Method of Sentinels for Packing Items within Arbitrary Convex Regions

E. G. Birgin [*] J. M. Martínez [†] W. F. Mascarenhas [‡] D. P. Ronconi [§]

June 23, 2005

Abstract

A new method is introduced for packing items in convex regions of the Euclidian n-dimensional space. By means of this approach the packing problem becomes a global finite-dimensional continuous optimization problem. The strategy is based on the new concept of *sentinels*. Sentinels sets are finite subsets of the items to be packed such that, when two items are superposed, at least one sentinel of one item is in the interior of the other. Minimal sets of sentinels are found in simple $2-$dimensional cases. Numerical experiments and pictures showing the potentiality of the new technique are presented.

Keywords: Sentinels, packing problems, cutting problems, models, nonlinear programming.

1 Introduction

The problem of packing a given set of pieces into defined regions maximizing the total number of pieces or the used area occurs in a large range of practical situations, including manufacturer's pallet loading, packing of ship containers and establishing of layout in clothing industry. Many papers have been published dealing with packing problems. A useful classification has been given by (Dyckhoff, 1990). One of the most popular and useful problem in this area is to find the maximum number of rectangles that can be orthogonally packed into a larger rectangle. Polynomial algorithms for the guillotine version of the problem exist (Tarnowski *et al*, 1994)

[*] Department of Computer Science, IME, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. This author was supported by PRONEX-CNPq/FAPERJ E-26/171.164/2003 - APQ1, CNPq (PROSUL 490333/2004-4) and FAPESP (Grants 03/09169-6 and 01/04597-4). Corresponding author. FAX: +55(11)3091-6134. e-mail: egbirgin@ime.usp.br

[†] Department of Applied Mathematics, IMECC, University of Campinas, CP 6065, 13081-970 Campinas, SP, Brazil. This author was supported by PRONEX-CNPq/FAPERJ E-26/171.164/2003 - APQ1, FAPESP (Grant 01/04597-4), CNPq and FAEP-UNICAMP. e-mail: martinez@ime.unicamp.br

[‡] Department of Computer Science, IME, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. e-mail: walterfm@ime.usp.br

[§] Department of Production Engineering, EP, University of São Paulo, Av. Prof. Almeida Prado, 128, Cidade Universitária, 05508-900, São Paulo, SP, Brazil. This author was supported by CNPq (PROSUL 490333/2004-4) and FAPESP (Grant 01/02972-2). e-mail: dronconi@usp.br

whereas the NP-completeness of the non-guillotine problem has been conjectured (Dowsland, 1987; Morabito and Morales, 1998). In (Lins *et al*, 2003; Birgin *et al*, 2005^c) a very efficient heuristic to solve this problem was introduced. The authors conjectured that their method always finds an optimal solution and solved hard instances that remained unsolved by other heuristics.

Several mixed 0-1 integer linear and nonlinear programming models have been proposed for 2D and 3D packing problems in which identical or non-identical boxes must be packed into a bigger box. In general, the packed boxes can be rotated, but always with its sides parallel to the sides of the bigger box. Sometimes constraints are imposed related to the number of each type of box being packed. In general, the proposed models are difficult. When an exact method is used, large-scale problems become impossible to solve. Fortunately, heuristic methods find optimal or good quality solutions in many cases. See, for example, (Beasley, 1995; Beasley 2004; Chen *et al*, 1995; Christofides and Hadjiconstantinou, 1995; Hadjiconstantinou and Christofides 1995; Tsai *et al*, 1993).

In (Beasley, 2004) a nonlinear formulation for the constrained two-dimensional non-guillotine cutting problem is presented and the model is used for the elaboration of a populational heuristic. The proposed model has two main difficulties: (i) the presence of integer variables; and (ii) the non-differentiability of the nonlinear constraints used to avoid overlapping. The first difficulty may be circumvented by replacing constraints of type $x \in \{0,1\}$ by $x(1-x)=0$. However, this kind of constraints leads to hard-to-solve nonlinear problems with many local-nonglobal solutions. Several strategies for fixing integer variables and solving the problem as a sequence of continuous nonlinear problems have also been proposed. The second inconvenience of the formulation can be overcome by approximating the nonsmooth constraints by smooth ones. These two ideas lead to nonlinear formulations that are suitable for classical nonlinear optimization solvers.

Several works related to packing irregular polygons have been also published. Most of them are based on placement policies previously defined (for example, bottom-left) and aim to find optimal solutions related to the pre-established policy. In (Bennell, 2001) an easy way to compute a placement policy known as "non-fit polygon" has been proposed. This idea is used in (Dowsland *et al*, 2002), among other strategies, to develop a bottom-left placement algorithm for polygon packing. See also (Amaral *et al*, 1990; Dowsland *et al*, 1998; Lamousin and Waggenspack Jr., 1997; Li and Milenkovic, 1995).

The Method of Sentinels introduced in this paper is a procedure for packing items inside a convex region without overlapping. The main idea is to define a set of points called sentinels for each small item in such a way that two items are superposed if and only if at least a sentinel of one of the items is in the interior of the other item. Based on the sentinels sets, a smooth nonlinear decision model to determine if a fixed set of items can be packed inside the convex region without overlapping is defined. The variables of the model are the ones that define a displacement in the Euclidian space. In 2D problems, the displacement is defined by a vector of translation and an angle of rotation. So, the method is not restricted to orthogonal (parallel to the axes) patterns. The new concept of sentinels introduces some difficult theoretical problems related to identification and minimality. Here we present minimal sets of sentinels for the case of identical rectangles and regular polygons. More general items, convex or non-convex are also considered. The non-linear decision model is solved using the well-established bound-constrained solver GENCAN (Birgin and Martínez, 2002).

This paper is organized as follows. In Section 2 the Method of Sentinels for packing items on arbitrary regions is introduced. In Section 3 we define an optimal set of sentinels for rectangular items and regular polygons. In Section 4 we present numerical experiments using the Method of Sentinels for packing rectangular items on arbitrary convex regions. In Section 5 we state conclusions and lines for future research.

2 Using Sentinels for Packing

We begin with an informal and general description of the Method of Sentinels. Let A_1 and A_2 be nonempty, open, bounded and convex sets of $I\!R^n$. Sets with these characteristics will be called *items* from now onwards. Define $B_1 = \bar{A}_1$ the closure of A_1 and $B_2 = \bar{A}_2$ the closure of A_2. Let $D_1, D_2 : I\!R^n \to I\!R^n$ be two displacement operators. So, D_1 and D_2 transform items in items preserving distances, angles and orientation. If $D_1(A_1) \cap D_2(A_2) \neq \emptyset$ then we say that $D_1(B_1)$ and $D_2(B_2)$ (or $D_1(A_1)$ and $D_2(A_2)$) are *superposed*.

Let S_1 and S_2 be finite subsets of B_1 and B_2 respectively. We say that S_1 and S_2 are *sentinels sets* relatively to B_1 and B_2 if the following property holds:

For all displacements D_1, D_2, if $D_1(B_1)$ and $D_2(B_2)$ are superposed, then

$$D_1(S_1) \cap D_2(A_2) \neq \emptyset \textit{ or } D_2(S_2) \cap D_1(A_1) \neq \emptyset. \tag{1}$$

Roughly speaking, if, after the displacements, the items B_1 and B_2 are superposed, then at least one sentinel of B_1 becomes interior to B_2 or one sentinel of B_2 becomes interior to B_1.

