Deterministic and stochastic global optimization techniques for planar covering with ellipses problems*

M. Andretta † E. G. Birgin ‡

November 28, 2011

Abstract

Problems of planar covering with ellipses are tackled in this work. Ellipses can have a fixed angle or each of them can be freely rotated. Deterministic global optimization methods are developed for both cases, while a stochastic version of the method is also proposed for large instances of the latter case. Numerical results show the effectiveness and efficiency of the proposed methods.

Key words: Planar covering with ellipses, deterministic global optimization, algorithms.

1 Introduction

In this work, a covering problem on the plane is considered. A finite set of demand points is given and the problem consists on covering the most valuable subset of demand points using at most k or exactly k ellipses from a set of m given ellipses. This problem was tackled in [11], where a detailed bibliographical review is given and the application to the optimization of wireless transmitters coverage is highlighted. The problem has many applications in the location of facilities and geographical systems. The case in which the distance between demand points is a fuzzy variable was approached in [12]. In [11] the authors present a MINLP formulation of the problem and show with numerical experiments that available solvers are not able to deliver a solution to small or medium-size instances within a reasonable amount of CPU time. More specifically, the problem is modeled in GAMS and several instances submitted to be solved by SBB [13] and BARON [14] trough the NEOS server [15]. Motivated by this fact, an heuristic approach based on Simulated Annealing is introduced in [11]. In the problem considered in [11],

*This work was supported by PRONEX-CNPq/FAPERJ E-26/111.449/2010-APQ1, CNPq (304484/2007-5) and FAPESP (2006/53768-0, 2009/10241-0, 2010/10133-0, and 2010/18980-3).

†Department of Applied Mathematics and Statistics, Institute of Mathematical and Computer Sciences, University of São Paulo, Avenida Trabalhador São-carlense, 400, Centro, 13566-590, São Carlos, SP, Brazil. e-mail: andretta@icmc.usp.br

‡Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. e-mail: egbirgin@ime.usp.br

ellipses are restricted to have their axes parallel to the Cartesian axes, while it is pointed out that allowing the ellipses to rotate would be the subject for future research.

A deterministic global optimization technique capable of finding a global solution to the instances whose data is available in [11] is introduced. Numerical results confirm that the heuristic approach presented in [11] also finds a global solution to those instances. The proposed method associates a clever semi-enumerative approach with nonlinear programing (NLP) subproblems to compute all maximal coverings using the ellipses individually. This information is later analysed and combined to compute an optimal solution for the original problem. When ellipses' axes are restricted to be parallel to the Cartesian axes, NLP subproblems are continuous, smooth, and convex, and, hence, their global solutions are easy to compute using any local NLP solver. The problem's extension to consider individual angles of rotation for each ellipse is also presented. In this case, the NLP subproblems are non-convex. The αBB [1, 2, 3, 7] method is considered to solve the subproblems and, due to the subproblems' complexity, only small instances can be solved to optimality within a tolerable limit of CPU time. Therefore, a stochastic global optimization approach is used to solve the non-convex NLP suproblems and solutions (not necessarily optimal) to a large set of considered instances are reported.

The paper is organized as follows. Section 2 presents the MINLP models for the considered problems. The introduced methods are fully described in Section 3. Numerical experiments are described in Section 4. Conclusions and lines for future research are stated in Section 5.

2 Problem description

The problem consists of n fixed points $p_i = (p_i^x, p_i^y)^T \in \mathbb{R}^2$ with profits $\widehat{w}_i$, $i = 1, \ldots, n$, that should be covered by a set of ellipses. The ellipses chosen to cover the points must belong to a given set of m ellipses with semi major and semi minor axes a_j and b_j, respectively, and costs $\widetilde{w}_j$, $j = 1, \ldots, m$. Let $S \subseteq \{1, \ldots, m\}$ be the set of indices of the chosen ellipses. Different versions of the problem exist depending on whether, for a given constant $k \leq m$, constraint $|S| = k$ or $|S| \leq k$ is considered. Moreover, another source of alternative problems lies on whether ellipses are restricted to be positioned with their axes parallel to the Cartesian axes, or they can be freely rotated. In any case, the objective is to maximize the income (profit of covered points minus cost of allocated ellipses).

A mixed integer nonlinear programming (MINLP) formulation of the problem follows. In the model below, the number of allocated ellipses is restricted to be equal to the given non-negative integer constant k, while the ellipses are restricted to be with their axes parallel to the Cartesian axes. This version of the problem will be called P1 from now on. Variables of the formulation are: $x_{ij} \in \{0, 1\}, i = 1, \ldots, n, j = 1, \ldots, m$, $y_j \in \{0, 1\}, j = 1, \ldots, m$, and $c_j = (c_j^x, c_j^y)^T \in \mathbb{R}^2, j = 1, \ldots, m$. In the model, $x_{ij} = 1$ if point p_i is assigned to be covered by ellipse j, $x_{ij} = 0$ otherwise; $y_j = 1$ if ellipse j was selected to be used, $y_j = 0$ otherwise; and, when $y_j = 1$, c_j stands for the position on the plane of ellipse j.

$$\text{Maximize} \sum_{j=1}^{m}\sum_{i=1}^{p} \widehat{w}_i x_{ij} - \sum_{j=1}^{m} \widetilde{w}_j y_j \tag{1}$$

$$\text{subject to} \sum_{j=1}^{m} y_j = k, \tag{2}$$

$$\sum_{j=1}^{m} x_{ij} \leq 1, \qquad i = 1, \ldots, n, \tag{3}$$

$$x_{ij} \leq y_j, \qquad i = 1, \ldots, n,\ j = 1, \ldots, m, \tag{4}$$

$$(c_j^x - p_i^x)^2/a_j^2 + (c_j^y - p_i^y)^2/b_j^2 \leq 1 + (1 - x_{ij})M, \quad i = 1, \ldots, n,\ j = 1, \ldots, m. \tag{5}$$

In the formulation above, the objective function (1) represents the covering income, i.e. the profit of the covered points minus the cost of the allocated ellipses. Constraint (2) determines the number of ellipses to be used. Constraint (3) says that a point can be covered by no more than one ellipse. More precisely, this constraint allows the profit of covering a point by an ellipse to be considered at most once. Constraint (4) says that points can only be attributed to be covered by the selected ellipses. Constraint (5) determines that a point covered by an ellipse must be physically located inside or in the border of the ellipse. In this constraint, M is a sufficiently large positive number.

If constraint (2) is substituted by

$$\sum_{j=1}^{m} y_j \leq k, \tag{6}$$

we obtain another version of the problem that will be called P2 from now on. Note that imposing an upper bound on the number of selected ellipses, instead of fixing the number of ellipses that must be used, may be profitable in the particular situation of having ellipses more costly than the profit of every possible set of points covered by them.

If, in P1 or P2, we substitute constraint (5) by

$$\left\| \begin{pmatrix} 1/a_j & 0 \\ 0 & 1/b_j \end{pmatrix} \begin{pmatrix} \cos\theta_j & -\sin\theta_j \\ \sin\theta_j & \cos\theta_j \end{pmatrix} \begin{pmatrix} c_j^x - p_i^x \\ c_j^y - p_i^y \end{pmatrix} \right\|_2^2 \leq 1+(1-x_{ij})M, i = 1, \ldots, n, j = 1, \ldots, m, \tag{7}$$

we obtain the counter-parts of P1 and P2 in which ellipses can be freely rotated. We call those problems P3 and P4, respectively, from now on. In P3 and P4, the additional variable θ_j represents the counter-clockwise angle of rotation of ellipse j, for $j = 1, \ldots, m$.

Notation. From now on,

$$c = (c_1, c_2, \ldots, c_m) = \begin{pmatrix} c_1^x & c_2^x & \ldots & c_m^x \\ c_1^y & c_2^y & \ldots & c_m^y \end{pmatrix} \in \mathbb{R}^{2\times m},$$

$$x = (x_1, x_2, \ldots, x_m) = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1m} \\ x_{21} & x_{22} & \ldots & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \ldots & x_{pm} \end{pmatrix} \in \{0,1\}^{p \times m},$$

$y \in \{0,1\}^m$, and $\theta \in \mathbb{R}^m$ may be used to denote the multi-dimensional variables of problems P1, P2, P3, and P4.

3 Deterministic global optimization method

We will start by describing a solution method for problem P1. Assume for a moment that values x and y such that (2–4) hold are given. Let

$$J(y) = \{j \mid y_j = 1\} \subseteq \{1, \ldots, m\}$$

and

$$I(x_j) = \{i \mid x_{ij} = 1\} \subseteq \{1, \ldots, p\}.$$

Clearly, $J(y)$ is the set of selected ellipses according to y and, for $j \in J(y)$, $I(x_j)$ is the set of points to be covered by ellipse j according to x_j. Note that the value of the objective function in (1) is already determined by (x, y) and is given by

$$\phi(x, y) = \sum_{j \in J(y)} \sum_{i \in I(x_j)} \widehat{w}_i - \sum_{j \in J(y)} \widetilde{w}_j. \tag{8}$$

The question that arises is whether, given x and y satisfying (2–4), c satisfying (5) exists, i.e. c such that (x, y, c) is a feasible point for problem P1.

Deciding whether such c exists is equivalent to answering, for each $j \in J(y)$, whether $c_j \in \mathbb{R}^2$, a position for the center of ellipse j, exists such that p_i belongs to ellipse j, for all $i \in I(x_j)$. If the answer is positive for all $j \in J(y)$, then c satisfying (5) exists. (For $j \notin J(y)$ we have that $x_{ij} = 0$, $i = 1, \ldots, n$, and any value for c_j satisfies (5) provided that constant M in (5) is large enough.) If the answer is negative for some $j \in J(y)$, then c satisfying (5) does not exist. Therefore, for given x and y, determining whether c such that (x, y, c) satisfies (2–5) exists reduces to solving the $|J(y)|$ independent bivariate feasibility problems of the form

$$(c_j^x - p_i^x)^2/a_j^2 + (c_j^y - p_i^y)^2/b_j^2 \leq 1, \ \forall\, i \in I(x_j), \tag{9}$$

for all $j \in J(y)$, or finding that there is at least one $j \in J(y)$ such that (9) is infeasible. Furthermore, if we define

$$f_{I(x_j)}(c_j) \equiv \sum_{i \in I(x_j)} \max\{0, (c_j^x - p_i^x)^2/a_j^2 + (c_j^y - p_i^y)^2/b_j^2 - 1\}^2,$$

for a given $j \in J(y)$, solving (9), or deciding (9) is infeasible, is equivalent to finding the global solution to the optimization problem

$$\mathrm{Min}_{c_j \in \mathbb{R}^2} f_{I(x_j)}(c_j). \tag{10}$$

Problem (10) is a bivariate, convex, continuous and differentiable unconstrained minimization problem whose global minimizer c_j^* can be found using any unconstrained minimization method devised to find stationary points. If $f_{I(x_j)}(c_j^*) > 0$, then the feasibility problem (9) is infeasible. Otherwise, if $f_{I(x_j)}(c_j^*)$ vanishes, then c_j^* satisfies (9) and it represents a feasible location of ellipse j to cover points p_i for all $i \in I(x_j)$.

Summing up, for a given (x, y) satisfying (2–4), finding c such that (x, y, c) satisfies (2–5), or determining that such c does not exist, is an easy problem. Hence, the proposed deterministic global optimization method to solve problem P1 given by (1–5) basically consists in an efficient semi-enumerative approach that checks every combination (x, y, c) that satisfies (2–5) to find one that minimizes the objective function (1). The efficiency of the proposed approach lies on several procedures described below, built with the intention of reducing the size of the search tree of combinations of (x, y, c).

3.1 Solving problems with a single ellipse

We start the description of the semi-enumerative approach for solving problem P1 by considering the case of a single ellipse, i.e. $k = 1$ and $S = \{\ell\}$ for some $\ell \in \{1, \ldots, m\}$. Therefore, to satisfy (2), we must have $y_\ell = 1$ and $y_j = 0$, for all $j \neq \ell$, while (3) and (4) are automatically satisfied by setting $x_{ij} = 0$ for all $i = 1, \ldots, n$ and for all $j \neq \ell$ and for any choice of $x_{i\ell} \in \{0, 1\}, i = 1, \ldots, n$. The idea consists of potentially considering all possible 2^n choices for $x_\ell \in \{0, 1\}^n$, discarding those that certainly produce sub-optimal or infeasible solutions. The strategy that follows was developed with the intention of being used to solve the problem with multiple ellipses, i.e. $k > 1$. If the original problem has, in fact, $k = 1$, many simplifications can be done, that will be mentioned at the end of Subsection 3.3.

