manuscript No.
(will be inserted by the editor)

Monotonic Optimization for Sensor Cover Energy Problem

Pham Thi Hoai · **Hoang Tuy**

Received: date / Accepted: date

Abstract We study the Sensor Cover Energy Problem (SCEP) in wireless communication − a difficult nonconvex problem with nonconvex constraints. In [2] a local approach based on DC programming called DCA was proposed for solving this problem. In the present paper, we propose a global approach to SCEP based on the theory of monotonic optimization. Using an appropriate reformulation of SCEP we propose an algorithm for finding quickly a local optimal solution along with an efficient algorithm for computing a global optimal solution. Computational experiments are reported which demonstrate the practicability of the approach.

Keywords Wireless sensor network · Monotonic optimization · Nonconvex programming

1 Introduction

Wireless sensor network has become a subject of intensive study in recent years due to its important applications in communication. An overview of the main fields of application, the relevant factors in designing sensor devices and the main issues arising from adopting different network topologies, has been given by Yick et al. in [12]. An important issue in this research area is the Sensors Coverage Problem discussed in [8] about minimizing the number of sensors from a given set in a network such that each target point in the monitored region is sensed by at least one or more sensors. Various aspects of this problem have been investigated by many authors. In particular, saving energy is one of the most important issues attracting researchers. In the papers [1,3,5,6] the authors consider the "energy-efficient" covers by finding disjoint sets of sensors to be activated in different time periods. By this way the consumption energy of sensors can be saved when the different covers are activated alternately. Another way to achieve energy efficiency as proposed in [4, 7], based on the adjustable sensing range property of the sensors, consists in finding the set covers with minimum sensing range. Using the relationship between the energy cost of a sensor and its sensing range, the problem studied in [2,13] is to determine the sensing range so that the total consumption energy is minimized. Zhou et al. [13] proposed a heuristic algorithm while Astorino et al.[2] treated the "Sensor Cover Energy Problem" (SCEP) by

Pham Thi Hoai
School of Applied Mathematics and Informatics,
Hanoi University of Science and Technology, Hanoi, Vietnam
E-mail: hoai.phamthi@hust.edu.vn

Hoang Tuy (✉)
Institute of Mathematics, 18 Hoang Quoc Viet Road,
10307 Hanoi, Vietnam
E-mail: htuy@math.ac.vn

selecting a set of sensing radii, covering a certain number of targets, so that the total energy cost is minimized.

Mathematically, the problem (SCEP) can be stated as follows.

Consider a wireless network deployed in a certain geographical region, with n sensors at points $s^i \in \mathbb{R}^p, i = \{1,\dots,n\}$, and m target nodes at points $t^j \in \mathbb{R}^p, j = \{1,\dots,m\}$, where p is generally set to 2 or 3. The energy consumption per unit time of a sensor i, denoted by $E_i(r_i)$, is a monotonically non decreasing function of its sensing radius r_i. As assumed in many studies,

$$E_i(r_i) = \alpha_i r_i^{\beta_i} + \gamma, \quad l_i \le r_i \le u_i,$$

where $\alpha_i > 0, \beta_i > 0$ are constants depending on the specific device and γ represents the idle-state energy cost. We wish to determine $r_i \in [l_i, u_i] \subset \mathbb{R}_+, i = 1,\dots,n$, such that each target node $t^j, j = 1,\dots,m$, is covered by at least one sensor and the total energy consumption is minimized. That is,

$$(SCEP) \qquad \begin{array}{ll} \min & \sum_{i=1}^n E_i(r_i) \\ \text{s.t.} & \max_{1\le i\le n}(r_i - \|s^i - t^j\|) \ge 0 \quad j = 1,\dots,m, \\ & l \le r \le u, \end{array}$$

where the last inequalities are componentwise understood.

Setting $x_i = \alpha_i r_i^{\beta_i}, a_{ij} = \alpha_i \|s^i - t^j\|^{\beta_i}$ and $a_i = \alpha_i l_i^{\beta_i}, b_i = \alpha_i u_i^{\beta_i}$ we can rewrite $(SCEP)$ in the form

$$\left. \begin{array}{l} \text{minimize } f(x) = \sum_{i=1}^n x_i \text{ subject to} \\ g_j(x) := \max_{1\le i\le n}(x_i - a_{ij}) \ge 0, \quad j = 1,\dots,m, \\ a \le x \le b. \end{array} \right| \qquad (1)$$

Clearly each $g_j(x), j = 1,\dots,m$, is a convex function, so the problem is a linear optimization under a reverse convex constraint and can in principle be solved by currently available methods of reverse convex programming (see [Tuy, 2016], Chapter 7, Section 7.3). In [Astorino and Miglionico, 2014], the authors use a penalty function to shift the reverse convex constraint to the objective function, transforming (SCEP) into an equivalent dc optimization problem which is then solved by the DCA algorithm of P. D. Tao. Aside from the need of a penalty factor which is not easy to determine even approximately, a drawback of this method is that it can at best provide an approximate local optimal solution which is not guaranteed to be global.

In this paper we propose a global approach to problem SCEP based on monotonic optimization. In fact, each $g_j(x), j = 1,\dots,m$, is an increasing function, i.e. $g_j(x) \ge g_j(x')$ whenever $x \ge x'$, so the problem is actually a standard monotonic optimization problem: minimize the increasing function $f(x) = \sum_{i=1}^n x_i$ under the monotonic constraints $g_j(x) \ge 0,\ j = 1,\dots,m$.

Using an appropriate discrete reformulation of the problem we propose a new solution method for problem SCEP. This includes an algorithm for quickly finding a local optimal solution and an efficient algorithm for computing a global optimal solution.

The practicality of the algorithms is demonstrated by numerical tests on instances of problems with up to 1000 variables for finding a local optimal solution and instances of problems with up to 75 variables for computing a global optimal solution.

The paper is organized as follows. After a review in Section 2 of the basic concepts of monotonic optimization we show in Section 3 that the problem SCEP can be reformulated and studied as a discrete monotonic optimization. Solution methods based on this approach are proposed and numerical results are reported in Section 4. Finally, the paper is closed with some conclusions in Section 5.

2 Preliminaries: Basic Concepts of Monotonic optimization

We first review the basic concepts of monotonic optimization as has been developed in the works [10] and [11].

Throughout the sequel, for any two vectors $x, y \in \mathbb{R}^n$ we write $x \leq y$ ($x < y$, resp.) to mean $x_i \leq y_i$ ($x_i < y_i$, resp.) for every $i = 1, ..., n$. 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.). We write $u = x \vee y$, to mean $u_i = \max\{x_i, y_i\}, i = 1, ..., n$ and $v = x \wedge y$, to mean $v_i = \min\{x_i, y_i\}, i = 1, ..., n$; As usual e^i denotes the ith unit vector of $\mathbb{R}^n$, i.e., a vector such that $e^i_i = 1, e^i_j = 0\ \forall j \neq i$, while $e \in \mathbb{R}^n$ is a vector of all ones, i.e., $e = \sum_{i=1}^{n} e^i$. The symbol 0 will denote the zero number or the zeros vector in $\mathbb{R}^n$, depending upon the context.

A function $f : [a, b] \to \mathbb{R}$ is said to be *increasing* (*decreasing,* resp.) if

$$a \leq x \leq y \leq b \Rightarrow f(x) \leq f(y)\ (f(x) \geq f(y),\ \text{resp.}) \tag{2}$$

A function which is either increasing or decreasing is said to be *monotonic*.

A set $G \subset [a, b]$ is said to be *normal* if $x \in G \Rightarrow [a, x] \subset G$. A set $H \subset [a, b]$ is said to be *conormal* if $x \in H \Rightarrow [x, b] \subset H$. Thus, with $g(x), h(x)$ increasing functions in $[a, b]$ the set $G = \{x \in [a, b] | g(x) \leq 0\}$ is normal and $H = \{x \in [a, b] | h(x) \leq 0\}$ is conormal.

Given a set $A \subset [a, b]$, the *normal hull* of A, written $A\rceil$, is the smallest normal set containing A. The *conormal hull* of A, written $\lfloor A$, is the smallest conormal set containing A.

