate (FP) (for a finite set S) as a discrete monotonic optimization problem (DM). Section 3 discusses some basic properties of the problem. Section 4 describes an operation called the Z-adjustment which will play a fundamental role in our solution method. Section 5 presents the discrete polyblock method for solving (DM) based on ideas developed earlier in [5]. Section 6 is devoted to an alternative branch and bound algorithm, combining ideas of d.c. optimization with polyblock approximation. The last section presents an illustrative example and some preliminary computational results.

2. Discrete monotonic reformulation

For convenience let us recall the problem we are concerned with:

(FP) *Given m balls in $\mathbb{R}^n$, of centres $a^j \in \mathbb{R}^n_{++}$ and radii $\alpha_j \geq 0$, and a finite set $S \subset \mathbb{R}^n_+$, find a point $x \in S$ lying outside all these m balls, if there is one.*

In terms of location theory, the problem is to check whether there is a point in S outside all the polluted spheres of m given obnoxious facilities located at a^j, $j = 1, \ldots, m$, with pollution radii α_j. Obviously, when $|S|$ is small, the simplest method for solving problem (FP) is to check $|S|$ inequalities:

$$\min_{j=1,\ldots,m} (\|x - a^j\| - \alpha_j) > 0.$$

However, this approach becomes inefficient or even impracticable when $|S|$ is very large.

As was established in [15] (FP) can be solved by solving the following mathematical program

$$\text{(Q)} \qquad \max\{\|x\|^2 - \varphi(x) |\ x \in S\},$$

where $\varphi(x)$ is a convex piecewise affine function defined by

$$\varphi(x) = \max_{j=1,\dots,m}(2\langle a^j, x\rangle + \alpha_j^2 - \|a^j\|^2). \tag{1}$$

If the optimal value of this program is positive, an optimal solution of (Q) will solve (FP). Otherwise, (FP) is infeasible.

Let us introduce some notations and definitions. For any two vectors $x, y \in \mathbb{R}^n$ we write $x \leq y$ ($x < y$, resp.) and say that x is *dominated* (strictly dominated, resp.) by y to mean that $x_i \leq y_i$ ($x_i < y_i$, resp.) for every $i = 1, \dots, n$. If $a \leq b$ then the box $[a, b]$ ($(a, b]$, resp.) is the set of all $x \in \mathbb{R}^n$ satisfying $a \leq x \leq b$ ($a < x \leq b$, resp.). A function $u : [a, b] \to \mathbb{R}$ is said to be *increasing* if

$$a \leq x \leq y \leq b \quad \Rightarrow \quad u(x) \leq u(y).$$

Now, since $a^j, x \in \mathbb{R}^n_{++}$, both functions $\|x\|^2$ and $\varphi(x)$ are increasing, and the objective function in problem (Q) is a difference of two monotonic functions (a d.m. function).

Let B be a box known to contain at least an optimal solution, if any, and define $[a, b] \subset B$ such that

$$a_i = \min_{x \in B \cap S} x_i, \quad b_i = \max_{x \in B \cap S} x_i \quad i = 1, \dots, n \tag{2}$$

Introducing an additional variable t we can rewrite (Q) as

$$\max\{\|x\|^2 + t |\ \varphi(x) + t \leq 0,\ x \in S \cap [a, b],\ a_{n+1} \leq t \leq b_{n+1}\} \tag{3}$$

where $a_{n+1} = -\varphi(b)$, $b_{n+1} = -\varphi(a)$. This problem belongs to the class of discrete monotonic optimization problems studied in [5].

By setting

$$\begin{aligned} z = (x, t) \in \mathbb{R}^n \times \mathbb{R},\ \bar{a} = (a, a_{n+1}),\ \bar{b} = (b, b_{n+1}), \\ Z = \{(x, t) |\ x \in S,\ t = -\varphi(x)\} \\ f(z) = \|x\|^2 + t,\ g(z) = \varphi(x) + t,\ G = \{z |\ g(z) \leq 0\} \end{aligned}$$

problem (Q) can be restated in the usual monotonic format:

$$\max\{f(z) |\ z \in G,\ z \in Z \subset [\bar{a}, \bar{b}]\}. \tag{DM}$$

Here the functions $f(z)$ and $g(z)$ are increasing in $[\bar{a}, \bar{b}]$, and the set Z is a discrete subset of $[\bar{a}, \bar{b}]$.

3. Basic properties

We first investigate some basic properties of problem (DM).

Proposition 1. (i) *G is closed and contains $\bar{a}$ in its interior.*

(ii) *If $z \in G$ and $z' \leq z$ then $z' \in G$.*

Proof. (i) follows from the fact that $g(z)$ is continuous and $g(\bar{a}) = \varphi(a) + a_{n+1} = \varphi(a) - \varphi(b) < 0$.

(ii) is also obvious because the function $g(z)$ is increasing, so if $g(z) \leq 0$ and $z' \leq z$ then $g(z') \leq g(z) \leq 0$. □

The property (ii) is expressed by saying that the set G is *normal*. For any set $A \subset [\bar{a}, \bar{b}]$ the set $A^{\rceil} := \cup_{z \in A}[\bar{a}, z]$ is called the *normal hull* of A. Clearly $G^{\rceil} = G$.

Proposition 2. *If A is compact then so is $A^{\rceil}$.*

Proof. If A is compact then A is contained in a ball B around $\bar{a}$, and if $x^k \in A^{\rceil}$, $k = 1, 2, \dots$, then since $x^k \in [\bar{a}, z^k] \subset B$, there exists a subsequence $\{k_\nu\} \subset \{1, 2, \dots\}$ such that $z^{k_\nu} \to z^0 \in A$, $x^{k_\nu} \to x^0 \in [\bar{a}, z^0]$, hence $x^0 \in A^{\rceil}$, proving the compactness of $A^{\rceil}$. □

For any finite set $T \subset [\bar{a}, \bar{b}]$, the set $P = T^{\rceil}$ is called a *polyblock* in $[\bar{a}, \bar{b}]$ and each point $v \in T$ is called a vertex of the polyblock P. We also write $T = \text{vert}P$ to mean that T is the vertex set of P. A vertex v of a polyblock P is called *proper* if it is not dominated by any other vertex v', i.e. if $v' = v$ whenever $v \leq v'$ and $v' \in \text{vert}P$. The proper vertex set of a polyblock P is denoted by $\text{pvert}\,(P)$.