Consider first the case in which it is already known that there exists a position c_ℓ for the center of ellipse ℓ such that it covers points $p_{i_1}, p_{i_2}, \ldots, p_{i_t}$. In this case, it makes no sense to check whether a position exists such that the ellipse covers any proper subset of $\{p_{i_1}, p_{i_2}, \ldots, p_{i_t}\}$, since the answer is clearly true and those proper subsets are associated to sub-optimal solutions. Another example of a combination of variables $x_{i\ell}$ that does not need to be checked is the case in which there are indices i_1 and i_2 such that $x_{i_1,\ell} = x_{i_2,\ell} = 1$ and

$$|p_{i_1}^x - p_{i_2}^x| > 2a_\ell \;\text{ or }\; |p_{i_1}^y - p_{i_2}^y| > 2b_\ell \;\text{ or }\; \|p_{i_1} - p_{i_2}\|_2 > 2a_\ell, \tag{11}$$

since, clearly, there is no way for ellipse ℓ to cover any subset of points containing p_{i_1} and p_{i_2}, and, therefore, problem (9) is infeasible, or equivalently, the objective function of problem (10) does not vanish at a global minimizer. There is also another test involving three points p_{i_1}, p_{i_2}, and p_{i_3}. Consider the three pairs (r_1, h_1), (r_2, h_2), and (r_3, h_3) of possible bases and heigths of the triangle determined by those three points. If

$$r_i \geq 2b_\ell \text{ and } h_i \geq 2b_\ell \text{ for } i = 1, 2, 3, \tag{12}$$

then there is no way for ellipse ℓ to cover any subset of points containing p_{i_1}, p_{i_2}, and p_{i_3}.

Summing up, considering all possible choices for $x_\ell \in \{0, 1\}^n$ means that, for all combinations that can not be discarded by pre-processing strategies as the ones already described, the global solution c_ℓ of the bivariate continuous nonlinear convex optimization problem (10) is computed.

If c_ℓ is such that $f_{I(x_\ell)}(c_\ell)$ vanishes, then (x, y, c) satisfies (2–5) and it is a feasible point for problem P1. In this case, the value of the objective function (1) of problem P1 is computed and, if necessary, the incumbent solution is updated. At the end of the process, the strategy obtains the optimal solution for problem P1 for the particular case of considering only ellipse ℓ. By repeating this strategy for all possible values of $\ell \in \{1, \ldots, m\}$, the optimal solution for problem P1 with $k = 1$ can be computed. We now describe the way in which combinations of $x_{i\ell} \in \{0, 1\}, i = 1, \ldots, n$, are generated and stored.

3.2 Data structure and algorithms for the single ellipse case

All maximal subsets of points that can be covered by a given ellipse ℓ are stored in a tree structure named T_ℓ in which each node represents an appearance of a point $p_i, i \in \{1, \ldots, n\}$. In the tree, a subset S of points, $S \subseteq \{p_1, \ldots, p_n\}$, is represented by a walk from a leaf to the root, that can be transversed in both directions. From the root to a leaf, points appear in increasing order of their indices. Still regarding the ordering of the tree's nodes, sibling nodes are saved in a linked list in increasing order of their indices. Each leaf saves important information related to the subset it represents: (a) the value of the covering, and (b) the position of ellipse's center. In addition, the tree data structure contains several linked lists. There is a linked list of leaves, sorted in decreasing order by the value of the covering. It means that the first leaf in the list represents the most valuable covering for ellipse ℓ. There are also n linked list, one for each $p_i, i = 1, \ldots, n$. Each list i links all appearances of point p_i in the tree. The tree starts with its root node, that is a special node unrelated to any point.

Figure 1 illustrates the tree data structure T_ℓ for an ellipse ℓ with cost $\widetilde{w}_\ell = 1.2$. In this example, points' profits are $\widehat{w}_1 = \widehat{w}_6 = \widehat{w}_7 = 0.5$, $\widehat{w}_2 = \widehat{w}_4 = \widehat{w}_5 = \widehat{w}_8 = \widehat{w}_9 = 1.0$, and $\widehat{w}_3 = 2.0$. In Figure 1, each path from the root to a leaf represents a maximal set of points covered by the ellipse. For example, the path most to the left represents the set given by points p_1, p_2, p_3, and p_4. The associated leaf shows that the value of this covering is 3.3 and that it corresponds to placing ellipse ℓ with its center at coordinates $(3.0, 1.0)^T$. The figure also shows that there is a linked list of leaves sorted in decreasing order by the value of the covering they represent, and that there are also lists linking the appearances of each point. The list for point p_3 is shown as an example. The lists of the other points are omitted in the figure to avoid an overabundance of information. Figure 2 illustrates the maximal coverings related to the tree T_ℓ of Figure 1. Ellipse ℓ has semi major axis $a_\ell = 5$ and semi minor axis $b_\ell = 3.5$. The set of demand points is given by $\{p_1, \ldots, p_9\} = \{(12.5, 7.5)^T, (17.5, 7.5)^T, (10, 5)^T, (20, 5)^T, (12.5, 10)^T, (10, 2.5)^T, (25, 0)^T, (0, 5)^T, (5, 6.5)^T\}$.

The described data structure T_ℓ facilitates two important tasks in the process of generating all maximal subsets of points that can be covered by a given ellipse ℓ: (a) to verify whether, given a subset of points I, there exists another subset S in T_ℓ such that $I \subseteq S$ or not, and (b) in case that such S does not exist, to insert I in T_ℓ. Algorithm 1 presents the pseudo-code for computing all maximal subsets of points that can be covered by a given ellipse ℓ. Algorithm 1 uses recursive Algorithm 2 to compute all subsets of fixed cardinality γ, for $\gamma = n, n-1, \ldots, 1$. A new subset is included in the tree structure only if it is not a subset of another subset already included. In this way only maximal subsets are generated. Algorithm 2 uses Algorithms 3–4 to check whether a new subset I should be included in the tree structure or not and, if this is the

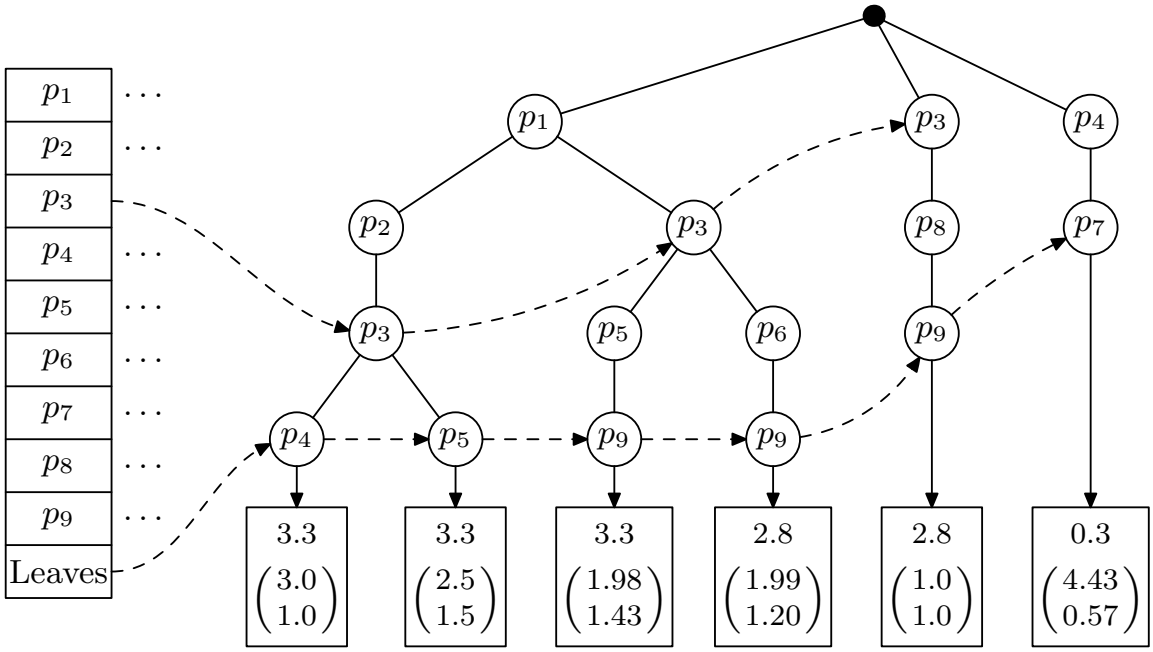

Figure 1: Tree T_ℓ in the figure represents all possible maximal coverings of an ellipse ℓ with cost $\widetilde{w}_\ell = 1.2$. Points' profits are $\widehat{w}_1 = \widehat{w}_6 = \widehat{w}_7 = 0.5$, $\widehat{w}_2 = \widehat{w}_4 = \widehat{w}_5 = \widehat{w}_8 = \widehat{w}_9 = 1.0$, and $\widehat{w}_3 = 2.0$.

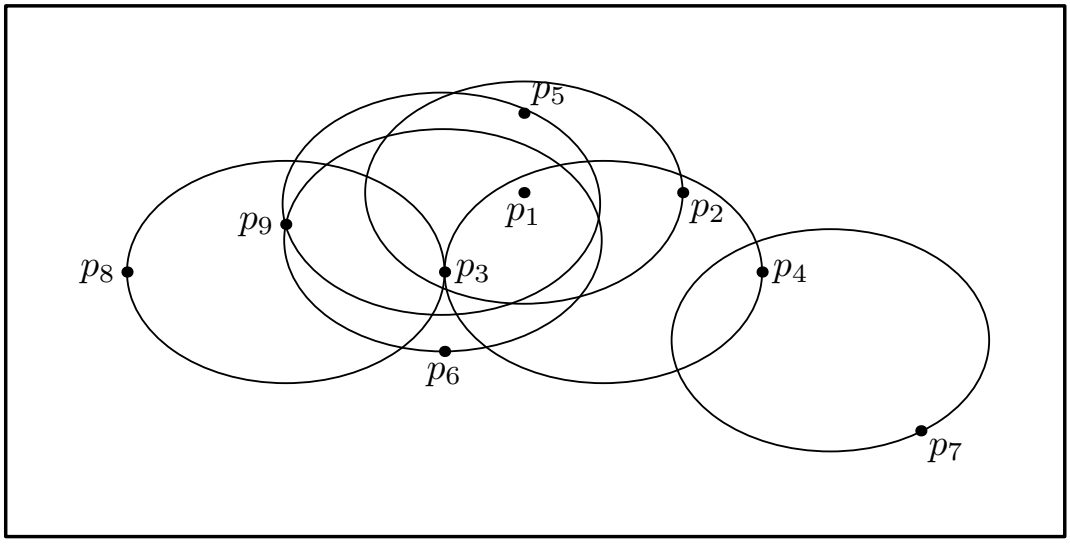

Figure 2: This figure shows the maximal coverings represented by the tree T_ℓ of Figure 1. Ellipse ℓ has semi major axis $a_\ell = 5$ and semi minor axis $b_\ell = 3.5$. The set of demand points is given by $\{p_1, \ldots, p_9\} = \{(12.5, 7.5)^T,\ (17.5, 7.5)^T,\ (10, 5)^T,\ (20, 5)^T,\ (12.5, 10)^T,\ (10, 2.5)^T,\ (25, 0)^T,\ (0, 5)^T,\ (5, 6.5)^T\}$.

case, it uses Algorithm 5 to insert it. In Algorithm 5, every time a new node s associated to a point p_i is created and inserted in the tree, it is added to the list of appearances of p_i. Moreover, if the new node is a leaf, it is also added to the list of leaves.

Algorithm 1: All maximal subsets of points $p_1, \ldots, p_n$ with profits $\widehat{w}_1, \ldots, \widehat{w}_n$ that can be covered by a given ellipse with semi major axis a, semi minor axis b, and cost $\widetilde{w}$.

Input: a, b, $\widetilde{w}$, n, p_i, $\widehat{w}_i (i = 1, \ldots, n)$.
Output: Each maximal subset is represented by a walk from the root to a leaf of tree T. The income (profit minus cost) and the ellipse center's position associated to each maximal subset are stored in the corresponding leaf.