The normal hull P of a finite set $T \subset [a, b]$ is called a *polyblock* with *vertex set* T. It is easily seen that $P = \cup_{z \in T}[a, z]$. A vertex z of a polyblock is called *proper* if there is no vertex $z' \neq z$ dominating z, i.e., such that $z' \geq z$. An *improper* vertex is a vertex which is not proper. Obviously, a polyblock is fully determined by its proper vertex set; more precisely, *a polyblock is the normal hull of its proper vertices.*

Analogously, the conormal hull Q of a finite set $T \subset [a, b]$ is called a *copolyblock* with *vertex set* T. It is easily seen that $P = \cup_{z \in T}[a, z]$. A vertex z of a copolyblock is called *proper* if there is no vertex $z' \neq z$ dominated by z, i.e., such that $z' \leq z$. An *improper* vertex is a vertex which is not proper. Obviously, a copolyblock is fully determined by its proper vertex set; more precisely, *a copolyblock is the normal hull of its proper vertices.*

Proposition 1 (i) *The intersection of finitely many polyblocks is a polyblock.*
(ii) *The intersection of finitely many copolyblocks is a copolyblock.*

Proof (i) It suffices to consider the case of 2 polyblocks; the case for more than 2 polyblocks is derived by induction. If $P_1 = \cup_{y \in T_1}[a, y], P_2 = \cup_{z \in T_2}[a, z]$ then we have $P = P_1 \cap P_2 = \cup_{y \in T_1, z \in T_2}[a, y \wedge z]$. Thus P is a polyblock with vertex set $\{y \wedge z, y \in T_1, z \in T_2\}$.

(ii) Similarly, if V_1, V_2 are the vertex sets of copolyblocks Q_1, Q_2 then $Q_1 \cap Q_2$ is a copolyblock with vertex set $\{y \vee z, y \in V_1, z \in V_2\}$. □

Proposition 2 (i) *The maximum of an increasing function $f(x)$ over a polyblock is achieved at a proper vertex of this polyblock.*

(ii) *The minimum of an increasing function $f(x)$ over a copolyblock is achieved at a proper vertex of this copolyblock.*

Proof It suffices to prove (i). The maximum of an increasing function $f(x)$ over a box $[0, z]$ is obviously achieved at z. Hence the maximum of $f(x)$ over a polyblock $P = \cup_{z \in T}[a, z]$ is achieved at an element of T, i.e. at a vertex of the polyblock. Moreover, if a point $y \in [a, b]$ is dominated by a point $z \in [a, b]$ then $[a, y] \subset [a, z]$, so the maximum of $f(x)$ over $[a, y] \cup [a, z]$. Therefore the maximum of $f(x)$ over a polyblock P is achieved at a proper vertex. □

Lemma 1 (i) *If $a \le x \le b$ then the set $[a,b]\setminus(x,b]$ is a polyblock with vertices*

$$u^i = b + (x_i - b_i)e^i, i = 1, ..., n. \tag{3}$$

In other words, $[a,b]\setminus(x,b] = \cup_{i=1}^n [a,u^i]$.

(ii) *If $a \le x \le b$ then the set $[a,b]\setminus[a,x)$ is a copolyblock with vertices*

$$v^i = a + (x_i - a_i)e^i, i = 1, ..., n. \tag{4}$$

In other words, $[a,b]\setminus[a,x) = \cup_{i=1}^n [v^i,b]$.

Proof If there exists i such that $b_i = x_i$ then $(x,b] = \emptyset$ and $u^i = b + (x_i - b_i)e^i = b$, hence $[a,b]\setminus(x,b] = [a,b] = \cup_{i=1}^n [a,u^i]$. Otherwise, $a \le x < b$. Let $K_i = \{z \in [a,b] | x_i < z_i\}$. Since $(x,b] = \cap_{i=1}^n K_i$ we have $[a,b]\setminus(x,b] = \cup_{i=1}^n ([a,b]\setminus K_i)$, proving assertion (i) because $[a,b]\setminus K_i = \{z | a_i \le z_i \le x_i, a_j \le z_j \le b_j \forall j \ne i\} = [a,u^i]$. The proof of (ii) is analogous. □

For our purpose we should also note the following property:

Proposition 3 *Let $f(x)$ be an increasing function in $[a,b]$.*

(i) *Any proper vertex of a polyblock P in $[a,b]$ is a local optimal solution of the problem* $\max\{f(x) |\ x \in P\}$.

(ii) *Any proper vertex of a copolyblock Q in $[a,b]$ is a local optimal solution of the problem* $\min\{f(x) |\ x \in Q\}$.

Proof We need only prove (i). Let z be an arbitrary proper vertex of P. Since the set of proper vertices of P is finite, there exists a ball $B(z,\varepsilon)$ of radius ε around z not containing any proper vertex. Noting that z achieves the maximum of $f(x)$ on the box $[a,z]$ we must have $f(y) \le f(z)$ for all $y \in B(z,\varepsilon) \cap [a,z] = B(z,\varepsilon) \cap P$. So z is a local minimum of $f(x)$ over P. □

3 The Sensor cover energy problem (SCEP)

By translating if necessary we can assume that $a = 0$ in problem (1). So the problem we are concerned with is

$$\min \quad f(x) = \sum_{i=1}^n x_i, \tag{5}$$

$$\text{s.t.} \quad g_j(x) = \max_{1\le i\le n} (x_i - a_{ij}) \ge 0,\ j = 1, \dots, m, \tag{6}$$

$$0 \le x \le b.$$

For every $j = 1, \dots, m$ let $a^j := (a_{1j}, \dots, a_{nj}),\ H_j := \{x \in [0,b] |\ g_j(x) \ge 0\}$.

Proposition 4 (i) *We have $H_j = [0,b]\setminus[0,a^j)$ and $H = \cap_{j\in J} H_j$ with $J = \{j \in \{1,\dots,m\} |\ a^j \ge 0\}$;*

(ii) *If J_1 denotes the set of all $j \in J$ for which there is no $k \in J\setminus\{j\}$ satisfying $a^j \le a^k$ then also $H = \cap_{j\in J_1} H_j = \{x \in [0,b] | g_j(x) \ge 0, j \in J_1\}$.*

(iii) *For every $j \in J$ we have $H_j = \cup_{i=1}^n [s^{ij}, b]$ with $s^{ij} = \{(a_{ij} - a_i)e^i, i = 1, \dots, n\}$.*

Proof (i) If $j \in J$ then clearly $H_j = \{x \in [0,b] |\ \max_{1\le i\le n}(x_i - a_{ij}) \ge 0\} = [0,b]\setminus[0,a^j)]$. If $j \notin J$ then since $g_j(0) \ge 0$ while $g_j(x)$ is increasing, we must have $\{x \in [0,b] | g_j(x) \ge 0\} = [0,b]$. So $H_j = [0,b]$ for $j \notin J$, hence $H = \cap_{j\in J} H_j$.

(ii) If $j_1 \in J\setminus J_1$ there exists $j_2 \in J, j_2 \ne j_1$ such that $a^{j_1} \le a^{j_2}$, hence, $[0,a_{j_1}) \subset [0,a_{j_2})$. From (i) we then have

$$\{x \in [0,b] | g_{j_2}(x) \ge 0\} = [0,b]\setminus[0,a^{j_2}) \subset [0,b]\setminus[0,a^{j_1}) = \{x \in [0,b] | g_{j_1}(x) \ge 0\}.$$

That is, every constraint $g_{j_1} \geq 0$ with $j_1 \in J \setminus J_1$ is implied by some constraint $g_{j_2}(x) \geq 0$. Therefore $H = \{x \in [0,b] | g_j(x) \geq 0, j \in J\} = \{x \in [0,b] | g_j(x) \geq 0, j \in J_1\}$.

(iii) Since $H_j = [0,b] \setminus [0,a^j)$, it follows from Proposition 1 that $H_j = \cup_{i=1}^n [s^{ij}, b]$ with $s^{ij} = a_i + (a_{ij} - a_i)e^i, i = 1, \ldots, n$. □

Proposition 5 (i) *The set* $H := \{x \in [0,b] | g_j(x) \geq 0, j \in J_1\}$ *is a copolyblock with vertex set* $S = \{\vee_{j\in J_1} y^j |\ y^j \in S_j\}$, *where* $z = \vee_{j\in J_1} y^j$ *means that* $z_i = \min_{j\in J_1} y_i^j$ *for every* $i = 1, \ldots, n$.

(ii) *Every proper vertex of* H *is a local optimal solution of (SCEP).*

Proof This follows from Propositions 1 and 3. □

4 Solution method for (SCEP)

Proposition 5 shows that a simple procedure for computing a whole set of local optimal solutions of the problem is to generate the proper vertex set of the copolyblock H. Finally the problem is reduced to the following discrete monotonic optimization problem

$$\left.\begin{array}{l} \text{minimize } f(x) = \sum_{i=1}^n x_i \text{ subject to} \\ g_j(x) := \max_{1\leq i\leq n}(x_i - a_{ij}) \geq 0, \quad j \in J_1, \\ x \in S \subset [0,b]. \end{array}\right| \tag{7}$$

where $S = \{\vee_{j\in J_1} y^j |\ y^j \in S_j\}$, and $S_j = \{s^{ij} = \{(a_{ij} - a_i)e^i, i = 1, \ldots, n\}$.