Proposition 3. *For any point z in a polyblock P there is a proper vertex v of P such that $z \leq v$.*

Proof. If $z \in \text{pvert}\,P$ one can take $v = z$. Otherwise, if T denotes the vertex set of P then, since $P = \bigcup_{v \in T} [\bar{a}, v]$, there is a $v \in T$ such that $z \in [\bar{a}, v]$, i.e. $v \geq z$. Either $v \in \text{pvert}P$ and we are done, or by definition of proper vertices, there is at least one $y \in T$ satisfying $y \geq z$. Let $V = \{y \in T |\ y \geq z\}$. Since this is a finite set ordered by the dominance relation, it has a maximal element v in this ordering. Clearly v is proper, since any $v' \in T$ such that $v' \geq v$ would satisfy $v' \geq z$ and hence would equal v by maximality of v. □

This proposition shows that the proper vertex set fully determines the polyblock and that it is the minimal set having this property.

Now define $D = (G \cap Z)^{\rceil}$, i.e. D is the normal hull of the set $G \cap Z$. Since $G \cap Z$ is a finite set, D is a polyblock and $\text{pvert}(D) \subset G \cap Z$.

Proposition 4. *An optimal solution of* (DM) *is achieved at a proper vertex of the polyblock D with vertex set $G \cap Z$.*

Proof. Let $\bar{z}$ be an optimal solution. Obviously $\bar{z} \in D$. Since D is a polyblock there exists by Proposition 3 a proper vertex v of D such that $\bar{z} \in [\bar{a}, v]$. We have $v \in G \cap Z$ and $f(v) \geq f(\bar{z})$ while $\bar{z}$ is an optimal solution, therefore v is also an optimal solution. □

We have thus reduced the problem to maximizing the increasing function $f(z)$ over the proper vertex set of the polyblock D. The difficulty, of course, is that this proper vertex set of D, though finite, is not easily computable.

In the next section we show how to get round this difficulty.

4. The S-adjustment operation

Proposition 5. *For any $u \in G$ and $v \in [\bar{a}, \bar{b}]$ such that $u \leq v$ let $z_{uv} = u + \lambda(v - u) \in G$, with*

$$\lambda = \sup\{\alpha \geq 0 |\ u + \alpha(v - u) \in G\}. \tag{4}$$

Then $v \in G$ if $\lambda \geq 1$, and $v \notin G$, if $\lambda < 1$.

Proof. Since G is closed, $z_{uv} \in G$. By normality of G, if $u + \alpha(v - u) \in G$ then $u + \alpha'(v - u) \in G$ for all $\alpha' \in [0, \alpha]$. Hence $v \in G$ if and only if $\lambda \geq 1$. □

We also write $z_{uv} = \pi_u(v)$ and if $u = \bar{a}$ we simply write $\pi(v)$ instead of $\pi_{\bar{a}}(v)$. Since $v \notin G$ and $u \leq v$ we always have $\pi_u(v) \leq v$.

Proposition 6. *If $v \notin G$ and $z = \pi(v)$ then $(z, \bar{b}] \cap G = \emptyset$, i.e. the cone $\{z' |\ z' > z\}$ separates v from G.*

Proof. Indeed, let $z = \bar{a} + \lambda(v - \bar{a})$, so that $\bar{a} + \alpha(v - \bar{a}) \notin G\ \forall \alpha > \lambda$. If there exists $y \in G \cap (z, \bar{b}]$ then, since $y \in G$ we have, by normality of G, $[\bar{a}, y] \subset G$, hence, since $z < y$, $\bar{a} + \alpha(z - \bar{a}) \in G$ for $\alpha > \lambda$, contradiction. □

Lemma 1. *If $\bar{a} < v < \bar{b}$, then the set $[\bar{a}, \bar{b}] \setminus (v, \bar{b}]$ is a polyblock with vertices*

$$u^i = \bar{b} + (v_i - \bar{b}_i)e^i, \quad i = 1, \dots, n+1 \tag{5}$$

where e^i denotes the i-th unit vector of $\mathbb{R}^{n+1}$.

Proof. Let $K_i = \{u \in [\bar{a}, \bar{b}] |\ v_i < u_i\}$. Clearly $(v, \bar{b}] = \bigcap_{i=1,\dots,n+1} K_i$, so $[\bar{a}, \bar{b}] \setminus (v, \bar{b}] = \bigcup_{i=1,\dots,n+1} ([\bar{a}, \bar{b}] \setminus K_i)$. But

$$[\bar{a}, \bar{b}] \setminus K_i = \{u |\ \bar{a}_i \leq u_i \leq v_i,\ \bar{a}_j \leq u_j \leq \bar{b}_j\ \forall j \neq i\} = [\bar{a}, u^i],$$

completing the proof of the Lemma. □

Note that $u^1, \dots, u^{n+1}$ are the $n+1$ vertices of the hyperrectangle $[v, \bar{b}]$ that are adjacent to $\bar{b}$. For convenience, if $z, u \in \mathbb{R}^{n+1}$, we also write $y = z \wedge u$ to mean that $y_i = \min\{z_i, u_i\}$ for every $i = 1, \dots, n+1$. With this notation it is clear that $[\bar{a}, z] \cap [\bar{a}, u] = [\bar{a}, z \wedge u]$. Also define $J(z, u) = \{j |\ z_j > u_j\}$.