GenTree(a, b, $\widetilde{w}$, n, p_i, $\widehat{w}_i (i = 1, \ldots, n)$)
1 **begin**
2 Set T as an empty tree containing only its special root node.
3 **for** $\gamma = n, \ldots, 1$ **do**
4 $I \leftarrow \emptyset$
5 GenTreeR(T, a, b, $\widetilde{w}$, n, p_i, $\widehat{w}_i (i = 1, \ldots, n)$, γ, n, I)
6 **return** T
7 **end**

3.3 Solving problems with multiple ellipses

In the previous subsections it was tackled the problem of finding all maximal coverings for every ellipse $j \in \{1, \ldots, m\}$. It remains to describe how to compute the optimal solution to problem P1 using exactly k ellipses for a given k, as required by constraint (2). This optimal solution is computed as a combination of the maximal coverings for the individual ellipses as we now describe.

For each ellipse $j \in \{1, \ldots, m\}$, a tree T_j with all the maximal coverings is computed. Each tree includes a list of the coverings in decreasing order of their value. The idea is to combine those coverings to build a covering using k ellipses. Algorithm 6 shows how to compute the optimal combined covering for a fixed given set of k ellipses. Note that the value of the combination of k coverings, one for each ellipse $\ell_1, \ldots, \ell_k$, is bounded from above by the sum of the values of each individual covering. Both quantities are not necessarily equal since in the combined covering some points may be covered by more than one ellipse, in which case their profits must be considered only once. This bound and the fact that the coverings for each individual ellipse are sorted by decreasing order of their value help to reduce the number of combinations in Algorithm 6. Algorithm 7 uses Algorithm 6 to check all possible $C_k^m = m!/(k!(m-k)!)$ combinations of k ellipses and to report the optimal one. If constraint (2) is substituted by (6) then we obtain problem P2 in which the optimal solution with at most k ellipses is required. In this case, the most valuable solution among the solutions with exactly k' ellipses for $k' = 1, \ldots, k$ is computed.

For the particular case in which $k = 1$, no combination has to be done and the optimal solution is given by the feasible solution with maximum income among the m optimal solutions with a single ellipse, i.e. suboptimal solutions for the individual ellipses play no role in the solution strategy. It means that, for each ellipse j, only the most valuable covering needs to

Algorithm 2: All maximal subsets of size γ of points $p_1, \ldots, p_n$ with profits $\widehat{w}_1, \ldots, \widehat{w}_n$ that can be covered by a given ellipse with semi major axis a, semi minor axis b, and cost $\widetilde{w}$.

Input: T, a, b, $\widetilde{w}$, n, p_i, $\widehat{w}_i (i = 1, \ldots, n)$, γ, n_r, I.
Output: All maximal subsets of size γ are added to T.
GenTreeR(T, a, b, $\widetilde{w}$, n, p_i, $\widehat{w}_i (i = 1, \ldots, n)$, γ, n_r, I)

1 **begin**
2 **if** $|I| < \gamma$ **then**
3 **if** $\gamma \leq n_r + |I|$ **then**
4 Given $i < j \in I$, let $r^{ij}_{\min}$ and $h^{ij}_{\min}$ be the smallest basis and height of the triangle determined by points p_i, p_j, and p_{n+1-n_r}, respectively. If there exists $i < j \in I$ such that $\min\{r^{ij}_{\min}, h^{ij}_{\min}\} \geq 2b$ then set Big$\triangle$ =True. Otherwise, set Big$\triangle$ =False.
5 Let $d^x_{\max} \leftarrow \max_{i \in I} |p^x_i - p^x_{n+1-n_r}|$, $d^y_{\max} \leftarrow \max_{i \in I} |p^y_i - p^y_{n+1-n_r}|$, and $d_{\max} \leftarrow \max_{i \in I} \|p_i - p_{n+1-n_r}\|_2$.
6 **if** $d^x_{\max} \leq 2a$ *and* $d^y_{\max} \leq 2b$ *and* $d_{\max} \leq 2a$ *and* Big$\triangle$ =False **then**
7 $I \leftarrow I \cup \{n + 1 - n_r\}$
8 GenTreeR(T, a, b, $\widetilde{w}$, n, p_i, $\widehat{w}_i (i = 1, \ldots, n)$, γ, $n_r - 1$, I)
9 $I \leftarrow I \setminus \{n + 1 - n_r\}$
10 GenTreeR(T, a, b, $\widetilde{w}$, n, p_i, $\widehat{w}_i (i = 1, \ldots, n)$, γ, $n_r - 1$, I)
11 **else**
12 **if** SearchTree*(I,T)* = false **then**
13 Compute the global minimizer $\bar{c} = \arg\min f_I(c)$.
14 **if** $f_I(\bar{c}) = 0$ **then**
15 Compute the profit $\bar{\alpha} = \sum_{i \in I} \widehat{w}_i - \widetilde{w}$ of covering points p_i with $i \in I$ by the given ellipse with cost $\widetilde{w}$.
16 Insert(I,$\bar{c}$,$\bar{\alpha}$,T)
17 **end**

Algorithm 3: Check whether there exists a subset S in the tree T such that $I \subseteq S$.

Input: I, T.
Output: true, if there exists S in T such that $I \subseteq S$, false, otherwise.
SearchTree(I, T)

1 **begin**
2 **if** $I = \emptyset$ **then**
3 **return** true
4 Let $i_{\min} = \min\{i \mid i \in I\}$ and let $L_{i_{\min}}$ be the list of nodes associated to appearances of $p_{i_{\min}}$ in T.
5 **foreach** $s \in L_{i_{\min}}$ **do**
6 **if** SearchTreeR*(I, T, s)* **then**
7 **return** true
8 **return** false
9 **end**

be stored in T_j. To perform this task, only Algorithms 1, 2, and 5 are used and a few simple modifications has to be done to Algorithms 1 and 2 in order to: (a) Save the incumbent solution with value $\bar{\alpha}_{\max}$ related the best covering so far obtained; and (b) In lines 12-16 of Algorithm 2,

Algorithm 4: Check whether there exists a subset S in the subtree of T with root s such that $I \subseteq S$.

Input: I, T, s.
Output: TRUE, if there exists S in the subtree of T with root node s such that $I \subseteq S$, FALSE, otherwise.

SEARCHTREER(I, T, s)

```
1  begin
2      if I = ∅ then
3          return TRUE
4      Let p_k be the point associated to node s and let i_min = min{i | i ∈ I}.
5      if i_min > k then
6          return FALSE
7      if i_min = k then
8          I ← I \ {i_min}
9      Let L_s be the list of sons of s.
10     foreach t ∈ L_s do
11         if SEARCHTREER(I, T, t) then
12             return TRUE
13     return FALSE
14 end
```

Algorithm 5: Insert subset I in tree T.

Input: I, $\bar{c}$, $\bar{\alpha}$, T.
Output: Updated data structure of T with the new subset I inserted in it.

INSERT(I, $\bar{c}$, $\bar{\alpha}$, T)

```
1  begin
2      Let i_1 < i_2 < ··· < i_|I| be the elements of I.
3      Let s_0 be the root node of T and let L_{s_0} be the list of sons of node s_0.
4      j ← 1
5      while ∃ s_j ∈ L_{s_{j-1}} such that node s_j is associated to p_{i_j} do
6          Let L_{s_j} be the list of sons of node s_j
7          j ← j + 1
8      for k = j, ..., |I| do
9          Create a new node s_k associated to p_{i_k} and add it to the list L_{s_{k-1}} of sons of node s_{k-1}.
           Also add s_k to the list of appearances of p_{i_k} in T.
10     Save c̄ and ᾱ in the new leaf s_|I| and add s_|I| to the leaves' list of T.
11 end
```

given I, first compute $\bar{\alpha}$ and only compute $\bar{c}$ if $\bar{\alpha} > \bar{\alpha}_{\max}$. Then, if $f_I(\bar{c}) = 0$, insert $(I, \bar{c}, \bar{\alpha})$ in T and update the incumbent solution. The fact that $\bar{\alpha} > \bar{\alpha}_{\max}$ assures that there is no set S in T such that $I \subseteq S$, which greatly simplifies the strategy and dismiss the usage of Algorithms 3 and 4.

Algorithm 6: Compute the covering with maximum income using the given fixed set of k ellipses $K = \{\ell_1, \ldots, \ell_k\} \subseteq \{1, \ldots, m\}$ whose costs are $\widetilde{w}_{\ell_1}, \ldots, \widetilde{w}_{\ell_k}$. All maximal subsets of points $p_1, \ldots, p_n$ that can be covered by ellipse ℓ_j are stored in the tree T_{ℓ_j}, for $j = 1, \ldots, k$. The profits of points $p_1, \ldots, p_n$ are given by $\widehat{w}_1, \ldots, \widehat{w}_n$.

Input: k, T_{ℓ_j}, $\tilde{w}_{\ell_j}(j = 1, \ldots, k)$, m, n, $\widehat{w}_i(i = 1, \ldots, n)$.
Output: Maximum-income covering using the given fixed subset of k ellipses $K = \{\ell_1, \ldots, \ell_k\} \subseteq \{1, \ldots, m\}$. This is a feasible solution for problem P1 and, on output, it is represented by $(\bar{x}, \bar{y}, \bar{c})$. Its value is given by $\bar{\phi}$.

CombinedCovering(k, T_{ℓ_j}, $\tilde{w}_{\ell_j}(j = 1, \ldots, k)$, m, n, $\widehat{w}_i(i = 1, \ldots, n)$)

begin
  Let L_{ℓ_j} be the leaves' list of tree T_{ℓ_j} for $j = 1, \ldots, k$. Let u_{ℓ_j} be a pointer to a leaf in list L_{ℓ_j}. This leaf is associated to a certain maximal subset of points being covered by ellipse ℓ_j. Let $I[u_{\ell_j}]$ be the set of indices of the covered points, $\alpha[u_{\ell_j}]$ be the covering value (profit of the covered points minus cost of ellipse ℓ_j), and $c[u_{\ell_j}]$ be the associated position of ellipse ℓ_j.
  for $j = 1, \ldots, k$ **do**
    $u_{\ell_j} \leftarrow \text{FIRST}(L_{\ell_j})$
  $\bar{\phi} \leftarrow -\infty$
  while $u_{\ell_1} \neq \text{NULL}$ **do**
    $\phi_{\text{UB}} \leftarrow \sum_{j=1}^{k} \alpha[u_{\ell_j}]$
    if $\phi_{\text{UB}} < \bar{\phi}$ **then**
      $u_1 \leftarrow \text{NEXT}(L_{\ell_1})$
      for $j = 2, \ldots, k$ **do**
        $u_{\ell_j} \leftarrow \text{FIRST}(L_{\ell_j})$
    else
      for $t = 1, \ldots, m$ **do**
        if $\exists\ \ell_j \in K$ *such that* $t = \ell_j$ **then**
          $c_t \leftarrow c[u_{\ell_j}],\ y_t \leftarrow 1$
          for $i = 1, \ldots, n$ **do**
            if $i \in I[u_{\ell_j}]$ *and* $i \notin I[u_{\ell_h}],\ \forall\ \ell_h \in K$ *such that* $\ell_h < \ell_j$ **then**
              $x_{it} \leftarrow 1$
            else
              $x_{it} \leftarrow 0$
        else
          $c_t \leftarrow 0,\ y_t \leftarrow 0,\ x_t \leftarrow 0$
      Compute the value of the objective function $\phi(x, y)$ of (1–5) given by (8)
      if $\phi(x, y) > \bar{\phi}$ **then**
        $\bar{\phi} \leftarrow \phi(x, y),\ (\bar{x}, \bar{y}, \bar{c}) \leftarrow (x, y, c)$
      $u_{\ell_k} \leftarrow \text{NEXT}(L_{\ell_k})$
      $q \leftarrow k$
      while $q > 1$ *and* $u_{\ell_q} = \text{NULL}$ **do**
        $u_{\ell_q} \leftarrow \text{FIRST}(L_{\ell_q}),\ u_{\ell_{q-1}} \leftarrow \text{NEXT}(L_{\ell_{q-1}}),\ q \leftarrow q - 1$
  return $(\bar{\phi}, \bar{x}, \bar{y}, \bar{c})$
end

3.4 Allowing individual ellipses' rotations

Problems P3 and P4 are the counterparts of problems P1 and P2, respectively, for the case in which rotations in the ellipses are allowed. Therefore, P3 corresponds to minimize (1) subject

Algorithm 7: Compute the solution for problem P1.