We now apply the branch-reduce-and-bound (BRB) algorithm proposed in [11] for solving this discrete monotonic optimization problem. As a byproduct, the bounding operation in this BRB algorithm will provide a procedure for finding quickly a local solution of problem (7).

To discribe our algorithm in detail, we need first to define a basic operation called S-adjustment operation devised for accommodating the discrete constraint $x \in S$ to the continuous structure of the problem.

Definition 1 (The S-adjustment operation [11]) Consider a box $[p,q] \subset [a,b]$. Given any point $x \in [p,q]$, the *lower S-adjustment* of x is the point

$$\begin{aligned} \lfloor x \rfloor_S = \tilde{x}, \text{ with } \tilde{x}_i &= \max\{y_i | y \in S \cup \{p\}, y_i \leq x_i\}, i \in I \\ &= \max\{y_i | y_i \in \{p_i, 0, a_{ij}, j \in J_1\}, y_i \leq x_i\}, i \in I \end{aligned} \tag{8}$$

and the *upper S-adjustment* of x is the point

$$\begin{aligned} \lceil x \rceil_S = \hat{x}, \text{ with } \hat{x}_i &= \min\{y_i | y \in S \cup \{q\}, y_i \geq x_i\}, i \in I \\ &= \min\{y_i | y_i \in \{q_i, 0, a_{ij}, j \in J_1\}, y_i \geq x_i\}, i \in I. \end{aligned} \tag{9}$$

Next we define the branching, reduction and bounding operations involved in our BRB algorithm.

I. Branching

This is the operation of subdividing a box $B = [p,q]$ (the current most promising box) into two or more subboxes. The most popular (though not always the most efficient) subdivision method is the standard bisection: determine the index $i_B \in \{1, \ldots, n\}$ such that $q_{i_B} - p_{i_B} = \max_{i=1,\ldots,n}(q_i - p_i)$; set $r_{i_B} = (q_{i_B} + p_{i_B})/2$ and divide B into two subboxes

$$B^+ = \{x \in B | x_{i_B} \geq r_{i_B}\},$$

$$B^- = \{x \in B | x_{i_B} \leq r_{i_B}\}.$$

In other words, divide the box $B = [p,q]$ into two equal subboxes by the hyperplane perpendicular to a longest edge of the box at its midpoint.

II. Reduction

Let $B = [p, q]$ be any box that remains for exploration at the current iteration. Before computing a lower bound of $f(x)$ over the box it is important to reduce it as far as possible without losing any feasible point better than the current best. Specifically, if γ is the current best value (CBV) of $f(x)$, i.e. the value of $f(x)$ at the best feasible point known so far, then replace the box $[p, q]$ by a smaller box $[p', q'] = \text{red}_\gamma[p, q]$ still containing all feasible solutions of B satisfying $f(x) \le \gamma$; after that, perform an S-adjustment to get $\text{red}^S_\gamma[p, q] = [\tilde{p}, \tilde{q}]$, where $\tilde{p} = \lceil p' \rceil_S$ and $\tilde{q} = \lfloor q' \rfloor_S$.

The following Proposition shows how to determine the box $[p', q'] = \text{red}_\gamma[p, q]$.

For convenience set $g(x) := \min_{j \in J_1} g_j(x) = \min_{j \in J_1} \max_{i=1}^n (x_i - a_{ij})$.

Proposition 6 *There exists a feasible solution $x \in [p, q]$ such that $f(x) \le \gamma$ only if $g(q) \ge 0$ and $f(p) \le \gamma$. Any such x must be contained in the box $[p', q']$ defined by*

$$p' = q - \sum_{i=1}^{n} \eta_i (q_i - p_i) e^i, \quad q' = p' + \sum_{i=1}^{n} \mu_i (q_i - p'_i) e^i, \tag{10}$$

where, for $i = 1, ..., n$

$$\eta_i = \sup\{\eta | 0 \le \eta \le 1, g(q - \eta_i (q_i - p_i) e^i) \ge 0\} \tag{11}$$

$$\mu_i = \sup\{\mu | 0 \le \mu \le 1, f(p' + \mu_i (q_i - p'_i) e^i) \le \gamma\} \tag{12}$$

Proof Consider any $x \in [p, q]$ satisfying $g(x) \ge 0$ and $f(x) \le \gamma$. Since $f(x), g(x)$ are increasing then $g(q) \ge g(x) \ge 0$ and $f(p) \le f(x) \le \gamma$.

If $x \not\ge p'$ there exists an index i such that $x_i < p'_i = q_i - \eta_i (q_i - p_i)$, i.e., $x_i = q_i - \eta(q_i - p_i)$ with some $\eta > \eta_i$. Clearly then $x \le q - \eta(q_i - p_i) e^i$, hence $g(x) \le g(q - \eta(q_i - p_i) e^i) < 0$ by definition of η_i. Consequently $x \ge p'$. Similarly, $x \le q'$.

So we get

$$p'_i = \begin{cases} p_i & \text{if } \ g(q - (q_i - p_i) e^i) \ge 0, \\ \max\limits_{j \in J_1} \{a_{ij} | q + (a_{ij} - q_i) e^i \in [0, a^j]\} & \text{otherwise.} \end{cases}$$

$$q'_i = \begin{cases} q_i & \text{if } \ f(p' + (q_i - p'_i) e^i) \le \gamma, \\ p'_i + \gamma - f(p') & \text{otherwise.} \end{cases}$$

III. Bounding

This is the operation of computing a lower bound and an upper bound of $f(x)$ over the box $B = [p, q]$. Of course, this box is supposed to have been reduced by the above defined procedure, so p, q must be in S.

Clearly if $g(q) < 0$ then no point $x \in [p, q]$ is feasible and $\min\{f(x) | \ x \in [p, q] = +\infty$; if $g(p) \ge 0$ then every $x \in [p, q]$ is feasible and $\min\{f(x) | \ x \in [p, q]\} = f(p)$. If $f(p) > \gamma$ then $f(x) > \gamma \ \forall x \in [p, q]$. Barring these trivial cases we can state

Proposition 7 *Assume that $g(q) \ge 0 > g(p)$ and $f(p) \le \gamma$.*

(i) *Let $x(B)$ be the intersection of the line joining p and q with the surface $g(x) = 0$. Then $x(B) = p + \lambda(q - p)$ where $\lambda = \max\limits_{j \in J_p} \min\limits_{\substack{i=1,\dots,n \\ q_i - p_i \neq 0}} \dfrac{a_{ij} - p_i}{q_i - p_i}$, $J_p = \{j \in J_1 | p \in [0, a^j)\} = \{j \in J_1 | g_j(p) < 0\}$).*

(ii) *Let x^1 be any point in $[p, q]$ satisfying $g(x^1) = 0$. Setting $y^i = p + (x^1_i - p_i) e^i, i = 1, \dots, n$ let $I = \{i | \ \text{red}^S_\gamma [y^i, q] = [\tilde{y}^i, \tilde{q}^i] \neq \emptyset\}$. If $I = \emptyset$ there is no feasible point in $[p, q]$ satisfying $f(x) < \gamma$. Otherwise, a lower bound of $f(x)$ over B is given by*

$$\beta(B) = \min_{i \in I} f(\tilde{y}^i) \tag{13}$$

(iii) *Let* $\tilde{x}^1 := \lfloor x^1 \rfloor_S$ *be the lower S-adjustment of* x^1 *in* $[p,q]$, *i.e.* $\tilde{x}^1_i = \max\{y_i | y_i \in \{p_i, a, a_{ij}, j \in J_1\}, y_i \leq x^1_i\}, i = 1,\dots,n$. *Then* $\tilde{x}^1 \in [p,q] \cap S$ *and* $g(\tilde{x}^1) = 0$.

(iv) *Let* $I_1 = \{i | \tilde{x}^1_i > 0\}$ *and* $w_i = \max\{y_i | y_i \in \{a, a_{ij}, j \in J_1\}, y_i < \tilde{x}^1_i\}, i \in I_1$. *We shall refer to each* $\bar{z}^i = \tilde{x}^1 + (w_i - \tilde{x}^1_i)e^i, i \in I_1$, *as the* i^{th} point around $\tilde{x}^1$. *If* $g(\bar{z}^i) < 0\ \forall i \in I_1$ *then* $\tilde{x}^1$ *is a proper vertex of* H, *i.e.,* $\tilde{x}^1 \in S$.

Proof (i) It is easily checked that λ is a solution of the equation $g(p + \lambda(q-p) = 0$.