Now, assume that $B_1, \ldots, B_m \subset I\!R^n$, $\Omega \subset I\!R^n$ and we want to *pack* the items $B_1, \ldots, B_m$ into the region Ω. This means that we want to find displacements $D_1, \ldots, D_m$ such that

$$D_j(B_j) \subset \Omega \; \forall \, j = 1, \ldots, m \tag{2}$$

and

$$D_i(B_i) \text{ and } D_j(B_j) \text{ are not superposed } \; \forall \, i, j = 1, \ldots, m, \; i \neq j. \tag{3}$$

Assume that $S_1 \subset B_1, \ldots, S_m \subset B_m$ are such that S_i and S_j are *sentinels sets* relatively to B_i and B_j. For all $i, j = 1, \ldots, m, \; i \neq j$, define

$$\kappa(D_i, D_j) = \#\{[D_i(S_i) \cap D_j(A_j)] \cup [D_j(S_j) \cap D_i(A_i)]\}. \tag{4}$$

Then, condition (3) can be formulated as follows:

$$\kappa(D_i, D_j) = 0 \; \forall \, i, j = 1, \ldots, m, \; i \neq j. \tag{5}$$

So, the packing problem defined by (2)-(3) is related to the optimization problem

$$\text{Minimize } \sum_{i \neq j} \kappa(D_i, D_j) \;\; \text{s. t. } \; D_k(B_k) \subset \Omega \; \forall \, k = 1, \ldots, m. \tag{6}$$

If a global solution of (6) is found such that the objective function value vanishes, then the packing problem (2)-(3) is solved. The objective function of (6) represents the total number of sentinels of one item that, after the displacements, fall in the interior of some other item.

The optimization problem (6) defines the Method of Sentinels. However, this minimization problem needs to be reformulated in order to transform it into a solvable nonlinear programming problem. Let us consider the case in which Ω is a closed and convex set defined by a set of inequalities. So,

$$\Omega = \{x \in I\!\!R^n \mid g_i(x) \leq 0,\ i = 1, \ldots, p\}. \tag{7}$$

Moreover, assume that each item B_k is a bounded polytope, so it is the convex hull of its vertices $V_1(B_k), \ldots, V_{\nu(k)}(B_k)$. Then, the constraints of (6) take the form

$$g_i(D_k[V_\ell(B_k)]) \leq 0 \ \forall \ \ell = 1, \ldots, \nu(k), k = 1, \ldots, m, i = 1, \ldots, p. \tag{8}$$

The displacements D_k can always be described by a finite set of parameters. For example, displacements in $I\!\!R^2$ are given by three parameters, the first two representing a translation and the third the angle of rotation. Displacements in $I\!\!R^3$ are given by three translation parameters and two angles of rotation, and so on. Therefore, the constraints (8) have the usual form adopted in nonlinear programming problems.

Let us analyze now the objective function of (6). This function depends on the continuous variables that define the displacements but it takes only discrete integer nonnegative values. For nonlinear programming reformulations we need to replace it by a continuous function of the displacement variables. As before, we restrict ourselves to the case in which the sets B_k are bounded polytopes. In this case, B_k is described by a set of linear inequalities:

$$\langle c_{k,j}, x\rangle \leq b_{k,j},\ j = 1, \ldots, \mu(k). \tag{9}$$

If s is a sentinel of B_i and $D_i(s)$ is in $D_k(A_k)$ with $k \neq i$, then $D_k^{-1} D_i(s)$ belongs to A_k and, so, it satisfies:

$$\langle c_{k,j}, D_k^{-1} D_i(s)\rangle < b_{k,j},\ j = 1, \ldots, \mu(k). \tag{10}$$

Thus, the displaced sentinel s belongs to the displaced A_k if, and only if,

$$\prod_{j=1}^{\mu(k)} \max\{0, b_{k,j} - \langle c_{k,j}, D_k^{-1} D_i(s)\rangle\} > 0. \tag{11}$$

Therefore, a degree of the superposition of B_i and B_k under the displacements D_i and D_k is given by

$$\Phi(D_i, D_k) = \tag{12}$$

$$\sum_{s \in S_i} \prod_{j=1}^{\mu(k)} [\max\{0, b_{k,j} - \langle c_{k,j}, D_k^{-1} D_i(s)\rangle\}]^2 + \sum_{s \in S_k} \prod_{j=1}^{\mu(i)} [\max\{0, b_{i,j} - \langle c_{i,j}, D_i^{-1} D_k(s)\rangle\}]^2.$$

The function $\Phi(D_i, D_k)$ is nonnegative and continuously differentiable with respect to the parameters that define the displacements D_i and D_k and it vanishes if, and only if, $D_i(B_i)$ and $D_k(B_k)$ are not superposed. Therefore, it can replace the function κ in the optimization problem (6).

Summing up, in the case in which Ω is a convex set defined by inequalities and the items are bounded polytopes, the packing problem can be formulated as the following nonlinear programming problem:

$$\text{Minimize } \sum_{i \neq j} \Phi(D_i, D_j) \tag{13}$$

subject to (8). Moreover, since we are only interested in global solutions of (13) where the objective function must vanish, the problem can be reformulated as the feasibility problem given below:

$$\sum_{i \neq j} \Phi(D_i, D_j) = 0, \tag{14}$$

$$g_i(D_k[V_\ell(B_k)]) \leq 0 \; \forall \; \ell = 1, \ldots, \nu(k), k = 1, \ldots, m, i = 1, \ldots, p. \tag{15}$$

Finally, (14)-(15) is equivalent to the following unconstrained continuously differentiable global optimization problem:

$$\text{Minimize } \sum_{i \neq j} \Phi(D_i, D_j) + \sum_{i=1}^{p} \sum_{k=1}^{m} \sum_{\ell=1}^{\nu(k)} [\max\{0, g_i(D_k[V_\ell(B_k)])\}]^2. \tag{16}$$

3 Sentinels for Planar Polygons

In this section we restrict ourselves to 2D packing problems of polygons. We present examples of polygons for which any set of sentinels is infinite and characterize a class of polygons that have finite sets of sentinels. We explain how to define sentinels for polygons in this last class and also provide optimal sentinels sets for rectangles and regular polygons. The results in this sections are based on a mathematical theory described in (Birgin *et al*, 2004). For the sake of brevity, here we explain only the intuitive aspects of (Birgin *et al*, 2004) and ask the interested reader to look at this reference for a more rigorous treatment of this subject.

In this section we work with *polygons*, by which we mean a sequence of points $p_0, p_1, \ldots, p_{n-1}$ connected by segments $p_i p_{i+1}$ that do not cross. The only contact the segments have is at the vertices p_i. We also assume that no three consecutive points p_i are aligned. According to this definition, polygons are not open sets. However, in order to simplify the language, we will say that $S_1, \ldots, S_n$ are sets of sentinels for a family of polygons $P_1, \ldots, P_n$ even when, to be one hundred percent rigorous, we should say that $S_1, \ldots, S_n$ are sets of sentinels for *the interior* of $P_1, \ldots, P_n$.

We are now ready to state a negative result: triangles do not have finite sentinels sets. In fact, any two polygons with an internal angle smaller than $\pi/2$ that could be superposed like in Figure 1, with contact only at the acute tips, cause problems. In this case, one can place any finite number of sentinels near the touching tips and it will always be possible to move the polygons slightly and have them to intersect in such way that one polygon does not touch the sentinels of the other.

Figure 1 by here.

On the other hand, this is as bad as things may get, as stated in the next theorem:

Theorem 1 *If $P_1, \ldots, P_n$ is a family of n planar polygons such that all internal angles of all P_i's are bigger than or equal to $\pi/2$ then there exist a family of finite sets of sentinels $S_1, \ldots, S_n$ for $P_1, \ldots, P_n$.*

The sentinels sets used to prove this theorem are built in two steps. First we use the next two lemmas to reduce the problem to the convex case:

Lemma 1 *Every polygon with internal angles bigger than or equal to $\pi/2$ can be decomposed as the union of convex polygons with internal angles bigger than or equal to $\pi/2$.*

Lemma 2 *Let $A_1, \ldots, A_n$ be a family of open sets such that each A_i is decomposed as $A_i = \cup_{j=1}^{n_i} A_{ij}$, for some open sets A_{ij}. Consider the set of n-uples $K = \{(k_1, \ldots, k_n)$ with $1 \leq k_i \leq n_i$ for $i = 1, \ldots, n\}$. If $S_{1k_1}, S_{2k_2}, \ldots, S_{nk_n}$ are sentinels sets for $A_{1k_1}, A_{2k_2}, \ldots, A_{nk_n}$ for all $(k_1, \ldots, k_n) \in K$ then the sets $S_i = \cup_{j=1}^{n_i} S_{ij}$ are sentinels sets for $A_1, \ldots, A_n$.*

In words, Lemma 2 states that we can obtain sentinels sets for the whole sets by combining the sentinels sets for their parts. As a consequence of the last two lemmas, we care mostly about convex polygons and the next lemma is of interest:

Lemma 3 *If $P_1, \ldots, P_n$, $n \geq 2$, is a family of strictly convex polygons and $S_1, \ldots, S_n$ is a family of finite sets of sentinels for the P_i's then each S_i contains all the vertices of P_i.*

Thus, vertices are natural candidates to sentinels. Moreover, in the case when some P_i are repeated we must have internal sentinels. Otherwise we could just put one copy right above the other. This observation proves the following lemma.