Input: k, m, a_j, b_j, $\tilde{w}_j (j = 1, \ldots, m)$, n, p_i, $\hat{w}_i (i = 1, \ldots, n)$.
Output: Solution (x^*, y^*, c^*) for problem P1.
SolveP1(k, m, a_j, b_j, $c_j (j = 1, \ldots, m)$, n, p_i, $w_i (i = 1, \ldots, n)$)

```
1  begin
2      for ℓ = 1, ..., m do
3          T_ℓ ← GenTree(a_ℓ, b_ℓ, w̃_ℓ, n, p_i, ŵ_i (i = 1, ..., n))
4      φ* ← −∞
5      foreach {ℓ_1, ..., ℓ_k} ⊆ {1, ..., m} do
6          (φ̄, x̄, ȳ, c̄) ← CombinedCovering(k, T_{ℓ_j}, w̃_{ℓ_j} (j = 1, ..., k), m, n, ŵ_i (i = 1, ..., n))
7          if φ̄ > φ* then
8              (φ*, x*, y*, c*) ← (φ̄, x̄, ȳ, c̄)
9      return (φ*, x*, y*, c*)
10 end
```

to (2–4,7), while P4 corresponds to minimize (1) subject to (3,4,6,7).

Defining

$$\begin{pmatrix} v_1(c_j, \theta_j) \\ v_2(c_j, \theta_j) \end{pmatrix} \equiv \begin{pmatrix} 1/a_j & 0 \\ 0 & 1/b_j \end{pmatrix} \begin{pmatrix} \cos\theta_j & -\sin\theta_j \\ \sin\theta_j & \cos\theta_j \end{pmatrix} \begin{pmatrix} c_j^x - p_i^x \\ c_j^y - p_i^y \end{pmatrix}, \tag{13}$$

equation (7) can be re-written as

$$v_1(c_j, \theta_j)^2 + v_2(c_j, \theta_j)^2 \leq 1 + (1 - x_{ij})M, i = 1, \ldots, n, j = 1, \ldots, m. \tag{14}$$

Therefore, for given x and y satisfying (2–4) (or (3,4,6)), to determine whether c and θ exist such that (x, y, c, θ) satisfies (2–4,7) (or (3,4,6,7)) reduces to solving the $|J(y)|$ independent trivariate feasibility problems of the form

$$v_1(c_j, \theta_j)^2 + v_2(c_j, \theta_j)^2 \leq 1, \forall\, i \in I(x_j), \tag{15}$$

for all $j \in J(y)$. Defining

$$g_{I(x_j)}(c_j, \theta_j) \equiv \sum_{i \in I(x_j)} \max\{0, v_1(c_j, \theta_j)^2 + v_2(c_j, \theta_j)^2 - 1\}^2, \tag{16}$$

solving (15) is equivalent to finding the global solution to

$$\mathrm{Min} g_{I(x_j)}(c_j, \theta_j). \tag{17}$$

The main difference with respect to the previous situation (for problems P1 and P2) is that $g_I(x_j)(c_j, \theta_j)$ is a *non-convex* continuous and differentiable function whose global minimizer must be found. However, a property of the objective function (16) makes problem (17) easier to be solved: objective function (16) is a convex function of c_j when θ_j is fixed.

There is another small difference between problems P1-P2 and P3-P4 that is related to the ellipses' rotation. In problems P1 and P2, inequalities (11,12) are used to discard subsets

of points that can not be covered by a given ellipse, avoiding the task of solving feasibility problem (9). When rotations are allowed, it is trivial to see that feasibility problem (15) is infeasible in the case in which there exist points p_{i_1} and p_{i_2} such that

$$\|p_{i_1} - p_{i_2}\|_2 > 2a_\ell. \tag{18}$$

Therefore, lines 5–6 in Algorithm 2 must be modified to replace (11) by (18), while keeping (12). Other than that, the whole strategy depicted by Algorithms 1–7, can also be used to solve problems P3 and P4 with no further modifications.

4 Numerical experiments

Algorithms 1–7 were coded in C language and compiled with gcc (GCC 4.4.5). The compiler optimization option -O4 was adopted. All the experiments were run on a 3.0GHz Intel Core 2 Quad Q950 with 4.0GB of RAM memory and a 64bits Linux Operating System (Ubuntu 11.10). Codes are available at *http://www.ime.usp.br/∼egbirgin/* for benchmarking purposes.

In our implementation, the unconstrained convex continuous optimization problem (10) is solved with Algencan [4, 5]. Algencan is an NLP solver for smooth nonlinear programming problems with general constraints that, when applied to (10), roughly speaking reduces to the Newton method (see [6, 9, 10] for details). Algencan is part of the TANGO Project and is also available at *http://www.ime.usp.br/∼egbirgin/tango/*. The global solution to the unconstrained non-convex continuous optimization problem (17) is computed with the αBB method introduced in [1, 2, 3, 7] and implemented in [8] for bound-constrained minimization. Since bound constraints for the variables $\theta_j \in \mathbb{R}$ and $c_j \in \mathbb{R}^2$ are needed, we considered $-\pi/2 \leq \theta_j \leq \pi/2$, $\max_{i \in I(x_j)}\{p_i^x\} - a_j \leq c_j^x \leq \min_{i \in I(x_j)}\{p_i^x\} + a_j$, and $\max_{i \in I(x_j)}\{p_i^y\} - a_j \leq c_j^y \leq \min_{i \in I(x_j)}\{p_i^y\} + a_j$. These bound constraints define a three-dimensional box such that, if $g_{I(x_j)}(\cdot)$ vanishes at the global minimizers to problem (17), then the box contains a global minimizer. Bounds for θ_j are straightforward, while bounds for the center of the ellipse are not very intuitive and require further comments. The argument that follows shows that the given bounds do not eliminate any feasible point of (17) at which the objective function $g_{I(x_j)}(\cdot)$ vanishes. Consider, for example, the bound $c_j^x \leq \min_{i \in I(x_j)}\{p_i^x\} + a_j$ and let $p_t^x = \arg\min_{i \in I(x_j)}\{p_i^x\}$. If c_j is such that $c_j^x > \min_{i \in I(x_j)}\{p_i^x\} + a_j = p_t^x + a_j$ then the ellipse can not contain p_t and, in consequence, the objective function does not vanishes at c_j. The other three cases are analogous.

A few words regarding the precision of the computed solutions are in order. We will refer to problem P1 but the same claims apply to the other problems too. In problem P1, the objective function (1) and constraints (2–4) involve only binary variables and, hence, the objective function value will be computed up to the machine precision as well as the constraints will be satisfied up to the machine precision. On the other hand, constraints (5) involves continuous variables. By using the default values for Algencan and the αBB method, constraint (5) will be satisfied with precision 10^{-8}, i.e. reported solutions to instances of problem P1 will satisfy

$$(c_j^x - p_i^x)^2/a_j^2 + (c_j^y - p_i^y)^2/b_j^2 \leq 1 + (1 - x_{ij})M + \varepsilon, \; i = 1, \ldots, n, \; j = 1, \ldots, m,$$

where $\varepsilon = 10^{-8}$.

4.1 Instances of problems P1 and P3 using data taken from [11]

In a first set of experiments, we considered six instances of problem P1 presented in [11] for which the availability of their data allows reproducibility. These instances are based on three sets of points P_1, P_2, and P_3 with $|P_1| = 25$, $|P_2| = 50$, and $|P_3| = 100$. Points are randomly generated with uniform distribution within the two-dimensional box $[0, 50]^2$ and all of them have unitary profit. Three ellipses with semi major and semi minor axes $(a_1, b_1) = (6, 4)$, $(a_2, b_2) = (8, 5)$, and $(a_3, b_3) = (10, 6)$, and costs $\widetilde{w}_1 = 2.0$, $\widetilde{w}_2 = 3.2$, and $\widetilde{w}_3 = 4.8$, respectively, are considered in all the six instances, i.e. $m = 3$ in all cases. The six instances consist on considering each one of the three sets of points, P_1, P_2, and P_3, with $k = 1$ and $k = 2$. In addition, we also considered the case $k = 3$. The sets of points that complete the definition of the instances can be found in [11] (pp. 207–208). Table 1 shows the results. In the table, the first four columns are related to the instances definition and are self-explanatory; "Selected ellipses" means ellipses that are selected to be used in the optimal solution, and "Optimal income" corresponds to the value of the global minimum. The remaining columns show some figures related to the performance of the introduced method. Columns "Trees' figures: # nodes and # leaves" display the sum of the number of nodes and leaves in the three generated trees (one for each ellipse). Columns "# NLP subproblems: feasible and infeasible" show the number of times an NLP solver was called to solve feasible and infeasible NLP feasibility problems, respectively. Finally, "CPU Time" shows the CPU time in seconds needed to solve each instance. Figure 3 completes the description of the solutions found by illustrating the location of the selected ellipses and which point is covered by each ellipse.

It is important to notice that the solutions found in this experiment to the six instances taken from [11] are the same as the ones reported in [11]. It shows that the heuristic method present in [11] is capable to find optimal solutions for the present set of instances. Regarding the performance of the introduced method, figures in the table show that the size of the generated trees is very moderate. In principle, the number of feasible NLP solved subproblems should coincide with the number of leaves in the trees. The table shows that the former number is a little bit smaller than the later. For instances of problem P1, in which rotations of the ellipses are not allowed, the position of the ellipse to cover a given set of points can be trivially determined in at least two situation: (i) when the set of points consists in a solely point; and (b) when the set of points consists in two points whose distance is less than or equal to twice the semi minor axis of the ellipse. In this cases, the NLP solver is not used to solve those trivial feasibility problems. The same arguments applies to problem P2. In addition to those cases, in problems P3 and P4 it also trivial the case of a set of two points whose distance is not grater than twice the semi major axis of the ellipse. Finally, regarding the figures in the table, the huge amount of infeasible feasibility subproblems being solved by the NLP solver shows that this is the bottleneck of the introduced method. In this sense, more sophisticated strategies than the ones represented by inequalities (11), (12), and (18), developed to determine in advance that a feasible subproblem is infeasible, would help to improve the performance of the method.

In a second set of experiments, we considered the same data of the instances described in the paragraph above, but as input data for instances of problem P3, i.e. allowing individual angles of rotation for the ellipses[1]. Table 2 shows the results. In the table, "–" means that the

[1]Note the abuse of notation as we will use the same name for instances of problems P1 and P3 based on the

method attained the CPU time limit of 12 hours. As expected, the optimal incomes for the instances of problem P3 are greater than or equal to the optimal incomes of the corresponding instances of problem P1. In the case of problem P3, unconstrained nonlinear minimization subproblem (17) is non-convex and, even being a problem with a small number of variables, finding a global solution is much more expensive than in the previous case. For that reason, instances CM8 and CM9 were not solved within a CPU time limit of 12 hours. As an alternative, we considered an heuristic approach: a global solution of subproblem (17) was approximated by the stationary point computed with only one call the local solver Algencan. Table 3 and Figure 4 show the results. Comparing Tables 3 and 2 it can be seen that this simple heuristic strategy found the known global optimal solution for instances CM1–CM7. In case the results would have been unsatisfactory, more than a solely run of the local solver Algencan could have been used, configuring a multistart stochastic global optimization strategy. Therefore, we refer to this method as a stochastic global optimization approach in contrast with the determinstic global optimization approach that uses the αBB method to find a global solution to the subproblems.