(ii) It suffices to show that every feasible point y in the box B belongs to some box $[y^i, q], i \in \{1,\dots,n\}$. Indeed, otherwise for any $i = 1,\dots,n$ one would have $y_j < y^i_j\ \ \forall j = 1\dots,n$, hence $y_i < y^i_i = x^1_i$ and then

$$g(y) = \min_{j \in J_1} \max_{i=1,\dots,n} (y_i - a_{ij}) < \min_{j \in J_1} \max_{i=1,\dots,n} (x(B)_i - a_{ij}) = g(x(B)) = 0,$$

contradicting the feasibility of y. Thus $\{x \in B | g(x) \geq 0\} \subset \cup_{i=1,\dots,n}[y^i, q]$, whence,

$$\{x \in B \cap S | g(x) \geq 0, f(x) \leq \gamma\} \subset \cup_{i=1,\dots,n} \text{red}^S_\gamma [y^i, q] = \cup_{i \in \bar{I}} [\tilde{y}^i, \tilde{q}^i].$$

Consequently, $\beta(B) = \min_{i \in I} f(\tilde{y}^i)$ is a lower bound of $f(x)$ over B if $I \neq \emptyset$ and

$$\{x \in B \cap S | g(x) \geq 0, f(x) \leq \gamma\} = \emptyset$$

if $I = \emptyset$.

(iii) If $g(\tilde{x}^1) < 0$ there is an index $j_1 \in J_1$ such that $g_{j_1}(\tilde{x}^1) = \max_{i=1,\dots,n}(\tilde{x}^1_i - a_{ij_1}) < 0$, i.e., $\tilde{x}^1_i < a_{ij_1}\ \forall i = 1,\dots,n$. Since $g(x^1) = 0$, we have $g_{j_1}(x^1) \geq 0$. So $\max_{i=1,\dots,n}(x^1_i - a_{ij_1}) \geq 0$, therefore there exists $i_1 \in I$ such that $x^1_{i_1} \geq a_{i_1 j_1}$. On the other hand, from the definition $\tilde{x}^1_i := \max\{y_i | y_i \in \{0, a_{ij}, j \in J_1\}, y_i \leq x^1_i\}, i = 1,\dots,n$, it follows that $\tilde{x}^1_{i_1} \geq a_{i_1 j_1}$, conflicting with $\tilde{x}^1_i < a_{ij_1}$ for all $i = 1,\dots,n$. Thus $0 \leq g(\tilde{x}(B)) \leq g(x^1) = 0$, hence $g(\tilde{x}^1) = 0$.

(iv) From (iii), because $g(\tilde{x}^1) = 0$ we have $\tilde{x}^1 \in H = \cup_{z \in S}[z, b]$. Consequently, there is an $z^1 \in S$ such that $\tilde{x}^1 \in [z^1, b]$. This implies $z^1 \leq \tilde{x}^1$. For $i = 1,\dots,n \setminus I_1, 0 \leq z^1_i \leq \tilde{x}^1_i = 0$ clearly $z^1_i = \tilde{x}^1_i = 0$. Now assume there exists $i_1 \in I_1$ satisfying $z^1_{i_1} < \tilde{x}^1_{i_1}$. Because $z^1 \in S$ there exists $j_1 \in J_1$ such that $z^1_{i_1} = a_{i_1 j_1} \leq w_{i_1}$ by the definition of w_{i_1}. We have $\bar{z}^{i_1}$, $\bar{z}^{i_1}_i = \tilde{x}^1_i \geq z^1_i\ \forall i \neq i_1$ and $\bar{z}^{i_1}_{i_1} = w_{i_1} \geq z^1_{i_1}$. So $g(\bar{z}^{i_1}) \geq g(z^1) = 0$, a contradicition. Therefore, $\tilde{x}^1 = z^1 \in S$. □

Remark 1 (i) Based on Proposition 7 (iv), starting from $\tilde{x}^1 \in S \cap [p,q]$ satisfying $g(\tilde{x}^1) = 0$ we can find a better feasible point $z \in [p,q] \cap S$ as follows.

– Step 1: We first construct the function $[l, y] = arround(\tilde{x}^1, p, q)$ defined by

  $l := 0$;
  for $i = 1 : n$ *do*
   if $\tilde{x}^1_i > 0$ *then compute* $\bar{z}^i$ *as "the* i^{th} *arround point of* $\tilde{x}^1$*";*
    if $g(\bar{z}^i) = 0$ *and* $\bar{z}^i \geq p$ *then*
        $y := \bar{z}^i$; $l := 1$;
    end if
   end if
    if $l = 1$ *then exit the "for" loop;*
  end for

  The value of l shows the albility of improvement of $\tilde{x}^1$. If $l = 1$, i.e we can substitute the first feasible *point arround* $\tilde{x}^1$ belonging to $[p,q]$ from the index 1, $\bar{z}^i$ (i.e., the first index $i \in I$ from 1 satisfying $p \leq \bar{z}^i$, $g(\bar{z}^i) = 0$) for $\tilde{x}^1$ and it is obvious that $\bar{z}^i$ is really better than $\tilde{x}^1$, i.e., $f(\bar{z}^i) < f(x^1)$. After that we exit the loop "for" and stop considering the indices after i. Otherwise, $l = 0$, i.e., any *arround point* of $\tilde{x}^1$ belonging to $[p,q]$ is not feasible and we can not improve it to a better point in $[p,q] \cap S$. As a result, when $[p,q] = red^S_{\gamma_0}[\mathbf{0}, b]$ (γ_0 is the first CBV) by using Lemma 7 (iv) we obtain $\tilde{x}^1$ to be a proper vertex if $l = 0$.

- Step 2: Next, the following function leads us to z
  $z = pro - ver(\tilde{x}^1, p, q)$ (we mean "**pro**per **ver**tex")
  $[l, z] = arround(\tilde{x}^1, p, q);$
  while $l > 0$ *do*
  $[l, z] = arround(z, p, q);$
  end while

  With the function $pro - ver(\tilde{x}^1, p, q)$ we move $\tilde{x}^1$ to $z \in [p, q] \cap S$ by carrying out the function *arround* successively until $l = 0$. The obtained point z obviously satisfies: there doesn't exist any *arround point* of z in $[p, q]$ so that it is a feasible point. Analogously, if $[p, q] = red^S_{\gamma_0}[\mathbf{0}, b]$ (γ_0 is the first CBV) then we get a proper vertex z from $\tilde{x}^1$ after using the above procedures.

(ii) Besides, if we find a proper vertex z by using the mentioned procedures with $[p, q] = red^S_{\gamma_0}[\mathbf{0}, b]$ we can adjust it to get a better feasible point z' as follows:
- Step 1: Firstly, if $z_{i_1} > 0$ for some $i_1 \in I$, choose an index $j_1 \in J_1$ satisfying $a_{i_1 j_1} = z_{i_1}$. Let $i_2 \in I$ such that $a_{i_2 j_1} = \min_{i \in I} a_{i j_1}$. If $a_{i_2 j_1} < a_{i_1 j_2}$ then go to Step 2.
- Step 2: Define z' by $z'_i = z_i \ \forall i \in I \setminus \{i_1, i_2\}, z'_{i_1} = 0, z'_{i_2} = a_{i_2 j_1}$. If $g(z') \geq 0$ then z' is a feasible point and better than z in sense that $f(z') < f(z)$.

In fact, in some few cases such z' might not be proper vertex. So we should continue using the function $pro - ver(z', p, q)$ to get a better new proper vertex.

Implementation: From Proposition 7 and Remark 1 we see that: to find a lower bound and an upper bound of a box B we first need finding a point $x^1 \in B$ satisfying $g(x^1) = 0$ and then find a lower bound by applying Lemma 7 (ii) and an upper bound by using Remark 1 for $\tilde{x}^1 = \lfloor x^1 \rfloor_S$. The first candidate for x^1 is $x(B)$ obtained from Lemma 7 (i). We may find an other point x^2 in B to increase the quality of lower bound and upper bound by using the following way. The main idea of this way is to find a vertex u of C belonging to B, then determine x^2 as the intersection of the line segment $\{p + \alpha(u - p), 0 \leq \alpha \leq 1\}$ and the facet $g(x) = 0$. We consider two cases:

- Case 1: If $T_j^{[p,q]} = \{v^{kj} = a_{kj} e^k, k \in I | p_k \leq a_{kj} \leq q_k\} \neq \emptyset$ for all $j \in J_1$ then we take $u^j = \text{argmin}\{||z||, z \in T_j^{[p,q]}\}, j \in J_1$ and $u = \vee_{j \in J_1} u^j \in [p, q]$ is a vertex of C. We next compute x^2 by using Lemma 7 (i) to find the intersection of $\{p + \alpha(u - p), 0 \leq \alpha \leq 1\}$ and the facet $g(x) = 0$.
- Case 2: If the set $J_2 = \{j \in J_1 | T_j^{[p,q]} = \emptyset\} \neq \emptyset$ then we take $u^j = \text{argmin}\{||z||, z \in T_j^{[p,q]}\}, j \in J_1 \setminus J_2$ and $u = \vee_{j \in J_1 \setminus J_2} u^j \in [p, q]$. Because we miss the indices in J_2 hence u may not feasible solution, that why we need to check the condition $g(u) \geq 0$. If this is true then we can use u, otherwise we can't use it to find x^2.