Proposition 7. *Let P be a polyblock with proper vertex set $T \subset [\bar{a}, \bar{b}]$, let $v \in [\bar{a}, \bar{b}]$ such that $T_* = \{z \in T |\ z \geq v\} \neq \emptyset$. For every $z \in T_*$ define $z^i = z + (v_i - z_i)e^i$, $i = 1, \dots, n+1$. Then the set*

$$T' = (T \setminus T_*) \cup \{z^i |\ z \in T_*,\ z_i > v_i, i = 1, \dots, n+1\}. \tag{6}$$

is the vertex set of the polyblock $P \setminus (v, \bar{b}]$. An element $z^i \in T'$ is improper if and only if there exists $y \in T_*$ such that $J(z, y) = \{i\}$.

Proof. Since $[\bar{a},z]\cap(v,\bar{b}]=\emptyset$ for every $z\in T\setminus T_*$, it follows that $P\setminus(v,\bar{b}]=P_1\cup P_2$, where P_1 is the polyblock with vertex $T\setminus T_*$ and $P_2=(\cup_{z\in T_*}[\bar{a},z])\setminus(v,\bar{b}]=\bigcup_{z\in T_*}([\bar{a},z]\setminus(v,\bar{b}])$. Noting that $[\bar{a},\bar{b}]\setminus(v,\bar{b}]$ is a polyblock with vertices given by (5), we can then write $[\bar{a},z]\setminus(v,\bar{b}]=[\bar{a},z]\cap([\bar{a},\bar{b}]\setminus(v,\bar{b}])=[\bar{a},z]\cap(\bigcup_{i=1,\dots,n+1}u^i)=\bigcup_{i=1,\dots,n+1}[\bar{a},z]\cap[\bar{a},u^i]=\bigcup_{i=1,\dots,n+1}[\bar{a},z\wedge u^i]$, hence $P_2=\cup\{[\bar{a},z\wedge u^i]|\ z\in T_*, i=1,\dots,n+1\}$, which shows that the vertex set of $P\setminus(v,\bar{b}]$ is the set T' given by (6).

It remains to show that every $y\in T\setminus T_*$ is proper, while a z^i with $z\in T_*$ is improper if and only if $J(z,y)=\{i\}$ for some $y\in T_*$.

Since every $y\in T\setminus T_*$ is proper in T, if $z'\geq y$ for some $z'\in T'$, then z' must be some z^i with $z\in T_*,\ i\in\{1,\dots,n+1\}$. But then $z\geq z\wedge u^i=z^i\geq y$, conflicting with y being proper in T. Therefore, every $y\in T\setminus T_*$ is proper. On the other hand, if $z^i\leq y$ for some $y\in T\setminus T_*$ then $v_i=z^i_i\leq y_i$, while $v_j\leq z_j=z^i_j\leq y_j\ \forall j\neq i$, hence $v\leq y$. i.e. $y\in T_*$, conflicting with $y\notin T_*$. Consequently, if z^i is improper then $z^i\leq y^l$ for some $(y,l)\neq(z,i)$, $y\in T_*$, $l\in\{1,\dots,n+1\}$. We cannot have $y=z, l\neq i$ for then the relation $z^i\leq z^l$ would imply $z_l=z^i_l\leq z^l_l=v_l$, conflicting with (6). So $y\neq z$ and $z^i_j\leq y^l_j$ $\forall j=1,\dots,n+1$, which means that $v_i\leq y_i$, $z_l\leq v_l$, $z_j\leq y_j\ \forall j\notin\{i,l\}$. Since $v_l\leq y_l$, we thus have $z_j\leq y_j\quad\forall j\neq i$, hence, noting that $z\not\leq y$ (z is proper), we derive $z_i>y_i$, so that $J(z,y)=\{i\}$. Thus any improper z^i must satisfy $J(z,y)=\{i\}$ for some $y\in T_*$. Conversely, if $J(z,y)=\{i\}$ for some $y\in T_*$ then $z_j\geq y_j\quad\forall j\neq i$, hence $z^i\leq y^i$, i.e. z^i is improper. This completes the proof of the Proposition. $\square$

We now introduce the operation $\lceil\cdot\rceil_S$, by defining for any $x\in[a,b]$:

$$\lceil x\rceil_S=\tilde{x}\quad\text{with}\quad\tilde{x}_i=\max_{y\in S}\{y_i|\ y_i<x_i\}\quad i=1,\dots,n. \tag{7}$$

In view of (4), $\lceil x\rceil_S\in[a,b]$ and $\lceil x\rceil_S<x$; we shall refer to the vector $\lceil x\rceil_S$ as the S-*adjustment* of x. Remembering that $Z=\{z=(x,t)|\ x\in S,\ t=-\varphi(x)\}$ we also define, for $z=(x,t)\in[\bar{a},\bar{b}]$:

$$\lceil z\rceil_Z=(\tilde{x},\tilde{t}),\quad\text{with}\quad\tilde{x}=\lceil x\rceil_S,\quad\tilde{t}=-\varphi(\tilde{x}) \tag{8}$$

and call $\lceil z\rceil_Z$ the Z-*adjustment* of z.

A special case frequently encountered is when $S=S_1\times\cdots\times S_n$, and every S_i is a finite set of real numbers. In this case for any $x\in[a,b]$ we have $\tilde{x}=\lceil x\rceil_S\in S$ because $\tilde{x}_i\in S_i\ \forall i=1,\dots,n$. (For example if S_i is the set of natural numbers, then $\tilde{x}_i$ is the largest integer still less than x_i.)

Also if $\lceil x\rceil_S\in S$ then it is the maximal element of the set $S_{[a,x)}:=\{x'\in S\cap[a,b]|\ x'<x\}$, i.e. $x'\leq\lceil x\rceil_S\ \ \forall x'\in S_{[a,x)}$ (but the converse may not be true).