Lemma 4 *Let $A_1, \ldots, A_n$ be a family of open sets and $S_1, \ldots, S_n$ be a family of finite sets of sentinels for the A_i's. If $A_i = A_j$ for $i \neq j$ then $(S_i \cap A_i) \cup (S_j \cap A_j) \neq \emptyset$.*

We are now ready to explain the construction used to prove Theorem 1. First we decompose each polygon as the union of convex parts with internal angles bigger than or equal to $\pi/2$. We take the vertices of such parts as sentinels. Next, for each convex part $C = c_0 c_1 \ldots c_k$, $c_k = c_0$, we take an arbitrary internal sentinel in the set

$$I = \bigcap_{i=1}^{k} C_i,$$

where C_i is the convex polygon with vertices c_j with $j \neq i$. Each C_i is the convex set obtained from C by the removal of the ear $c_{i-1} c_i c_{i+1}$ and their intersection is illustrated in Figure 2.

Figure 2 by here.

Since each C has at least four vertices, Helly's theorem implies that I is not empty, because the intersection of any three C_i's, for $i \in \{i_1, i_2, i_3\}$, contains at least a vertex c_j for some

$j \notin \{i_1, i_2, i_3\}$. Therefore, we can always pick an internal sentinel for C as described above. Finally we take a small δ and populate the sides of the parts with sentinels which are at most δ apart, as in Figure 5.

Figure 3 by here.

If no polygon has parallel sides which are close to each other then we can take δ as the length of the smallest side among all the polygons. In cases similar to the rectangle, when there are two consecutive internal angles of $\pi/2$, we must take δ slightly smaller than the side connecting the corresponding vertices. In (Birgin *et al*, 2004) we provide a precise value for δ and a formal justification of this procedure.

Lemmas 3 and 4 show that the sets of sentinels we propose are optimal for families of regular polygons with sides of the same length and with more than four vertices. However, rectangles are different. Since they are so important in practice we devote the last part of this section to them. Using the theory developed in (Birgin *et al*, 2004), we found the following optimal sentinels sets when all the P_i's equal the rectangle R_λ with vertices $(\pm\lambda, \pm 1)$ and $\lambda \geq 1$.

Figure 4 by here.

For technical reasons, we must look at two ranges for λ: $\lambda < 4$ and $\lambda \geq 4$, and the horizontal distance δ^λ among the points is at most

$$\delta_\lambda = \lambda/2 \qquad \text{if } 1 \leq \lambda < 4, \tag{17}$$

$$\delta_\lambda = \frac{2\lambda}{\lfloor\lambda\rfloor + 1} \text{ if } \lambda \geq 4. \tag{18}$$

The optimal sentinels set S_λ is defined in terms of the auxiliary sets M_λ and B_λ as follows

$$B_\lambda = \{(-\lambda,-1),(-\lambda,0),(-\lambda,1),(\lambda,-1),(\lambda,0),(\lambda,1)\}, \tag{19}$$

$$n_\lambda = 1 \text{ for } \lambda < 4 \text{ and } n_\lambda = \lfloor\lambda\rfloor - 2 \text{ otherwise}, \tag{20}$$

$$M_\lambda = \{(k\delta_\lambda - \lambda, 0), \text{ for } 2 \leq k \leq n_\lambda + 1\}, \tag{21}$$

$$S_\lambda = B_\lambda \cup M_\lambda \cup \{(-\lambda + \delta_\lambda, 1), (\lambda - \delta_\lambda, -1)\}. \tag{22}$$

In (Birgin *et al*, 2004) we give a formal proof that these configurations S_λ are indeed sentinels sets and that they are minimal. A reader that is not interested in these proofs could still get a heuristic verification of the validity of S_λ by copying either of the rectangles in Figure 4 in two pieces of paper and trying to rotate and translate these copies in such a way that they overlap and no sentinel of a rectangle falls inside of the other. The reader will soon find out that this is impossible.

4 Numerical results

We solved 2D problems of packing identical rectangles within arbitrary convex regions using the Method of Sentinels (equivalent to the unconstrained continuously differentiable global optimization problem (16)). The unknowns of (16) are the translation vectors and the rotation angles for each rectangle. We use as sentinels the points of S_λ given by (22).

4.1 Basic optimization procedure

Given a convex set Ω and an unlimited number of identical rectangles, the problem of packing as many rectangles as possible is equivalent to finding a null-cost global solution of problem (16) for the biggest possible value of m. Most nonlinear optimization methods find first-order stationary points (very likely, local minimizers) of problem (16). An alternative to enhance the probability of finding a global minimizer is to start the method from many random initial guesses.

In practice, we proceed as follows. First, we establish the maximum effort we are able to do (defining, for example, a maximum CPU-time T). If the allowed computer time is exhausted, the algorithm stops returning the best packing found. In the algorithmic context defined below, this means that $m \equiv r - 1$ rectangles were packed. On the other hand, while our maximum effort is not exhausted, we carry out the following steps:

Step 1: Set $r \leftarrow m_{\rm lb}$, where $m_{\rm lb}$ is a known lower bound on the number of rectangles that can be packed (note that $m_{\rm lb} = 0$ is a feasible option).

Step 2: For solving problem (16) of packing r rectangles, we consider a random initial guess from which we start a local minimization, with the expectation of finding a global solution that annihilates the objective function.

Step 3: If a solution with null optimal cost was found and $r = m_{\rm ub}$, we stop the execution declaring "An optimal packing with $m \equiv m_{\rm ub}$ rectangles was found".

Step 4: If a solution with null optimal cost was found but $r < m_{\rm ub}$, we increase r by 1 and return to Step 2. In this case, the packing problem with r rectangles was solved and, now, we will try to solve a larger one.

Step 5: If a solution with null optimal cost was not found, we return to Step 2. In this case, the packing problem we are trying to solve remains the same and we will try to solve it choosing another random initial guess.

Figure 5 gives a graphical representation of these steps, including the stopping criterion based on exhaustion. If the algorithm stops with $m \equiv m_{\rm lb} - 1$ we conclude that even a packing with $m_{\rm lb}$ rectangles was not found. Moreover, when the method stops declaring "A packing with m rectangles was found", this means that it was the best the method could do within the available time. There is no optimality certification apart from the one that might be given attaining a known upper bound.

Figure 5 by here.

We chose GENCAN (Birgin and Martínez, 2002) as the local solver. GENCAN is a recently introduced active-set method for smooth box-constrained minimization. For a description of basic techniques of continuous optimization and active-set methods see, for example, (Dennis Jr and Schnabel, 1983) and (Luenberger, 1984) (pp. 326–330). GENCAN adopts the leaving-face criterion of (Birgin and Martínez, 2001), that employs the spectral projected gradients defined in (Birgin *et al*, 2000; Birgin *et al*, 2001). For the internal-to-the-face minimization GENCAN uses a general algorithm with a line search that combines backtracking and extrapolation. In the present available implementation, GENCAN employs, for the direction chosen at each step inside the faces, a truncated-Newton approach. This means that the search vector is an approximate minimizer of the quadratic approximation of the function in the current face. Conjugate gradients are used to find this direction. The method is fully described in (Birgin and Martínez, 2002) where extensive numerical experiments assess its reliability. GENCAN is available as a part of the TANGO project (see www.ime.usp.br/~egbirgin/tango/).