If we can find x^2 from the above way, we get the lower of B by getting the maximum of two lower bounds taking from x^1 and x^2, converserly, the upper bound of B obtained by choosing the minimum of two upper bounds taking from x^1 and x^2.

Finally, we give the local algorithm (LA) and global algorithm (GA) (or BRB algorithm) for solving problem (**??**). As a matter of fact, the local algorithm is process of finding a proper vertex of Q, or carrying out procedure of searching an upper bound discribed in Remark 1 with the box $\text{red}^S_{\gamma_0}[\mathbf{0}, b]$.

Local algorithm (LA) for problem (7)

If $g(b) < 0$ then problem (7) is infeasible. Otherwise, $g(b) \geq 0$ then $\gamma_0 := g(b)$ as the first CBV of our problem. Apply process of finding a feasible solution for the box $\text{red}^S_{\gamma_0}[\mathbf{0}, b]$. As said before, this feasible point is a proper vertex of Q, in other words, it is a local solution of problem (**??**) by Proposition 3.

Global algorithm BRB (GA) for problem (7)

Initialization. If $g(b) < 0$ then problem (7) is infeasible. Otherwise, $g(b) \geq 0$ set $\gamma = \gamma_0 := g(b)$ as the first CBV of our problem. Let $\mathcal{P}_1 := \{B_1\}$, $B_1 = [\mathbf{0}, b]$, $\mathcal{R}_1 = \emptyset$. Set $k := 1, lb := 0, x := b$.

Step 1. For each box $B \in \mathcal{P}_k$ we consider three cases:

- Case 1: If $g(q) < 0$ or $f(p) > \gamma$ then delete B.

- Case 2: If $g(q) \geq g(p) \geq 0$ and $f(p) \leq \gamma$ then updating the new feasible solution $x = p$ and new $CBV = f(p)$ and delete B.

- Case 3: If $g(q) \geq 0 > g(p)$ and $f(p) \leq \gamma$ then reducing $[p, q]$ to be $[\tilde{p}, \tilde{q}]$ by reduction procedure.

Let $\mathcal{P}'_k$ be the resulting collection of reduced boxes.

Step 2. Apply bounding procedure for each box B of $\mathcal{P}'_k$ to have $\beta(B)$. Updating the new $CBV \leq \gamma$ and the new feasible solution corresponding.

Step 3. Reset $\gamma := CBV$ and set $\mathcal{R}_{k+1} = \{B \in \mathcal{R}_k \cup \mathcal{P}'_k | \beta(B) \leq \gamma\}$, $lb := \min\{\beta(B) | B \in \mathcal{R}_{k+1}\}$.

Step 4. If $\mathcal{R}_{k+1} \neq \emptyset$ then choosing $B_k \in$ argmin $\{\beta(B) | B \in \mathcal{R}_{k+1}\}$. Divide B_k into two boxes B_{k_1} and B_{k_2} according to branching procedure. Let $\mathcal{P}_{k+1} = \{B_{k_1}, B_{k_2}\}$. Increment k and return to *Step 1.*

Step 5. If $\mathcal{R}_{k+1} = \emptyset$ then Stop and x is an optimal solution of our problem with optimal value γ.

Theorem 1 *GA for problem (7) terminates after finitely many iterations, yielding an globally optimal solution of the problem.*

Proof It follows from the fact that the set S is finite, therefore since $S-$adjustment reduction, the total number of nodes of the branch and bound tree is finite, which implies finiteness of the algorithm itself. □

5 Computational results

In [2] the authors tested on generated randomly problems satisfying some conditions:

- the sensor and target nodes are randomly located in a 100×100 area;

- 5 different values of n, the number of sensors, have been considered: $25, 75, 125, 175, 225$;

- the number of targets m is calculated by considering the two cases of "Nondense" and "Dense" areas. In the former case $n = 5m$, while in the latter $m = 2n$;

- the values of the maximum and minimum sensing radius are respectively $r_i^{max} = u_i = 30$ and $r_i^{min} = l_i = 0$.

- the values of α_i, β_i are: $\alpha_i = 1, \beta_i = 2$ for all $i \in I$.

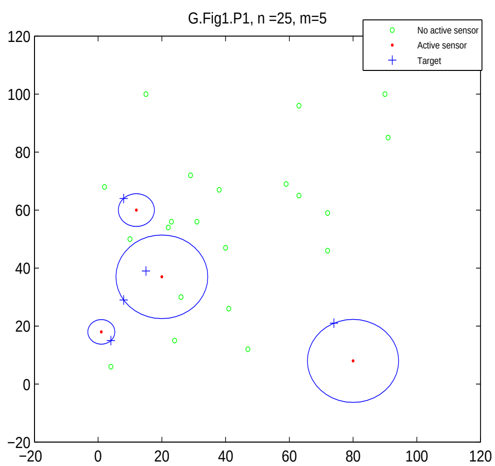
However, their work only give a local solution based on DCA. In this paper, we test our local algorithm (LA) for not only all instances randomly generated similar to [2] but also other ones in larger dimension (with up to 1000). The results are shown in Table 5 and inlustrated by figures from L.Fig1 to L.Fig12. We see that LA is efficient even for large scales. Especially, in "Nondense" cases LA give a local solution in a small time.

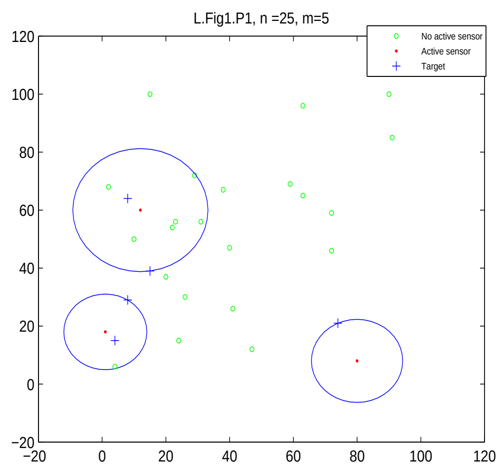
For comparision purposes we test global algorithm BRB (GA) and local algorithm (LA) in three cases: $n = 25, m = 5; n = 25, m = 50; n = 75, m = 15$, where each case includes five problems randomly generated. To reduce running times for GA we add a terminating condition that is $re < \varepsilon$, where $re := \frac{\gamma - lb}{\gamma}$ is relative error and $\varepsilon = 0.05$. The results are shown in Table 1, 2, 3 and attached figures. Bold numbers mean that the optimal value of GA is really better than optimal value of LA although GA is certainly more expensive than LA. We inlustrate these cases by corresponding pairs of figures. Some other ones with italic numbers indicate that GA and LA provide the same optimal values. We have such two problems P_2, P_3 in case $n = 25, m = 5$ and P_2, P_5 in case $n = 75, m = 15$. These are belong to "Nondense" cases. Is this true that in these cases LA sometimes gives us a global optimal solution? In Table 2 we obtain the infeasibility of P_1, P_2 immediately by checking whether $g(b)$ should be negative.

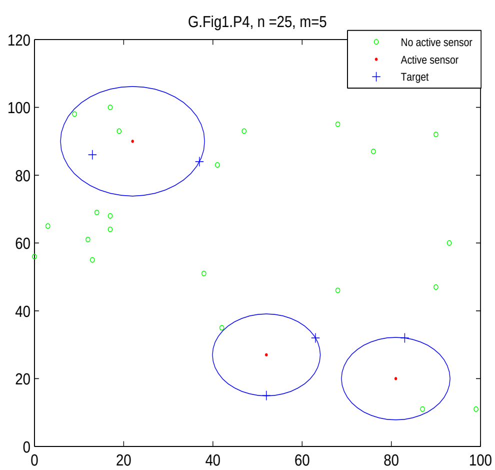
We coded our algorithms in Matlab R2012a, Win7-64 bit and Laptop Dell (RAM 8G, Intel core i7 2.26 GHz).