For our purpose the most useful property of S-adjustment is the following.

Lemma 2. *Let $z = (x, t)$ with $t = -\varphi(x)$. If $[z, \bar{b}] \cap (G \cap Z) = \emptyset$ and $\tilde{z} = \lceil z \rceil_Z$ then $(\tilde{z}, \bar{b}] \cap (G \cap Z) = \emptyset$.*

Proof. Suppose there is $z' \in (\tilde{z}, \bar{b}] \cap (G \cap Z)$. Then $z' = (x', t')$ with $x' \in S$, $t' = -\varphi(x')$, $x' > \tilde{x}$, $t' > \tilde{t}$. On the other hand, since $z' \in G \cap Z$ while $[z, \bar{b}] \cap (G \cap Z) = \emptyset$, there is at least one $i \in \{1, \dots, n+1\}$ such that $z'_i < z_i$. The fact $x' > \tilde{x}$ (i.e. $z'_i > \tilde{x}_i \ \forall i = 1, \dots, n$) then implies that $t' < z_{n+1} = t$, and noting that $t = -\varphi(x) < -\varphi(\tilde{x})$ (because $\tilde{x} < x$), we have $t' < -\varphi(\tilde{x}) = \tilde{t}$, a contradiction. □

Proposition 8. *For $v \notin G$, if $z = \pi(v)$ and*

$$\tilde{v} = \begin{cases} z & \text{if } z \in Z \\ \lceil z \rceil_Z & \text{if } z \notin Z. \end{cases} \tag{9}$$

then $(\tilde{v}, \bar{b}] \cap (G \cap Z) = \emptyset$, i.e. the cone $\{z |\ z \geq \tilde{v}\}$ separates v from $D = (G \cap Z)^{\rceil}$.

Proof. By Proposition 6 we always have $(z, \bar{b}] \cap G = \emptyset$. If $z \in Z$ then, by (9) $\tilde{v} = z$, hence $(\tilde{v}, \bar{b}] \cap (G \cap Z) = \emptyset$. If $z \notin Z$, then $\tilde{v} = \lceil z \rceil_Z$, hence, by Lemma 2, $(\tilde{v}, \bar{b}] \cap (G \cap Z) = \emptyset$. □

5. The discrete polyblock algorithm

With the above backgound we now describe a method for solving (DM). This method consists in constructing a sequence of polyblocks, $P_0 \supset P_1 \supset \cdots$ together with a sequence of numbers $\gamma_0 \leq \gamma_1 \leq \cdots$, such that:

(i) $\gamma_k = f(\bar{z}^k)$, for some $\bar{z}^k \in G \cap Z$ (current best feasible solution) if $\gamma_k > 0$;

(ii) $P_k \supset G \cap Z_{\gamma_k}$ where

$$Z_{\gamma_k} = \{(x, t)|\ x \in S_{\gamma_k}, t = -\varphi(x)\},$$
$$S_{\gamma_k} = \{x \in S |\ \|x\|^2 - \varphi(x) > \gamma_k\}.$$

We will show that the sequence $P_0 \supset P_1 \supset \cdots$ can be so constructed that it will terminate at some $P_k = \emptyset$: then the corresponding γ_k will be the optimal value of the problem, unless the latter has no optimal solution.

As initial polyblock P_0 we can take any $P_0 \supset G \cap Z$, e.g. $P_0 = [\bar{a}, \bar{b}]$, with vertex set $T_0 = \{\bar{b}\}$, and $\gamma_0 = 0$. At iteration $k = 0, 1, \dots$, let P_k be the current polyblock, T_k its vertex set, γ_k (current best value) and $\bar{z}^k \in G \cap Z$ (current best feasible solution), satisfying (i) and (ii).

Then we prune T_k by droping every improper element, and every $v \in T_k$ such that $f(v) \leq \gamma_k$. Letting $\tilde{T}_k$, $\tilde{P}_k$ be the resulting set and polyblock, we reset $T_k \leftarrow \tilde{T}_k$, $P_k \leftarrow \tilde{P}_k$. If $T_k = \emptyset$ the procedure terminates: $\bar{z}^k$ is optimal (if $\gamma_k > 0$), or the problem is infeasible (if $\gamma_k = 0$). If $T_k \neq \emptyset$ select (by an arbitrary rule) $v^k \in T_k$. Two cases may arise:

<u>Case 1</u>: $v^k \in G \cap Z$. Then we let $\tilde{v}^k = v^k$.

Since $\tilde{v}^k$ is feasible, we use it to define the new current best value γ_{k+1} and new current best feasible solution $\bar{z}^{k+1}$ (reset $\gamma_{k+1} = \max\{\gamma_k, f(\tilde{v}^k)\}$, $\bar{z}^{k+1} = \tilde{v}^k$

if $\gamma_{k+1} > \gamma_k$, $\bar{z}^{k+1} = \bar{z}^k$ otherwise), then define P_{k+1} to be the polyblock with vertex set $T_{k+1} = T_k \setminus \{\tilde{v}^k\}$ and go to the next iteration.

Case 2: $v^k \notin G \cap Z$. In this case we compute $z^k = \pi(v^k)$ then set $\tilde{v}^k = z^k$ if $z^k \in S_{\gamma_k}$, $\tilde{v}^k = \lceil z^k \rceil_{Z_{\gamma_k}}$ if $z^k \notin S_{\gamma_k}$. If $\tilde{v}^k$ is feasible we use it to define γ_{k+1} and $\bar{z}^{k+1}$.