4.2 Improving the initial random configuration

The whole packing procedure consists of solving a potentially large number of unconstrained optimization problems. Each time we try to solve an instance of the smooth nonlinear unconstrained optimization problem (16), a random configuration is used as an initial guess to run a local solver. No information of the previous trials is considered. Information of preceding problems with a small number of rectangles is not useful because the optimal configuration may vary a lot when the number of rectangles increases. Information of preceding problems with the same number of rectangles is not useful either. Note that, if we are still trying to solve a problem repeating the number of rectangles, this is because we failed in the previous trials. So, the final points of those trials are undesired stationary points of the nonlinear model, and to start near to them would trap us in the basin of convergence of a spurious solution. Starting from random initial guesses hopefully provides the method with the necessary diversification to find the desired global solution.

Objective functions having products like the ones in (11) exhibit deep valleys that make the task of finding its global minimizer very hard. This is the case of function (12) which is part of the objective function of our nonlinear optimization problem (16). So, starting from a random initial guess will probably lead us to the basin of convergence of a near local non-global minimizer.

With the purpose of overcoming this inconvenience, we formulate another smoother (and hopefully easier) optimization problem that, in some sense, approximates the optimization problems (16). Consider some circles that "almost cover" the rectangle, as shown in Figure 6. (The number of circles and their distribution inside the rectangle are completely arbitrary and based on empirical evidences.)

Figure 6 by here.

Given a random initial guess, we first solve the problem of finding displacements such that the above mentioned circles that belong to different rectangles do not overlap. The model for this auxiliary problem follows closely the model for packing cylinders introduced in (Birgin *et al*, 2005^b). This problem turns out to be much easier than (16). Its minimizers (global or not) are not necessarily solutions of (16) but almost always provide improved initial points for solving the rectangle packing problem.

Of course, there is a trade-off between spending time for improving the initial random guess (solving the auxiliary problem) and using the time for solving more problems, starting from (cheaply generated) initial random guesses. In the next subsection we will see that the auxiliary problem helps to find better solutions with less global time.

4.3 Experiments

We consider the set of rectangle packing problems in arbitrary convex regions described in Table 1. Table 1 shows the description of each problem (inequalities that describe the convex region Ω, the area of Ω, dimensions of the rectangles to be packed, and the area of the rectangles).

Problems 26 to 30, in which some special items are packed, deserve some explanations. The special black items in problems 26 to 29 are items that will be fixed within the convex regions. The special items in problem 30 will be packed as regular items, allowing free displacements and rotations inside the convex region. All these items are unions of rectangles (sometimes overlapped and sometimes not). The special item in problems 26 and 29 corresponds to two non-overlapped rectangles and its area is 4. The special item in problem 27 corresponds to 8 non-overlapped rectangles and its area is 16. The special item in problem 28 corresponds to two overlapped rectangles and its area is 3. Finally, the first, the second and the fourth special items in problem 30 correspond to 2, 2 and 3 non-overlapped rectangles and their areas are 0.665, 0.665 and 0.9975, respectively, while the third special item corresponds to two overlapped rectangles and its area is 0.5425. In all these cases, we first try to pack the special items in the order and quantities that appear in Table 1. Then, we continue packing as many rectangles as possible.

Table 1 by here.

Table 1 (cont.) by here.

All the experiments were run on an 1.8GHz AMD Opteron 244 processor, 2Gb of RAM memory and Linux operating system. Codes are in Fortran77 and the compiler option “-O” was adopted.

In the experiments we set our maximum effort in terms of CPU-time, fixing $T = 6$ hours. Since, for arbitrary convex regions, it is not easy to find tight bounds, we set $m_{\rm lb} = 1$ and $m_{\rm ub}$ as the upper bound based on the areas showed in Table 1. Table 2 shows, for each problem, the number of items that were packed, the total number of nonlinear programming problems that were solved and the elapsed CPU-time until the best solution was found. (The remaining time, to complete the 6 hours, was spent just to confirm that a better solution could not be found.) Figures 7 and 8 illustrate the solutions. In Table 2, the column "Upper bound" deserves some explanation. Assuming that the special items (in problems 26 to 30) can be packed, the upper bound on the number of items that can be packed is computed as follows:

$$\text{Upper bound} = \#\text{ special items} + \left\lfloor \frac{\text{Convex region area} - \text{Sum of the special items areas}}{\text{rectangular item area}} \right\rfloor$$

In these cases, the column "Number of packed items" also corresponds to the number of special items packed plus the number of rectangles packed.

Problems 1–10 show how the method packs rectangles in arbitrary convex regions. In (Mahoney, 1985) the problem of packing containers in airplanes is analyzed. This practical problem resembles the geometrical problems studied here. Problems 11–15 consist in packing unitary squares in equilateral triangles. Problems 16–23 consist in packing unitary squares in bigger squares. Both problems are related to classical combinatorial and geometrical problems (see (Erdös and Graham, 1975; Friedman, 2000); namely, to find the smaller "big" square (or equilateral triangle) that contains a fixed number of "small" squares. The duality relationships between this problem and the one considered in the present paper can be used to obtain approximations to the solution of the first. Problems 24 and 25 show that our approach is able to obtain non-guillotine and guillotine solutions of the problem of packing rectangles within a rectangle. In Problems 26–29 we show that the method can be used to pack items in regions that have prohibited zones (Viana and Arenales, to appear; Bischoff and Dowsland, 1982), when these zones can be described as unions of rectangles. In this case, the corresponding problem is handled using constraints that impose a fixed position to the rectangles that represent the forbidden regions. Finally, observe that, at a first sight, in problem 30, the packed items are not rectangles, as in (Amaral *et al*, 1990; Bennell *et al*, 2001; Dowsland *et al*, 1998; Dowsland *et al*, 2002; Lamousin and Waggenspack Jr., 1997; Li and Milenkovic, 1995). However, each item is also a union of rectangles and the desired configuration is obtained using the basic rectangle-packing procedure and imposing additional constraints to force that the packed rectangles preserve adequate relative positions. In other words, the problem of packing items that are union of rectangles can be handled employing, essentially, the same technique introduced for rectangles.

Table 2 by here.

Figure 7 by here.

Figure 8 by here.

In addition to the previous set of problems, we also consider a few problems of packing (l, w)-rectangles within an (L, W)-rectangle from the problems set Cover I. We apply the Method of Sentinels, the orthogonal packing procedure GENPACK described in (Birgin *et al*, 2005^a) and the L-Approach introduced in (Lins *et al*, 2003, Birgin *et al*, 2005^c). Table 3 shows the obtained results and Figure 9 illustrates the solutions.

Table 3 by here.

Figure 9 by here.

From these results it becomes evident that the nonlinear approaches described in this work and in (Birgin *et al*, 2005^a) are not competitive with clever methods developed for this specific problem. The Method of Sentinels found the optimal solution in 3 cases and GENPACK found the optimal solution in 9 problems, while the L-Approach found the optimal solution in all the 10 problems. Moreover, the nonlinear approaches used a CPU-time several orders of magnitude bigger than the L-Approach. Remembering that we are using a CPU-time limit of 6 hours, and noting that without a time limit the nonlinear approaches find the optimal solution of any problem with probability one, we can concentrate the "comparison" in the efficiency of the methods instead of in their robustness. Then, we can say that the Method of Sentinels is the less efficient method while the L-Approach is the more efficient one for this type of problems. This is due to the amount of information about the specific problem that the method uses.

In fact, for the special case of packing rectangles within rectangles, and for this set of problems, at least two properties are well known:

1. An optimal solution with the sides of the (l, w)-rectangles parallel to the sides of the (L, W)-rectangle exists.

2. Consider the set $C = \{c = a\ l + b\ w \mid c \leq \min(L, W)\} \subset I\!N$, with a and b non-negative integer numbers. Assume,without loss of generality, that the (L, W)-rectangle has its left-down corner in the origin. Then an optimal solution with the left-down corner (x_i, y_i) of each (l, w)-rectangle such that $x_i, y_i \in C$ exists.