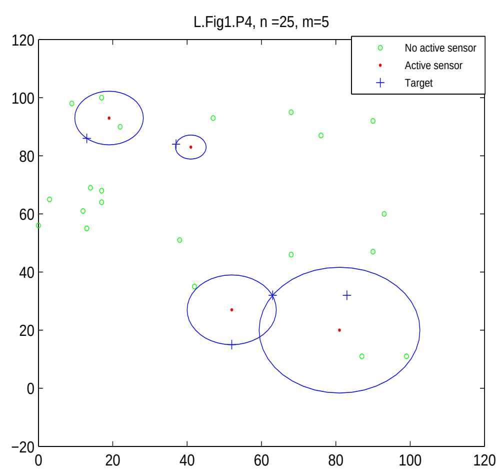
Table 1 $n = 25, m = 5$

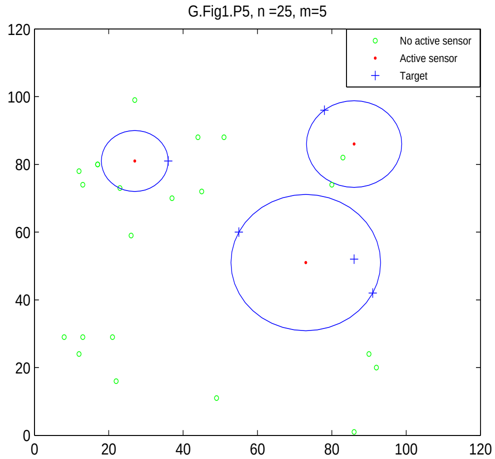
Prob	$\|J_1\|$	GA				LA		
		Opt.val	#Iter	Time(s)	Fig.	Opt.val	Time(s)	Fig.
P_1	5	**462.9994**	11	0.4992	G.Fig1.P1	824.9991	0.0312	L.Fig1.P1
P_2	5	*842.9992*	11	0.3276	-	*842.9992*	0.0156	-
P_3	5	*468.9988*	8	0.2028	-	*468.9988*	0.0312	-
P_4	5	**554.9985**	6	0.2028	G.Fig1.P4	713.9988	0.0000	L.Fig1.P4
P_5	5	**649.9983**	12	0.2808	G.Fig1.P5	903.0002	0.0312	L.Fig1.P5