Letting $T_{k,*} = \{z \in T_k | \ z \geq \tilde{v}^k\}$, we further compute

$$T_{k+1} = (T_k \setminus T_{k,*}) \cup \{z^{k,i} | \ z \in T_{k,*}, \ z_i > \tilde{v}_i^k, i = 1, \ldots, n+1\}$$

where $z^{k,i} = z + (\tilde{v}_i^k - z_i)e^i$, $i = 1, \ldots, n+1$. Defining then P_{k+1} to be the polyblock with vertex set

$$T_{k+1} = (\tilde{T}_k \setminus \tilde{T}_{*k}) \cup \{z^{k,1}, \ldots, z^{k,n+1}\}. \tag{10}$$

we go to the next iteration.

Proposition 9. *The polyblock P_{k+1} satisfies $G \cap Z_{\gamma_{k+1}} \subset P_{k+1} \subset P_k \setminus (\tilde{v}^k, \bar{b}]$. In particular, conditions* (i), (ii) *still hold for $k \leftarrow k+1$.*

Proof. We have $P_{k+1} \subset P_k \setminus (\tilde{v}^k, \bar{b}]$ by Proposition 6. To show that $G \cap Z_{\gamma_{k+1}} \subset P_{k+1}$, observe that in case 1, $(v^k, \bar{b}] \cap G = \emptyset$ because $v^k \in \text{pvert}(P_k)$. Noting that $f(z) \leq \gamma_{k+1}$ $\forall x \in [a, v^k]$, we can then write $G \cap S_{k+1} \subset (G \cap S_k) \setminus [a, v^k] \subset P_k \setminus [a, v^k] \subset P_{k+1}$. In case 2, if $v^k \in G \setminus Z$ then, since $P_k \supset G \cap Z_{\gamma_k}$ whereas $[v^k, \bar{b}] \cap P_k = \{v^k\}$ we must have $[v^k, \bar{b}] \cap G \cap Z_{\gamma_k} = \emptyset$. Therefore, by Proposition 7, $(\tilde{v}^k, \bar{b}] \cap G \cap Z_{\gamma_k} = \emptyset$, and consequently, $P_{k+1} \supset G \cap Z_{\gamma_k} \supset G \cap Z_{\gamma_{k+1}}$. On the other hand, if $v^k \notin G$ (in case 2), Proposition 9 implies that $(\tilde{v}^k, \bar{b}] \cap G \cap Z_{\gamma_k} = \emptyset$, and again $P_{k+1} \supset G \cap Z_{\gamma_k} \supset G \cap Z_{\gamma_{k+1}}$. $\square$

Thus, P_{k+1}, γ_{k+1} and $\bar{z}^{k+1}$ still satisfies (i), (ii) for $k \leftarrow k+1$, and so the above described procedure continues further if $T_{k+1} \neq \emptyset$. In a formal way we can state

Algorithm A. *Initialization.* Let $T_0 = \{\bar{b}\}$. Let $\bar{z}$ be the best feasible solution available (the current best feasible solution), $\bar{\gamma} = \max\{0, f(\bar{z})\}$ (current best value). If no feasible solution is available, let $\bar{\gamma} = 0$. Set $k = 0$

Step 1. From T_k remove: all $z \in T_k$ such that $f(z) \leq \bar{\gamma}$ and all improper elements. Let $\tilde{T}_k$ be the resulting set. Reset $T_k \leftarrow \tilde{T}_k$.

Step 2. If $T_k = \emptyset$, terminate: if $\bar{\gamma} = 0$, the problem is infeasible; if $\bar{\gamma} > 0$, $\bar{z}$ is an optimal solution.

Step 3. If $T_k \neq \emptyset$, select $v^k \in T_k$.

If $v^k \in G \cap S$ define $T_{k+1} = T_k \setminus \{v^k\}$, update $\bar{\gamma}$ and $\bar{z}$ (using v^k), set $k \leftarrow k+1$ and go back to Step 1.

Step 4. a) If $v^k \in G \setminus S$, compute $\tilde{v}^k = \lceil v^k \rceil_{Z_{\bar{\gamma}}}$ (using formula (7) with S replaced by $S_{\bar{\gamma}} := \{x \in S | \ \|x\|^2 - \varphi(x) > \bar{\gamma}\}$).

b) If $v^k \notin G$ compute $z^k = \pi(v^k)$ and define $\tilde{v}^k = z^k$ if $z^k \in S_{\gamma_k}$, and $\tilde{v}^k = \lceil z^k \rceil_{Z_{\bar{\gamma}}}$ if $z^k \notin S_{\gamma_k}$.

Update $\bar{\gamma}$ and $\bar{z}$ if $\tilde{v}^k \in G \cap Z$.

Step 5. Let $T_{k,*} = \{z \in T_k |\ z \geq \tilde{v}^k\}$. Compute

$$T'_k = (T_k \setminus T_{k,*}) \cup \{z^{k,i} |\ z \in T_{k,*}, z_i > \tilde{v}^k_i,\ i = 1, \dots, n+1\}.$$

where $z^{k,i} = z + (\tilde{v}^k_i - z_i)e^i$. Let T_{k+1} be the set that remains from T'_k after removing every z^i such that $\{j |\ z_j > y_j\} = \{i\}$ for some $y \in T_{k,*}$.

Set $k \leftarrow k + 1$ and go back to Step 1.