The L-Approach uses both properties, while GENPACK uses only the orthogonality property and the Method of Sentinels does not use any of them. So, the advantage of the Method of Sentinels of being able to deal with a wide range of problems becomes a disadvantage when known properties of particular problems (not being used by the method) exist.

Finally, to verify the usefulness of the auxiliary problem in the generation of improved initial solutions, we also applied the Method of Sentinels to the 10 problems of packing rectangles within a rectangle, but without using the auxiliary model. In problem $P_{23}(25, 13, 7, 2)$, 21 rectangles were packed instead of 18. In 3 problems the number of packed rectangles was the same and in the other 6 problems 3.5 less rectangles were packed in average. In the 3 problems in which the same number of rectangles was packed, the time was more than 4 times the time used by the Method of Sentinels plus the auxiliary problem. So, the application of the auxiliary problem for the generation of improved initial solutions seems to be very useful in practice.

5 Conclusions

In this paper we introduced the concept of *sentinels*. We proved the existence of finite sets of sentinels for polygons with their internal angles larger than or equal to $\pi/2$. Moreover, we found minimal sets of sentinels to pack identical rectangles. An interesting and challenging problem is to find minimal sets of sentinels for arbitrary non-regular polygons or for 3D items.

We introduced a nonlinear-programming oriented algorithm for packing items within arbitrary regions. The method is based in the new concept of sentinels. To improve the efficiency of the proposed method, the initial approximations are chosen using an auxiliary simpler nonlinear-programming procedure. However, the initial procedure alone is not able to find solutions of the packing problem and using sentinels is necessary.

The main limitation of the proposed model for solving packing problems relies in the complexity of the continuous optimization problem. Basically, this is a global optimization problem with many local-nonglobal minimizers. General and specific novel strategies for global optimization must be useful for improving the efficiency of the new approach.

Acknowledgement: The authors are indebted to three anonymous referees whose comments helped a lot to improve this paper.

References

Amaral C, Bernardo J and Jorge J (1990). Marker making using automatic placement of irregular shapes for the garment industry. *Computers and Graphics* **14**: 41–46.

Beasley J E (1995). An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research* **33**: 49–64.

Beasley J E (2004). A population heuristic for constrained two-dimensional non-guillotine cutting. *European Journal of Operational Research* **156**: 601–627.

Bennell J A, Dowsland K A and Dowsland W B (2001). The irregular cutting-stock problem - a new procedure for deriving the non-fit polygon. *Computers & Operations Research* **28**: 271–287.

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

Birgin E G and Martínez J M (2002). Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications* **23**: 101–125.

Birgin E G, Martínez J M, Mascarenhas W F and Ronconi D P (2004). Method of Sentinels for Packing Items whitin Arbitrary Convex Regions. Technical Report, IME – USP (available at www.ime.usp.br/˜egbirgin).

Birgin E G, Martínez J M, Nishihara F H and Ronconi D P (2005^a). Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization. *Computers & Operations Research.* To appear.

Birgin E G, Martínez J M and Raydan M (2000). Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization* **10**: 1196–1211.

Birgin E G, Martínez J M and Raydan M (2001). Algorithm 813 SPG: Software for convex-constrained optimization. *ACM Transactions on Mathematical Software* **27**: 340–349.

Birgin E G, Martínez J M and Ronconi D P (2005^b). Optimizing the Packing of Cylinders into a Rectangular Container: A Nonlinear Approach. *European Journal of Operational Research* **160**: 19–33.

Birgin E G, Morabito R and Nishihara F H (2005^c). A note on an L-approach for solving the manufacturer's pallet loading problem. *Journal of the Operational Research Society.* To appear.

Bischoff E and Dowsland W B (1982). An Application of the Micro to Product Design and Distribution. *Journal of the Operational Research Society* **33**: 271–280.

Chen C S, Lee S M and Shen Q S (1995). An analytical model for the container loading problem. *European Journal of Operational Research* **80**: 68–76.

Christofides N and Hadjiconstantinou E (1995). An exact algorithm for orthogonal 2-D cutting problems using guillotine cuts. *European Journal of Operational Research* **83**: 21–38.

Dennis Jr. J E and Schnabel R B (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall: Englewoods Cliffs.

Dowsland K A (1987). An exact algorithm for the pallet loading problem. *European Journal of Operational Research* **84**: 78–84.

Dowsland K A, Dowsland K B and Bennell J A (1998). Jostling for position - Local improvement for irregular cutting patterns. *Journal of the Operational Research Society* **49**: 647–658.

Dowsland K A, Vaid S and Dowsland W B (2002). An algorithm for polygon placement using a bottom-left strategy. *European Journal of Operational Research* **141**: 371–381.

Dyckhoff H (1990). A typology of cutting and packing problems. *European Journal of Operational Research* **44**: 145–159.

Erdös P and Graham R L (1975). On packing squares with equal squares. *Journal of Combinatorial Theory A* **19**: 119–123.

Friedman E (2000). Packing unit squares into squares. *The electronic Journal of Combinatorics* **7**: DS#7.

Hadjiconstantinou E and Christofides N (1995). An exact algorithm for general, orthogonal two-dimensional knapsack problems. *European Journal of Operational Research* **83**: 39–56.

Lamousin H and Waggenspack Jr. W N (1997). Nesting of two-dimensional irregular parts using a shape reasoning heuristic. *Computer-Aided Design* **29**: 221–238.

Li Z and Milenkovic V (1995). Compaction and separation algorithms for non convex polygons and their application. *European Journal of Operational Research* **84**: 539–561.

Lins L, Lins S and Morabito R (2003). An L-approach for packing (l,w)- rectangles into rectangular and L-shaped pieces. *Journal of the Operational Research Society* **54**: 777–789.

Luenberger D G (1984). *Linear and Nonlinear Programming.* Addison-Wesley: California, London, Amsterdam, Ontario, Sydney.

Mahoney J H (1985). Intermodal Freight Transportation. Eno Foundation for Transportation, Inc.: Westport, Connecticut.

Morabito R and Morales S (1998). A simple and effective recursive procedure for the manufacturer's pallet loading problem. *Journal of the Operational Research Society* **49**: 819–828.

Tarnowski A, Terno J and Scheithauer G (1994). A polynomial-time algorithm for the guillotine pallet loading problem. *INFOR* **32**: 275–287.

Tsai R D, Malstrom E M and Kuo W (1993). Three dimensional palletization of mixed box sizes. *IIE Transactions* **25**: 64–75.

Viana A C G and Arenales M. O Problema de Corte de Placas Defeituosas. *Pesquisa Operacional.* To appear.

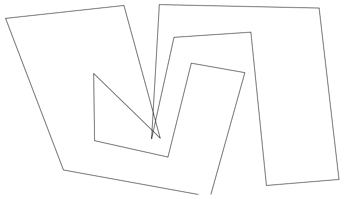


Figure 1: Polygons with internal angles smaller than $\pi/2$ may not have finite sets of sentinels.

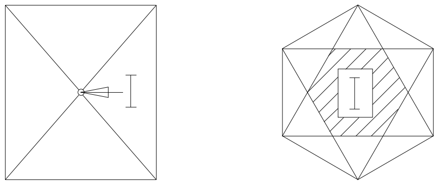


Figure 2: The set I for the rectangle (just one point) and the hexagon (hatched).

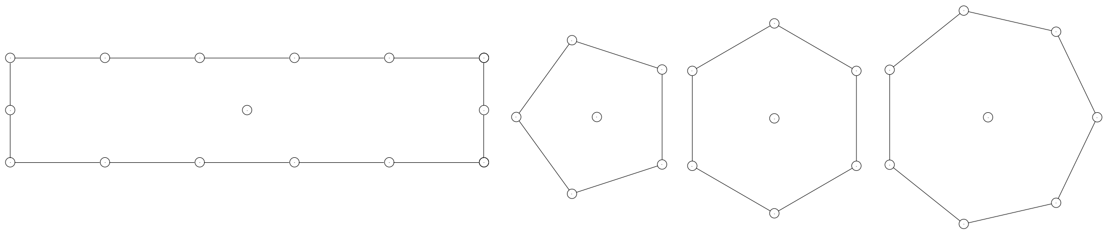


Figure 3: A family of polygons and their sentinels. The rectangle has smallest side of length slightly bigger than 1 and the other polygons have sides of length 1. The sentinels in the borders are 1 apart, except for the ones in the smallest side of the rectangle, which are roughly 1/2 apart.