4.2 Instances of problems P2 and P4 using data taken from [11]

Problem P1 corresponds to require that the number of used ellipses must be equal to k. As it can be seen in Table 1, instances CM1–CM3 are given by the set of points P_1 with $n = |P_1| = 25$, the same $m = 3$ ellipses, and $k = 1, 2, 3$, respectively. As more ellipses are allowed to be used in CM2 than in CM1, it might be expected the optimal income of CM2 to be larger than the optimal income of CM1; and the same situation with CM3 with respect to CM2 (and also with the subsets CM4–CM6 and CM7–CM9). However, the numerical experiments in Table 1 show that the optimal income of CM3 is smaller than the optimal income of CM2. This is due to the fact that, in CM3, being forced to use $k = 3$ ellipses, the third ellipse with $(a_3, b_3) = (10, 6)^T$ and $\widetilde{w}_3 = 4.8$ is used to cover four points with unitary profit (see Figure 3), presenting a negative profit of $4 - 4.8 = -0.8$, that is the difference between the optimal incomes of instances CM2 and CM3. In all the other situations, using more ellipses is in fact more profitable and the described unexpected situation does not occur. Consider now the case of problem P2, in which the number of used ellipses is not forced to be equal to k, but to be not greater than k. The described scenario implies that the optimal solution of instance CM3 of problem P2 coincide with the optimal solution of instance CM2 of problem P1; while for the other eight instances the solutions to problems P1 and P2 coincide. For the nine considered instances, Table 3 shows that the optimal solutions to instances of problem P4 coincide with the optimal solutions to the corresponding instances of problems P3.

4.3 Other instances

To further analyse the performance of the proposed methods, new sets of instances like the ones suggested in [11] were generated. We considered instances based on ten different sets of points, namely, $Q_1, \ldots, Q_{10}$, with $n \in \{10, 20, \ldots, 100\}$ unitary-profit points randomly generated with uniform distribution within the two-dimensional box $[0, 50]^2$. The five considered ellipses have semi major and minor axes $(a_1, b_1) = (1, 1)$, $(a_2, b_2) = (3, 2)$, $(a_3, b_3) = (5, 3)$, $(a_4, b_4) = (7, 4)$,

same data.

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
CM1				1	1	2.0	4	1	1	1	0.00
CM2	P_1	25	3	2	1,2	3.8	151	59	49	158	0.11
CM3				3	1,2,3	3.0	151	59	49	158	0.11
CM4				1	3	4.2	15	2	2	34	0.10
CM5	P_2	50	3	2	1,3	8.2	514	140	133	4,925	4.59
CM6				3	1,2,3	10.0	514	140	133	4,925	4.60
CM7				1	3	12.0	40	3	3	212	71.61
CM8	P_3	100	3	2	2,3	20.0	2,388	410	409	1,432,878	2,772.33
CM9				3	1,2,3	27.0	2,388	410	409	1,432,878	2,786.42

Table 1: Numerical results of the deterministic global optimization method applied to the six instances of problem P1 taken from [11] (CM1, CM2, CM4, CM5, CM7, CM8) plus the three additional instances with $k = 3$ (CM3, CM6, CM9).

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
CM1				1	2	2.8	10	2	2	2	1.66
CM2	P_1	25	3	2	1,2	4.8	237	88	64	174	152.42
CM3				3	1,2,3	5.0	237	88	64	174	177.27
CM4				1	2	5.8	15	2	2	346	200.87
CM5	P_2	50	3	2	2,3	10.0	1,038	270	214	26,048	27,146.31
CM6				3	1,2,3	13.0	1,038	270	214	26,048	27,302.67
CM7				1	3	13.2	42	3	3	774	665.33
CM8	P_3	50	3	2	–	–	–	–	–	–	–
CM9				3	–	–	–	–	–	–	–

Table 2: Numerical results of the deterministic global optimization method applied to instances CM1–CM9 of problem P3.

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
CM1				1	2	2.8	10	2	2	2	0.00
CM2	P_1	25	3	2	1,2	4.8	249	93	71	197	0.33
CM3				3	1,2,3	5.0	249	93	71	197	0.33
CM4				1	2	5.8	15	2	2	346	1.16
CM5	P_2	50	3	2	2,3	10.0	1,124	295	287	26,107	58.64
CM6				3	1,2,3	13.0	1,124	295	287	26,107	69.80
CM7				1	3	13.2	42	3	3	774	369.68
CM8	P_3	100	3	2	2,3	22.0	8,452	1,564	1,561	7,678,669	27,185.65
CM9				3	1,2,3	28.0	8,452	1,564	1,561	7,678,669	27,053.48

Table 3: Numerical results of the heuristic or stochastic global optimization method applied to instances CM1–CM9 of problem P3.

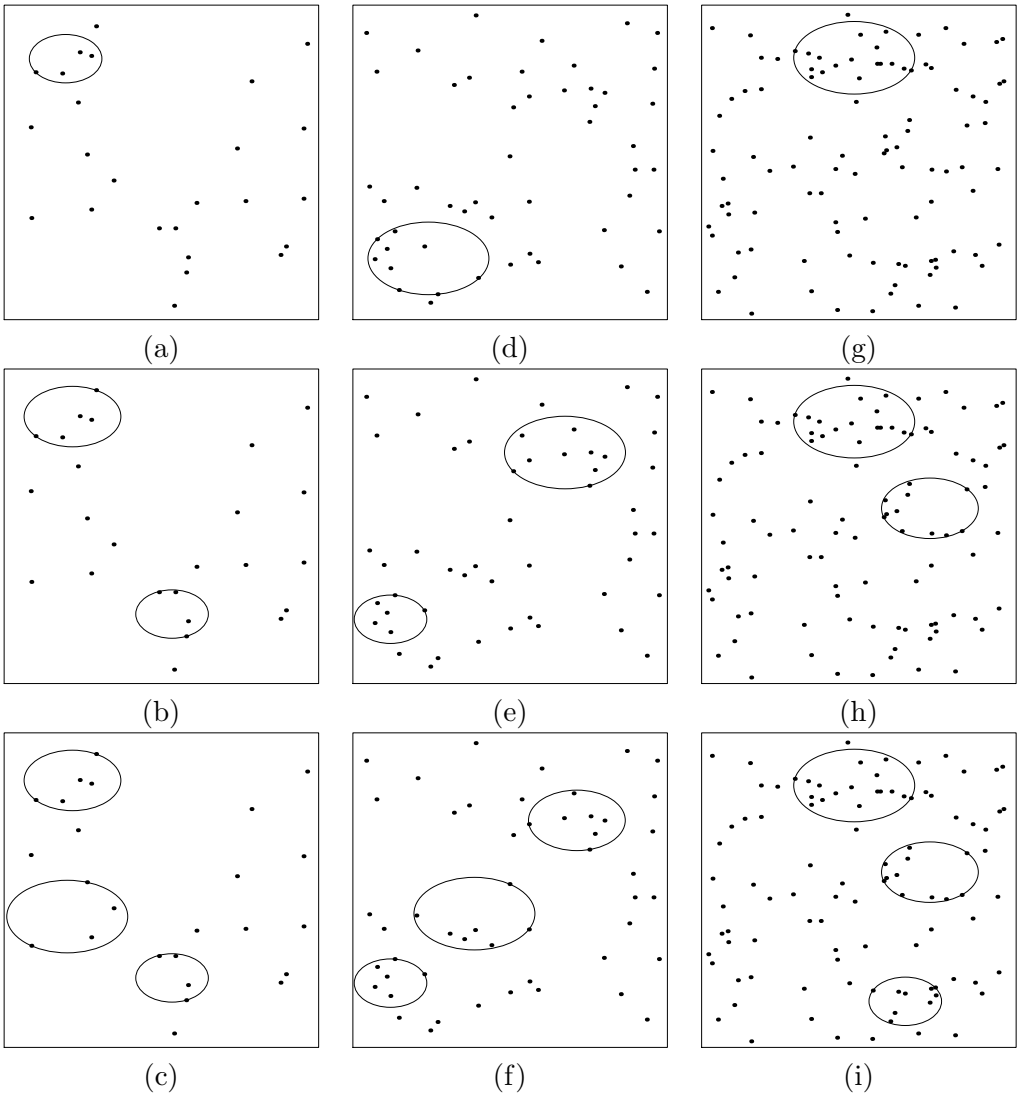

Figure 3: Graphical representation of the solutions found by the deterministic global optimization method applied to the six instances of problem P1 taken from [11] (CM1, CM2, CM4, CM5, CM7, CM8) plus the three additional instances with $k = 3$ (CM3, CM6, CM9). (a)–(c) correspond to instances CM1–CM3, (d)–(f) correspond to instances CM4–CM6, and (g)–(i) correspond to instances CM7–CM9.

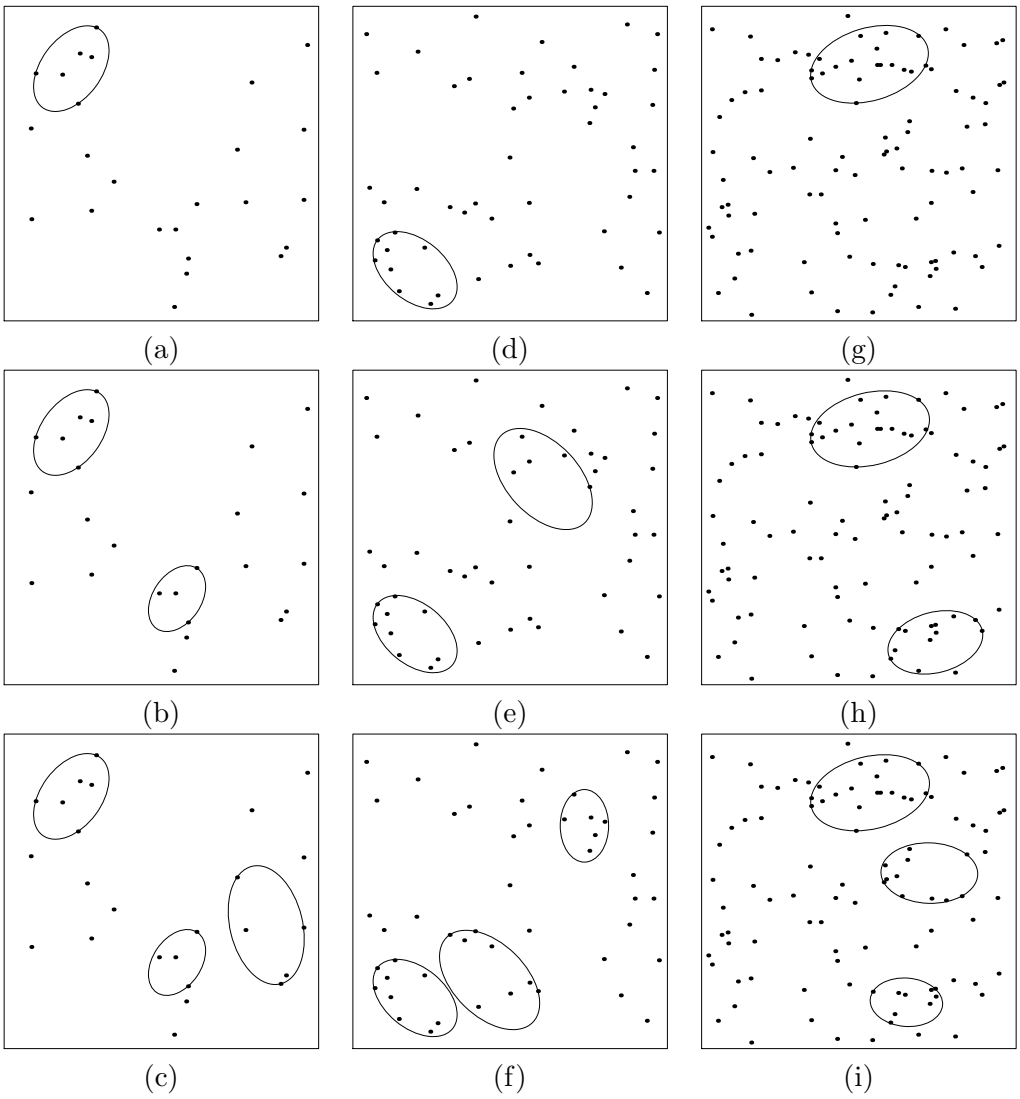

Figure 4: Graphical representation of the solutions found by the heuristic or stochastic global optimization method applied to the six instances of problem P1 taken from [11] (CM1, CM2, CM4, CM5, CM7, CM8) plus the three additional instances with $k = 3$ (CM3, CM6, CM9). (a)–(c) correspond to instances CM1–CM3, (d)–(f) correspond to instances CM4–CM6, and (g)–(i) correspond to instances CM7–CM9.

and $(a_5, b_5) = (9, 5)$, and profits $\widetilde{w}_1 = 0.1$, $\widetilde{w}_2 = 0.6$, $\widetilde{w}_3 = 1.5$, $\widetilde{w}_4 = 2.8$, and $\widetilde{w}_5 = 4.5$, respectively. Combining the ten sets of points, with $m \in \{3, 4, 5\}$ (i.e, the first three, the first four, and all the five ellipses) and $k \in \{1, \ldots, m\}$, we totalized 120 instances. Instances will be named AB001, ..., AB120 from now on, and are available at *http://www.ime.usp.br/∼egbirgin/* for benchmarking purposes.