Theorem 1. *Algorithm A is finite.*

Proof. Since for every $i = 1, \dots, n+1$ the set $X_i = z_i(Z) = \{\xi \in \mathbb{R} |\ \xi = z_i, z \in Z\}$ is finite, so is the set $X = \prod_{i=1,\dots,n+1} X_i$. At each iteration k a point $v^k \in P_k$ together with $\tilde{v}^k \leq v^k$ are generated such that either $\tilde{v}^k < v^k$, $P_{k+1} \subset P_k \setminus (\tilde{v}^k, \bar{b}]$, or $\tilde{v}^k = v^k$, $P_{k+1} \subset P_k \setminus [v^k, \bar{b}]$. Hence, $[\tilde{v}^k, \bar{b}]$ cannot contain any $\tilde{v}^l$ with $l > k$. That is, $\tilde{v}^k$ is distinct from all $\tilde{v}^l$ with $l > k$, and so there can be no repetition in the sequence $\{\tilde{v}^0, \tilde{v}^1, \dots, \tilde{v}^k, \dots\} \subset X$. The finiteness of the algorithm then follows from the finiteness of the set X. □

Remark 1. If in Step 3 we always select $v^k \in \operatorname{argmax}\{f(z) |\ z \in T_k\}$ then when $v^k \in G \cap S$ the algorithm terminates, with v^k being optimal. However, this rule for selecting v^k may not always be the best one. Sometimes the rule $v^k \in \operatorname{argmin}\{\|z\| |\ z \in T_k\}$ may help to reach a feasible solution more rapidly. If storage problem is a matter of concern, v^k can be selected according to the depth-first rule, in order to minimize memory requirements.

6. Alternative branch and bound algorithm

Algorithm A can be interpreted as a branch and bound algorithm in which a node z of the monitoring tree represents a box $[\bar{a}, z]$ and branching is performed by splitting a node into $n+1$ descendants while the bound over a node z is taken to be $f(z)$. A positive feature of this algorithm is that the bounding operation is straightforward. However, since each node has $n + 1$ descendants, storage problems may arise with the growth of the number of iterations. Therefore, an alternative rectangular branch and bound algorithm in a more conventional format such as the following one may be more efficient.

Since the basic variables are $x = (x_1, \dots, x_n)$ we branch upon x (and not upon $z = (x, t)$) and use rectangular subdivision. Thus a partition set is a rectangle $M = [p, q] \subset [a, b]$.

Bounding. An upper bound $\mu(M)$ for the value

$$\text{SP}(M) \qquad \max\{\|x\|^2 - \varphi(x) :\ x \in M\}$$

is computed as follows.

• Step 1: For $i = 1, \dots, n$ let $\hat{p}_i = \min_{x \in S \cap M} x_i$, $\hat{q}_i = \max_{x \in S \cap M} x_i$. Set $[p, q] \leftarrow [\hat{p}, \hat{q}]$. (the replacement of $[p, q]$ by $\hat{p}, \hat{q}]$ will be referred to as a box *reduction operation*).

• Step 2: If $\|q\|^2 - \varphi(p) \le 0$, set $\mu(M) = 0$. In fact, for all $x \in [p, q]$ we have $\|p\|^2 \le \|x\|^2 \le \|q\|^2$ while $\varphi(p) \le \varphi(x) \le \varphi(q)$, hence $\|x\|^2 - \varphi(x) \le \|q\|^2 - \varphi(p) \le 0$, hence $\max\{\|x\|^2 - \varphi(x)|\ p \le x \le q\} \le 0$.

• Step 3: (entered with $\|q\|^2 - \varphi(p) > 0$) Compute an *upper bound* $\mu(M)$ of the objective function value over the feasible solutions of (FP) in M. For this, solve the relaxed subproblem

$$\mathrm{LP}(M) \qquad \begin{array}{ll} \text{maximize} & \sum_{i=1}^{n}[(p_i+q_i)x_i - p_iq_i] + t \\ \text{s.t.} & 2\langle a^j, x\rangle + \alpha_j^2 - \|a^j\|^2 + t \le 0,\ j = 1, \dots, s \\ & x \in M \end{array}$$

Take $\mu(M)$ to be the optimal value of this linear program.

• Step 4: (when $\mu(M) > 0$) compute a *lower bound* $\nu(M)$ of the objective function value over the feasible solutions of (FP) in M and if $\nu(M) > 0$ a point $\bar{x}^M$ such that $\nu(M) = \|\bar{x}^M\|^2 - \varphi(\bar{x}^M)$. Two options are proposed for this computation:

Option 1. (to be used if S satisfies the following condition

$$x_i = \min\{z_i^i|\ z^i \in S\} \quad (i = 1, \dots, n) \Rightarrow x \in S, \tag{11}$$

e.g. if $S = S_1 \times \dots \times S_n$). If $\mu(M) > 0$, let x^M be an optimal solution of the linear progam LP(M). Compute $\lceil x^M \rceil_S$, the S-adjustment of x^M. If $\|\lceil x^M \rceil_S\|^2 - \varphi(\lceil x^M \rceil_S) > 0$, then set $\bar{x}^M = \lceil x^M \rceil_S$, and $\nu(M) = \|\bar{x}^M\|^2 - \varphi(\bar{x}^M)$. Otherwise, set $\nu(M) = 0$.

Option 2. Write SP(M) in the monotonic format:

$$\max\{\|x\|^2 + t|\ \varphi(x) + t \le 0,\ x \in S,\ -\varphi(q) \le t \le -\varphi(p)\},$$

that is,

$$\mathrm{SD}(M) \qquad \max\{f(z)|\ z \in G,\ z \in Z \cap [\tilde{p}, \tilde{q}]\}$$

where $z = (x, t)$, $f(z) = \|x\|^2 + t$, $G = \{z|\ \varphi(x) + t \le 0\}$, $Z = \{(x, t)|\ x \in S, t = -\varphi(x)\}$, $\tilde{p} = (-\varphi(q), a)$, $\tilde{q} = (-\varphi(p), q)$. (see Section 2). Apply Algorithm A to SD(M) until evidence of infeasibility or a feasible solution $\bar{x}^M$ is obtained. In the former case reset $\mu(M) = 0$; in the latter case, set $\nu(M) = \|\bar{x}^M\|^2 - \varphi(\bar{x}^M)$.

Thus in option 2, a feasible solution of (FP) in M will always be found, provided there is one; on the other hand, option 2 is computationally more expensive than option 1.