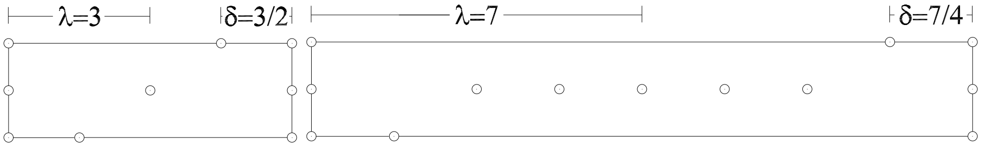


Figure 4: Minimal sentinels sets for the rectangles R_λ, with $\lambda = 3$ and 7.

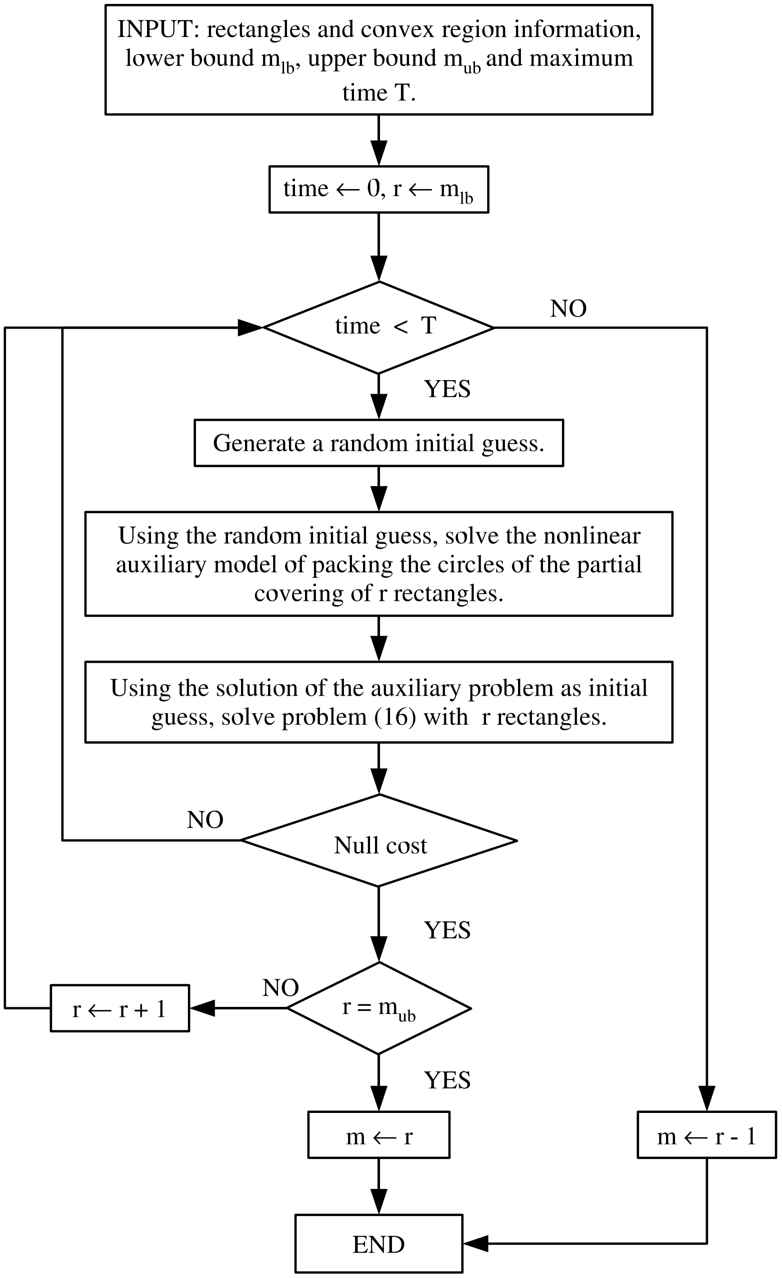

INPUT: rectangles and convex region information, lower bound m_{lb}, upper bound m_{ub} and maximum time T.
time ← 0, r ← m_{lb}
time < T
NO
YES
Generate a random initial guess.
Using the random initial guess, solve the nonlinear auxiliary model of packing the circles of the partial covering of r rectangles.
Using the solution of the auxiliary problem as initial guess, solve problem (16) with r rectangles.
NO
Null cost
YES
NO
r ← r + 1
r = m_{ub}
YES
m ← r
m ← r - 1
END

Figure 5: Fluxogram of the algorithm. When the method stops, m represents the number of packed items.

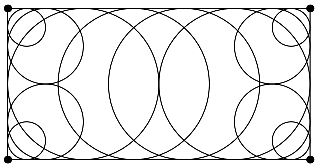


Figure 6: Partial covering of the rectangle by circles used to generate initial configurations.