Tables 4 and 5 show the results of solving instances AB001–AB120 of problem P1 by the deterministic global optimization approach, while Tables 6 and 7 show the results of solving instances AB001–AB120 of problem P3. As expected, optimal incomes of instances of problem P3 are larger than or equal to the optimal incomes of the corresponding instances of problem P1. Note that optimal solutions of some large instances of problem P3 were not found by the method within the CPU time limit of twelve hours. Tables 8 and 9 show the results of solving the same instances of problem P3 by the stochastic approach. Much less CPU time was needed by the latter method to find the known optimal solutions of *all* the instances the deterministic approach was able to solve.

As already explained, the observation of Tables 4–9 allows us to identify the instances for which solving problem P2 or P4 delivers an optimal income larger than its corresponding instance of problem P1 or P3, respectively. Those are the instances for which the constraint that requires the number of used ellipses to be equal to k forces the usage of an ellipse whose cost does not compensate the profit of the points it covers. Tables 10 and 11 show the results of solving those selected instances of problems P2 and P4, respectively, using the deterministic global optimization approach, while Table 12 shows the results of solving those selected instances of problems P4 by the heuristic approach. For the remaining instances, numerical results would coincide with the ones already reported in the previous tables. Figure 5 shows solutions to instance AB024 of problems P1–P4. Tables 4–12 indicate that the optimal incomes related to solutions depicted on Figures 5(a–d) are 2.5, 3.5 4.8, and 5.0, respectively. The tables and the graphics illustrate that, when the problem formulation allows at most k ellipses to be used instead of exactly k, optimal solutions associated to larger incomes may be found.

5 Concluding remarks

Covering problems with ellipses were considered in the present work. Deterministic and stochastic global optimization methods were developed for different variants of the covering problem. The introduced deterministic method is able to solve instances of the problem presented in [11] in which rotations of the ellipses are not allowed. Numerical experiments show that the heuristic method presented in [11] found a global solution in all the six considered instances.

As suggested in [11], the situation in which each ellipse has an individual angle of rotation was also tackled in the present work. Since NLP subproblems are non-convex in this case, the problem is harder than the one in which ellipses axes are fixed (being parallel to the Cartesian axes or fixed at any arbitrary angle). The introduced deterministic global optimization method was able to find an optimal solution for small and moderate-size instances of the problem, while the stochastic global optimization version of the method found the optimal solutions to all the small and moderate-size instances and delivered "reasonable" solutions to the larger instances. As the quality of the solution delivered in the larger instances remains to be determined, those

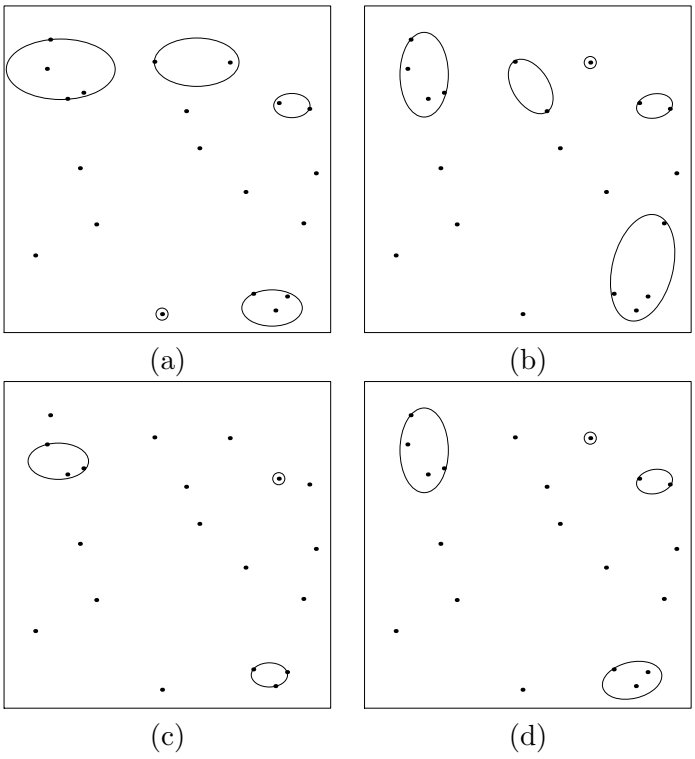

Figure 5: Graphical representation of the solutions found to instance AB024. (a) Problem P1, i.e. no rotations and exactly 5 ellipses, (b) Problem P3, i.e. rotations allowed and exactly 5 ellipses, (c) Problem P2, i.e. no rotations and at most 5 ellipses, and (d) Problem P4, i.e. rotations allowed and at most 5 ellipses.

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
AB001				1	2	1.4	4	2	2	0	0.00
AB002	Q_1	10	3	2	1,2	2.3	30	27	3	2	0.00
AB003				3	1,2,3	2.8	30	27	3	2	0.00
AB004				1	1	0.9	5	3	0	1	0.00
AB005	Q_1	10	4	2	1,3	1.4	40	36	0	2	0.00
AB006				3	1,2,3	1.8	40	36	0	2	0.00
AB007				4	1,2,3,4	1.0	40	36	0	2	0.00
AB008				1	1	0.9	8	4	2	6	0.00
AB009				2	1,3	1.4	60	48	11	14	0.00
AB010	Q_1	10	5	3	1,2,3	1.8	60	48	11	14	0.00
AB011				4	1,2,3,4	1.0	60	48	11	14	0.00
AB012				5	1,2,3,4,5	-1.5	60	48	11	14	0.00
AB013				1	2	1.4	4	2	1	0	0.00
AB014	Q_2	20	3	2	1,2	2.3	61	53	4	1	0.00
AB015				3	1,2,3	2.8	61	53	4	1	0.00
AB016				1	3	1.5	6	2	2	2	0.00
AB017	Q_2	20	4	2	2,3	2.9	91	65	16	14	0.00
AB018				3	1,2,3	3.8	91	65	16	14	0.00
AB019				4	1,2,3,4	4.0	91	65	16	14	0.00
AB020				1	2	2.4	13	4	4	5	0.00
AB021				2	2,3	3.9	112	75	20	20	0.02
AB022	Q_2	20	5	3	1,2,3	4.8	112	75	20	20	0.01
AB023				4	1,2,3,4	4.0	112	75	20	20	0.01
AB024				5	1,2,3,4,5	2.5	112	75	20	20	0.01
AB025				1	3	2.5	4	1	1	0	0.00
AB026	Q_3	30	3	2	2,3	4.9	101	67	16	10	0.00
AB027				3	1,2,3	6.8	101	67	16	10	0.01
AB028				1	3	2.5	9	2	2	4	0.00
AB029	Q_3	30	4	2	2,3	4.9	156	93	31	52	0.03
AB030				3	2,3,4	6.1	156	93	31	52	0.04
AB031				4	1,2,3,4	7.0	156	93	31	52	0.03
AB032				1	3	2.5	15	3	3	8	0.00
AB033				2	2,3	4.9	225	110	51	196	0.14
AB034	Q_3	30	5	3	2,3,4	7.1	225	110	51	196	0.15
AB035				4	1,2,3,4	9.0	225	110	51	196	0.15
AB036				5	1,2,3,4,5	9.5	225	110	51	196	0.15
AB037				1	3	2.5	4	1	1	6	0.00
AB038	Q_4	40	3	2	2,3	4.9	148	95	29	36	0.02
AB039				3	1,2,3	6.8	148	95	29	36	0.02
AB040				1	4	5.2	8	1	1	0	0.00
AB041	Q_4	40	4	2	2,4	7.1	216	114	43	191	0.16
AB042				3	1,3,4	8.6	216	114	43	191	0.16
AB043				4	1,2,3,4	10.0	216	114	43	191	0.16
AB044				1	5	3.5	8	1	1	0	0.01
AB045				2	2,5	7.0	323	139	75	227	0.17
AB046	Q_4	40	5	3	3,4,5	9.2	323	139	75	227	0.16
AB047				4	1,3,4,5	11.1	323	139	75	227	0.16
AB048				5	1,2,3,4,5	12.5	323	139	75	227	0.16
AB049				1	3	5.5	7	1	1	1	0.01
AB050	Q_5	50	3	2	1,3	7.9	218	113	49	210	0.18
AB051				3	1,2,3	9.8	218	113	49	210	0.17
AB052				1	4	5.2	8	1	1	2	0.02
AB053	Q_5	50	4	2	3,4	8.7	336	150	79	1,020	0.80
AB054				3	2,3,4	11.1	336	150	79	1,020	0.80
AB055				4	1,2,3,4	13.0	336	150	79	1,020	0.80
AB056				1	5	3.5	8	1	1	7	0.02
AB057				2	2,5	6.9	482	198	118	1,137	0.89
AB058	Q_5	50	5	3	2,3,5	9.4	482	198	118	1,137	0.90
AB059				4	2,3,4,5	11.6	482	198	118	1,137	0.91
AB060				5	1,2,3,4,5	13.5	482	198	118	1,137	0.93

Table 4: Numerical results of the deterministic global optimization method applied to instances AB001–AB120 of problem P1.

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
AB061				1	3	3.5	5	1	1	0	0.00
AB062	Q_6	60	3	2	2,3	5.9	248	134	55	81	0.07
AB063				3	1,2,3	7.8	248	134	55	81	0.06
AB064				1	4	5.2	8	1	1	8	0.05
AB065	Q_6	60	4	2	3,4	8.7	427	185	99	1,116	0.87
AB066				3	2,3,4	12.1	427	185	99	1,116	0.88
AB067				4	1,2,3,4	14.0	427	185	99	1,116	0.87
AB068				1	5	4.5	9	1	1	41	0.16
AB069				2	3,5	9.0	665	258	165	4,393	3.90
AB070	Q_6	60	5	3	2,3,5	12.4	665	258	165	4,393	3.88
AB071				4	2,3,4,5	14.6	665	258	165	4,393	3.90
AB072				5	1,2,3,4,5	16.5	665	258	165	4,393	4.12
AB073				1	3	4.5	6	1	1	2	0.02
AB074	Q_7	70	3	2	2,3	7.9	314	161	75	268	0.20
AB075				3	1,2,3	9.8	314	161	75	268	0.21
AB076				1	4	5.2	8	1	1	4	0.03
AB077	Q_7	70	4	2	3,4	9.7	469	201	114	918	0.80
AB078				3	2,3,4	13.1	469	201	114	918	0.73
AB079				4	1,2,3,4	16.0	469	201	114	918	0.79
AB080				1	5	5.5	10	1	1	78	0.30
AB081				2	3,5	10.0	808	266	187	6,220	6.75
AB082	Q_7	70	5	3	3,4,5	14.2	808	266	187	6,220	6.76
AB083				4	2,3,4,5	17.6	808	266	187	6,220	7.12
AB084				5	1,2,3,4,5	19.5	808	266	187	6,220	7.07
AB085				1	3	4.5	6	1	1	0	0.02
AB086	Q_8	80	3	2	2,3	7.9	343	178	81	173	0.14
AB087				3	1,2,3	10.8	343	178	81	173	0.17
AB088				1	4	7.2	10	1	1	4	0.20
AB089	Q_8	80	4	2	3,4	12.7	713	260	161	4,851	5.11
AB090				3	2,3,4	16.1	713	260	161	4,851	5.25
AB091				4	1,2,3,4	18.0	713	260	161	4,851	5.10
AB092				1	4	6.2	19	2	2	94	0.59
AB093				2	4,5	10.7	1,079	365	251	13,882	17.20
AB094	Q_8	80	5	3	3,4,5	15.2	1,079	365	251	13,882	17.49
AB095				4	2,3,4,5	18.6	1,079	365	251	13,882	18.64
AB096				5	1,2,3,4,5	19.5	1,079	365	251	13,882	17.42
AB097				1	3	5.5	7	1	1	1	0.03
AB098	Q_9	90	3	2	2,3	9.9	451	216	110	520	0.61
AB099				3	1,2,3	11.8	451	216	110	520	0.50
AB100				1	4	6.2	9	1	1	31	0.24
AB101	Q_9	90	4	2	3,4	10.7	739	292	188	3,537	4.30
AB102				3	2,3,4	14.1	739	292	188	3,537	4.26
AB103				4	1,2,3,4	17.0	739	292	188	3,537	4.32
AB104				1	4	8.2	23	2	2	18	1.80
AB105				2	4,5	12.7	1,410	417	311	41,337	56.37
AB106	Q_9	90	5	3	3,4,5	16.2	1,410	417	311	41,337	56.71
AB107				4	2,3,4,5	19.6	1,410	417	311	41,337	58.52
AB108				5	1,2,3,4,5	21.5	1,410	417	311	41,337	58.13
AB109				1	3	5.5	7	1	1	11	0.08
AB110	Q_{10}	100	3	2	2,3	10.9	541	249	139	781	1.05
AB111				3	1,2,3	13.8	541	249	139	781	0.96
AB112				1	4	7.2	10	1	1	35	0.76
AB113	Q_{10}	100	4	2	3,4	12.7	1,011	339	239	13,614	17.87
AB114				3	2,3,4	17.1	1,011	339	239	13,614	17.84
AB115				4	1,2,3,4	20.0	1,011	339	239	13,614	17.58
AB116				1	5	8.5	13	1	1	287	13.16
AB117				2	3,5	16.0	1,814	495	380	183,136	293.94
AB118	Q_{10}	100	5	3	3,4,5	22.2	1,814	495	380	183,136	298.43
AB119				4	2,3,4,5	25.6	1,814	495	380	183,136	291.37
AB120				5	1,2,3,4,5	27.5	1,814	495	380	183,136	293.73

Table 5: Numerical results of the deterministic global optimization method applied to instances AB001–AB120 of problem P1 (cont.).