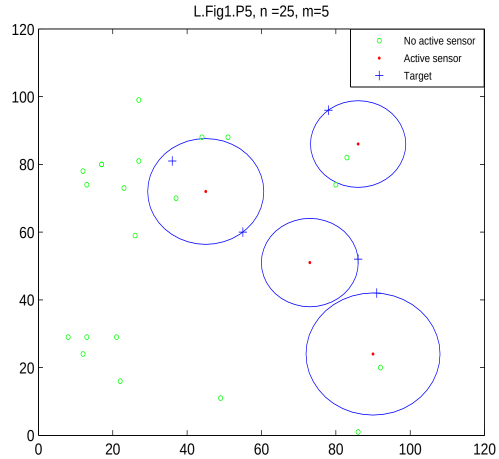


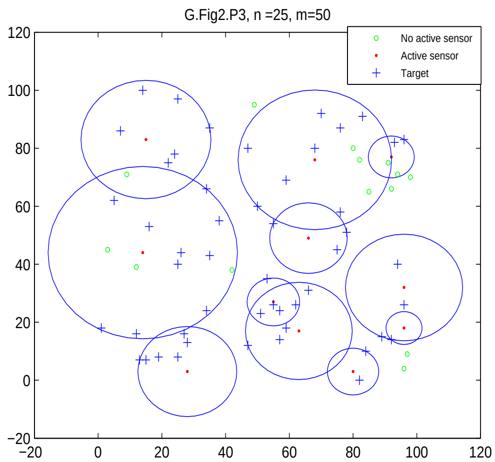


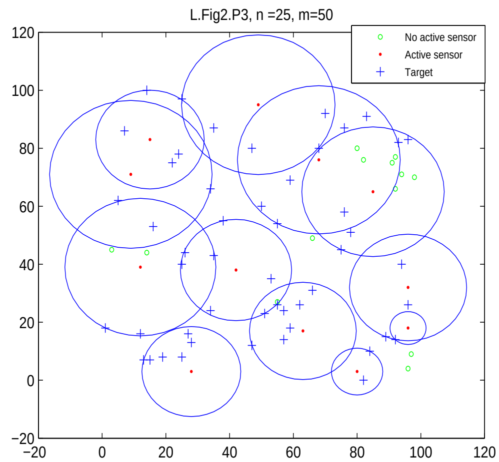


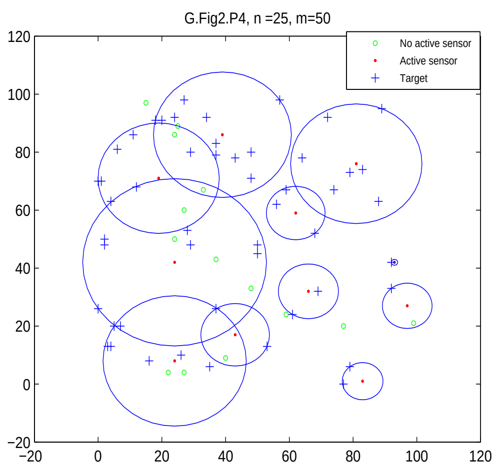


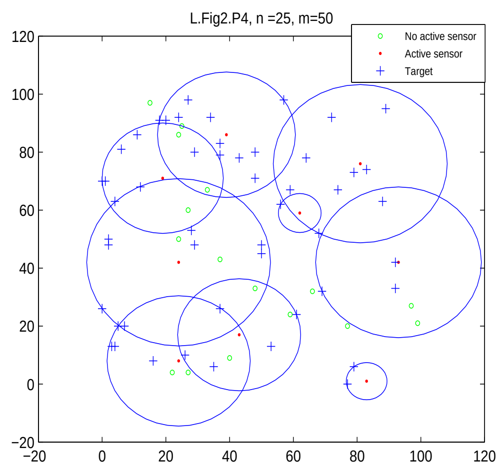

G.Fig1.P5, n =25, m=5
No active sensor
Active sensor
Target

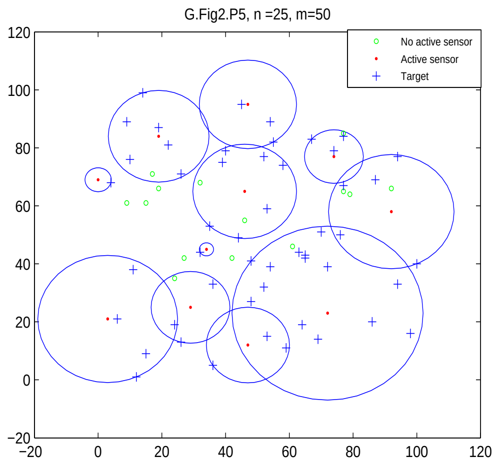

L.Fig1.P5, n =25, m=5
No active sensor
Active sensor
Target

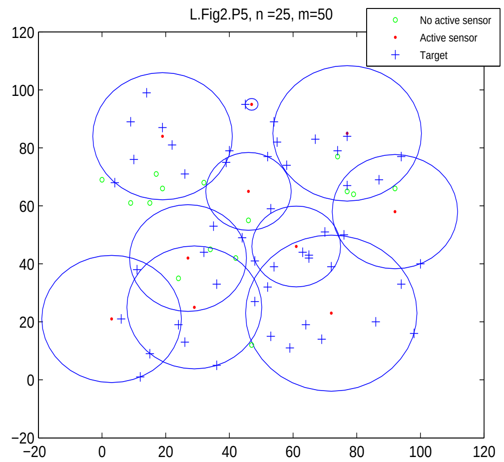

G.Fig2.P3, n =25, m=50
No active sensor
Active sensor
Target

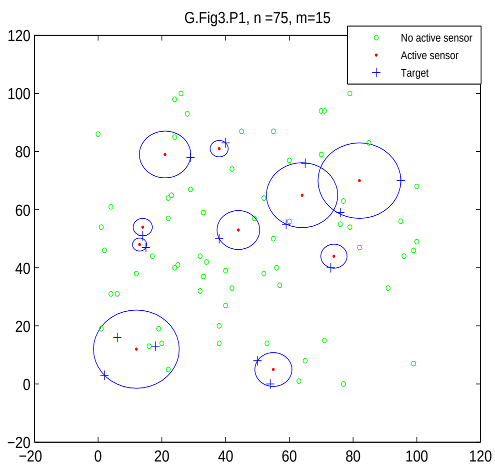

L.Fig2.P3, n =25, m=50
No active sensor
Active sensor
Target

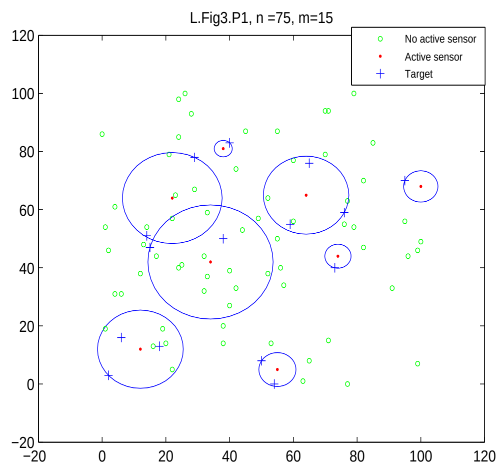

G.Fig2.P4, n =25, m=50
No active sensor
Active sensor
Target

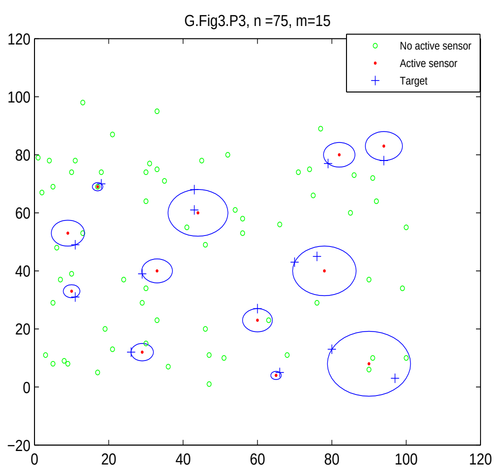

L.Fig2.P4, n =25, m=50
No active sensor
Active sensor
Target


G.Fig2.P5, n =25, m=50
No active sensor
Active sensor
Target


L.Fig2.P5, n =25, m=50
No active sensor
Active sensor
Target


G.Fig3.P1, n =75, m=15
No active sensor
Active sensor
Target


L.Fig3.P1, n =75, m=15
No active sensor
Active sensor
Target


G.Fig3.P3, n =75, m=15
No active sensor
Active sensor
Target

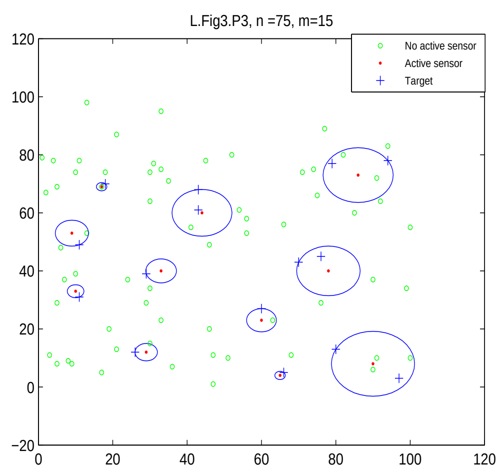
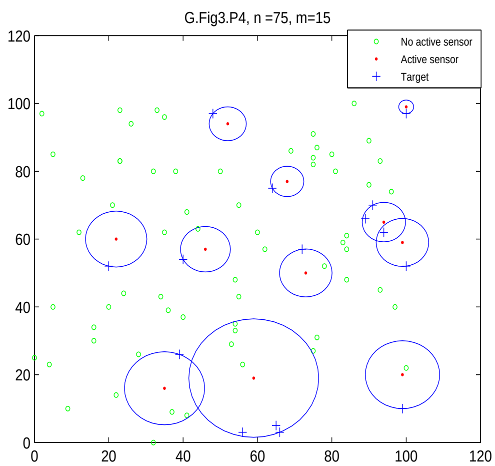

L.Fig3.P3, n =75, m=15
No active sensor
Active sensor
Target

Table 2 $n = 25, m = 50$.

Prob	$\|J_1\|$	GA				LA		
		Opt.val	#Iter	Time(s)	Fig.	Opt.val	Time(s)	Fig.
P_1	39	infeasible	-	-	-	infeasible	-	-
P_2	47	infeasible	-	-	-	infeasible	-	-
P_3	46	**3105.0027**	843	121.7744	G.Fig2.P3	4497.0046	0.1248	L.Fig2.P3
P_4	40	**2984.9956**	256	35.5838	G.Fig2.P4	4046.9989	0.1248	L.Fig2.P4
P_5	38	**2946.9961**	1196	247.8856	G.Fig2.P5	3784.9955	0.1716	L.Fig2.P5

Table 3 $n = 75, m = 15$.

Prob	$\|J_1\|$	GA				LA		
		Opt.val	#Iter	Time(s)	Fig.	Opt.val	Time(s)	Fig.
P_1	14	**657.9996**	655	195.8749	G.Fig3.P1	1080.0013	0.0468	L.Fig3.P1
P_2	15	*499.0025*	2863	954.5857	-	*499.0025*	0.0468	-
P_3	15	**376.9991**	76	36.3950	G.Fig3.P3	422.9998	0.0468	L.Fig3.P3
P_4	15	**816.9986**	816	337.3210	G.Fig3.P4	1461.9977	0.0780	L.Fig3.P4
P_5	15	*637.9994*	106	38.7506	-	*637.9994*	0.0624	-



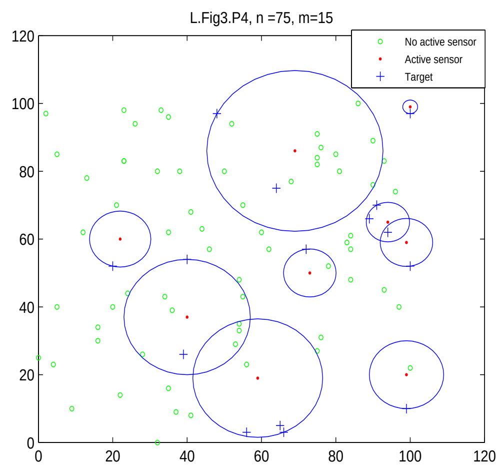
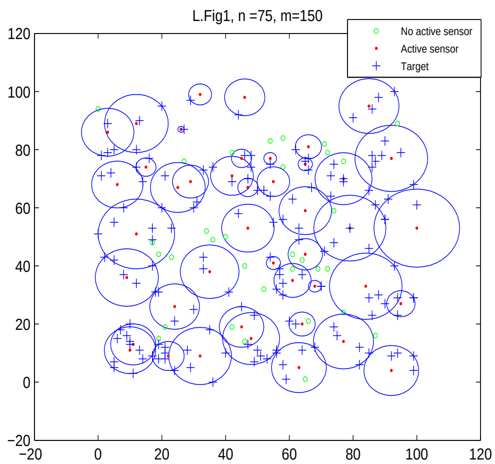
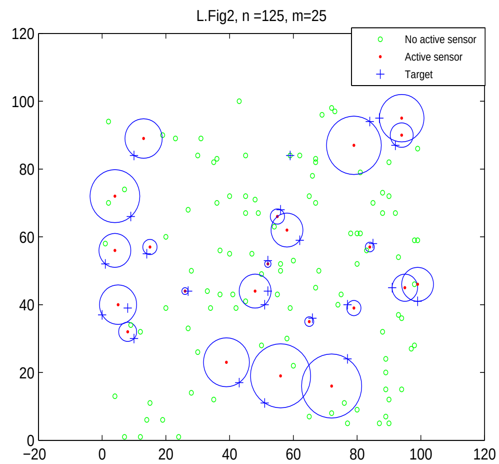
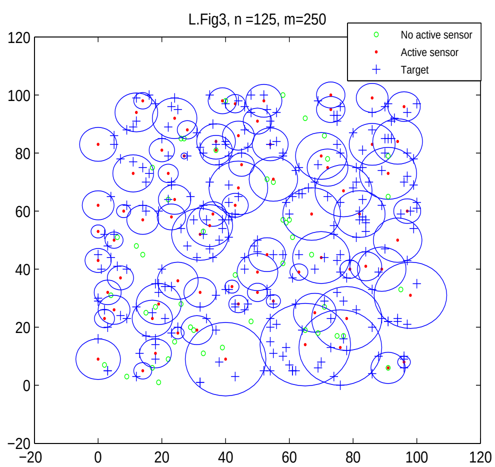


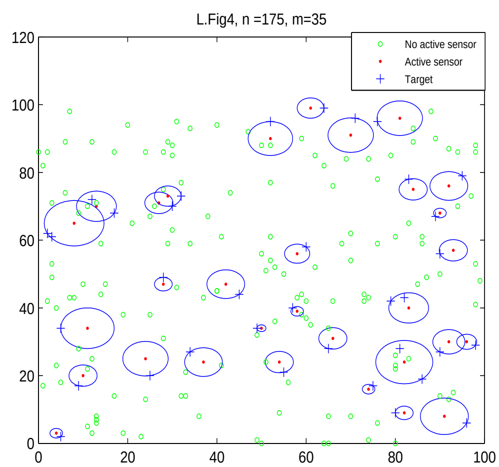
Table 4 Numerical results for local algorithm (LA)