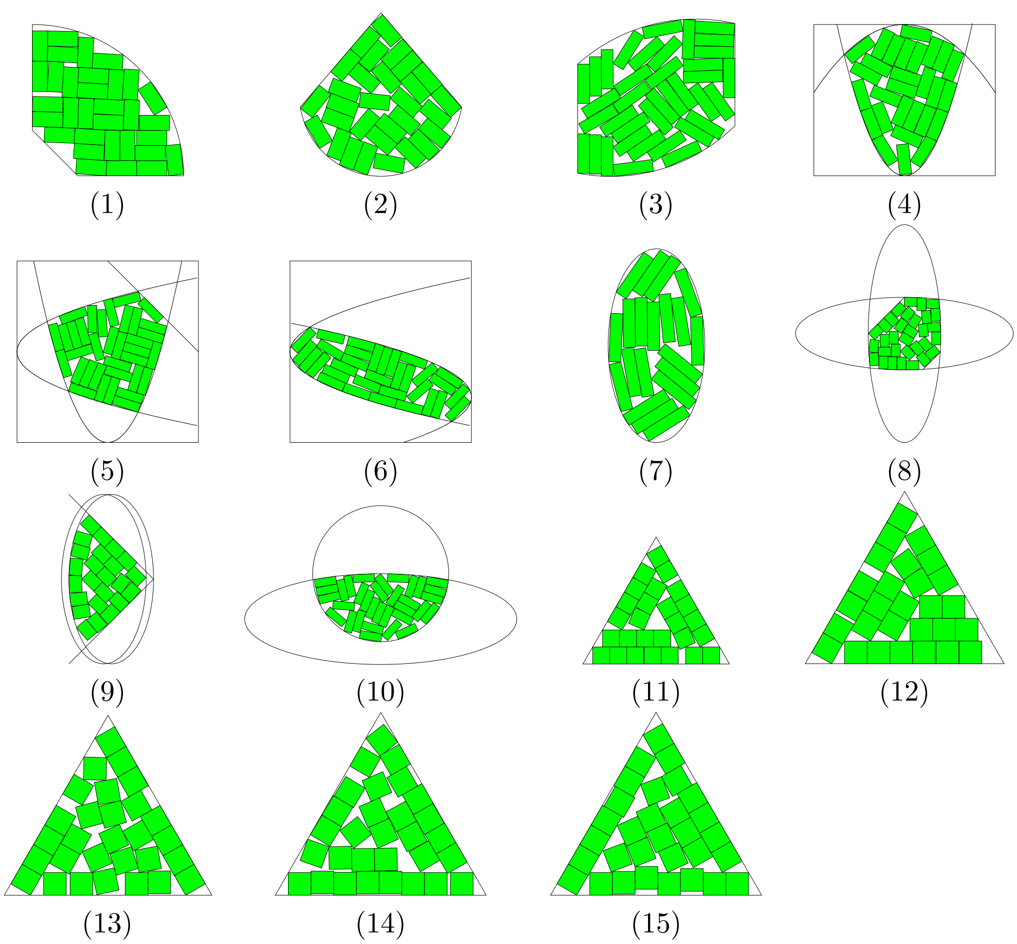
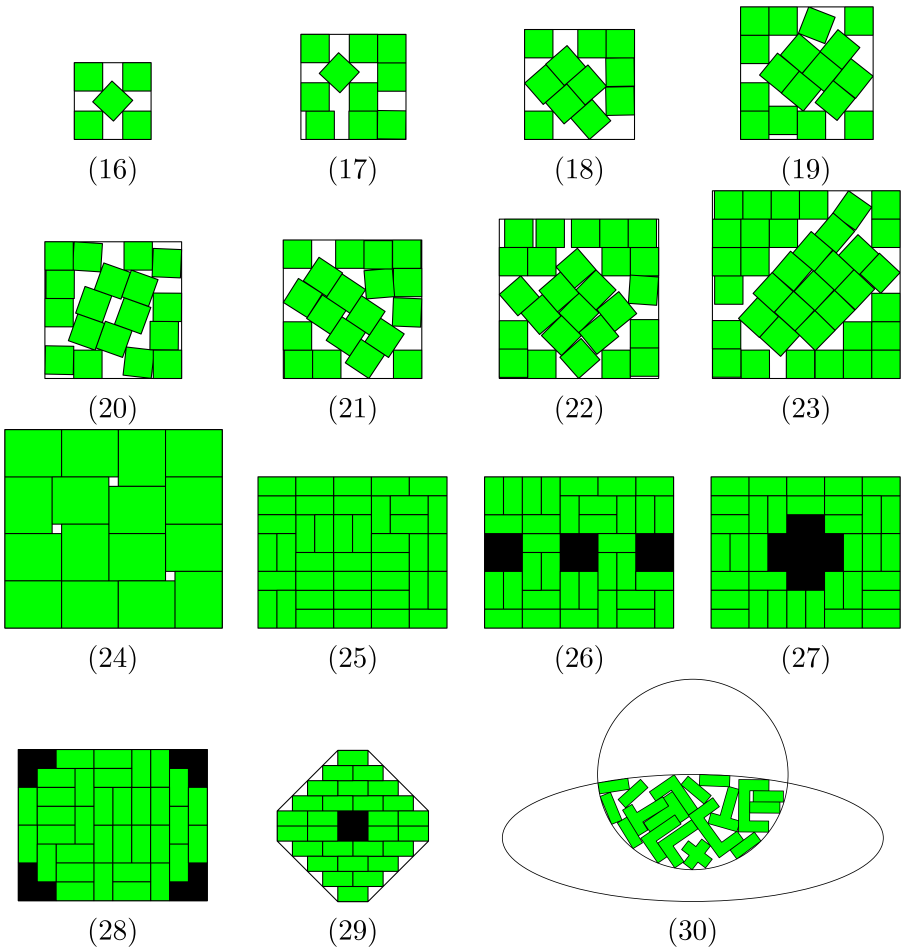


Figure 7: Graphical representation of the solutions. The pictures are automatically generated by the software using MetaPost.


(16) (17) (18) (19)

(20) (21) (22) (23)

(24) (25) (26) (27)

(28) (29) (30)

Figure 8: Graphical representation of the solutions. The pictures are automatically generated by the software using MetaPost (cont.).

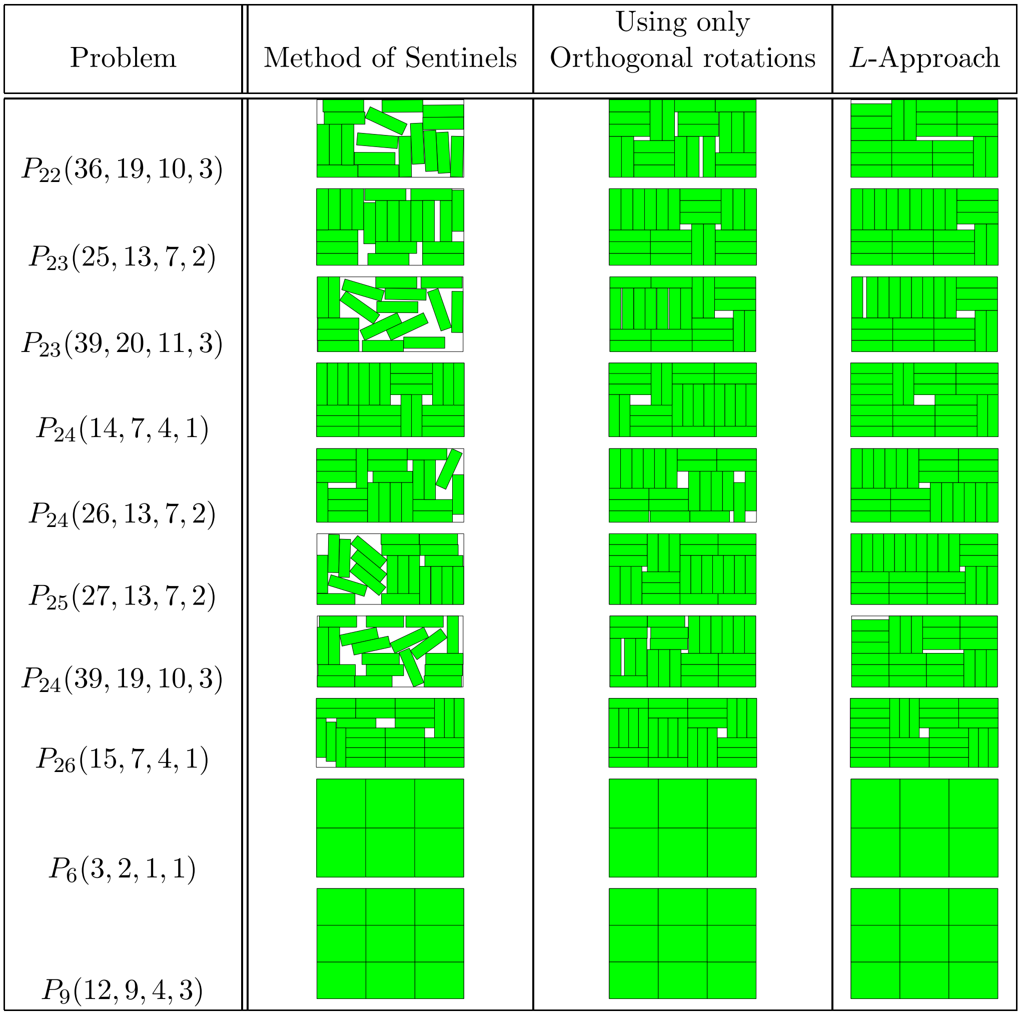


Figure 9: Comparison between the Method of Sentinels, GENPACK (for orthogonal packing) and the L-Approach in the particular problem of packing identical rectangles within a rectangle. When more information of the problem the method uses, better is its performance. This is the reason why the nonlinear approaches for packing rectangles within arbitrary convex regions are not competitive with clever methods developed for this particular case.