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
AB001				1	2	1.4	4	2	0	0	0.00
AB002	Q_1	10	3	2	1,2	2.3	33	28	0	0	0.00
AB003				3	1,2,3	2.8	33	28	0	0	0.00
AB004				1	2	1.4	6	3	0	0	0.00
AB005	Q_1	10	4	2	1,2	2.3	42	34	0	0	0.00
AB006				3	1,2,3	2.8	42	34	0	0	0.00
AB007				4	1,2,3,4	2.0	42	34	0	0	0.00
AB008				1	2	1.4	10	4	2	8	6.53
AB009				2	1,2	2.3	74	53	9	12	9.03
AB010	Q_1	10	5	3	1,2,3	2.8	74	53	9	12	9.70
AB011				4	1,2,3,4	2.0	74	53	9	12	8.20
AB012				5	1,2,3,4,5	-0.5	74	53	9	12	8.20
AB013				1	3	1.5	3	1	1	0	0.00
AB014	Q_2	20	3	2	2,3	2.9	65	52	2	0	0.00
AB015				3	1,2,3	3.8	65	52	2	0	0.00
AB016				1	3	1.5	7	2	2	3	2.36
AB017	Q_2	20	4	2	2,3	2.9	116	74	16	9	7.70
AB018				3	2,3,4	4.1	116	74	16	9	7.62
AB019				4	1,2,3,4	5.0	116	74	16	9	7.78
AB020				1	2	2.4	14	4	4	3	2.10
AB021				2	2,3	3.9	141	85	20	21	13.74
AB022	Q_2	20	5	3	1,2,3	4.8	141	85	20	21	14.43
AB023				4	1,2,3,4	5.0	141	85	20	21	13.40
AB024				5	1,2,3,4,5	3.5	141	85	20	21	14.45
AB025				1	3	3.5	5	1	1	0	0.00
AB026	Q_3	30	3	2	2,3	5.9	119	74	15	10	8.27
AB027				3	1,2,3	7.8	119	74	15	10	8.68
AB028				1	3	2.5	9	2	2	15	11.43
AB029	Q_3	30	4	2	2,3	4.9	216	115	46	127	121.16
AB030				3	2,3,4	7.1	216	115	46	127	122.57
AB031				4	1,2,3,4	8.0	216	115	46	127	123.31
AB032				1	5	2.5	7	1	1	4	5.51
AB033				2	3,5	5.0	350	151	80	288	313.50
AB034	Q_3	30	5	3	2,3,5	7.4	350	151	80	288	320.39
AB035				4	1,2,3,5	9.3	350	151	80	288	312.91
AB036				5	1,2,3,4,5	9.5	350	151	80	288	321.95
AB037				1	3	3.5	5	1	1	1	1.22
AB038	Q_4	40	3	2	2,3	6.9	165	97	23	26	30.46
AB039				3	1,2,3	8.8	165	97	23	26	29.66
AB040				1	4	5.2	8	1	1	0	0.00
AB041	Q_4	40	4	2	3,4	7.7	297	144	64	189	234.73
AB042				3	1,3,4	9.6	297	144	64	189	227.60
AB043				4	1,2,3,4	11.0	297	144	64	189	233.56
AB044				1	5	3.5	8	1	1	8	10.40
AB045				2	3,5	7.0	571	219	130	1144	1313.65
AB046	Q_4	40	5	3	2,3,4	10.1	571	219	130	1144	1280.58
AB047				4	1,2,3,4	12.0	571	219	130	1144	1296.32
AB048				5	1,2,3,4,5	13.5	571	219	130	1144	1269.12
AB049				1	3	7.5	9	1	1	1	0.83
AB050	Q_5	50	3	2	2,3	9.9	292	139	55	700	762.71
AB051				3	1,2,3	11.8	292	139	55	700	749.34
AB052				1	4	5.2	8	1	1	35	33.84
AB053	Q_5	50	4	2	3,4	9.7	568	214	127	2524	2307.74
AB054				3	2,3,4	12.1	568	214	127	2524	2371.22
AB055				4	1,2,3,4	14.0	568	214	127	2524	2310.90
AB056				1	5	4.5	9	1	1	7	9.92
AB057				2	3,5	8.0	903	313	212	2977	3271.87
AB058	Q_5	50	5	3	2,3,5	11.4	903	313	212	2977	3377.17
AB059				4	2,3,4,5	14.6	903	313	212	2977	3275.01
AB060				5	1,2,3,4,5	16.5	903	313	212	2977	3392.49

Table 6: Numerical results of the deterministic global optimization method applied to instances AB001–AB120 of problem P3.

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
AB061				1	3	4.5	6	1	1	6	8.40
AB062	Q_6	60	3	2	2,3	7.9	357	179	81	145	171.71
AB063				3	1,2,3	9.8	357	179	81	145	166.30
AB064				1	4	6.2	9	1	1	34	57.23
AB065	Q_6	60	4	2	3,4	10.7	819	298	176	6827	7857.43
AB066				3	2,3,4	14.1	819	298	176	6827	7795.17
AB067				4	1,2,3,4	16.0	819	298	176	6827	7601.76
AB068				1	5	6.5	11	1	1	35	32.24
AB069				2	3,5	11.0	1308	417	286	22871	24766.92
AB070	Q_6	60	5	3	3,4,5	14.2	1308	417	286	22871	25462.20
AB071				4	2,3,4,5	16.6	1308	417	286	22871	26526.68
AB072				5	1,2,3,4,5	18.5	1308	417	286	22871	25217.38
AB073				1	3	5.5	7	1	1	3	3.86
AB074	Q_7	70	3	2	2,3	8.9	492	220	110	497	517.73
AB075				3	1,2,3	10.8	492	220	110	497	502.58
AB076				1	4	6.2	9	1	1	5	6.66
AB077	Q_7	70	4	2	3,4	11.7	786	296	179	1667	1853.19
AB078				3	2,3,4	16.1	786	296	179	1667	1888.11
AB079				4	1,2,3,4	19.0	786	296	179	1667	1865.49
AB080				1	5	7.5	12	1	1	14	12.18
AB081				2	4,5	12.7	1739	501	377	29086	36013.01
AB082	Q_7	70	5	3	3,4,5	17.2	1739	501	377	29086	36439.96
AB083				4	2,3,4,5	20.6	1739	501	377	29086	36264.07
AB084				5	1,2,3,4,5	23.5	1739	501	377	29086	38871.69
AB085				1	3	5.5	7	1	0	1	0.03
AB086	Q_7	80	3	2	3,2	8.9	500	229	110	245	221.65
AB087				3	3,2,1	11.8	500	229	110	245	251.58
AB088				1	4	8.2	11	1	0	18	26.42
AB089	Q_7	80	4	2	4,3	13.7	1491	459	312	19438	22026.24
AB090				3	4,3,2	17.1	1491	459	312	19438	22167.69
AB091				4	4,3,2,1	19.0	1491	459	312	19438	22824.37
AB092				1	5	6.5	11	1	1	691	721.11
AB093				2	–	–	–	–	–	–	–
AB094	Q_7	80	5	3	–	–	–	–	–	–	–
AB095				4	–	–	–	–	–	–	–
AB096				5	–	–	–	–	–	–	–
AB097				1	3	5.5	7	1	0	9	5.34
AB098	Q_7	90	3	2	3,2	9.9	738	319	179	1022	1055.14
AB099				3	3,2,1	11.8	738	319	179	1022	1043.77
AB100				1	4	7.2	10	1	0	3	2.45
AB101	Q_7	90	4	2	4,3	12.7	1417	489	350	7173	8284.02
AB102				3	4,3,2	16.1	1417	489	350	7173	8154.56
AB103				4	4,3,2,1	19.0	1417	489	350	7173	8122.51
AB104				1	5	10.5	15	1	1	72	120.50
AB105				2	–	–	–	–	–	–	–
AB106	Q_7	90	5	3	–	–	–	–	–	–	–
AB107				4	–	–	–	–	–	–	–
AB108				5	–	–	–	–	–	–	–
AB109				1	3	7.5	9	1	1	4	4.36
AB110	Q_7	100	3	2	3,2	12.9	842	363	214	1721	1789.56
AB111				3	3,2,1	15.8	842	363	214	1721	1748.79
AB112				1	4	8.2	11	1	1	270	366.45
AB113	Q_7	100	4	2	–	–	–	–	–	–	–
AB114				3	–	–	–	–	–	–	–
AB115				4	–	–	–	–	–	–	–
AB116				1	5	9.5	14	1	1	7316	6725.64
AB117				2	–	–	–	–	–	–	–
AB118	Q_7	100	5	3	–	–	–	–	–	–	–
AB119				4	–	–	–	–	–	–	–
AB120				5	–	–	–	–	–	–	–

Table 7: Numerical results of the deterministic global optimization method applied to instances AB001–AB120 of problem P3 (cont.).