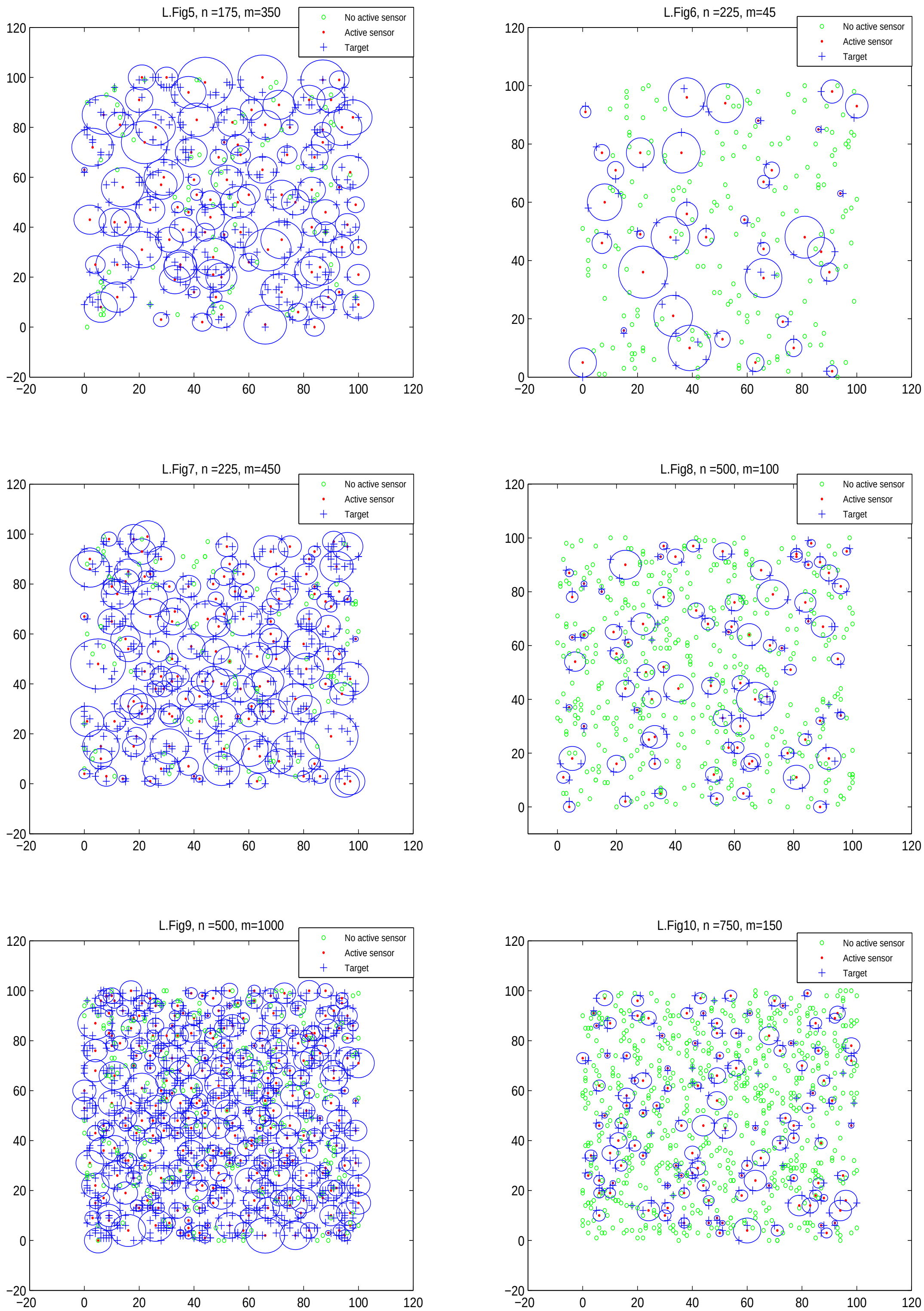
LA					
n	m	$\|J_1\|$	Opt.val	Time(s)	Fig.
75	150	137	2494.0047	1.5756	L.Fig1
125	25	25	640.0011	0.0780	L.Fig2
125	250	243	3058.0048	6.1308	L.Fig3
175	35	35	468.9988	0.1248	L.Fig4
175	350	343	2857.0123	21.6373	L.Fig5
225	45	45	756.0011	0.3120	L.Fig6
225	450	436	2981.0108	48.0015	L.Fig7
500	100	100	615.0027	1.7472	L.Fig8
500	1000	953	2924.0077	388.0681	L.Fig9
750	150	149	574.0034	5.9592	L.Fig10
750	1000	951	2316.0118	356.5871	L.Fig11
1000	500	492	1232.0070	123.8804	L.Fig12

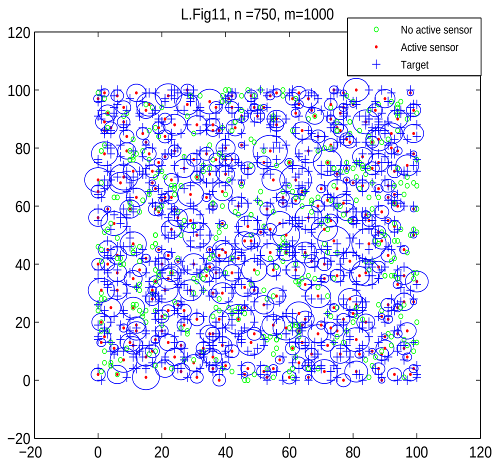


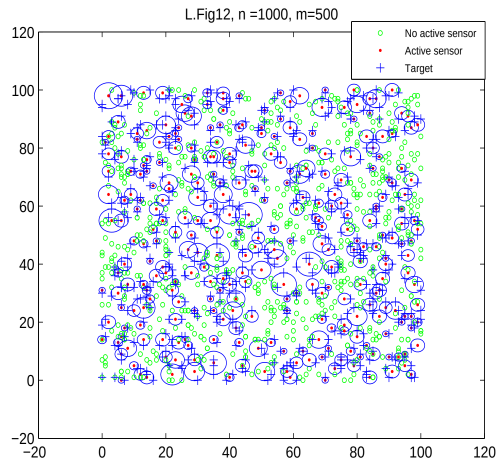







L.Fig5, n =175, m=350
L.Fig6, n =225, m=45
L.Fig7, n =225, m=450
L.Fig8, n =500, m=100
L.Fig9, n =500, m=1000
L.Fig10, n =750, m=150
No active sensor
Active sensor
Target





6 Conclusion

We propose new solution methods for solving (SCEP) by using theory of monotonic optimization. We determine the conditions so that our problem is feasible and give the set of local solutions in this case. Global and local algorithms are proposed and verified throughout many problems with up to 75 variables for first one and 1000 variables for second one. The numerical results show the efficience of these algorithms. Naturally, this paper provides a solution for general problem of minimizing an increasing function over the intersection of finitely given copolyblocks in a box.

Acknowledgements The first author would like to thank Dr. Ta Anh Son for sending her some useful material on sensor cover energy problem.

References

1. Alfieri, A., Bianco, A., Brandimarte, P., Chiasserini, C.F.: Maximizing system lifetime in wireless sensor networks. Eur. J. Operat. Res. 181, 390-402 (2007)
2. Astorino, A., Miglionico, G.: Optimizing sensor cover energy via DC programming. Optim. Lett. DOI 10.1007/s11590-014-0778-y (2014)
3. Bartolini, N., Calamoneri, T., La Porta, T., Petrioli, C., Silvestri, S.: Sensor activation and radius adaptation (SARA) in heterogeneous sensor networks. ACM Trans. Sensor Netw. (TOSN) 8, 24 (2012)
4. Cardei, M., Mingming, L., Pervaiz, M.O.: Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In: Proceedings of the IEEE international conference on wireless and mobile computing, networking and communications (WiMob) 3, 438-445 (2005)
5. Cardei, M., Du, D.: Improving wireless sensor network lifetime through poweraware organization. Wirel. Netw. 11, 333-340 (2005)
6. Cardei, M., Wu, J.: Energy-efficient coverage problems in wireless ad-hoc sensor networks. Comput. Commun. 29, 413-420 (2006)
7. Cerulli, R., De Donato, R., Raiconi, A.: Exact and heuristic methods to maximize network lifetime in wireless sensor networks with adjustable sensing ranges. Eur. J. Operat. Res. 220, 58-66 (2012)
8. Mulligan,R.:Coverage in wireless sensor networks: a survey. Netw. Protoc. Algorithms. 2, 27-53 (2010)
9. Pattem, S., Poduri, S., Krishnamachari, B., Energy-quality tradeoffs for target tracking in wireless sensor networks. Proc. of ACM International Conference on Information Processing in Sensor Networks (IPSN) (2003)
10. Tuy, H.: Monotonic optimization: Problems and solution approaches, SIAM J. Optim. 11, 464-494, (2000)
11. Tuy, H., Minoux, M., Phuong, N.T.H.: Discrete monotonic optimization with application to a discrete location problem, SIAM J. Optim. 17, 78-97 (2006)
12. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw. 52, 2292-2330 (2008)
13. Zhou, Z., Da,s S., Gupta, H., Variable radii connected sensor cover in sensor networks. ACM Transactions on Sensor Networks. 5, 8-36 (2009)