Problem	Convex Region		Rectangular item	
	Description	Area	Dimensions	Area
1	$g_1(x_1, x_2) = -x_1 - x_2 + 3$ $g_2(x_1, x_2) = x_1^2 + x_2^2 - 100$ $g_3(x_1, x_2) = -x_1$ $g_4(x_1, x_2) = -x_2$	74.1	2×1	2
2	$g_1(x_1, x_2) = -7x_1 + 6x_2 - 24$ $g_2(x_1, x_2) = 7x_1 + 6x_2 - 108$ $g_3(x_1, x_2) = (x_1 - 6)^2 + (x_2 - 8)^2 - 9$	21.7	1.1×0.55	0.61
3	$g_1(x_1, x_2) = (x_1 - 6)^2 + x_2^2 - 81$ $g_2(x_1, x_2) = (x_1 - 1.7)^2 + (x_2 - 10)^2 - 81$ $g_3(x_1, x_2) = -x_1$ $g_4(x_1, x_2) = x_1 - 8$	54.4	2×0.6	1.2
4	$g_1(x_1, x_2) = x_1^2 - x_2$ $g_2(x_1, x_2) = x_1^2/4 + x_2 - 5$	13.3	1×0.4	0.40
5	$g_1(x_1, x_2) = x_1^2 - x_2$ $g_2(x_1, x_2) = -x_1 + x_2^2 - 6x_2 + 6$ $g_3(x_1, x_2) = x_1 + x_2 - 6$	10.9	0.9×0.3	0.27
6	$g_1(x_1, x_2) = -x_1 + x_2^2 - 6x_2 + 6$ $g_2(x_1, x_2) = x_1 + x_2^2 - 3x_2 - 3/4$	10.2	0.9×0.3	0.27
7	$g_1(x_1, x_2) = (x_1 - 2)^2/4 + (x_2 - 4)^2/16 - 1$	25.1	2×0.5	1
8	$g_1(x_1, x_2) = (x_1 - 6)^2/4 + (x_2 - 6)^2/36 - 1$ $g_2(x_1, x_2) = (x_1 - 6)^2/36 + (x_2 - 6)^2/4 - 1$ $g_3(x_1, x_2) = x_1 - x_2 - 3$ $g_4(x_1, x_2) = -x_1 + x_2 - 2$	13.2	0.7×0.5	0.35
9	$g_1(x_1, x_2) = (x_1 - 3)^2/4 + (x_2 - 4)^2/16 - 1$ $g_2(x_1, x_2) = (x_1 - 2.65)^2/4 + (x_2 - 4)^2/16 - 1$ $g_3(x_1, x_2) = x_1 - x_2 - 1$ $g_4(x_1, x_2) = x_1 + x_2 - 9$ $g_5(x_1, x_2) = -x_1 + 1$	13.7	0.8×0.6	0.48
10	$g_1(x_1, x_2) = (x_1 - 6)^2/36 + (x_2 - 6)^2/4 - 1$ $g_2(x_1, x_2) = (x_1 - 6)^2/9 + (x_2 - 8)^2/9 - 1$	13.6	0.95×0.35	0.33
11	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(3/2 + \sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	32.2	1×1	1
12	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(2 + 4/\sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	33.3	1×1	1
13	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(3 + 4/\sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	36.3	1×1	1
14	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(2 + 2\sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	37.5	1×1	1
15	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(4 + 4/\sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	37.5	1×1	1
16	$g_1(x_1, x_2) = -x_1$ $g_2(x_1, x_2) = x_1 - (2 + \sqrt{2}/2)$ $g_3(x_1, x_2) = -x_2$ $g_4(x_1, x_2) = x_2 - (2 + \sqrt{2}/2)$	7.3	1×1	1
17	$g_1(x_1, x_2) = -x_1$ $g_2(x_1, x_2) = x_1 - (3 + \sqrt{2}/2)$ $g_3(x_1, x_2) = -x_2$ $g_4(x_1, x_2) = x_2 - (3 + \sqrt{2}/2)$	13.7	1×1	1

Table 1: Problems definition.

Problem	Convex Region		Rectangular item	
	Description	Area	Dimensions	Area
18	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - 3.878$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - 3.878$	15.0	1 × 1	1
19	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - 4.676$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - 4.676$	21.9	1 × 1	1
20	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - (7 + \sqrt{7}/2)$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - (7 + \sqrt{7}/2)$	23.3	1 × 1	1
21	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - (3 + 4\sqrt{2}/3)$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - (3 + 4\sqrt{2}/3)$	23.9	1 × 1	1
22	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - (7/2 + 3\sqrt{2}/2)$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - (7/2 + 3\sqrt{2}/2)$	31.6	1 × 1	1
23	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - 6.621$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - 6.621$	43.9	1 × 1	1
24	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - 23$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - 21$	483.0	6 × 5	30
25	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - 10$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - 8$	80.0	2 × 1	2
26	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - 10$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - 8$	80.0	3 fixed items plus 2 × 1 rectangles	2
27	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - 10$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - 8$	80.0	the fixed item plus 2 × 1 rectangles	2
28	$g_1(x_1,x_2) = -x_1$ $g_2(x_1,x_2) = x_1 - 10$ $g_3(x_1,x_2) = -x_2$ $g_4(x_1,x_2) = x_2 - 8$	80.0	4 fixed items plus 2 × 1 rectangles	2
29	$g_1(x_1,x_2) = -x_1 + x_2 - 6$ $g_2(x_1,x_2) = x_1 - x_2 - 6$ $g_3(x_1,x_2) = -x_1 - x_2 + 6$ $g_4(x_1,x_2) = x_1 + x_2 - 18$ $g_5(x_1,x_2) = -x_1 + 1$ $g_6(x_1,x_2) = x_1 - 11$ $g_7(x_1,x_2) = -x_2 + 1$ $g_8(x_1,x_2) = x_2 - 11$	68.0	the fixed item plus 2 × 1 rectangles	2
30	$g_1(x_1,x_2) = (x_1 - 6)^2/36 + (x_2 - 6)^2/4 - 1$ $g_2(x_1,x_2) = (x_1 - 6)^2/9 + (x_2 - 6)^2/9 - 1$	13.6	the following non-fixed items: 3 , 3 , 1 , 2 , plus 0.95 × 0.35 rectangles	0.3325

Table 1: Problems definition (cont.)

Problem	Upper bound	Number of packed items	Number of trials	CPU-time in seconds
1	37	32	1990	14078.67
2	35	30	91	958.27
3	45	37	332	7887.49
4	33	28	5862	18177.95
5	40	35	927	6951.79
6	37	32	976	5801.68
7	25	20	391	557.09
8	37	32	197	530.12
9	28	24	4021	3750.76
10	40	33	4965	17772.39
11	32	27	278	267.53
12	33	28	74	85.37
13	36	29	38	22.29
14	37	30	69	97.28
15	37	31	53	56.47
16	7	5	6	0.02
17	13	10	12	0.28
18	15	11	408	53.43
19	21	17	74	23.28
20	23	18	25	5.58
21	23	19	521	268.89
22	31	26	68	45.04
23	43	37	1253	2841.36
24	16	16	225	405.03
25	40	40	156	249.24
26	37	37	89	146.03
27	35	35	143	191.25
28	38	38	482	670.25
29	33	29	32	12.68
30	31	18	4144	10228.52

Table 2: Performance of the Method of Sentinels.

Problem	Method of Sentinels			Using only orthogonal rotations			L-approach	
	n	# trials	CPU-time	n	# trials	CPU-time	n	CPU-time
$P_{22}(36, 19, 10, 3)$	17	533	5083.22	22	66710	4429.52	22	0.25
$P_{23}(25, 13, 7, 2)$	18	1835	9923.40	23	320554	8464.60	23	0.15
$P_{23}(39, 20, 11, 3)$	21	212	1730.03	23	66347	5046.09	23	0.32
$P_{24}(14, 7, 4, 1)$	24	9030	14313.55	24	62359	1199.32	24	0.08
$P_{24}(26, 13, 7, 2)$	22	2035	12055.44	23	62439	2659.11	24	0.23
$P_{25}(27, 13, 7, 2)$	21	607	4640.60	25	60378	2434.14	25	0.22
$P_{24}(39, 19, 10, 3)$	19	754	8766.18	24	62611	5357.32	24	0.52
$P_{26}(15, 7, 4, 1)$	25	581	1105.93	26	60766	1451.70	26	0.13
$P_6(3, 2, 1, 1)$	6	7	0.03	6	8	0.00	6	0.00
$P_9(12, 9, 4, 3)$	9	303	28.38	9	114996	224.24	9	0.00

Table 3: Performance of the Method of Sentinels, GENPACK and the L-Approach in the particular problem of packing identical rectangles within a rectangle.