Algorithm B

Step 0. Start with $\mathcal{P}_1 = \mathcal{S}_1 = \{M_1 = [a, b]\}$ (reduced box). Set $k = 1$.
Step 1. For each box $M \in \mathcal{P}_k$ compute $\mu(M)$ and $\nu(M)$ (when $\mu(M) > 0$). Let $CBV = \max\{\nu(M)|\ M \in \mathcal{P}_k\}$, and when $CBV > 0$ let $CBS = \bar{x}^k$ be the best among all $\bar{x}^M$, $M \in \mathcal{P}_k$, i.e. such that $\|\bar{x}^k\|^2 - \varphi(\bar{x}^k) = CBV$.

Step 2. Delete every $M \in \mathcal{S}_k$ such that $\mu(M) \leq CBV$. Let $\mathcal{R}_k$ be the collection of remaining members of $\mathcal{S}_k$. If $\mathcal{R}_k = \emptyset$, then terminate: $\bar{x}^k$ is an optimal solution if $\nu^k > 0$, or the problem is infeasible otherwise.

Step 3. Select $M_k \in \mathcal{R}_k\}$. Choose $j_k \in \text{argmax}_i\{q_i - p_i\}$ and divide M_k into two subboxes via the hyperplane $y_{j_k} = (p^k_{j_k} + q^k_{j_k})/2$. Let $\mathcal{P}_{k+1}$ be the partition of M_k.

Step 4. Set $\mathcal{S}_{k+1} = (\mathcal{R}_k \setminus \{M_k\}) \cup \mathcal{P}_{k+1}$. Set $k \leftarrow k+1$ and go back to Step 1.

Proposition 10. *Algorithm B terminates after finitely many iterations, yielding an optimal solution of* (FP) *or establishing that the problem is infeasible.*

Proof. For every $i = 1, \ldots, n$ the set $X_i = \{\xi \in \mathbb{R} | \ \xi = x_i, x \in S\}$ is finite. Since every box M can be assumed to be of the form $[p, q]$ with $p_i \in S$, $q_i \in S$ $\forall i = 1, \ldots, n$, (see the box reduction operation in the Bounding process), it follows that the total number of nodes of the monitoring tree in each iteration is also finite. Whence the finiteness of the algorithm □

7. Solution method for problem (DC)

So far we have been concerned with problem (FP). Turning now to problem (DC) let us examine how Algorithm 1 (or 2) can be used to solve (DC).

Given a number $r \geq 0$, we say that the value r is feasible if there exists a point $x(r) \in S$ such that the ball of centre $x(r)$ and radius r does not intersect any one of the m given balls of centres a^j, radii θ_j. Clearly checking whether r is feasible and if yes, finding $x(r)$, amounts to solving the following subproblem

(*) FP(r) *Find a point $x(r) \in S$ lying outside any one of the m balls of centres a^j and radii $\alpha_j = \theta_j + r$ $(j = 1, \ldots, m)$.*

Since this problem can be solved by either Algorithm A or Algorithm B, and since solving (DC) amounts to finding the *maximal feasible value* $r_{\max}$, to solve (DC) we can proceed as follows:

Algorithm C (for solving (DC)

Let s be an estimated upper bound for $r_{\max}$.

Step 0. Solve FP(0) by Algorithm 2. If FP(0) is infeasible then stop: (DC) is infeasible. Otherwise, we obtain a point $x(0) \in S$. If $\min_j\{\|x(0) - a^j\| - \theta_j\} = 0$, stop: x^0 is a solution of (DC) with optimal value 0. If $\min_j\{\|x(0) - a^j\| - \theta_j\} > 0$, set $z^\ell = \min_j\{\|x(0) - a^j\| - \theta_j\}$, $z^u = s$, $\bar{x} = x(0)$.

Step 1. If $z^u - z^\ell \leq \varepsilon$, stop: $\bar{x}$ is ε-optimal solution of (DC) with optimal value $\min_j(\|\bar{x} - a^j\| - \theta_j - r)$. Otherwise, go to Step 2.

Step 2. Solve Q$((z^\ell + z^u)/2)$ to obtain the optimal value $\bar{\rho}$ and an optimal solution $\bar{x}$. If $\bar{\rho} = 0$, stop: $\bar{x}$ is optimal solution of (DC). If $\bar{\rho} > 0$, reset $z^\ell \leftarrow \min_j(\|\bar{x} - a^j\| - \theta_j)$ and go back to Step 1. If $\bar{\rho} < 0$, reset $z^u \leftarrow (z^\ell + z^u)/2$ and go back to Step 1.

8. Illustrative example

To illustrate the method we consider a problem (DC) in $\mathbb{R}^2$ with $m = 10$ balls of centres a^j and radii θ_j, $j = 1, \dots, 10$ as given in the following table and with S being the set of points with integral coordinates in the rectangle $1 \le x_1 \le 12$, $1 \le x_2 \le 12$. (so $a = (1, 1)$, $b = (12, 12)$).

j	a^j	θ_j
1	(1, 5)	3
2	(3, 12)	2
3	(12.5, 11.5)	2.5
4	(14.5, 5)	3.5
5	(5, 8)	1
6	(6, 2)	2
7	(7, 10)	1
8	(10, 8)	1
9	(9, 2)	1
10	(6.5, 5.5)	0.5

The solution of this problem by Algorithm C yields the following results:

Running time (on a PC Pentium II 300 MHz): 5 Seconds

Number of solved subproblems FP: 7 (see Table 1)

Number of interations: 65

Maximal number of active nodes in branch and bound trees: 5

Optimal Solution: ball of centre $\bar{x} = (9, 5)$ and radius $r_{\max} = 2.00$.

The example is illustrated by Figure 1.