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
AB001				1	2	1.4	4	2	0	0	0.00
AB002	Q_1	10	3	2	1,2	2.3	33	28	0	0	0.00
AB003				3	1,2,3	2.8	33	28	0	0	0.00
AB004				1	2	1.4	6	3	0	0	0.00
AB005	Q_1	10	4	2	1,2	2.3	42	34	0	0	0.00
AB006				3	1,2,3	2.8	42	34	0	0	0.00
AB007				4	1,2,3,4	2.0	42	34	0	0	0.00
AB008				1	2	1.4	10	4	2	8	0.01
AB009				2	1,2	2.3	74	53	11	10	0.01
AB010	Q_1	10	5	3	1,2,3	2.8	74	53	11	10	0.01
AB011				4	1,2,3,4	2.0	74	53	11	10	0.01
AB012				5	1,2,3,4,5	-0.5	74	53	11	10	0.01
AB013				1	3	1.5	3	1	1	0	0.00
AB014	Q_2	20	3	2	2,3	2.9	65	52	2	0	0.00
AB015				3	1,2,3	3.8	65	52	2	0	0.00
AB016				1	3	1.5	7	2	2	3	0.00
AB017	Q_2	20	4	2	2,3	2.9	116	75	15	10	0.01
AB018				3	2,3,4	4.1	116	75	15	10	0.02
AB019				4	1,2,3,4	5.0	116	75	15	10	0.01
AB020				1	2	2.4	14	4	4	3	0.00
AB021				2	2,3	3.9	141	85	23	18	0.03
AB022	Q_2	20	5	3	1,2,3	4.8	141	85	23	18	0.03
AB023				4	1,2,3,4	5.0	141	85	23	18	0.02
AB024				5	1,2,3,4,5	3.5	141	85	23	18	0.03
AB025				1	3	3.5	5	1	1	0	0.00
AB026	Q_3	30	3	2	2,3	5.9	119	74	15	10	0.02
AB027				3	1,2,3	7.8	119	74	15	10	0.02
AB028				1	3	2.5	9	2	2	15	0.04
AB029	Q_3	30	4	2	2,3	4.9	216	115	48	125	0.27
AB030				3	2,3,4	7.1	216	115	48	125	0.29
AB031				4	1,2,3,4	8.0	216	115	48	125	0.29
AB032				1	5	2.5	7	1	1	4	0.01
AB033				2	3,5	5.0	353	154	86	288	0.64
AB034	Q_3	30	5	3	2,3,5	7.4	353	154	86	288	0.65
AB035				4	1,2,3,5	9.3	353	154	86	288	0.63
AB036				5	1,2,3,4,5	9.5	353	154	86	288	0.66
AB037				1	3	3.5	5	1	1	1	0.00
AB038	Q_4	40	3	2	2,3	6.9	165	97	23	26	0.04
AB039				3	1,2,3	8.8	165	97	23	26	0.06
AB040				1	4	5.2	8	1	1	0	0.00
AB041	Q_4	40	4	2	3,4	7.7	297	144	64	189	0.46
AB042				3	1,3,4	9.6	297	144	64	189	0.40
AB043				4	1,2,3,4	11.0	297	144	64	189	0.44
AB044				1	5	3.5	8	1	1	8	0.03
AB045				2	3,5	7.0	566	217	141	1,136	2.79
AB046	Q_4	40	5	3	2,3,4	10.1	566	217	141	1,136	2.73
AB047				4	1,2,3,4	12.0	566	217	141	1,136	2.66
AB048				5	1,2,3,4,5	13.5	566	217	141	1,136	2.76
AB049				1	3	7.5	9	1	1	1	0.02
AB050	Q_5	50	3	2	2,3	9.9	289	137	53	703	1.58
AB051				3	1,2,3	11.8	289	137	53	703	1.65
AB052				1	4	5.2	8	1	1	35	0.14
AB053	Q_5	50	4	2	3,4	9.7	576	216	141	2,522	6.10
AB054				3	2,3,4	12.1	576	216	141	2,522	5.67
AB055				4	1,2,3,4	14.0	576	216	141	2,522	5.96
AB056				1	5	4.5	9	1	1	7	0.08
AB057				2	3,5	8.0	909	316	231	2,994	7.27
AB058	Q_5	50	5	3	2,3,5	11.4	909	316	231	2,994	7.04
AB059				4	2,3,4,5	14.6	909	316	231	2,994	7.33
AB060				5	1,2,3,4,5	16.5	909	316	231	2,994	7.15

Table 8: Numerical results of the heuristic or stochastic global optimization method applied to instances AB001–AB120 of problem P3.

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
AB061				1	3	4.5	6	1	1	6	0.02
AB062	Q_6	60	3	2	2,3	7.9	357	179	81	145	0.33
AB063				3	1,2,3	9.8	357	179	81	145	0.34
AB064				1	4	6.2	9	1	1	34	0.36
AB065	Q_6	60	4	2	3,4	10.7	828	306	206	6,853	18.96
AB066				3	2,3,4	14.1	828	306	206	6,853	16.82
AB067				4	1,2,3,4	16.0	828	306	206	6,853	18.00
AB068				1	5	6.5	11	1	1	35	1.35
AB069				2	3,5	11.0	1342	425	325	22,930	63.88
AB070	Q_6	60	5	3	3,4,5	14.2	1342	425	325	22,930	59.98
AB071				4	2,3,4,5	16.6	1342	425	325	22,930	61.59
AB072				5	1,2,3,4,5	18.5	1342	425	325	22,930	65.22
AB073				1	3	5.5	7	1	1	3	0.03
AB074	Q_7	70	3	2	2,3	8.9	502	224	120	498	0.91
AB075				3	1,2,3	10.8	502	224	120	498	1.10
AB076				1	4	6.2	9	1	1	5	0.07
AB077	Q_7	70	4	2	3,4	11.7	788	297	195	1,656	4.07
AB078				3	2,3,4	16.1	788	297	195	1,656	3.81
AB079				4	1,2,3,4	19.0	788	297	195	1,656	4.08
AB080				1	5	7.5	12	1	1	14	0.81
AB081				2	4,5	12.7	1,859	542	445	29,228	81.52
AB082	Q_7	70	5	3	3,4,5	17.2	1,859	542	445	29,228	83.33
AB083				4	2,3,4,5	20.6	1,859	542	445	29,228	82.76
AB084				5	1,2,3,4,5	23.5	1,859	542	445	29,228	82.19
AB085				1	3	5.5	7	1	1	0	0.03
AB086	Q_8	80	3	2	2,3	8.9	501	231	120	239	0.57
AB087				3	1,2,3	11.8	501	231	120	239	0.58
AB088				1	4	8.2	11	1	1	17	1.21
AB089	Q_8	80	4	2	3,4	13.7	1,540	479	365	19,665	50.17
AB090				3	2,3,4	17.1	1,540	479	365	19,665	50.46
AB091				4	1,2,3,4	19.0	1,540	479	365	19,665	50.03
AB092				1	5	6.5	11	1	1	691	7.71
AB093				2	4,5	12.7	2,689	786	654	100,394	298.73
AB094	Q_8	80	5	3	3,4,5	18.2	2,689	786	654	100,394	293.76
AB095				4	2,3,4,5	22.6	2,689	786	654	100,394	299.28
AB096				5	1,2,3,4,5	23.5	2,689	786	654	100,394	294.53
AB097				1	3	5.5	7	1	1	8	0.06
AB098	Q_9	90	3	2	2,3	9.9	733	318	188	1,018	2.30
AB099				3	1,2,3	11.8	733	318	188	1,018	2.21
AB100				1	4	7.2	10	1	1	7	0.57
AB101	Q_9	90	4	2	3,4	12.7	1,471	506	392	7,326	18.63
AB102				3	2,3,4	16.1	1,471	506	392	7,326	18.55
AB103				4	1,2,3,4	19.0	1,471	506	392	7,326	19.70
AB104				1	5	10.5	15	1	1	72	26.53
AB105				2	4,5	15.7	4,182	1,032	918	396,579	1,196.00
AB106	Q_9	90	5	3	3,4,5	20.2	4,182	1,032	918	396,579	1,172.51
AB107				4	2,3,4,5	23.6	4,182	1,032	918	396,579	1,189.70
AB108				5	1,2,3,4,5	25.5	4,182	1,032	918	396,579	1,183.69
AB109				1	3	7.5	9	1	1	4	0.16
AB110	Q_{10}	100	3	2	2,3	12.9	849	366	225	1,728	4.19
AB111				3	1,2,3	15.8	849	366	225	1,728	4.18
AB112				1	4	8.2	11	1	1	270	3.94
AB113	Q_{10}	100	4	2	3,4	14.7	2,257	648	532	63,504	166.33
AB114				3	2,3,4	19.1	2,257	648	532	63,504	163.69
AB115				4	1,2,3,4	22.0	2,257	648	532	63,504	165.36
AB116				1	5	9.5	14	1	1	7316	189.01
AB117				2	4,5	17.7	6,221	1,494	1,352	1,763,651	5,476.20
AB118	Q_{10}	100	5	3	5,4,3	25.2	6,221	1,494	1,352	1,763,651	5,493.64
AB119				4	2,3,4,5	29.6	6,221	1,494	1,352	1,763,651	5,433.90
AB120				5	1,2,3,4,5	31.5	6,221	1,494	1,352	1,763,651	5,442.50

Table 9: Numerical results of the heuristic or stochastic global optimization method applied to instances AB001–AB120 of problem P3 (cont.).

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
AB007	Q_1	10	4	4	1,2,3	1.8	40	36	0	2	0.00
AB011	Q_1	10	5	4	1,2,3	1.8	60	48	11	14	0.00
AB012				5	1,2,3	1.8	60	48	11	14	0.00
AB023	Q_2	20	5	4	1,2,3	4.8	112	75	20	20	0.01
AB024				5	1,2,3	4.8	112	75	20	20	0.02

Table 10: Numerical results of the deterministic global optimization method applied to selected instances within AB001–AB120 of problem P2.

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
AB007	Q_1	10	4	4	1,2,3	2.8	42	34	0	0	0.00
AB011	Q_1	10	5	4	1,2,3	2.8	74	53	9	12	8.15
AB012				5	1,2,3	2.8	74	53	9	12	8.14
AB024	Q_2	20	5	5	1,2,3,4	5.0	141	85	20	21	11.02

Table 11: Numerical results of the deterministic global optimization method applied to selected instances within AB001–AB120 of problem P4.

Problem					Solution		Performance metrics				
Name	Points	n	m	k	Selected ellipses	Optimal income	Trees' figures # nodes	Trees' figures # leaves	# NLP subproblems feasible	# NLP subproblems infeasible	CPU Time in secs.
AB007	Q_1	10	4	4	1,2,3	2.8	42	34	0	0	0.00
AB011	Q_1	10	5	4	1,2,3	2.8	74	53	11	10	0.01
AB012				5	1,2,3	2.8	74	53	11	10	0.01
AB024	Q_2	20	5	5	1,2,3,4	5.0	141	85	23	18	0.02

Table 12: Numerical results of the heuristic or stochastic global optimization method applied to selected instances within AB001–AB120 of problem P4.

instances would be used to benchmark other heuristic methods in the future. Codes and instances are fully available for benchmarking purposes.

Developing an efficient deterministic global optimization technique for the situation in which the ellipses' semi major and minor axes are variables of the problem, the ellipse cost being a function of their axes, like, for example, the ellipse's area, would be the subject of future research.

References

[1] C. S. Adjiman, I. P. Androulakis, C. D. Maranas and C. A. Floudas, A Global Optimization Method αBB for Process Design, *Computers and Chemical Engineering* 20, pp. S419–424, 1996.

[2] C. S. Adjiman, S. Dallwig, C. A. Floudas and A. Neumaier, A global optimization method, αBB, for general twice-differentiable constrained NLPs – I. Theoretical Advances, *Computers & Chemical Engineering* 22, pp. 1137–1158, 1998.

[3] C. S. Adjiman, I. P. Androulakis and C. A. Floudas, A global optimization method, αBB, for general twice-differentiable constrained NLPs – II. Implementation and computational results, *Computers & Chemical Engineering* 22, pp. 1159–1179, 1998.

[4] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification, *Mathematical Programming* 111, pp. 5–32, 2008.

[5] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, On Augmented Lagrangian methods with general lower-level constraints, *SIAM Journal on Optimization* 18, pp. 1286–1309, 2007.

[6] M. Andretta, E. G. Birgin, and J. M. Martínez, Practical active-set Euclidian trust-region method with spectral projected gradients for bound-constrained minimization, *Optimization* 54, pp. 305–325, 2005.

[7] I. P. Androulakis, C. D. Maranas and C. A. Floudas, αBB: A Global Optimization Method for General Constrained Nonconvex Problems, *Journal of Global Optimization* 7, pp. 337–363, 1995.

[8] E. G. Birgin, C. A. Floudas, and J. M. Martínez, Global minimization using an Augmented Lagrangian method with variable lower-level constraints, *Mathematical Programming* 125, pp. 139–162, 2010.

[9] E. G. Birgin and J. M. Martínez, A box-constrained optimization algorithm with negative curvature directions and spectral projected gradients, *Computing [Suppl]* 15, pp. 49–60, 2001.

[10] E. G. Birgin and J. M. Martínez, Large-scale active-set box-constrained optimization method with spectral projected gradients, *Computational Optimization and Applications* 23, pp. 101–125, 2002.

[11] M. S. Canbolat and M. von Massow, Planar maximal covering with ellipses, *Computers and Industrial Engineering* 57, pp. 201–208, 2009.

[12] S. Davari, M. H. F. Zarandi, and A. Hemmati, Maximal covering location problem (MCLP) with fuzzy travel times, *Expert Systems with Applications* 38, pp. 14535-14541, 2011.

[13] GAMS Development Corp., *GAMS - The Solver Manuals*, Washington DC, 2009.

[14] M. Tawarmalani and N. V. Sahinidis, A polyhedral branch-and-cut approach to global optimization, *Mathematical Programming* 103, pp. 225–249, 2005.

[15] *http://www.neos-server.org/neos/*