Table 1. $z^l = 1.08$, z^u =3.89

Subprob	z	$\bar{x}$	z^t	N	# Iter	Time
1	2.49	infeasible		3	8	0.52
2	1.78	(9,5)	2.0	1	2	0.08
3	2.24	infeasible		3	8	0.51
4	2.12	infeasible		3	9	0.53
5	2.06	infeasible		4	12	0.68
6	2.03	infeasible		5	13	0.70
7	2.02	infeasible		5	13	0.71

9. Computational experiments

Algorithm B has been coded in C^{++} and tested on a number of problem instances of dimension ranging from 2 to 10. The computations have been carried out on a PC Pentium 300 MHz with linear subproblems solved using the LP software CPLEX. The results are summarized in Table 2, where the following abbreviations are used:

n: dimension

m: number of given points a^j

sub:number of subproblems FP needed

N: maximal number of active nodes in the branch and bound trees

Time: running time in seconds.

Table 2

Problem	n	m	#sub	N	Time
1	2	100	11	7	7
2	2	200	12	10	12
3	2	300	11	18	28
4	2	400	12	21	43
5	3	50	9	11	9
6	3	150	10	17	19
7	3	250	10	34	54
8	4	80	9	9	10
9	4	120	9	10	11
10	4	300	9	33	54
11	5	100	8	17	13
12	5	200	9	28	35
13	6	80	8	13	13
14	6	250	8	27	37
15	7	100	8	29	19
16	7	300	8	20	34
17	8	150	8	39	48
18	8	250	8	25	40
19	9	100	8	10	12
20	9	250	8	24	48
21	10	100	8	24	24
22	10	200	8	20	33
23	10	300	8	25	47
24	10	400	9	33	112

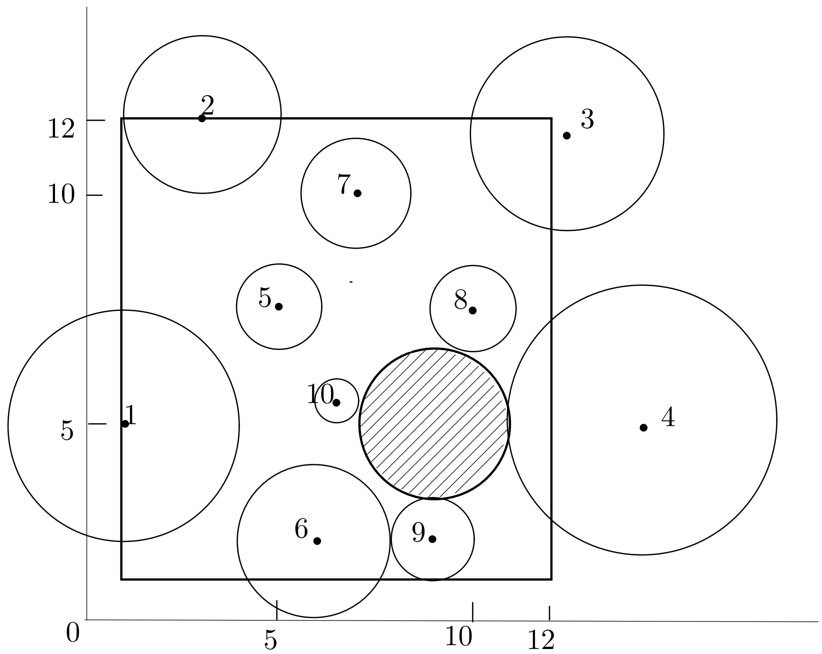

Fig. 1

References

[1] B. Dasarathy and L. J. White, *One some maximin location and classifier problems*, Computer Science Conference, Washington D. C., 1975.

[2] C. A. Floudas, *Deterministic Global Optimization*, Kluwer, 2000.

[3] R. Horst and H. Tuy, *Global Optimization–Deterministic Approaches*. 3rd edition, Springer, 1996.

[4] H. Konno, P. T. Thach and H. Tuy, *Optimization on Low Rank Nonconvex Structures*, Kluwer, 1997.

[5] M. Minoux and H. Tuy, *Discrete monotonic optimization*, Preprint, Institute of Mathematics, 2001.

[6] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer, 1985.

[7] A. Rubinov, H. Tuy and H. Mays, *Algorithm for a monotonic global optimization problem*, Optimization **11** (2000), 464-494.

[8] J. S. Shi and Y. Yamamoto, *A D.C. Approach to the largest empty sphere problem in higher dimension*, in State of the Art in Global Optimization, C. Floudas and P. Pardalos eds, Kluwer, 1996, 395-411.

[9] P. T. Thach, *The design centering problem as a D.C. programming problem*, Mathematical Programming **41** (1988), 229-248.

[10] P. T. Thach and H. Tuy, *Parametric approach to a class of nonconvex global optimization problems*, Optimization **19** (1988), 3-1.

[11] H. Tuy, *Normal sets, polyblocks, and monotonic optimization*, Vietnam J. Math. **27** (1999), 277-300.

[12] H. Tuy, *Monotonic optimization: problems and solution approaches*, SIAM J. Optim. **11** (2000), 464-494.

[13] H. Tuy and L. T. Luc, *A new approach to optimization under monotonic constraint*, J. Global Optim. **18** (2000), 1-15.
[14] H. Tuy and Ng. T. Hoai-Phuong, *A unified monotonic approach to generalized linear fractional programming*, Journal of Global Optimization, to appear.
[15] H. Tuy, N. D. Nghia and L. S. Vinh, *Efficient DC approach to maximin location and design centring problems*, Preprint, Institute of Mathematics, Hanoi, 2001.

Institute of Mathematics,
18 Hoang Quoc Viet Road
10307 Hanoi, Vietnam

E-mail address: htuy@thevinh.ac.vn

Faculty of Information Technology,
Hanoi University of Technology

E-mail address: nghiand@it-hut.edu.vn

Faculty of Information Technology,
Hanoi State University

E-mail address: VinhLS@ho.fpt.vn